CLIQUE-CONVERGENCE IS UNDECIDABLE FOR AUTOMATIC GRAPHS

C. CEDILLO[1,3,4,5,6] AND M.A. PIZAÑA[2,3,6]

Abstract. The *clique operator* transforms a graph G into its *clique graph* $K(G)$, which is the intersection graph of all the (maximal) cliques of G. *Iterated clique graphs* are then defined by: $K^n(G) = K(K^{n-1}(G))$, $K^0(G) = G$. If there are some $n \neq m$ such that $K^n(G) \cong K^m(G)$ then we say that G is *clique-convergent*. The clique graph operator and iterated clique graphs have been studied extensively, but no characterization for clique-convergence has been found so far.

Automatic graphs are (not necessarily finite) graphs whose vertices and edges can be recognized by finite automata. Automatic graphs (and *automatic structures*) have strong decidability properties inherited from the finite automata defining them.

Here we prove that clique-convergence is algorithmically undecidable for the class of automatic graphs. Moreover, the problem remains undecidable, if we reduce to the class that contain only quasi clique-Helly and bounded degree graphs. As a consequence, it follows that clique-convergence for automatic graphs is not first-order expressible.

1. Introduction

Let $\mathcal{G}$ be the class of all graphs. In *graph dynamics* [19], we have a graph operator $\Phi : \mathcal{G} \to \mathcal{G}$ and we are interested in the properties of the resulting discrete dynamic system. This setting is useful in certain approaches to loop quantum gravity [20–22] where the quantum spacetime foam is to be obtained as an emergent property from the (hypothetical) underlying discrete spacetime.

Given a graph operator Φ, we can define the corresponding *iterated* operators by $\Phi^0(G) = G$ and $\Phi^n(G) = \Phi(\Phi^{n-1}(G))$. One of the central topics of study in graph dynamics is that of Φ-*convergence*: A graph G is said to be Φ-convergent, if $\Phi^n(G) \cong \Phi^m(G)$, for some $n < m$; otherwise, G is Φ-*divergent*. Φ-convergence have been fully characterized for many graphs operators, ranging from the classic characterization of convergence for iterated line graphs and iterated path graphs [18] to the more recent characterization for iterated biclique graphs [7].

The clique operator K, however, is widely considered one of the most complex ones [19] and a characterization of K-convergence (or *clique-convergence*) has resisted all attempts during the 48 years since the notion of iterated clique graphs was

[1]Email: mc.cedilloc@gmail.com
[2]Email: mpizana@gmail.com
[3]Universidad Autónoma Metropolitana - Iztapalapa, Mexico City, Mexico.
[4]Centro Universitario UAEM Nezahualcóyotl, Nezahualcoyotl City, Mexico.
[5]Partially supported by CONACYT, scholarship 397900.
[6]Partially supported by SEP-CONACYT, grant A1-S-45528.
Key words and phrases. graph theory, graph dynamics, clique graphs, decidability, automatic structures.

introduced in [8]. No substantial progress has been made either on showing that clique-convergence might be algorithmically undecidable. The algorithmic undecidability of clique-convergence has been suggested in several places starting in [17], including [19]; the first explicit statement on the issue appeared in [15] and the first explicit statement in a formal publication appeared in [13]. The main objective of this research program is to prove that the problem of deciding clique-convergence is undecidable for the class of finite simple graphs, i.e. that there is no algorithm for deciding clique-convergence. This problem, however, still remains open.

Here we present the first undecidability result for clique-convergence by relaxing the condition that the class under study is only the class of *finite* simple graphs. Infinite graphs have been considered within clique graph theory in several papers including [4, 14]. Here we wanted to relax the finiteness condition as little as possible and hence we considered only (possibly infinite but) finitely presentable, quasi clique-Helly, locally finite graphs of bounded degree and with at most one dominated vertex. Indeed, striving for the most stringent conditions, we decided to consider only *automatic graphs* which are graphs whose vertices and edges can be recognized by deterministic finite automata (the formal definitions are in the next section). Automata are among the simplest models of computation, so automatic graphs have not only algorithms for determining vertices and edges, but also these algorithms are of the most simple form. Automatic graphs and *automatic structures* [1, 2, 11, 12, 23] have very strong decidability properties inherited from automata theory. Indeed, the first-order theory of any automatic structure is decidable (see Theorem 2.3), but also first-order theories expanded with certain generalized quantifiers are still decidable [23] and even some fragments of second-order theories are decidable [12]. Moreover clique-Helly graphs without dominated vertices are all known to be clique-convergent even in the infinite case (see Theorem 2.7) hence our undecidability result for quasi clique-Helly graphs (which become clique-Helly after removing only one vertex) with at most one dominated vertex is somewhat unexpected.

Let $\mathcal{AG}$ be the class of automatic graphs, $\mathcal{TM}$ the class of Turing machines and let $\mathcal{TMW}$ be the class of ordered pairs (M, w), where w is an input string for $M \in \mathcal{TM}$. We shall prove the following result:

Theorem 1.1. *Clique-convergence is algorithmically undecidable for the class of automatic graphs. Moreover, the problem remains undecidable, if we reduce to the class that contains only quasi clique-Helly and bounded degree graphs with at most one dominated vertex.*

The proof is obtained by reduction from the halting problem (well known to be undecidable), indeed the previous theorem follows immediately from the next one:

Theorem 1.2. *There is a computable function $\lambda : \mathcal{TMW} \to \mathcal{AG}$ such that for each $(M, w) \in \mathcal{TMW}$, M halts on input w if and only if $\lambda(M, w)$ is clique-convergent. Hence, clique-convergence is undecidable for any class containing $\mathcal{AG}_0 := \lambda(\mathcal{TMW})$. Furthermore, λ can be chosen such that the graphs in $\mathcal{AG}_0$ are quasi clique-Helly, of bounded degree and with at most one dominated vertex.*

Section 4 is devoted to the proof of Theorem 1.2. We considered it pertinent to present an informal overview of the proof in Section 3. All preliminary results, definitions and terminology are in Section 2. In Section 5, we explore several consequences of Theorem 1.2.

2. Preliminaries

This paper is mostly self-contained but it always helps to have some previous experience with the topics at hand: language theory (automata and Turing machines), undecidability, automatic presentations of structures, clique graph theory and algorithms. Most crucially, we will not explain why some constructions are *effective* (i.e. that the construction can be realized algorithmically) since this is usually obvious for those who have a previous background on algorithms, but it would require an enormous amount of space to explain this for those who do not.

We refer the reader to the literature for standard terminology and results on language theory and undecidability [9], automatic presentations of structures [23], graph theory [3] and clique graph theory [24]. For the reader's convenience, we overview here the required terminology and results.

2.1. **Language theory.** An *alphabet* Σ is a finite set of *symbols*. A *string* w over Σ is a finite sequence of symbols of Σ. The *length* of a string w is denoted by $|w|$. The *empty string* is the only string of length 0 and it is denoted by ε. We represent the concatenation of two strings by simple juxtaposition vw. Note that, for every string w, we have $\varepsilon w = w\varepsilon = w$. The set of all strings over Σ is denoted by Σ^*. A language L over Σ is any subset $L \subseteq \Sigma^*$. We shall use the special symbol $\diamond \notin \Sigma$ as right-padding for strings. The extended alphabet $\Sigma \cup \{\diamond\}$ is denoted by $\Sigma_\diamond$. If w is a string over Σ, we denote its i-th symbol (for $1 \leq i \leq |w|$) by $w[i]$. We also define $w[i] = \diamond$ for all $i > |w|$. For any set A, the Cartesian product of n copies of A is denoted by A^n. Given a sequence of strings $\overline{w} = (w_1, w_2, \ldots, w_n)$ over Σ, its *convolution* $u = \otimes\overline{w} = \otimes(w_1, w_2, \ldots, w_n)$ is the string of length $|u| = \max |w_i|$ over the alphabet $(\Sigma_\diamond)^n$ whose i-th symbol is $u[i] = (w_1[i], w_2[i], \ldots, w_n[i])$. For example $\otimes(1010, 01) = (1,0)(0,1)(1,\diamond)(0,\diamond)$ is a string of length 4 over the alphabet:

$$(\Sigma_\diamond)^2 = \{(0,0),(0,1),(0,\diamond),(1,0),(1,1),(1,\diamond),(\diamond,0),(\diamond,1),(\diamond,\diamond)\}.$$

Using column vectors instead of tuples, may better convey the underlying idea of the convolution (and right-padding):

$$\otimes\binom{1010}{01} = \binom{1}{0}\binom{0}{1}\binom{1}{\diamond}\binom{0}{\diamond}.$$

We shall use column vectors and tuples indistinctly, but we emphasize that the difference is only notational. Some standard operations on languages are: *union* denoted as $L_1 + L_2$, *intersection* $L_1 \cap L_2$, *complement* $\overline{L} = \Sigma^* - L$, *concatenation* $L_1 \cdot L_2 := \{vw : v \in L_1, w \in L_2\}$, *powers* $L^{(0)} := \{\varepsilon\}$, $L^{(n+1)} := L \cdot L^{(n)}$ (we use parenthesis in the exponents to avoid confusion with the Cartesian product powers, which we shall use much more often) and *Kleene closure* $L^* := \cup_{n=0}^{\infty} L^{(n)}$.

2.2. **Automata theory.** A deterministic finite automaton (DFA) is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is the finite set of *states*, Σ is the alphabet, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state* and $F \subseteq Q$ is the set of *final states*.

Automata are usually represented by diagrams like those in Figure 1. For instance, in the diagram on the left of Figure 1, $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$ and the initial state q_0 is indicated by an arrow not coming from any other state. The final states $F = \{q_1, q_2\}$ are indicated by a double circle in the state. The transition function δ is represented by the arrows in the diagram: we put an arrow from state q to state p labeled with symbol $a \in \Sigma$ whenever $\delta(q, a) = p$. Let us call this

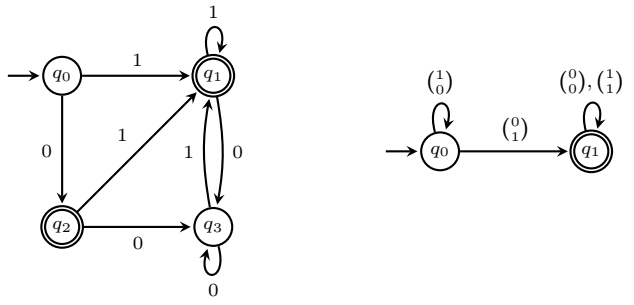

FIGURE 1. Automata $M_{\mathtt{N}}$ (left) and M_{+1} (right).

automaton $M_{\mathtt{N}}$. Then, given a string w, we start at q_0 and follow the arrows in the diagram which are labeled as the symbols in string w, in left-to-right order. Thus the string 00101 produces the walk $q_0 \to q_2 \to q_3 \to q_1 \to q_3 \to q_1$. Since this last state $q_1 \in F$ is a final state, we say that the automaton *accepts* the string 00101, otherwise, the string is *rejected*. The language *recognized* by an automaton $L(M)$ is the set of all strings accepted by the automaton. Any language recognized by a DFA, is said to be a *regular language*. It is not difficult to verify that $\mathtt{N} := L(M_{\mathtt{N}})$ is exactly the set of all non-empty strings ending in 1 plus the string 0, and hence $\mathtt{N}$ is regular. We shall use the strings in $\mathtt{N}$ as *big-endian* binary representations of the natural numbers $\mathbb{N}$, in this representation we start by the least significant bit on the left and end with the most significant bit on the right (so that 011 is the big-endian binary representation of 6). This big-endian representation is standard in automata theory but contrary to our everyday usage.

