Deciding whether there are infinitely many prime graphs with forbidden induced subgraphs

Robert Brignall[†], Hojin Choi[‡], Jisu Jeong[‡], and Sang-il Oum[*‡]

[†]School of Mathematics and Statistics
The Open University, Milton Keynes, UK
robert.brignall@open.ac.uk
[‡]Department of Mathematical Sciences
KAIST, Daejeon, Korea
hjchoi0330@gmail.com, jisujeong89@gmail.com, sangil@kaist.edu

August 2, 2018

Abstract

A *homogeneous set* of a graph G is a set X of vertices such that $2 \leq |X| < |V(G)|$ and no vertex in $V(G) - X$ has both a neighbor and a non-neighbor in X. A graph is *prime* if it has no homogeneous set. We present an algorithm to decide whether a class of graphs given by a finite set of forbidden induced subgraphs contains infinitely many non-isomorphic prime graphs.

Keywords: modular decomposition, induced subgraph, prime graph, homogeneous set

1 Introduction

All graphs in this paper are simple. We write $H \preceq_i G$ if a graph H is isomorphic to an induced subgraph of a graph G, which is a subgraph of G obtained by deleting some vertices. A class $\mathcal{C}$ of graphs is *hereditary* if for all graphs H and G, $H \in \mathcal{C}$ whenever $H \preceq_i G$ and $G \in \mathcal{C}$. For a set X of graphs, we say G is X*-free* if $H \not\preceq_i G$ for all $H \in X$. Let us write $\mathrm{Free}(X)$ to denote the class of X-free graphs. It is clear that for each hereditary class $\mathcal{C}$ of graphs, there exists a set X of graphs such that $\mathcal{C} = \mathrm{Free}(X)$, simply by taking X as $\preceq_i$-minimal graphs

[*]Supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2017R1A2B4005020).

not in $\mathcal{C}$. Note that this set X is not necessarily finite (for example, consider the class of forests, whose minimal forbidden set contains all cycles on three or more vertices).

A *homogeneous* set (also known in the literature as clans [12], intervals [17, 20], or modules [15, 21]) of a graph G is a set X of vertices such that $2 \leq |X| < |V(G)|$ and each vertex in $V(G) - X$ is either complete or anti-complete to X. A graph is *prime*[1] if it has no homogeneous set.

For positive integers n, let P_n be a path on n vertices and let $K_{1,n}$ be a complete bipartite graph on $n + 1$ vertices where one part consists of one vertex. In P_4-free graphs, also known as cographs [8], it is well known that they have no prime graphs on three or more vertices. However, in $K_{1,3}$-free graphs, commonly known as claw-free graphs, we can easily find infinitely many prime graphs, such as all cycle graphs on at least 5 vertices. Thus we may ask the following question: *for a given set L of finitely many graphs, can we decide whether there are infinitely many non-ismorphic L-free prime graphs?* We answer this question positively as follows.

Theorem 1.1. *For a given finite set L of graphs, there exists an algorithm to decide whether* $\mathrm{Free}(L)$ *contains infinitely many non-isomorphic prime graphs.*

Prime graphs form the 'building blocks' of all other graphs by means of the *modular decomposition* (See [3, Theorem 1.5.1]). The modular decomposition first appeared in the abstract of a talk by Fraïssé [13] in 1953, although its first appearance in an article seems to be Gallai [14]. It has since appeared in a number of contexts, ranging from game theory to combinatorial optimization.

The significance of Theorem 1.1 is that if a hereditary class $\mathcal{C} = \mathrm{Free}(L)$ of graphs has only finitely many non-isomorphic prime graphs, then the class has a number of desirable properties. For example, $\mathcal{C}$ is well-quasi-ordered by the induced subgraph relation [18, Theorem 6] (in other words, $\mathcal{C}$ contains no infinite set of graphs no one of which is an induced subgraph of any other), and every graph in $\mathcal{C}$ has bounded *clique-width* [11], which itself gives rise to a number of desirable algorithmic properties, via the meta-theorem of Courcelle, Makowsky, and Rotics [10].