A *dead state* of an automaton, is a non-final state q such that there is no arrow leaving that state, that is, such that $q \notin F$ and $\delta(q, a) = q$ for all $a \in \Sigma$. It is a standard convention not to draw the dead state of automata as in the diagram on the right of Figure 1: There, an extra (implicit) dead state q_2 must exist which is the destination of all arrows not present in the drawing (like $\delta(q_0, \binom{0}{0}) = q_2$ and $\delta(q_1, \binom{0}{1}) = q_2$).

Whenever $\circ$ is a binary (resp. unary) operation on languages, we say that *regular languages are effectively closed under* $\circ$ if whenever L_1 and L_2 are regular languages $L_1 \circ L_2$ (resp. $\circ(L_1)$) is also a regular language and the automaton for $L_1 \circ L_2$ (resp. $\circ(L_1)$) can be obtained algorithmically from the automata for L_1 and L_2. Then we have:

Theorem 2.1. [9] *Any finite language is regular. Regular languages are effectively closed under: complement, union, intersection, concatenation and Kleene closure.*

For auxiliary purposes, we shall also consider $\hat{\mathtt{N}}$, which contains $\mathtt{N}$ but also allows the strings to end in 0, formally $\hat{\mathtt{N}} = \mathtt{N} + \mathtt{N} \cdot \{0\}$ (recall that the union is denoted by "+" in language theory). It follows by Theorem 2.1, that $\hat{\mathtt{N}}$ is also regular. Note that each natural number has two representations in $\hat{\mathtt{N}}$ (7 is represented both, as 111 and as 1110). Although $\mathbb{N}$ is a set of numbers and $\hat{\mathtt{N}}$ is a set of strings, we shall treat the elements of $\hat{\mathtt{N}}$ as numbers too, so we can use expressions like "$t + a > t + b$" with $t \in \hat{\mathtt{N}}$ and $a, b \in \mathbb{N}$ (for instance).

Regular languages also have very strong decidability properties.

Theorem 2.2. [9] *Given automata M_1 and M_2 the following decision problems are algorithmically decidable: Is $L(M_1)$ empty? Is $L(M_1)$ finite (infinite)? Is $L(M_1)$ equal to $L(M_2)$? Also, given an automaton M we can algorithmically compute the minimal automaton M' such that $L(M) = L(M')$.*

Moreover, given an automaton M we can effectively find a string $w \in L(M)$ (if it exists). Also, given M and w, we can effectively find the next (in lexicographic order) string $w' \in L(M)$ (if it exists).

2.3. **Automatic presentations of structures.** A *relational structure* is a tuple $\mathcal{A} = (A, R_1, \ldots, R_n)$, where A is a set, called its *domain*, and R_i are relations on A, of some arity $r_i \in \mathbb{N}$, that is, $R_i \subseteq A^{r_i}$. When the domain is a set of strings of Σ^*, we can define the convolution of a relation $R \subseteq (\Sigma^*)^r$ as the set $\otimes R = \{\otimes\overline{w} : \overline{w} \in R\}$, that is, $\otimes R \subseteq ((\Sigma_\diamond)^r)^*$ is the set of convolutions of the tuples in R. Observe that $\otimes R$ is a language over alphabet $(\Sigma_\diamond)^r$ and that it is regular whenever there is an automaton recognizing it. We will also say that R is an *automatic relation* whenever $\otimes R$ is a regular language and that R is *recognized* by an automaton M if $\otimes R = L(M)$. We identify elements x of a set $R \subseteq A$ with 1-tuples $(x) \in R^1$, hence a set $R = R^1 \subseteq A^1$ is also a unary relation and then, in this case, $\otimes(R) = R$. Then R is a regular language if and only if it is an automatic relation, in particular, we may use either relational notation or set notation: $R(x) \Leftrightarrow x \in R$.

As an example, consider the successor relation $R_{+1} := \{(t, t+1) \in \hat{\mathbb{N}}^2, |t| = |t+1|\}$, that is, the set of all the pairs of strings $(t, t+1) \in \hat{\mathbb{N}}^2$, where t and $t+1$ have the same length. Recall that the strings in $\hat{\mathbb{N}}$ are big-endian, then all the strings in $\otimes R_{+1}$ are of the form:

$$\otimes\binom{t}{t+1} = \otimes\binom{11\cdots10a_1a_2\cdots a_s}{00\cdots01a_1a_2\cdots a_s} = \binom{1}{0}\binom{1}{0}\cdots\binom{1}{0}\binom{0}{1}\binom{a_1}{a_1}\binom{a_2}{a_2}\cdots\binom{a_s}{a_s}$$

with $a_i \in \{0, 1\}$. We emphasize that the initial prefix $\binom{1}{0}\cdots\binom{1}{0}$ may be empty, and also the final suffix $\binom{a_1}{a_1}\cdots\binom{a_s}{a_s}$ may be empty. But all the strings in $\otimes R_{+1}$ contain the symbol $\binom{0}{1}$ at some point.

Automatic relations have very strong decidability and closure properties, which are inherited from automata theory: Every finite relation is automatic, also automatic relations are closed under projection, instantiation, cylindrification, permutation of coordinates and many others [23]. In particular, automatic relations are effectively closed under logical constructs:

Theorem 2.3 (Thm. 4.4 in [11])**.** *If R_1, R_2 are automatic relations, then the following are also automatic relations: $R_1 \vee R_2$, $R_1 \wedge R_2$, $\neg R_1$, $\exists x(R_1)$ and $\forall x(R_1)$. Moreover, the automata for these relations can be effectively constructed from the automata for R_1 and R_2.*

When R_1 is an unary relation, both $\exists x(R_1)$ and $\forall x(R_1)$ are 0-ary relations which, strictly speaking, can not be automatic for technical reasons, but in [11] they use unary relations ($R = A$ for the `true` relation and $R = \varnothing$ for the `false` relation) to represent these 0-ary relations. We also point out that, for any automatic unary relation R_1 an example of a string w satisfying $R_1(w)$ (and thus a certificate for $\exists x(R_1)$) can be effectively found (if it exists). Moreover, given R_1 and w we can effectively find the next (in lexicographic order) string w' satisfying $R_1(w')$ (if it exists). Observe that $\otimes R$ is always a unary relation.

Note that $\hat{\mathbb{N}}^2$ is automatic by Theorem 2.3 since $\hat{\mathbb{N}}^2(w_1, w_2) = (w_1 \in \hat{\mathbb{N}} \wedge w_2 \in \hat{\mathbb{N}})$. Also, R_{+1} is an automatic relation since $\otimes R_{+1} = L(M_{+1}) \cap \otimes\hat{\mathbb{N}}^2$ is a regular language (here M_{+1} is the automaton on the right of Figure 1). Note that $(110, 001) \in R_{+1}$ but $(11, 001) \notin R_{+1}$, since we defined R_{+1} to require the strings to be of the same length.

Given a relational structure $\mathcal{A} = (A, R_1, \ldots, R_n)$, let $L \subseteq \Sigma^*$ be a language representing the elements of A. A mapping $\mu : L \to A$ specifies that an element $a \in A$ is represented by some string $w \in L$, whenever $\mu(w) = a$. Also, the relations $R_i \subseteq A^{r_i}$ will be represented by the corresponding relations $\mu^{-1}(R_i) \subseteq L^{r_i}$, defined by $\mu^{-1}(R_i) = \{(w_1, w_2, \ldots, w_{r_i}) \in L^{r_i} : (\mu(w_1), \mu(w_2), \ldots, \mu(w_{r_i})) \in R_i\}$. Formally, an *automatic presentation* of a relational structure $\mathcal{A} = (A, R_1, \ldots, R_n)$ is a mapping μ and a tuple of automata $(M_A, M_1, \ldots, M_n)$ such that

(1) $\mu : L(M_A) \to A$ is bijective.
(2) $L(M_i) = \otimes\mu^{-1}(R_i) \subseteq ((\Sigma_\diamond)^{r_i})^*$.

Usually, μ is not required to be injective and the identity relation in A is included among the relations R_i and hence, as part of the automatic presentation, an automaton $M_=$ is given, which is able to recognize two different representations w_1, w_2 of the same element $a \in A$, that is: $L(M_=) = \{\otimes(w_1, w_2) : \mu(w_1) = \mu(w_2)\}$. Here, we opt to use the equivalent approach in which μ is injective and hence $M_=$ is a very simple automaton ($L(M_=) = \{\otimes(w, w) : w \in L\}$) and the elements of A can be identified with their representations in $L = L(M_A)$.

Automatic presentations, have very strong decidability properties; the following is a classic result:

Theorem 2.4 (Cor. 4.2 in [11])**.** *The first-order theory of any automatic structure is algorithmically decidable.*

Moreover, the previous theorem has been generalized beyond first-order logic, to include several generalized quantifiers, including $\exists^\infty, \exists^{(k,m)}$ and $\exists^{k-ram}$ [23]: The first quantifier specifies an infinite number of elements in the domain satisfying the proposition, while the second quantifier specifies a number of elements that is congruent with k modulo m; the third, $\exists^{k-ram}$, is the k-Ramsey quantifier, whose definition is beyond the scope of this paper. The previous theorem have even been extended to some fragments of second-order logic called FSO [12].

An *automatic graph* is a pair of automata $G = (M_V, M_E)$, where the alphabet of M_V is Σ and the alphabet of M_E is $(\Sigma_\diamond)^2$. The vertices and edges of G are then given by $V(G) = L(M_V)$ and $\otimes E(G) = L(M_E)$. Note that an automatic graph is then an automatic presentation of the graph $(V(G), E(G))$ with $\mu = 1_{V(G)}$, but we prefer to consider an automatic graph to be a graph on its own right. We denote by $\mathcal{AG}$ the class of all automatic graphs.

2.4. **Turing machines.** Informally (see Figure 2), a Turing machine is a finite control with a read/write head, which operates on a one-way infinite tape. The finite control is, at any given time, in some specific state q among a finite number of possibilities $q \in Q$. Depending on this state q and the symbol a which is directly under the read/write head (the symbols on the tape belong to the tape alphabet Γ), the finite control decides which is the new state p for the finite control, which new symbol b is to be written at the current head position and whether the read/write head moves to the left ($\leftarrow$) or to the right ($\rightarrow$). An one-step Turing transition is

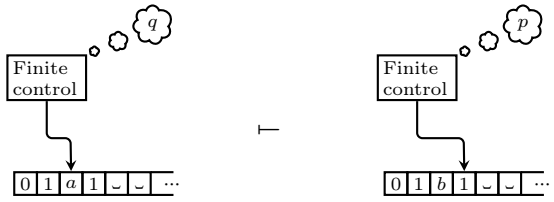

FIGURE 2. Turing machines and transitions.

denoted by the symbol $\vdash$ and it is illustrated in Figure 2, in the case where the head movement is to the right.

The decisions of the finite control are encoded in a (partial) transition function $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$. This function is *partial* since it is allowed to be undefined for some elements (q, a) of its domain $Q \times \Gamma$. A Turing machine always starts at an initial state q_0, with some input string w written on the leftmost part of its tape (the rest of the tape is filled with the blank symbol "␣") and with the read/write head at the leftmost cell of the tape. Then the Turing machine proceeds as dictated by δ until either it finds a pair (q, a) where δ is undefined or the read/write head falls off the left boundary of the tape; when any of these two conditions happens, the Turing machine *halts*. The *halting problem* consist in determining whether a given Turing machine M eventually halts on a given input string w. It is well known that the halting problem is algorithmically undecidable.

Formally a *Turing machine*, is a tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, ␣, F)$ where Q is the set of *states*, Σ is the *input alphabet* (the alphabet used to encode the input string w), Γ is the *tape alphabet* (the set of all symbols that can appear on the tape, $\Sigma \subseteq \Gamma$), $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ is the (partial) *transition function*, $q_0 \in Q$ is the *initial state*, $␣ \in \Gamma \setminus \Sigma$ is the *blank symbol* and $F \subseteq Q$ is the set of *final states*.