Brignall, Ruškuc, and Vatter [5] studied an analogous problem for permutations, under the 'containment' ordering. In the theory of permutations, simple permutations correspond to prime graphs in our context. They proved that there exists an algorithm to determine whether a given hereditary class of permutations described by finitely many forbidden permutations admits infinitely many simple permutations. To prove the existence of a decision algorithm, they utilise a theorem on unavoidable subpermutations in large simple permutations by Brignall, Huczynska, and Vatter [4].

For us, it is also necessary to understand unavoidable induced subgraphs in large prime graphs. Recently Chudnovsky, Kim, Oum, and Seymour [6] proved such a theorem, which states that every sufficiently large prime graph contains one of a few large prime graphs as an induced subgraph. We will review this theorem in detail in Theorem 2.2. Our algorithm will check whether all these unavoidable induced subgraphs are forbidden by the given set L

[1] Other terms that have been used for 'prime' include indecomposable, irreducible, and primitive.

of forbidden graphs. If all of them are forbidden, then Free(L) contains only finitely many non-isomorphic prime graphs and so the algorithm terminates with the answer NO. If at least one of them is not forbidden, then we prove that Free(L) contains arbitrarily large prime graphs and so the algorithm terminates with the answer YES.

One outcome of Theorem 2.2 dominates the work to prove Theorem 1.1, namely the case of 'chains' of length n, and this is covered in Section 3. In theory, to handle this case one could employ automata-theoretic arguments analogous to those used in [5] to handle 'pin sequences', the direct analogue of chains for permutations. Instead, we will present a purely combinatorial argument, using a few applications of the pigeonhole principle, to show that if a class Free(L) contains arbitrarily long chains, then it must contain arbitrarily long chains with a periodic construction, whose period is bounded by a function of the largest graph in L.

The remaining cases from Theorem 2.2 and hence the proof of Theorem 1.1 are covered in Section 4.

2 Unavoidable induced subgraphs in large prime graphs

Chudnovsky, Kim, Oum, and Seymour [6] proved that every sufficiently large prime graph contains one of a few large prime graphs as an induced subgraph. After a couple of preliminary concepts, we introduce definitions of those large prime graphs and the result in this section.

The 1-*subdivision* of a graph G is the graph H obtained from G by subdividing every edge once. The *line graph* of a graph G is the graph H whose vertex set is $V(H) = E(G)$ and two vertices e_1, e_2 are adjacent in H if two edges e_1, e_2 share an end in G. We are particularly interested in the 1-subdivision of $K_{1,n}$, and the line graph of $K_{2,n}$, both of which are prime for all $n \geq 3$, and illustrated in Figure 1(i) and (ii), respectively.

The *thin spider with n legs* is the graph H with vertex set $V(H) = \{v_1, v_2, \dots, v_n\} \cup \{u_1, u_2, \dots, u_n\}$ and edge set $E(H) = \{v_iu_i : 1 \leq i \leq n\} \cup \{u_iu_j : 1 \leq i < j \leq n\}$. The *half-graph of height n* is the graph H with vertex set $V(H) = \{v_1, v_2, \dots, v_n\} \cup \{u_1, u_2, \dots, u_n\}$ and edge set $E(H) = \{v_iu_j : 1 \leq i \leq j \leq n\}$. The graph $H'_{n,I}$ has vertex set $V(H'_{n,I}) = \{v_1, v_2, \dots, v_n\} \cup \{u_1, u_2, \dots, u_n\} \cup \{w\}$ and edge set $E(H'_{n,I}) = \{wv_i : 1 \leq i \leq n\} \cup \{v_iu_j : 1 \leq i \leq j \leq n\} \cup \{u_iu_j : 1 \leq i < j \leq n\}$. Finally, the graph H^*_n has vertex set $V(H^*_n) = \{v_1, v_2, \dots, v_n\} \cup \{u_1, u_2, \dots, u_n\} \cup \{w\}$ and edge set $E(H^*_n) = \{wv_1\} \cup \{v_iu_j : 1 \leq i \leq j \leq n\} \cup \{u_iu_j : 1 \leq i < j \leq n\}$. Examples of these graphs are illustrated in Figure 1(iii)–(vi), and it is easy to see that these graphs are prime.