Figure 2 illustrates two "snapshots" of a Turing machine at two consecutive moments in time. These "snapshots" are called *configurations* or *instantaneous descriptions* of the Turing machine. It is standard practice to represent instantaneous descriptions by strings $s = uqv \in \Gamma^* Q \Gamma^*$ where q is the state of the machine, u is the content of the tape to the left of the head, and v is the content of the tape starting at the head and to the right of it (not considering the infinite tail of blank symbols). For instance, the two configurations in Figure 2 are represented by the strings $01qa1$ and $01bp1$ respectively. The *transition relation* among configurations is written as $s \vdash s'$. That is, we write $s \vdash s'$ whenever the configuration s transits in one step to the configuration s' according to the transition function δ of M. In our example, we can write $01qa1 \vdash 01bp1$ assuming that $\delta(q, a) = (p, b, \rightarrow)$. The transition relation is automatic:

Theorem 2.5 (Prop. 2.6 in [11], Lemma 5.12 in [2])**.** *For every Turing machine the configuration-transition relation $uqv \vdash u'q'v'$ is automatic.*

Here we shall use *timestamped configurations* $d = ts = tuqv \in \hat{\mathbb{N}}\Gamma^* Q \Gamma^*$, thus t is a binary string representing the elapsed time of a given computation. Hence the starting timestamped configuration of a Turing machine with the string w on its tape is $0q_0w$. We shall write $ts \vdash t's'$ (or $d \vdash d'$) whenever $s \vdash s'$ and $t' = t + 1$; in this case, we say that d' is a *successor* of d and that d is a *predecessor* of d'. Also, for $n \geq 0$, we shall write $d \overset{n}{\vdash} d_1$ whenever d transits to d_1 in n steps: $d \vdash \cdots \vdash d_1$; when

the value of n is not relevant, we simply write $d \overset{*}{\vdash} d_1$ (meaning $d \overset{n}{\vdash} d_1$ for some $n \geq 0$) in which case we say that d_1 is a *descendant* of d and that d is an *ancestor* of d_1. Note that M halts on input w if and only if $0qw \vdash d$ for some *halting configuration* d (that is, there is no configuration d_1 satisfying $d \vdash d_1$). Without loss of generality, we assume Q, Γ and $\{0,1\}$ to be mutually disjoint sets. This convention allows unambiguous identification of the several parts of a timestamped configuration $d = tuqv$.

Recall that $\mathcal{TM}$ denotes the class of all Turing machines and that $\mathcal{TMW}$ denotes the class of pairs (M,w), where w is an input string for $M \in \mathcal{TM}$. The class of all pairs (M,d) where d is a timestamped configuration of M is denoted by $\mathcal{TMD}$.

Observe that, since the transitions in a Turing machine M are specified by the transition function $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$, the number of *predecessors* of a given configuration d is always finite: $|\{d_0 : d_0 \vdash d\}| \leq |Q||\Gamma|$. Thanks to timestamps, the number of ancestors of a configuration d is also finite: If t is the timestamp of d, we have that $|\{d_0 : d_0 \overset{*}{\vdash} d\}| \leq 1 + |Q||\Gamma| + (|Q||\Gamma|)^2 + \cdots + (|Q||\Gamma|)^t$. On the other hand, the number of successors is either 0 or 1. In particular for any two descendants d_1, d_2 of d we have either $d_1 \overset{*}{\vdash} d_2$ or $d_2 \overset{*}{\vdash} d_1$.

2.5. **Clique graphs.** We identify induced subgraphs with their corresponding vertex sets, in particular we prefer to write $x \in G$ instead of $x \in V(G)$. The neighborhood of a vertex is denoted by $N(x)$ and the closed neighborhood by $N[x] = N(x) \cup \{x\}$. We also write $N_G(x)$, $N_G[x]$ when we want to emphasize the graph that we are considering. A vertex x is *dominated* if there is some vertex $y \neq x$ such that $N[x] \subseteq N[y]$. A *clique* of a graph, is a maximal complete subgraph. Given a family F of sets, the *intersection graph* $G = \Omega(F)$ *of* F is the graph having F as the vertex set ($V(G) = F$) and where two vertices $X, Y \in F$ are *adjacent-or-equal* ($X \simeq Y$) whenever $X \cap Y \neq \varnothing$. In clique graph theory, the adjacency-or-equality relation ($\simeq$) is usually more useful than adjacency relation ($\sim$). The *clique graph* $K(G)$ of a graph G is the intersection graph of its set of cliques (considered as sets, as per our convention that identifies induced subgraphs with their corresponding vertex sets). *Iterated clique graphs* are defined by $K^0(G) = G$ and $K^{n+1}(G) = K(K^n(G))$. A graph G is *clique-convergent* if $K^n(G) \cong K^m(G)$ for some $n < m$; it is *clique-divergent* otherwise. A graph is a *cone* if it contains an *apex*, which is a vertex adjacent to all other vertices. A graph G is *clique-Helly* if every collection of pairwise intersecting cliques has a nonempty total intersection; G is *quasi clique-Helly* if $G - x$ is clique-Helly for some vertex $x \in G$. Given a triangle $T = \{x, y, z\}$ of a graph G, its *extended triangle* is $\hat{T} = \{u \in G : |N(u) \cap T| \geq 2\}$. Although "apex" is defined for graphs, we can apply the concept to extended triangles as per our convention. Given a vertex $x \in G$, its *star* is $x^* = \{q \in K(G) : x \in q\}$. Stars may or may not be *cliques of cliques* of G (i.e. cliques of $K(G)$ and vertices of $K^2(G)$) since they may not be maximal. Cliques of cliques which are not stars are called *neckties*. The following lemma has not been explicitly stated in this way before, but all its statements are well known in the literature.

Lemma 2.6. (1) $N[x] = \cup x^*$.

(2) *For q a clique, $q \in x^*$ if and only if $q \subseteq N[x]$.*

(3) *Let Q be a clique of cliques of a graph G. If $x \in \cap Q$ for some x, then Q is a star $Q = x^*$, otherwise, it is a necktie and $\cap Q = \varnothing$.*

(4) *Given $x \in G$, x^* is always a complete subgraph of $K(G)$, and hence it is a clique of $K(G)$ if and only if it is maximal.*
(5) *Given $x^*, y^* \in K^2(G)$, they are different if and only if $N[x] \neq N[y]$.*
(6) *$x^* \subseteq y^*$ if and only if $N[x] \subseteq N[y]$.*
(7) *Given $x \in G$, x^* is a clique of cliques if and only if there is no necktie Q of G containing x^* and there is no $y \in G$ with $N[x] \subsetneq N[y]$.*
(8) *Given $x^*, y^* \in K^2(G)$, we have that $x^* \simeq y^*$ if and only if $x \simeq y$. Note however that it may be the case that $x \sim y$, $x \neq y$ and $x^* = y^*$.*

Proof. (1) $y \in N[x]$ if and only if $y \in q \in x^*$ for some q.
(2) Clearly $q \in x^*$ implies $q \subseteq N[x]$. Conversely, $q \subseteq N[x]$ implies $x \in q$ since q is maximal and x is an apex of $N[x]$. Hence $q \subseteq N[x]$ implies $q \in x^*$.
(3) If $x \in \cap Q$, then $x \in q$ for all $q \in Q$ and hence $Q \subseteq x^*$, the maximality of Q gives the equality. The last statement is obvious.
(4) For any $q, q' \in x^*$, we have $q \cap q' \supseteq \{x\} \neq \varnothing$ and hence $q \simeq q'$ in $K(G)$. It follows that x^* is (induce) a complete subgraph of $K(G)$. Then x^* only needs to be maximal to be a clique of cliques.
(5) Immediate from (2)
(6) By (1), $x^* \subseteq y^*$ implies $N[x] = \cup x^* \subseteq \cup y^* = N[y]$. Conversely, by (2) $N[x] \subseteq N[y]$ implies that $q \in x^* \Rightarrow q \subseteq N[x] \subseteq N[y] \Rightarrow q \in y^*$ and hence $x^* \subseteq y^*$.
(7) By (4) x^* is not a clique of cliques, only when it is not maximal. By (3) any clique of cliques containing x^* properly, must be either a necktie or a star. By (5) and (6) x^* is contained properly by y^* if and only if $N[x] \subsetneq N[y]$.
(8) $x \simeq y$ if and only if $x, y \in q$ for some q if and only if $q \in x^* \cap y^*$ if and only if $x^* \simeq y^*$.

□

We shall need the following two theorems:

Theorem 2.7 ([6, 24])**.** *If a graph G is clique-Helly without dominated vertices, then $G \cong K^2(G)$. Moreover, the isomorphism $* : G \to K^2(G)$ is given by $*(x) = x^*$.*

Theorem 2.8 ([5, 24, 25])**.** *A graph G is clique-Helly if and only if, for every triangle T of G, its extended triangle, $\hat{T}$, is a cone.*

Both theorems were originally stated only for finite graphs. But it is straightforward to verify that the standard proofs are valid for the infinite case: Theorem 2.7 is valid for any infinite graph, and Theorem 2.8 is valid for locally finite graphs (i.e. the degree of every vertex is finite).

3. Overview of the Proof of Theorem 1.2

Let us begin with an informal description of the proof of Theorem 1.2. A detailed proof is given in the next section.

The key idea is to reduce the halting problem to the clique-convergence problem by emulating Turing machine transitions with iterated clique graphs. Then we shall have that whenever the Turing machine halts, the corresponding sequence $G, K^2(G), K^4(G), \ldots$ stabilizes (i.e. $K^{2a}(G) \cong K^{2b}(G)$ for some $b < a$), and vice versa. This last condition is equivalent to clique-convergence, since, for the family of graphs that we use in the reduction, $K^n(G) \cong K^m(G)$ is only possible when n and m have the same parity.

The emulation is obtained by constructing a computable function $\gamma : \mathcal{TMD} \to \mathcal{AG}$ such that whenever $d \vdash d'$, we have $K^2(\gamma(M,d)) \cong \gamma(M,d')$ (the λ in Theorem 1.2 will be defined later). That is, γ is constructed to make the following diagram commute:

$$\begin{array}{ccc} \mathcal{TMD} & \xrightarrow{\vdash} & \mathcal{TMD} \\ \downarrow{\scriptstyle \gamma} & & \downarrow{\scriptstyle \gamma} \\ \mathcal{AG} & \xrightarrow{K^2} & \mathcal{AG} \end{array}$$

Now, take some Turing machine M, then for any two $(M,d_1),(M,d_2) \in \mathcal{TMD}$, $\gamma(M,d_1)$ and $\gamma(M,d_2)$ do not differ much, indeed we first construct a computable function $\gamma_0 : \mathcal{TM} \to \mathcal{AG}$ and then, $\gamma(M,d)$ is obtained from $\gamma_0(M)$ by the addition of at most one vertex (and its adjacencies).

Let us describe $\gamma_0(M)$. Take a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, \llcorner, F)$, and take the set of all timestamped configurations of M: $D = \{tuqw : t \in \mathtt{N}; u, v \in \Gamma^*; q \in Q\}$. Then $\vdash$ is an asymmetric binary relation on D and hence we can consider D to be a digraph with the adjacency relation given by $\vdash$. We construct $\gamma_0(M)$ by producing, for each $d \in D$, six vertices and seven edges as in Figure 3(a), in particular, the six vertices produced are labeled as $1d, 2d, \ldots, 6d$. Note that these labels are strings over the alphabet $\{0,1,\ldots,6\} \cup Q \cup \Gamma$. Also, for each arrow $d \vdash d'$ of D, we add 6 additional edges to $\gamma_0(M)$ as in Figure 3(b). Note that γ_0 maps a Turing machine M (which is a finite tuple) to an automatic presentation of a graph $\gamma_0(M)$ (which is a pair of finite automata), both of which are finite objects regardless of the infinity of D or the infinity of the graphs represented by $\gamma_0(M)$. When the full definitions of γ_0, γ and λ are presented and the related theorems considered, it will be clear that all these functions are algorithmically computable.

In Figure 3(d) we depict all the types of cliques of $\gamma_0(M)$: 5 (red) edges for each d and for each $d \vdash d'$, two blue triangles and one clique on 4 vertices.

As an example, the fragment of the digraph D depicted in Figure 3(c)(left) produces the fragment of $\gamma_0(M)$ shown in Figure 3(c)(right); there, for clarity, the vertices of the form $4d$, $5d$ and $6d$ are omitted from the drawing.

In what follows, we shall omit vertices of the form $4d$, $5d$ and $6d$ from drawings. The role of these vertices is crucial but minor: they are present to prevent the existence of dominated vertices (so we can use Theorem 2.7) and also to prevent undesired isomorphisms, (see the proof of Lemma 4.16).

Hence, paths in D like $d \vdash d' \vdash d'' \vdash \cdots \vdash d^{\mathtt{v}}$ produce *strips* in $\gamma_0(M)$, like that in Figure 5. It is known since Escalante [6] that these kind of strips are K^2-invariant, and that certain local perturbations on these strips, like the red vertex in Figure 4(a), are *traveling perturbations*, in the sense that the second clique graph of the graph in Figure 4(a) is isomorphic to the graph in Figure 4(b). Our strips are more complicated than those of Escalante since Escalante's were only circular strips, but we can have a lot of branching, and our strips may also have loose ends (for instance, when the path in D ends at $d^{\mathtt{v}}$, i.e. when $d^{\mathtt{v}}$ is a halting configuration of M), in this last case, the traveling perturbation simply disappears at the end of the corresponding strip. All of this requires a new detailed analysis of the clique graph transformation which we will do in Lemma 4.14.

We use the properties mentioned above to simulate Turing machine transitions: Just define $\gamma(M,d)$ as the graph obtained from $\gamma_0(M)$ by the addition of the (red) vertex $0d$ (and its edges) as in Figure 4(a). Thus the desired property is obtained:

$K^2(\gamma(M,d)) \cong \gamma(M,d')$. We shall prove that $\gamma(M,d)$ is clique-convergent if and only if M reaches a halting configuration when it starts from configuration d (see the proof of Lemma 4.18). The computability of γ is inherited from γ_0, since the automata describing γ_0 can be easily (and algorithmically) amended to obtain the automata for γ.

The final step is to define $\lambda : \mathcal{TMW} \to \mathcal{AG}$ by $\lambda(M,w) = \gamma(M, 0q_0w)$ and the result follows. The next section contains a detailed proof of Theorem 1.2.

4. Proof of Theorem 1.2

Take a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$. We define $V = \{1,2,3,4,5,6\}\cdot\mathtt{N}\cdot\Gamma^*\cdot Q\cdot\Gamma^*$. We shall use V as the set of vertices of the automatic graph $\gamma_0(M)$. Thus, a vertex $xd = xts = xtuqv \in V$ is the concatenation of a label $x \in \{1,2,\dots,6\}$ and a timestamped configuration $d \in D$. We also define $\hat{V} = \{1,2,3,4,5,6\}\cdot\hat{\mathtt{N}}\cdot\Gamma^*\cdot Q\cdot\Gamma^*$. Thanks to Theorem 2.1, both V and $\hat{V}$ are regular languages (and automatic unary relations). The only difference between V and $\hat{V}$ is that the timestamps $t \in \hat{\mathtt{N}}$ occurring in $xts \in \hat{V}$ may have an additional 0 at the end of the timestamp. When this additional 0 is present, some misalignment of the rest of the information occurs among equivalent timestamped configurations:

$$\otimes\binom{611abqabc}{6110abqabc} = \binom{6}{6}\binom{1}{1}\binom{1}{1}\binom{a}{0}\binom{b}{a}\binom{q}{b}\binom{a}{q}\binom{b}{a}\binom{c}{b}\binom{\diamond}{c}.$$

The automaton recognizing this is perhaps the trickiest of all the automata considered here and hence we present it explicitly:

Lemma 4.1. *The relation $T_0 = \{(xt_1s, xt_2s) \in \hat{V}^2 : t_2 = t_10\}$ is effectively automatic.*

Proof. Let us consider first the relation:

$$T_0' = \{(uv, u0v) : u \in \{0,1,\dots,6\}^*, v \in (\Gamma \cup Q)^*\}.$$

This relation is recognized by the automaton $M' = (Q', \Sigma', \delta', q_0', F')$, where $Q' = \{q_0', q_1', q_2'\} \cup \Gamma \cup Q$, $\Sigma' = \{0,1,\dots,6\} \cup \Gamma \cup Q$, $q_0' = q_0'$, $F = \{q_1'\}$, and the transition function δ' is given by the following rules:

(1) $\delta'(q_0', \binom{a}{a}) = q_0'$ for $a \in \{0,1,\dots,6\}$.
(2) $\delta'(q_0', \binom{b}{0}) = b$ for $b \in \Gamma \cup Q$.
(3) $\delta'(b, \binom{b'}{b}) = b'$ for $b, b' \in \Gamma \cup Q$.
(4) $\delta'(b, \binom{\diamond}{b}) = q_1'$ for $b \in \Gamma \cup Q$.
(5) $\delta'(*, *) = q_2'$ otherwise.

Indeed, note that q_2' is the dead state and that everything not conforming to our pattern goes there (rule 5) and is rejected. The state q_0' consumes input until the extra 0 is found (rule 1), when the extra 0 is found, it comes as $\binom{b}{0}$ for some $b \in \Gamma \cup Q$, at this point the automaton transits to the state b (rule 2). Here, it is crucial that $0 \notin \Gamma \cup Q$ and $b \notin \{0,1,\dots,6\}$ (according to our assumption), so that there is no confusion on whether to use rule 1, rule 2 or rule 5. This state b has the purpose of remembering the unmatched "b" that came with the extra 0. Hence, when reading the next symbol $\binom{b'}{b}$ it checks that the bottom part is indeed a b (otherwise the automaton transits to the dead state q_2') and then it transits to the new state b' (rule 3), waiting to match this b' with the bottom part of the next

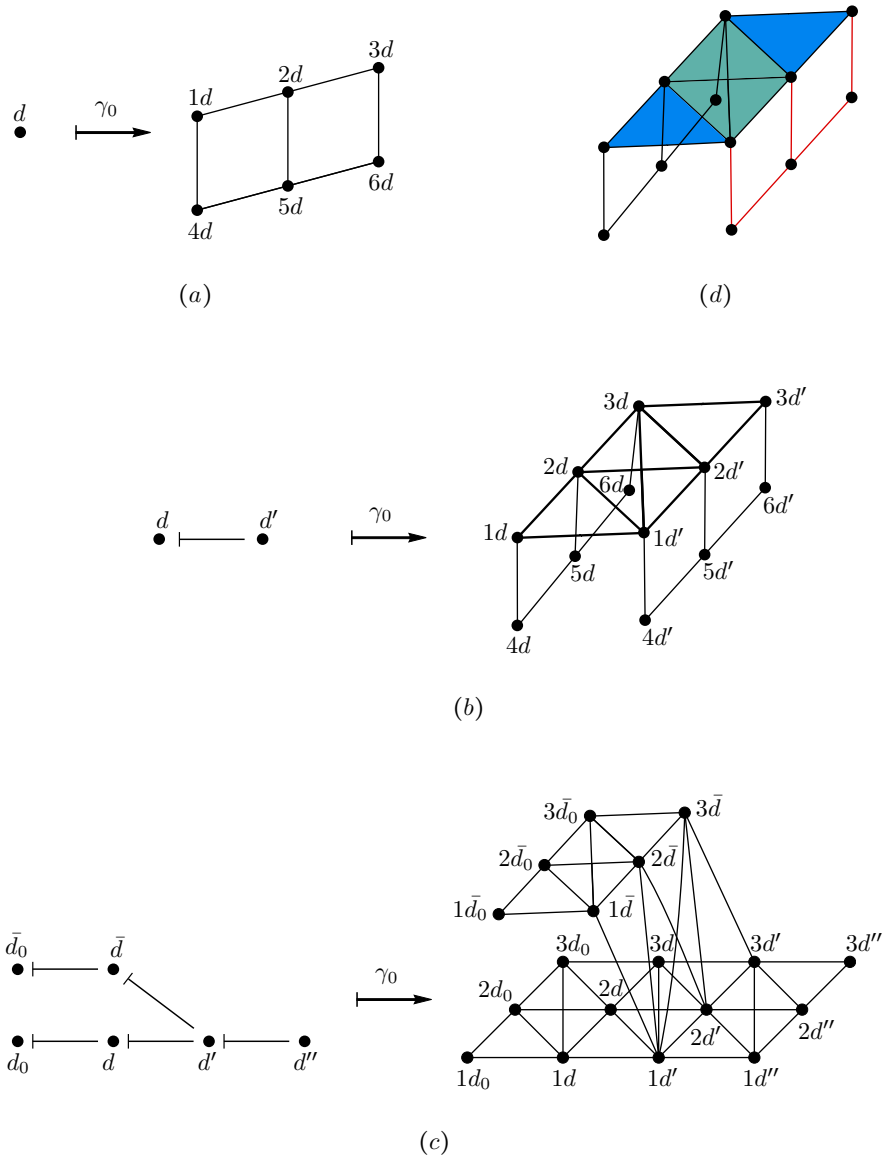

FIGURE 3. The construction of $\gamma_0(M)$. Vertices of the form $4d$, $5d$ and $6d$ are omitted in (c).

symbol. The automaton keeps using rule 3 until the end of input is found (rule 4) and then it enters the final state q_1' and accepts.

Recall that $\hat{V}$ is both a regular language and an automatic unary relation, so that "$u \in \hat{V}$" (viewing $\hat{V}$ as a set) is the same as "$\hat{V}(u)$" (viewing $\hat{V}$ as an unary relation). Now the required relation $T_0(u,v) = T_0'(u,v) \wedge u \in \hat{V} \wedge v \in \hat{V}$ is automatic by Theorem 2.3. It should be clear that all these constructions can be performed algorithmically from the original Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$. □

Definition 4.2. *We now define $\gamma_0 : \mathcal{TM} \to \mathcal{AG}$ (see Figure 3): The set of vertices of $\gamma_0(M)$ is $V(\gamma_0(M)) = V = \{1,2,3,4,5,6\}\cdot\mathtt{N}\cdot\Gamma^*\cdot Q\cdot\Gamma^*$. Two such vertices xd, x_1d_1 are adjacent in $\gamma_0(M)$ if and only if any of the following conditions holds:*

(1) *$d = d_1$ and $\{x, x_1\} \in \{\{1,2\},\{1,4\},\{2,3\},\{2,5\},\{3,6\},\{4,5\},\{5,6\}\}$*
(2) *$d \vdash d_1$ and $(x, x_1) \in \{(1,1),(2,1),(2,2),(3,1),(3,2),(3,3)\}$*
(3) *$d_1 \vdash d$ and $(x_1, x) \in \{(1,1),(2,1),(2,2),(3,1),(3,2),(3,3)\}$*

By Theorem 2.1, the set of vertices of $\gamma_0(M)$ is automatic. We show now that the set of edges of $\gamma_0(M)$ is also automatic:

Lemma 4.3. *Given a Turing machine M, $\gamma_0(M)$ is an automatic graph.*

Proof. We already know that the successor relation R_{+1} is automatic (when $t, t+1 \in \hat{\mathtt{N}}$ have the same length, see subsection 2.3) and that the transition relation $s \vdash s'$ is automatic for configurations $s = uqv, s' = u'q'v'$ by Theorem 2.5. Hence the transition relation is also automatic for timestamped configurations $d \vdash d'$ since regular languages (i.e. the convolutions of these relations) are effectively closed under concatenation by Theorem 2.1.