A *chain* C of length n is a sequence $v_0, v_1, \dots, v_n$ of distinct vertices such that for each $i \in \{2, \dots, n\}$, v_i is adjacent to all $v_0, v_1, \dots, v_{i-2}$ but not v_{i-1}, or non-adjacent to all $v_0, v_1, \dots, v_{i-2}$ but adjacent to v_{i-1}. We call v_0 the *first vertex* of the chain. The graph induced by a chain of length n is prime, or is prime after discarding one of the vertices v_0 or v_1, as shown by the following result.

Proposition 2.1 ([6, Corollary 2.3]). *Every chain of length $n > 3$ contains a chain of length $n - 1$ inducing a prime graph.*

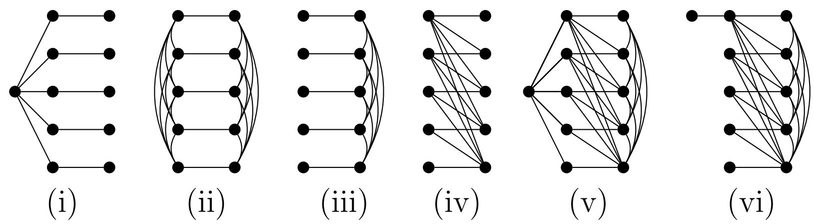

Figure 1: Examples of the unavoidable graphs of cases (i)–(vi) in Theorem 2.2.

Note that, in a slight departure from [6], we will not necessarily require that a chain is contained inside some specified graph. Instead, chains can be considered as sequences of vertices, which may or may not be embedded inside some larger graph, depending on the context. Additionally, we may from time to time abuse notation by referring to the chain when we mean the graph induced by a chain.

We are now ready to state the main result of [6], which provides the structural basis for our algorithm.

Theorem 2.2 (Chudnovsky, Kim, Oum and Seymour [6]). *For every integer $n \geq 3$, there exists N such that every prime graph with at least N vertices contains one of the following graphs as an induced subgraph.*

(i) The 1-subdivision of $K_{1,n}$ or its complement.

(ii) The line graph of $K_{2,n}$ or its complement.

(iii) The thin spider with n legs or its complement.

(iv) The half-graph of height n.

(v) The graph $H'_{n,I}$.

(vi) The graph H^*_n or its complement.

(vii) A prime graph induced by a chain of length n.

Note that in the characterization of Theorem 2.2, the complements of the half-graphs (case (iv)) and $H'_{n,I}$ (case (v)) both contain (as induced subgraphs) graphs of the same type, with two vertices removed. Since the graphs in cases (i)–(vi) of Theorem 2.2 admit regular structures, it is relatively straightforward to check whether a class Free(L) contains arbitrarily large ones. The details are provided in Section 4.

3 Chains and strings

In this section, we consider the chains that arise in case (vii) of Theorem 2.2. Note that the complement of a chain is again a chain.

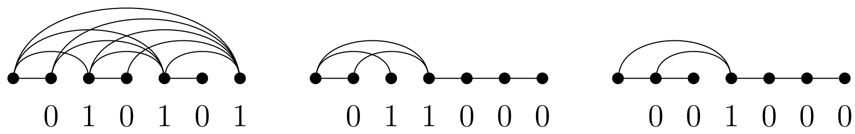

Figure 2: Examples of chains, and their encodings as strings via the bijection ϕ. Note the two examples on the right give rise to graphs that are isomorphic.

For convenience, we seek to describe an encoding of chains as strings over the alphabet $\{0, 1\}$. First, we introduce some elementary concepts about strings.

A $(0, 1)$-*string* (or simply a *string*) is an element of $\{0, 1\}^*$, where $\{0, 1\}^*$ is the set of all finite sequences of 0 and 1. The *length* of a string S is the number of 0's and 1's in the string and is denoted by $|S|$. Given strings S and T, we denote the *concatenation* (defined in the natural way) by ST. For example, if $S = 011$ and $T = 101$, then $ST = 011101$. Let S^t denote the concatenation of t copies of a string S. For example, $S^3 = SSS$.

We say that T is a *factor* of S, or S *contains* T as a factor, if there exist strings X and Y such that $S = XTY$. An *occurrence* of T in S is a pair (T, i) such that $S = XTY$ and $|X| = i - 1$ (that is, T is a factor of S that starts at the i-th letter). Furthermore, we say that the occurrences (S_1, i_1) and (S_2, i_2) of two (possibly equal) factors inside some string S with $i_1 \leq i_2$ are *1-disjoint* if $i_1 + |S_1| < i_2$ (in other words, there is at least one letter of S that is not used in either of the occurrences, but lies 'between' S_1 and S_2), and they *intersect* if $i_1 + |S_1| > i_2$.

We are now ready for the basic encoding of chains into strings, which we will denote by ϕ. For a chain $C = v_0, v_1, \ldots, v_k$ of length k, let $\phi(C) = s_1 s_2 \cdots s_k$ where $s_i = 0$ if v_i is adjacent to v_{i-1}, and $s_i = 1$ otherwise for each $i \in \{1, \ldots, k\}$. Note that ϕ is a bijection between chains and strings, but recall that the graphs induced by two distinct chains C_1 and C_2 can be isomorphic and so a graph that is induced by some chain does not necessarily have a unique representation as a string. Note also that if C contains $k + 1$ vertices, then $\phi(C)$ contains k letters, because the first vertex is not assigned a letter. See Figure 2.

We say that a graph G is *induced by* a string S if G is induced by $C = \phi^{-1}(S)$. Similarly, we say that a string S *contains* a graph G if the graph induced by S contains G as an induced subgraph.

In addition to encoding chains into strings, we also need to be able to encode subgraphs of strings, in order to identify when a given string contains graphs from the minimal forbidden set L. To this end, suppose that G is a graph on n vertices that embeds inside some string S. If $\phi^{-1}(S) = v_0, v_1, \ldots, v_k$, then G is isomorphic to the graph induced on the subsequence $v_{i_1}, v_{i_2}, \ldots, v_{i_n}$ that corresponds to the chosen embedding, where $0 \leq i_1 < i_2 < \cdots < i_n \leq k$. We now define a new encoding ψ from subsequences of chains (or embeddings of graphs into chains) into strings over the three-letter alphabet $\{0, 1, |\,\}$.

For each j ranging from 2 to n, the encoding ψ writes symbols according to the following rules: if $i_j = i_{j-1} + 1$, then write 0 if v_{i_j} is adjacent to $v_{i_{j-1}}$, and 1 otherwise. When $i_j > i_{j-1} + 1$, write $|\ 0$ if v_{i_j} is *not* adjacent to $v_{i_{j-1}}$ (and all earlier vertices), and $|\ 1$ otherwise. If G is isomorphic to the graph induced on the subsequence M of some chain,

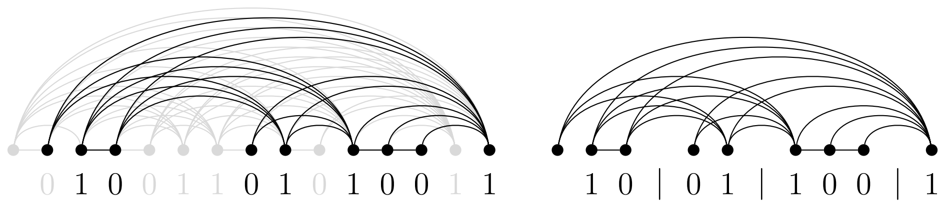

Figure 3: On the left, an embedding M of a graph inside the chain with string 01001101010011. On the right, the encoding of M is the representation $\psi(M) = 10 \mid 01 \mid 100 \mid 1$ with blocks 10, 01, 100 and 1.

then we call $\psi(M)$ a *representation* of G. A *block* of a representation is a maximal factor that contains only the letters 0 and 1. See Figure 3. Note that if a representation begins with the symbol $|$, then we will assume that there is an empty block preceding it.