We have defined other automatic relations: V, $\hat{V}$, and T_0 (Lemma 4.1). Also, $\hat{V}^2$ is an automatic relation by Theorem 2.3 since $\hat{V}^2(w_1, w_2) = (w_1 \in \hat{V} \wedge w_2 \in \hat{V})$. $\hat{V}^2$ can be viewed as a set of pairs or as a relation $((w_1, w_2) \in \hat{V}^2 \Leftrightarrow \hat{V}^2(w_1, w_2))$. It should be clear now that all the following relations are automatic (in part because of Theorems 2.1 and 2.3, and in part because it is easy to give the corresponding automata):

$$T_= = \{(x_0t_0s_0, x_1t_1s_1) \in \hat{V}^2 : t_0 = t_1\},$$

$$D_= = \{(x_0d_0, x_1d_1) \in \hat{V}^2 : d_0 = d_1\},$$

$$D_\vdash = \{(x_0t_0s_0, x_1t_1s_1) \in \hat{V}^2 : t_0s_0 \vdash t_1s_1 \text{ and } |t| = |t+1|\},$$

$$E_0 = \left\{(x_0d_0, x_1d_1) \in \hat{V}^2 : \{x_0, x_1\} \in \left\{ \begin{array}{l} \{1,2\},\{1,4\},\{2,3\},\{2,5\}, \\ \{3,6\},\{4,5\},\{5,6\} \end{array} \right\}\right\},$$

$$E_1 = \left\{(x_0d_0, x_1d_1) \in \hat{V}^2 : (x_0, x_1) \in \left\{ \begin{array}{l} (1,1),(2,1),(2,2), \\ (3,1),(3,2),(3,3) \end{array} \right\}\right\},$$

$$\overrightarrow{E} = (D_= \wedge E_0) \vee (D_\vdash \wedge E_1),$$

$$E(w_1, w_2) = (w_1 \in V) \wedge (w_2 \in V) \wedge \exists w_1' \exists w_2' \left(\begin{array}{l} (T_=(w_1, w_1') \vee T_0(w_1, w_1')) \wedge \\ (T_=(w_2, w_2') \vee T_0(w_2, w_2')) \wedge \\ (\overrightarrow{E}(w_1', w_2') \vee \overrightarrow{E}(w_2', w_1')) \end{array} \right).$$

This last relation is precisely the adjacency relation in $\gamma_0(M)$ and hence the result follows. □

It should be clear that the required automata for $\gamma_0(M)$ can be algorithmically constructed from the given Turing machine:

Remark 4.4. *$\gamma_0 : \mathcal{TM} \to \mathcal{AG}$ is a computable function.*

Definition 4.5. *A timestamped configuration d is* non-terminal *if there are $d', d_1 \neq d$ such that $d \vdash d'$ and that either $d' \vdash d_1$ or $d_1 \vdash d'$. It is* terminal *otherwise.*

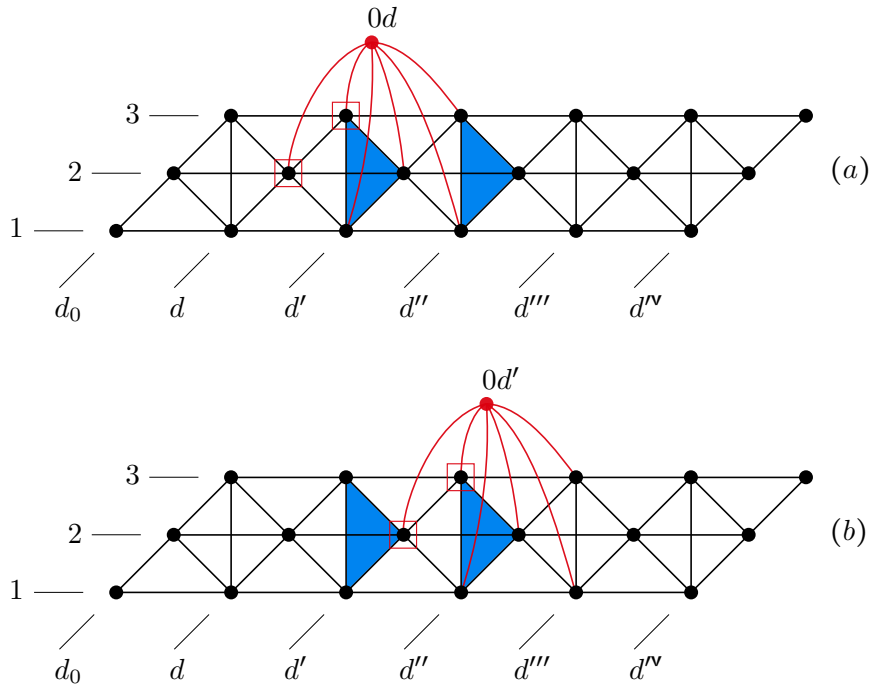

FIGURE 4. $\gamma(M,d)$ and $\gamma(M,d')$. Vertices of the form $4d$, $5d$ and $6d$ are omitted from the drawings.

Note that if d is terminal then either d or d' (when $d \vdash d'$) is a halting time-stamped configuration. Also, every halting timestamped configuration d is terminal.

We can now define $\gamma : \mathcal{TMD} \to \mathcal{AG}$ (see Figure 4(a)): $\gamma(M,d)$ is almost the same as $\gamma_0(M)$. In fact, we define $\gamma(M,d) := \gamma_0(M)$ whenever d is terminal. Otherwise, $\gamma(M,d)$ is obtained from $\gamma_0(M)$ by the addition of exactly one vertex (namely $0d$) and its adjacencies:

Definition 4.6. *Take $(M,d) \in \mathcal{TMD}$. If d is terminal, define $\gamma(M,d) := \gamma_0(M)$. Otherwise ($d$ is non-terminal), define $V(\gamma(M,d)) := V(\gamma_0(M)) \cup \{0d\}$; in this case, there is some d' with $d \vdash d'$, the adjacencies in $\gamma(M,d)$ are the same as in $\gamma_0(M)$, except that the neighbors of $0d$ are given by (if there is no d'' with $d' \vdash d''$, simply remove $1d''$ from the set):*

$$N(0d) := \{x\bar{d} : x \in \{2,3\}, \bar{d} \vdash d'\} \cup \{1d', 2d', 3d', 1d''\}.$$

The red squares in Figure 4 are there to remind us that multiple vertices like those may be present in $N(0d)$ whenever $|\{\bar{d} : \bar{d} \vdash d'\}| > 1$. Note that $N(0d)$ is exactly the extended triangle of $\{3d, 1d', 2d'\}$ in $\gamma_0(M)$ (the left blue triangle in Figure 4(a)). Also, note that $2d'$ is always an apex of this extended triangle in $\gamma_0(M)$. For future reference, we now state the following:

Remark 4.7. *If d is terminal, then $\gamma(M,d) = \gamma_0(M)$. If d is non-terminal, then $\gamma(M,d)$ is obtained from $\gamma_0(M)$, by the addition of exactly one vertex (namely: $0d$) and its edges as in Figure 4(a).*

The amendments made to $\gamma_0(M)$ to obtain $\gamma(M,d)$ are finite (one vertex and a finite number of edges). By Theorem 2.1 and Theorem 2.3 finite relations are

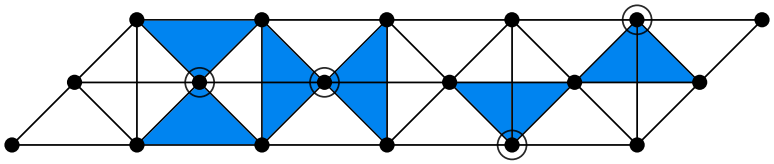

FIGURE 5. The types of triangles of $\gamma_0(M)$. Vertices of the form $4d$, $5d$ and $6d$ are omitted from the drawing.

effectively automatic and disjunctions of automatic relations are also effectively automatic, hence $\gamma(M,d)$ is effectively automatic and we have:

Remark 4.8. *$\gamma : \mathcal{TMD} \to \mathcal{AG}$ is a computable function.*

Note that, when present, $0d$ is always dominated by $2d'$ (for d' satisfying $d \vdash d'$) but no other vertex of $\gamma(M,d)$ is dominated as we shall see in the next lemma. Recall that the star morphism $* : G \to K^2(G)$ is given by $*(x) := x^*$.

Lemma 4.9. *$\gamma_0(M)$ is clique-Helly without dominated vertices. In particular we have that the star morphism $* : \gamma_0(M) \to K^2(\gamma_0(M))$ is an isomorphism and that, for all $n \geq 0$, $K^{2n}(\gamma_0(M)) \cong \gamma_0(M)$.*

Proof. By Theorems 2.7 and 2.8, it is sufficient to show that in $\gamma_0(M)$ every extended triangle has an apex and that there are no dominated vertices.

By construction (see Figure 3), every triangle of $\gamma_0(M)$, is contained in a set of the form $\{xd_1 : x \in \{1,2,3\} \text{ and } d_1 \in \{d,d'\}\}$ for some d, d' with $d \vdash d'$. It follows that all triangles of $\gamma_0(M)$ are of one the following forms:

$$\{1d, 2d, 1d'\}, \quad \{2d, 3d, 1d'\}, \quad \{2d, 3d, 2d'\},$$
$$\{2d, 1d', 2d'\}, \quad \{3d, 1d', 2d'\}, \quad \{3d, 2d', 3d'\}.$$

By direct inspection, we can see that the apices of the corresponding extended triangles are $2d$, $2d$, $3d$, $1d'$, $2d'$ and $2d'$ respectively: In Figure 5, for each blue triangle, the apex of its corresponding extended triangle have been marked with a circle (some extended triangles share apex). Note that these vertices are apices of their extended triangles regardless of the multiplicity of predecessors of d and d' and regardless of the existence of a successor of d'. Hence by Theorem 2.8, $\gamma_0(M)$ is clique-Helly.

There are no dominated vertices in $\gamma_0(M)$ since if we had $xd_1 \neq yd_2$ and $N[xd_1] \subseteq N[yd_2]$, then $N(xd_1)$ would be a cone and hence connected. However (thanks to the vertices of the form $4d, 5d$ and $6d$) every vertex of $\gamma_0(M)$ has a disconnected neighborhood.

It follows by Theorem 2.7 that $K^{2n}(\gamma_0(M)) \cong \gamma_0(M)$. □

Recall that the number of predecessors of any timestamped configuration is at most $|Q||\Gamma|$ and that the number of successors is at most one. Since $\gamma(M,d)$ is obtained from $\gamma_0(M)$ by adding (at most) one vertex and a finite number of edges, we have:

Remark 4.10. *$\gamma(M,d)$ is quasi clique-Helly with at most one dominated vertex and of bounded degree.*

Lemma 4.11. *If $d \overset{*}{\vdash} d_1$ and $d \overset{*}{\vdash} d_2$, with $d_1 \neq d_2$ and d_1 non-terminal, then*

$$\gamma(M, d_1) \ncong \gamma(M, d_2).$$

Proof. By way of contradiction, assume there is such an isomorphism $\alpha : \gamma(M, d_1) \to \gamma(M, d_2)$. Since d_1 is non-terminal, $0d_1$ is a vertex of $\gamma(M, d_1)$ and it is the only dominated vertex of $\gamma(M, d_1)$. Since dominated vertices go onto dominated vertices under any isomorphism, it follows that $0d_2$ must also be a vertex $\gamma(M, d_2)$ and that $\alpha(0d_1) = 0d_2$.