Before we go further, we need to make a few remarks about the strings created under the encoding ψ. Suppose that M is an embedding (or subsequence) of a graph on n vertices inside some chain.

- $\psi(M)$ has exactly $n-1$ symbols that are 0 or 1.
- $\psi(M)$ cannot contain the factor $||$, nor can it end with the symbol $|$. Therefore, there are at most $n-1$ instances of the symbol $|$ in $\psi(M)$.
- $\psi(M)$ therefore contains at most $2n-2$ letters.
- ψ is not a bijection, because it does not remember the specific positions of vertices of M in the chain.

At this point we make an important observation: one can view the reverse process of ψ, from words over the alphabet $\{0, 1, |\}$ to graphs, as a monadic second-order transduction, from which it is possible to conclude that the edge relation on subgraphs of chains is definable by a monadic second-order formula. This gives rise to a decision procedure for whether Free(L) contains arbitrarily long or not via the Backwards Translation Theorem (see [9, Theorem 7.10]), as the language over $\{0, 1, |\}$ corresponding to Free(L) is regular. This approach is essentially the same as the one given in the case of permutations, see [5], but it is not the approach we use here.

Instead, the decision procedure we present here comprises two parts and is elementary (in that it requires only the pigeonhole principle applied to the structures introduced in this section so far). First, we establish that if there exists a chain of a specified (large) length in Free(L), then there exists arbitrarily long chains with a periodic structure, where the size of the period is bounded above by a function of the largest forbidden graph in L (this may be compared to the 'pumping lemma' in the study of regular languages). Note that by exhaustively checking membership in Free(L) of all chains of the specified large length, this result is already sufficient for a decision procedure. However, the second part of our

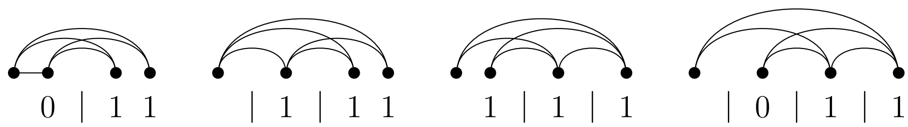

Figure 4: Four different representations of the same graph $K_4 - e$.

procedure gives us a simpler method, namely a check for whether a particular chain sequence can be repeated arbitrarily often.

Now we consider the total number of possible representations of graphs on n vertices. Each representation is obtained from a $(0,1)$-string of length $n-1$ by inserting at most $n-1$ copies of the symbol $|$. There are 2^{n-1} $(0,1)$-strings of length $n-1$, and there are 2^{n-1} choices of inserting the symbol $|$ or not at each position. Thus, we deduce the following observation.

Observation 3.1. *For each positive integer n, there are at most 2^{2n-2} representations of graphs on n vertices. Moreover, each such representation R has at most n blocks.*

Now consider a representation R of a graph with n vertices. Although we cannot recover the specific embedding of this graph in a chain that gave rise to the representation, we can reconstruct the graph from R in the natural way: create vertices $v_1, v_2, \dots v_n$, where $v_2, \dots, v_n$ correspond to the non-$|$ symbols in R, reading from left to right. For each i ($2 \le i \le n$), the adjacencies of v_i to the previous $i-1$ vertices is determined by the letter of R corresponding to v_i (which is either 0 or 1), and the letter (if it exists) immediately preceding this one in R (specifically, whether this symbol is $|$ or not).

Given the above reconstruction process, each representation R corresponds to a unique graph G. However, each graph G can have several corresponding representations – see Figure 4 for an example. We let $\mathcal{R}_G$ denote the set of all representations that correspond to a given graph G. Note that $|\mathcal{R}_G| \le 2^{2|V(G)|-2}$ by Observation 3.1.