Now, take d_1' and d_2' such that $d_1 \vdash d_1'$ and $d_2 \vdash d_2'$ and observe that the vertex $2d_1' \in \gamma(M, d_1)$ is the only apex of $N(0d_1)$ and that $2d_2' \in \gamma(M, d_2)$ is the only apex of $N(0d_2)$; it follows that $\alpha(2d_1') = 2d_2'$.

Also, $\{2\bar{d}_1 : \bar{d}_1 \vdash d_1'\}$ is exactly the set of vertices of $N(0d_1)$ which have degree 3 (in $N(0d_1)$) and belong to a complete on four vertices (and something analogous occurs in $\gamma(M, d_2)$). It follows that the set $\{2\bar{d}_1 : \bar{d}_1 \vdash d_1'\}$ is bijectively mapped by α onto $\{2\bar{d}_2 : \bar{d}_2 \vdash d_2'\}$. Similarly, whenever $\bar{d}_1$ is an ancestor of d_1', we must have $\alpha(2\bar{d}_1) = 2\bar{d}_2$ for some ancestor $\bar{d}_2$ of d_2', i.e. that $\{2\bar{d}_1 : \bar{d}_1 \overset{*}{\vdash} d_1'\}$ is bijectively mapped by α onto $\{2\bar{d}_2 : \bar{d}_2 \overset{*}{\vdash} d_2'\}$.

Since $d \overset{*}{\vdash} d_1 \vdash d_1'$, $d \overset{*}{\vdash} d_2 \vdash d_2'$ and $d_1 \neq d_2$, it follows that $d_1' \neq d_2'$ and that either $d_2' \overset{*}{\vdash} d_1'$ or $d_1' \overset{*}{\vdash} d_2'$, assume the latter without loss of generality. It follows that the (finite) number of ancestors of d_1' is less than the (also finite) number of ancestors of d_2'. This is a contradiction with the last statement in the previous paragraph. □

It is convenient to recall that all types of cliques of $\gamma_0(M)$ are depicted in Figure 3(d). The cliques of $\gamma(M, d)$ are mostly the same, and the few differences can be read from Figure 4. The explicit list of all these cliques is given in Table 1. In the table, the condition "$d_1 \vdash d_2$" means "$\exists d_1 \exists d_2,\ d_1 \vdash d_2$" , also, just to be explicit about it, "$d_1 \nvdash d_2$" means "$\neg(\exists d_1 \exists d_2,\ d_1 \vdash d_2)$". Lemma 4.12 shows that Table 1 is correct:

Lemma 4.12. *All the cliques of $\gamma_0(M)$ are listed in Table 1(top) parameterized by a generic configuration d. The cliques of $\gamma(M, d)$ are mostly the same and the variations are all listed in Table 1(bottom).*

Proof. Let us study the cliques of $\gamma_0(M)$ first. Consider any generic timestamped configuration d. By construction, it produces 6 vertices and 7 edges in $\gamma_0(M)$ as shown in Figure 3(a).

Note that the edges $e_{1d} := \{1d, 4d\}$, $e_{2d} := \{2d, 5d\}$, $e_{3d} := \{3d, 6d\}$, $e_{4d} := \{4d, 5d\}$, $e_{5d} := \{5d, 6d\}$ are always cliques. The edges $\overline{e}_{1d} := \{1d, 2d\}$, $\overline{e}_{2d} := \{2d, 3d\}$ are cliques only when d is an *isolated configuration*, that is, when there is no d_0 with $d_0 \vdash d$ and there is no d' with $d \vdash d'$.

Now, assume, d is not an isolated configuration. Then either $d_0 \vdash d$ or $d \vdash d'$ (or both) for some d_0, d'. Without loss of generality (renaming the configurations if necessary), we may assume $d \vdash d'$. By construction, every clique of $\gamma_0(M)$ (except $\overline{e}_{1d}$ and $\overline{e}_{2d}$ mentioned above) must be contained in a set of the form $\{xd_1 : x \in \{1, 2, \ldots, 6\}, d_1 \in \{d, d'\}\}$ for some d, d' with $d \vdash d'$. Figure 3(d) shows the three possible types of cliques which are not edges. We give them a name: $q_{1d} := \{1d, 2d, 1d'\}$, $q_{2d} := \{2d, 3d, 1d', 2d'\}$ and $q_{3d} := \{3d, 2d', 3d'\}$.

These are all the cliques of $\gamma_0(M)$ as indicated in Table 1 (top).

Now fix some d, and let us consider the cliques of $\gamma(M,d)$. By definition of $\gamma(M,d)$, it is identical to $\gamma_0(M)$ when d is terminal. Hence, assume from now on that d is not terminal.

Since only one vertex, $0d$, has been added to $\gamma_0(M)$, the cliques of $\gamma(M,d)$ are mostly the same as those of $\gamma_0(M)$, except *around* $0d$.

Around $0d$, we can see the differences in Figure 4(a). Recall that $N(0d) = \{x\bar{d} : x \in \{2,3\}, \bar{d} \vdash d'\} \cup \{1d', 2d', 3d', 1d''\}$ and that the red squares in Figure 4(a) are there to remind us that there may be other relevant vertices there when there are configurations $\bar{d}$ with $d \neq \bar{d} \vdash d'$.

Assume first that $d \vdash d' \vdash d''$. Then, we see in Figure 4(a) that three types of cliques of $\gamma_0(M)$ are extended with the addition of $0d$, namely: $q_{2\bar{d}}$ and $q_{3\bar{d}}$ whenever $\bar{d} \vdash d'$ (this includes d as one of those $\bar{d}$), and also the clique $q_{1d'}$, hence the *extended cliques* are:

$$\begin{aligned}
\hat{q}_{2\bar{d}} &:= q_{2\bar{d}} \cup \{0d\} = \{2\bar{d}, 3\bar{d}, 1d', 0d, 2d'\} \quad \forall \bar{d},\, \bar{d} \vdash d'.\\
\hat{q}_{3\bar{d}} &:= q_{3\bar{d}} \cup \{0d\} = \{3\bar{d}, 0d, 2d', 3d'\} \quad \forall \bar{d},\, \bar{d} \vdash d'.\\
\hat{q}_{1d'} &:= q_{1d'} \cup \{0d\} = \{1d', 0d, 2d', 1d''\}.
\end{aligned}$$

Also note that the triangle $\{2d', 3d', 1d''\}$ is not a clique of $\gamma_0(M)$ (since it is properly contained in $q_{2d'}$), but all of these vertices are adjacent to $0d$. Hence we have that this is a *new clique* of $\gamma(M,d)$:

$$q_{0d} := \{0d, 2d', 3d', 1d''\}.$$

We know that d is non-terminal, hence in the case when we do not have $d \vdash d' \vdash d''$, it is necessary, by the definition of "non-terminal" that $d \vdash d'$ and $\bar{d} \vdash d'$ for some d' and $\bar{d} \neq d$ (and $d' \nvdash d''$ for all d''). In this case, the vertex $1d''$ does not exist and hence, the above mentioned cliques $\hat{q}_{1d'}$ and q_{0d} also do not exist, since when we remove the vertex $1d''$ from these cliques the remaining vertices are contained in the cliques $\hat{q}_{2d}$ and $\hat{q}_{3d}$ respectively. Therefore in this case, only the extended cliques $\hat{q}_{2\bar{d}}$ and $\hat{q}_{3\bar{d}}$ prevail.

All of this is reported in Table 1(bottom). □

Now, we need to characterize the neckties of $\gamma(M,d)$:

Lemma 4.13. *$\gamma(M,d)$ has exactly one necktie when $d \vdash d'$ and d' is non-terminal. Otherwise $\gamma(M,d)$ has no neckties. The unique necktie of $\gamma(M,d)$ is given by*

$$Q_{0d} = \{q_{2\bar{d}'} : \bar{d}' \vdash d''\} \cup \{q_{0d}, q_{3d'}, q_{1d''}\} \quad \textit{when } d' \vdash d'' \vdash d''', \textit{ or}$$
$$Q_{0d} = \{q_{2\bar{d}'} : \bar{d}' \vdash d''\} \cup \{q_{0d}, q_{2d'}, q_{3d'}\} \quad \textit{when } d' \vdash d'' \nvdash d''' \textit{ and } d' \neq \bar{d}' \vdash d''.$$

Proof. The reader could find it convenient to follow the steps of this proof on Figure 4.

Let Q be a necktie of $\gamma(M,d)$, that is, $Q \in K^2(\gamma(M,d))$ and $\cap Q = \varnothing$. Since $\gamma_0(M)$ is clique-Helly it does not have any neckties (by Theorem 2.7), therefore Q must contain a clique q that contains the vertex $0d$: $0d \in q \in Q$. According to Table 1(bottom), the cliques containing $0d$ are $0d^* \subseteq \{\hat{q}_{2\bar{d}}, \hat{q}_{3\bar{d}} : \bar{d} \vdash d'\} \cup \{\hat{q}_{1d'}, q_{0d}\}$. Note that every clique containing $0d$ also contains $2d'$, moreover, for every $q \in K(\gamma(M,d))$ with $q \neq q_{0d}$ we have $(q - \{0d\}) \in K(\gamma_0(M))$. It follows that, whenever $q, q' \in K(\gamma(M,d))$ with $q, q' \neq q_{0d}$ and $q \cap q' \neq \varnothing$, we have also that $(q - \{0d\}) \cap (q' - \{0d\}) \neq \varnothing$.

Cliques of $\gamma_0(M)$ associated with a generic d	Required conditions
$q_{1d} \coloneqq \{1d, 2d, 1d'\}$ $q_{2d} \coloneqq \{2d, 3d, 1d', 2d'\}$ $q_{3d} \coloneqq \{3d, 2d', 3d'\}$	$d \vdash d'$
$\overline{e}_{1d} \coloneqq \{1d, 2d\}$, $\overline{e}_{2d} \coloneqq \{2d, 3d\}$.	$d_0 \nvdash d \nvdash d'$
$e_{1d} \coloneqq \{1d, 4d\}$, $e_{2d} \coloneqq \{2d, 5d\}$, $e_{3d} \coloneqq \{3d, 6d\}$, $e_{4d} \coloneqq \{4d, 5d\}$, $e_{5d} \coloneqq \{5d, 6d\}$.	None

Extended and new cliques of $\gamma(M, d)$		Required conditions
$\hat{q}_{2\bar{d}} \coloneqq \{2\bar{d}, 3\bar{d}, 1d', 0d, 2d'\}$	$\forall \bar{d},\ \bar{d} \vdash d'$	d is non-terminal
$\hat{q}_{3\bar{d}} \coloneqq \{3\bar{d}, 0d, 2d', 3d'\}$	$\forall \bar{d},\ \bar{d} \vdash d'$	
$\hat{q}_{1d'} \coloneqq \{1d', 0d, 2d', 1d''\}$		$d \vdash d' \vdash d''$
$q_{0d} \coloneqq \{0d, 2d', 3d', 1d''\}$		

TABLE 1. Cliques of $\gamma_0(M)$ and of $\gamma(M, d)$ and the conditions for their existence.

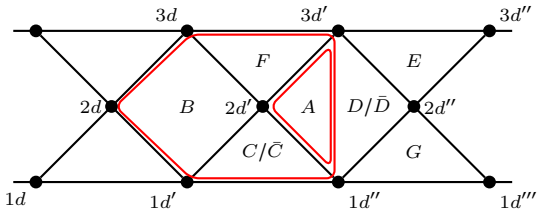

FIGURE 6. Cliques of $\gamma(M, d)$ adjacent to q_{0d}.