Our final preparatory task is to observe how a representation $R \in \mathcal{R}_G$ can be embedded in some given $(0,1)$-strings S. We say that the string S *contains* the representation R if

(1) each block of R is embedded as a factor in S, and

(2) every pair of distinct blocks B_i and B_j are embedded as 1-disjoint factors, with the factor corresponding to B_i preceding that of B_j if and only if B_i precedes B_j in R.

Now, we introduce two lemmas for the proof of Theorem 1.1.

Lemma 3.2. *Let L be a set of graphs having at most n vertices. If there exists a $(0,1)$-string T of length at least $\lceil \frac{(n-1)4^n+1}{3} \rceil (2^{n-2}+n-1)$ containing no graphs in L, then there exists a $(0,1)$-string S of length at most 2^n such that the $(0,1)$-string S^k contains no graph in L for all k.*

Proof. Let $\mathcal{R} = \cup_{G \in L} \mathcal{R}_G$ be the set of all representations of graphs from L. Note that, by Observation 3.1, we have $|\mathcal{R}| \leq \sum_{k=1}^{n} 2^{2k-2} \leq 4^n/3$. Furthermore, each representation in $\mathcal{R}$ has at most n blocks.

Let $s = \lceil \frac{(n-1)4^n+1}{3} \rceil$. We may assume that $|T| = s(2^{n-2} + n - 1)$. We can rewrite $T = T_1 \ell_1 T_2 \ell_2 \cdots T_s \ell_s$ where $|T_i| = 2^{n-2} + n - 2$ and $\ell_i = 0$ or 1 for all i. Thus, the T_i are pairwise 1-disjoint.

We claim that there exists j^* such that for every representation $R \in \mathcal{R}$, at least one block of R is not a factor of T_{j^*}. Suppose not. Then for each $j \in \{1, 2, \ldots, s\}$, there exists $R_j \in \mathcal{R}$ such that T_j contains each block of the representation R_j as a factor. Note that the blocks of R_j in T_j may overlap and may appear in the incorrect order. Since $s > (n-1)4^n/3$ and $|\mathcal{R}| \leq 4^n/3$, by the pigeonhole principle, at least n of the T_j must contain all the blocks from one particular representation $R^* \in \mathcal{R}$ as a factor. That is, there exists a subsequence $j_1, j_2, \ldots, j_n$ of $1, 2, \ldots, s$ such that $R_{j_1} = R_{j_2} = \cdots = R_{j_n} = R^*$. This means that by considering the factor of T_{j_1} equal to the first block of R^*, the factor of T_{j_2} equal to the second block, and so on, and recalling that the T_{j_k} are pairwise 1-disjoint, we find that T contains the representation R^*. Therefore T contains some $G \in L$, a contradiction which proves the claim.

Now, T_{j^*} does not contain at least one block of every representation $R \in \mathcal{R}$ as a factor. By the pigeonhole principle, since $|T_{j^*}| = 2^{n-2} + n - 2$, there exist two (not necessarily disjoint) occurrences (A, a_1) and (A, a_2) in T_{j^*} such that $|A| = n - 2$, and $a_1 < a_2$. That is, we find the same factor of $n - 2$ letters occurring at least twice in T_{j^*}.

Now, consider the occurrence (S, a_1) in T_{j^*} where S is a factor of T_{j^*} of length $a_2 - a_1$, in other words, $T_{j^*} = K_1 S A K_2$ for some (possibly empty) prefix K_1 and suffix K_2 of T_{j^*}. Note that $|S| \leq 2^{n-2}$ since T_{j^*} has length $2^{n-2} + n - 2$ and $|A| = n - 2$. We claim that $\phi^{-1}(S^k) \in \text{Free}(L)$ for all k.

Suppose to the contrary that there exists k such that S^k contains some representation $R \in \mathcal{R}$. By construction of T_{j^*}, there is some block B of R that is not contained in T_{j^*} as a factor, and therefore B is not contained in SA or in S as a factor. Moreover, by construction of S, we observe that either S^k is a factor of SA, or SA is a factor of S^k. See Figure 5.