Assume first that $q_{0d} \notin Q$. Then $P \coloneqq \{(q - \{0d\}) : q \in Q\}$ is a set of mutually intersecting cliques of $\gamma_0(M)$. Since $\gamma_0(M)$ does not have neckties (by Theorem 2.7 and Lemma 4.9), it follows that $\cap P \neq \varnothing$, and hence $\cap Q \neq \varnothing$ also, a contradiction. If follows that $q_{0d} \in Q$.

The other cliques of $\gamma(M, d)$ intersecting q_{0d} are: $\hat{q}_{2\bar{d}}$ and $\hat{q}_{3\bar{d}}$ (for each $\bar{d} \vdash d'$); $\hat{q}_{1\bar{d}'}$ and $q_{2\bar{d}'}$ (for each $\bar{d}' \vdash d''$); and also $q_{3d'}$ and $q_{1d''}$. To improve the readability of the subsequent case analysis, we prepared the following propositions (see Figure 6):

A: $q_{0d} \in Q$. B: $\hat{q}_{2\bar{d}} \in Q$ for some $\bar{d} \vdash d'$.

C: $\hat{q}_{1d'} \in Q$. $\overline{C}$: $\hat{q}_{1\bar{d}'} \in Q$ for some $\bar{d}' \vdash d''$ with $\bar{d}' \neq d'$.

D: $q_{2d'} \in Q$. $\overline{D}$: $q_{2\bar{d}'} \in Q$ for some $\bar{d}' \vdash d''$ with $\bar{d}' \neq d'$.

E: $q_{3d'} \in Q$. F: $\hat{q}_{3\bar{d}} \in Q$ for some $\bar{d} \vdash d'$.

G: $q_{1d''} \in Q$.

Then, we already know that A, and we have that $A \wedge F \wedge E$ implies $3d' \in \cap Q$ (a contradiction) since any other clique intersecting $q_{0d}, \hat{q}_{3\bar{d}}$ and $q_{3d'}$, also contains

$3d'$. Similarly we have:

$$A \wedge F \wedge (B \vee C) \Rightarrow 2d' \in \cap Q.$$

Therefore we conclude that $\neg F$ (since we know that A and since, by the maximality of Q, we have that $A \wedge F \Rightarrow (A \wedge F \wedge E) \vee (A \wedge F \wedge (B \vee C)) \Rightarrow (3d' \in \cap Q) \vee (2d' \in \cap Q)$ a contradiction). Moreover:

$$A \wedge B \Rightarrow 2d' \in \cap Q.$$

$$A \wedge C \Rightarrow 2d' \in \cap Q \vee 1d'' \in \cap Q.$$

$$A \wedge \overline{C} \Rightarrow 1d'' \in \cap Q.$$

It follows that $\neg(F \vee B \vee C \vee \overline{C})$. Now, we also observe that:

$$\neg(\overline{D} \vee G) \Rightarrow 3d' \in \cap Q \text{ and}$$

$$\neg E \Rightarrow 1d'' \in \cap Q.$$

We conclude that $A \wedge E \wedge (\overline{D} \vee G)$ and also, by the maximality of Q, that $A \wedge E \wedge D \wedge (\bar{D} \vee G)$. This last proposition characterizes Q and hence we have that:

$Q_{0d} = \{q_{2\bar{d'}} : \bar{d'} \vdash d''\} \cup \{q_{0d}, q_{3d'}, q_{1d''}\}$ when $d' \vdash d'' \vdash d'''$, or

$Q_{0d} = \{q_{2\bar{d'}} : \bar{d'} \vdash d''\} \cup \{q_{0d}, q_{2d'}, q_{3d'}\}$ when $d' \vdash d'' \nvdash d'''$ and $d' \neq \bar{d'} \vdash d''$,

as claimed. Finally, note that the above conditions for the existence of Q are equivalent to: d' is non-terminal. The precondition $d \vdash d'$ is necessary even for $0d$ to exist. □

Lemma 4.14. *$d \vdash d'$ implies $K^2(\gamma(M,d)) \cong \gamma(M,d')$. Otherwise ($d \nvdash d'$ for all d'), $K^2(\gamma(M,d)) \cong \gamma_0(M) = \gamma(M,d)$.*

Proof. When $d \nvdash d'$ for all d', d is terminal and then, by definition, $\gamma(M,d) = \gamma_0(M)$. By Lemma 4.9 we have $\gamma(M,d) = \gamma_0(M) \cong K^2(\gamma_0(M)) = K^2(\gamma(M,d))$ as claimed.

When $d \vdash d'$, but d is still terminal, so is d' and hence $\gamma(M,d') = \gamma_0(M) \cong K^2(\gamma_0(M)) = K^2(\gamma(M,d))$ as claimed.

Now suppose that $d \vdash d'$, d non-terminal, but d' is terminal. Now the vertex $0d$ does exist, but, by Lemma 4.13, Q_{0d} does not. Hence $K^2(\gamma(M,d)) \cong \gamma_0(M) = \gamma(M,d')$ as claimed.

Assume from now on, that $d \vdash d'$ and that d' is non-terminal (then d is also non-terminal). Observe that the vertex $0d$ is always dominated by $2d'$. By Lemma 4.13, there is exactly one necktie Q_{0d}. Note that this necktie does not contain any star xd_1^*. By Lemma 2.6(7), it follows that every xd_1^* (with $xd_1 \neq 0d$) is a clique of cliques of $\gamma(M,d)$, and since $0d$ is the only dominated vertex (by Remark 4.10), all these stars xd_1^* are different by Lemma 2.6(5). Moreover, by Lemma 2.6(8), all the stars induce in $K^2(\gamma(M,d))$ a subgraph isomorphic to $\gamma_0(M)$. It follows that $K^2(\gamma(M,d))$ can be obtained (up to isomorphism) by adding exactly one vertex (namely Q_{0d}) and its adjacencies to $\gamma_0(M)$. Now (see Figure 4) note that $\cup Q_{0d} = \{x\bar{d'} : x \in \{2,3\}, \bar{d'} \vdash d''\} \cup \{0d, 1d'', 2d'', 3d'', 1d'''\} = N_{\gamma(M,d')}(0d') \cup \{0d\}$ (if the vertex $1d'''$ is not present simply drop it from the formula). Then, for $xd_1 \neq 0d$ the adjacencies of Q_{0d} in $K^2(\gamma(M,d))$ are given by $Q_{0d} \sim xd_1^* \Leftrightarrow \exists q,\ q \in Q_{0d} \cap xd_1^*$ $\Leftrightarrow \exists q,\ xd_1 \in q \in Q_{0d} \Leftrightarrow xd_1 \in \cup Q_{0d} \Leftrightarrow xd_1 \in N_{\gamma(M,d')}(0d') \Leftrightarrow xd_1 \sim 0d'$ in $\gamma(M,d')$. Hence, it follows that $K^2(\gamma(M,d)) \cong \gamma(M,d')$. □

Lemma 4.15. *For all d and $a \geq 0$ there is some d_1 with $d \overset{*}{\vdash} d_1$ such that $K^{2a}(\gamma(M,d)) \cong \gamma(M,d_1)$. Moreover, when d_1 is non-terminal, we have $d \overset{a}{\vdash} d_1$ and when it is terminal, we have $d \overset{b}{\vdash} d_1$ for some $b \leq a$.*

Proof. If there is a d_1 such that $d \overset{a}{\vdash} d_1$ then, by iterated application of Lemma 4.14, we have $K^{2a}(\gamma(M,d)) \cong \gamma(M,d_1)$. Note that in this case, d_1 may or may not be terminal.

Otherwise, there is some $b < a$ and some d_1 with $d \overset{b}{\vdash} d_1$ and such that $d_1 \nvdash d_2$ for any d_2. In this case d_1 is terminal and $\gamma(M,d_1) = \gamma_0(M)$ by definition of $\gamma(M,d_1)$, (Definition 4.6). Hence, by Lemmas 4.14 and 4.9 we have $K^{2a}(\gamma(M,d)) = K^{2a-2b}(K^{2b}(\gamma(M,d))) = K^{2a-2b}(\gamma(M,d_1)) \cong K^{2a-2b}(\gamma_0(M)) \cong \gamma_0(M) = \gamma(M,d_1)$. □

Lemma 4.16. *If $K^n(\gamma(M,d)) \cong K^m(\gamma(M,d))$ for some $n < m$, then $K^{2b}(\gamma(M,d)) \cong K^{2a}(\gamma(M,d))$ for some $b < a$.*

Proof. By Lemma 4.15, for every natural number c, there is some d_1 such that $d \overset{*}{\vdash} d_1$ and $K^{2c}(\gamma(M,d)) \cong \gamma(M,d_1)$. This graph have many vertices whose neighborhoods are isomorphic to an edgeless graph on 3 vertices (those of the form $5d_2 \in V(\gamma(M,d_1))$. A direct examination of the cliques of $\gamma(M,d_1)$ (see Figure 3(d), also consider the extra clique $q_{0d} = \{0d, 2d', 3d', 1d''\}$ in Figure 4) and their intersections, shows that $K^{2c+1}(\gamma(M,d)) \cong K(\gamma(M,d_1))$ does not have such vertices. It follows that $K^n(\gamma(M,d)) \cong K^m(\gamma(M,d))$ is only possible when n and m have the same parity. Therefore, $K^{2b}(\gamma(M,d)) \cong K^{2a}(\gamma(M,d))$ for either $2b = n < m = 2a$ or $2b = n+1 < m+1 = 2a$. □

Lemma 4.17. *If $K^{2a}(\gamma(M,d)) \cong K^{2b}(\gamma(M,d))$ for some $b < a$ then $K^{2a}(\gamma(M,d)) \cong \gamma_0(M) \cong K^{2b}(\gamma(M,d))$.*

Proof. By Lemma 4.15, there are d_1 and d_2 with $d \overset{*}{\vdash} d_1$, $d \overset{*}{\vdash} d_2$ and such that $K^{2a}(\gamma(M,d)) \cong \gamma(M,d_1)$ and $K^{2b}(\gamma(M,d)) \cong \gamma(M,d_2)$.

By Lemma 4.14 if d_1 or d_2 was terminal, we would have either $\gamma(M,d_1) = \gamma_0(M)$ or $\gamma(M,d_2) = \gamma_0(M)$ and the result follows.

Assume now that d_1 and d_2 are non-terminal, then by Lemma 4.15, we have $d \overset{a}{\vdash} d_1$ and $d \overset{b}{\vdash} d_2$. These d_1 and d_2, can not be equal, because the timestamps must be different: if $d = ts$, $d_1 = t_1 s_1$ and $d_2 = t_2 s_2$ then we must have $t_1 = t + a > t + b = t_2$. Hence, by Lemma 4.11, we get $K^{2a}(\gamma(M,d)) \cong \gamma(M,d_1) \ncong \gamma(M,d_2) \cong K^{2b}(\gamma(M,d))$, a contradiction. □

Lemma 4.18. *M halts on input w if and only if $\lambda(M,w) := \gamma(M, 0q_0w)$ is clique-convergent.*

Proof. Recall that the starting configuration for M on input w is $0q_0w$ (subsection 2.4). We shall show that the following propositions are equivalent:

(1) M halts on input w.
(2) $0q_0w \overset{*}{\vdash} d$ for some terminal d.
(3) $K^{2b}(\gamma(M, 0q_0w)) \cong \gamma_0(M)$ for some $b \geq 0$.
(4) $K^{2a}(\gamma(M, 0q_0w)) \cong K^{2b}(\gamma(M, 0q_0w))$ for some $b < a$.
(5) $\lambda(M,w) := \gamma(M, 0q_0w)$ is clique-convergent.

(1)⇔(2) By definition, M halts on input w if and only if $0q_0w \overset{*}{\vdash} d$ for some halting (timestamped) configuration d (subsection 2.4). By definition of terminal configuration (Definition 4.5), every halting configuration d is terminal, and every terminal configuration is either a halting configuration or there is some halting configuration d' with $d \vdash d'$. It follows that M halts on input w if and only if $0q_0w \overset{a}{\vdash} d$ for some terminal configuration d and some integer $a \geq 0$.

(2)⇔(3) By iterated application of Lemma 4.14 and the definition of $\gamma(M,d)$ (Definition 4.6), we have that $0q_0w \overset{b}{\vdash} d$ and d terminal implies $K^{2b}(\gamma(M,0q_0w)) \cong \gamma(M,d) \cong \gamma_0(M)$. Conversely, by Lemma 4.15, $K^{2b}(\gamma(M,0q_0w)) \cong \gamma_0(M)$ implies that there is some d with $0q_0w \overset{*}{\vdash} d$ and $\gamma(M,d) \cong K^{2b}(\gamma(M,0q_0w)) \cong \gamma_0(M)$, then d is necessarily terminal (by Definition 4.6).

(3)⇔(4) (3) implies (4) with $a = b + 1$ by Lemma 4.9. (4) implies (3) by Lemma 4.17.

(4)⇔(5) This follows by definition of convergence and by Lemma 4.16. □

Hence we conclude that the first statement of Theorem 1.2 is true because of Lemma 4.18 and Remark 4.8. The second statement is an immediate corollary of the first, and the third statement follows by Remark 4.10.

This concludes the proof of Theorem 1.2.

5. Concluding Remarks

We have shown that clique-convergence is undecidable for automatic graphs. A number of consequences can be drawn. For instance, when the graphs we used here do converge, they converge to a 2-cycle of clique-Helly graphs (as defined in [19]: $K^n(G) \cong K^{n+2}(G)$ where both, $K^n(G)$ and $K^{n+1}(G)$, are clique-Helly). Thus the problem of deciding whether a graph is *eventually clique-Helly* (i.e. $K^n(G)$ is clique-Helly for some n) and the problem of deciding whether a graph is *eventually 2-self-clique* (i.e. $K^n(G) \cong K^{n+2}(G)$ for some n) are equivalent to the clique-convergence problem in the class $\mathcal{AG}_0$. In particular, these problems are undecidable for any class containing $\mathcal{AG}_0$. The clique-divergence problem is then also obviously undecidable for automatic graphs. The same is true for the problem of recognizing *eventually self-clique* graphs (i.e. $K^n(G) \cong K^{n+1}(G)$ for some n):

Theorem 5.1. *The problem of deciding whether a given graph is eventually self-clique is undecidable for automatic graphs.*

Proof. We shall freely use the notation and results of the previous section.

The required reduction from the halting problem, $\delta : \mathcal{TMW} \to \mathcal{AG}$, is given by $\delta(M,w) = \lambda(M,w) \boxtimes K(\lambda(M,w))$. Note that $\delta(M,w)$ is indeed automatic: strong products of automatic graphs are automatic and the clique graphs of our graphs in $\mathcal{AG}_0$ are easily describable as automatic graphs, and the corresponding automata can be effectively computed from those of $\lambda(M,w)$.

It is well known that the clique operator distributes over the strong product, that is, $K(A \boxtimes B) \cong K(A) \boxtimes K(B)$ (the proof in [16] is valid for infinite graphs). Also, connected graphs factorize in a unique way with respect to the strong product (up to repetitions of trivial factors) [10]. Moreover, each connected component of $\gamma(M,d)$ contains vertices (namely those of the form $5d_1$) whose closed neighborhood is isomorphic to $K_{1,3}$ (the complete bipartite graph with 1 vertex in one part and 3 vertices in the other). Since $K_{1,3}$ is no non-trivially factorizable, $\gamma(M,d)$ is also

no non-trivially factorizable. The same can be said about $K(\gamma(M,d))$, since the closed neighborhoods of the cliques of the form $\{5d_1, 6d_1\}$ in $K(\gamma(M,d))$ are also no non-trivially factorizable. Hence, it follows that the (essentially) unique factorization of $K^n(\delta(M,w))$, for even n, is $K^n(\delta(M,w)) = K^n(\lambda(M,w) \boxtimes K(\lambda(M,w))) \cong K^n(\lambda(M,w)) \boxtimes K^{n+1}(\lambda(M,w)) \cong \gamma(M,d) \boxtimes K(\gamma(M,d))$ for some d, with $0q_0w \overset{*}{\vdash} d$.

If M halts on w, d is terminal for some d satisfying $0q_0w \overset{*}{\vdash} d$. Hence, $\gamma(M,d) = \gamma_0(M)$ and

$$\begin{aligned} K^{n+1}(\delta(M,w)) &= K(K^n(\delta(M,w))) \cong K(\gamma(M,d) \boxtimes K(\gamma(M,d))) \\ &\cong K(\gamma_0(M) \boxtimes K(\gamma_0(M))) \cong K(\gamma_0(M)) \boxtimes K^2(\gamma_0(M)) \\ &\cong K(\gamma_0(M)) \boxtimes \gamma_0(M) \cong K^n(\delta(M,w)) \end{aligned}$$

and thus, $\delta(M,w)$ is eventually self-clique.

Reciprocally when $\delta(M,w)$ is eventually self-clique we have $K^n(\delta(M,w)) \cong K^{n+1}(\delta(M,w))$. Assume without loss that n is even, then $\gamma(M,d) \boxtimes K(\gamma(M,d)) \cong K(\gamma(M,d)) \boxtimes \gamma(M,d_1)$ for some d, d_1 with $0q_0w \overset{*}{\vdash} d$, and either $d \vdash d_1$ or d is a halting configuration and $d_1 = d$. But this is possible only when d is terminal, and hence when M halts on w.

It follows that M halts on w if and only if $\delta(M,w)$ is eventually self-clique. □

We point out that Theorem 1.2, can be strengthened in a number of ways, by redefining λ to reduce the class $\mathcal{AG}_0$. For instance, it is easy to amend the digraph of timestamped configurations D to reduce its in-degree to at most 2 (adding extra vertices). This allows us to claim that the clique-convergent problem is undecidable even for automatic graphs with maximum degree at most 10. Similarly, we may restrict the graphs in $\mathcal{AG}_0$ to be connected (adding extra edges between connected components, not forming triangles), with exactly one "exit point" (connecting all halting vertices in D with a (perhaps one-way-infinite) directed path), etc.

Thanks to Theorem 2.4, our Theorem 1.1 implies that:

Theorem 5.2. *The clique-convergence property is not first-order expressible for automatic graphs.* □

Moreover, in view of the generalization of Theorem 2.4 which appeared in [12, Theorem 3.2], the clique-convergence property is also not expressible when we extend the first-order logic to include several generalized quantifiers, including $\exists^{\infty}$, $\exists^{(k,m)}$ and $\exists^{k-ram}$. Also, the Theorem remains valid when we extend the first order logic to include the restricted second-order quantification allowed in FSO [12].

Which in turn, begs the question: Is clique-convergence first-order expressible for *finite* graphs? Is it expressible in FSO?

Acknowledgments We are grateful to the anonymous reviewers for their meticulous reading and many shrewd observations that made this a better paper.

References

[1] V. Bárány. *Automatic Presentations of Infinite Structures.* PhD thesis, RWTH Aachen University, Budapest, Hungary, 2007. `https://logic.rwth-aachen.de/~vbarany/diss.pdf`.

[2] A. Blumensath and E. Grädel. *Finite presentations of infinite structures: automata and interpretations.* Theory Comput. Syst. **37** (2004) 641–674. http://dx.doi.org/10.1007/s00224-004-1133-y.

[3] J.A. Bondy and U.S.R. Murty. *Graph theory with applications.* American Elsevier Publishing Co., Inc., New York, 1976.

[4] G.L. Chia and P.H. Ong. *On self-clique graphs all of whose cliques have equal size.* Ars Combin. **105** (2012) 435–449.

[5] F.F. Dragan. *Centers of graphs and the Helly property (in Russian).* PhD thesis, Moldava State University, Chisinău, Moldava, 1989.

[6] F. Escalante. *Über iterierte Clique-Graphen.* Abh. Math. Sem. Univ. Hamburg **39** (1973) 59–68.

[7] M. Groshaus, A.L.P. Guedes and L. Montero. *Almost every graph is divergent under the biclique operator.* Discrete Appl. Math. **201** (2016) 130–140. http://dx.doi.org/10.1016/j.dam.2015.07.022.

[8] S.T. Hedetniemi and P.J. Slater. *Line graphs of triangleless graphs and iterated clique graphs.* Springer Lecture Notes in Math. **303** (1972) 139–147.

[9] J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.

[10] W. Imrich and S. Klavžar. *Product graphs.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000. Structure and recognition, With a foreword by Peter Winkler.

[11] B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Logic and computational complexity (Indianapolis, IN, 1994)*, volume 960 of *Lecture Notes in Comput. Sci.*, pages 367–392. Springer, Berlin, 1995. http://dx.doi.org/10.1007/3-540-60178-3_93.

[12] D. Kuske. Theories of automatic structures and their complexity. In *Algebraic informatics*, volume 5725 of *Lecture Notes in Comput. Sci.*, pages 81–98. Springer, Berlin, 2009. http://dx.doi.org/10.1007/978-3-642-03564-7_5.

[13] F. Larrión, V. Neumann-Lara and M.A. Pizaña. *Clique divergent clockwork graphs and partial orders.* Discrete Appl. Math. **141** (2004) 195–207.

[14] F. Larrión, M.A. Pizaña and R. Villarroel-Flores. *On self-clique shoal graphs.* Discrete Applied Mathematics **205** (2016) 86 – 100. http://www.sciencedirect.com/science/article/pii/S0166218X16000275.

[15] J. Meidanis. *The clique operator.* Available (2001) at : http://www.ic.unicamp.br/~meidanis/research/clique/.

[16] V. Neumann-Lara. On clique-divergent graphs. In *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, volume 260 of *Colloq. Internat. CNRS*, pages 313–315. CNRS, Paris, 1978.

[17] V. Neumann-Lara. Clique divergence in graphs. In L. Lovász and V.T. Sós, editors, *Algebraic methods in graph theory, Coll. Math. Soc. János Bolyai, vol. 25 Szeged*, pages 563–569. North-Holland, Amsterdam, 1981.

[18] E. Prisner. *The dynamics of the line and path graph operators.* Graphs and Combinatorics **9** (1993) 335–352.

[19] E. Prisner. *Graph dynamics.* Longman, Harlow, 1995.

[20] M. Requardt. *(Quantum) spacetime as a statistical geometry of lumps in random networks.* Classical Quantum Gravity **17** (2000) 2029–2057.

[21] M. Requardt. Space-time as an order-parameter manifold in random networks and the emergence of physical points. In *Quantum theory and symmetries (Goslar, 1999)*, pages 555–561. World Sci. Publ., River Edge, NJ, 2000.

[22] M. Requardt. *A geometric renormalization group in discrete quantum space-time*. J. Math. Phys. **44** (2003) 5588–5615.

[23] S. Rubin. *Automata presenting structures: a survey of the finite string case*. Bull. Symbolic Logic **14** (2008) 169–209. http://doi.org/10.2178/bsl/1208442827.

[24] J.L. Szwarcfiter. A survey on clique graphs. In B.A. Reed and C. Linhares-Sales, editors, *Recent advances in algorithms and combinatorics*, volume 11 of *CMS Books Math./Ouvrages Math. SMC*, pages 109–136. Springer, New York, 2003.

[25] J.L. Szwarcfiter. *Recognizing clique-Helly graphs*. Ars Combin. **45** (1997) 29–32.