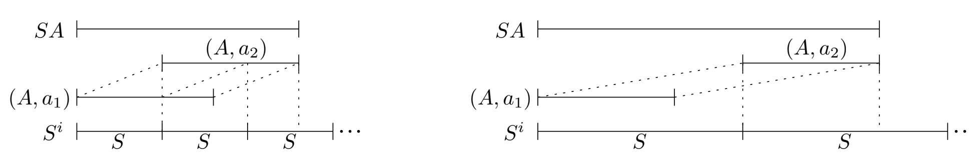

Figure 5: Since both occurrences (A, a_1) and (A, a_2) represent the same factor, we can deduce that for each positive integer i, either S^i is a factor of SA or SA is a factor of S^i.

If S^k is a factor of SA, then since the block B is a factor of S^k, it is also a factor of SA, which is a contradiction. Therefore, SA is a factor of S^k. We may assume that B is embedded as a factor in S^k starting from an entry in the first copy of S. Note that A contains precisely $n - 2$ letters, and B contains at most $n - 1$ letters. From this, we conclude that B embeds into SA (starting from an entry in the prefix S), another contradiction.

Thus we conclude that S^k contains no representation $R \in \mathcal{R}$ for all k, which completes the proof. $\square$

Lemma 3.2 tells us that if a class Free(L) contains arbitrarily long chains then it contains arbitrarily long chains with a periodic construction, whose period is at most 2^n. Our next lemma gives us the necessary practical condition for our decision procedure to test whether a string can be repeated arbitrarily many times or not.

Lemma 3.3. *Let L be a set of graphs having at most n vertices. Let S be a string. If S^{2n-1} contains none of the graphs in L, then* Free(L) *contains $\phi^{-1}(S^k)$ for all k.*

Proof. Suppose that the lemma is false. Let M be the minimum number such that S^M contains at least one graph $G \in L$. This means that there exists a representation $R \in \mathcal{R}_G$ which is contained in S^M. Fix one such embedding of R in S^M. Since $M \geq 2n$ and $|V(G)| \leq n$, there exist two consecutive copies of S neither of which is used in the embedding of R in S^M. We can therefore eliminate one of these two copies of S while still ensuring that the blocks of R are 1-disjoint (to ensure R can still be embedded in the resulting string). That is, S^{M-1} still contains R, which is a contradiction since M is the minimum number such that S^M contains at least one graph in L. $\square$

4 Proof of the main result

In this section, we prove our main result, Theorem 1.1. Recall the statement of our main theorem.

Theorem 1.1. *For a given finite set L of graphs, there exists an algorithm to decide whether* Free(L) *contains infinitely many non-isomorphic prime graphs.*

Let $\mathcal{G}_n$ be the set that consists of the 1-subdivision of $K_{1,n}$ and its complement, the line graph of $K_{2,n}$ and its complement, the thin spider with n legs and its complement, the half-graph of height n, the graph $H'_{n,I}$, and the graph H^*_n and its complement. In other words, $\mathcal{G}_n$ contains one representative of each type of graph in Theorem 2.2 except for chains. Note that it is routine to check that all the graphs in $\mathcal{G}_n$ are prime.

By Theorem 2.2, a large prime graph that does not contain a chain of length n must contain a graph in $\mathcal{G}_n$. For a graph in $\mathcal{G}_n$, it is easy to deduce the following lemma by the definition of $\mathcal{G}_n$. For an example, suppose that a graph G with n vertices is an induced subgraph of the 1-subdivision of $K_{1,N+1}$. Let v be a vertex of degree $N+1$ in the 1-subdivision of $K_{1,N+1}$, let $u_1, u_2, \ldots, u_{N+1}$ be neighbors of v, and let v_i be a neighbor of u_i other than v for each i. Now, there exist u_i and v_i such that neither u_i nor v_i are in G. We delete u_i and v_i from the 1-subdivision of $K_{1,N+1}$ to obtain the 1-subdivision of $K_{1,N}$ that contains G as an induced subgraph. We can prove similarly for other cases.

Lemma 4.1. *Let G be a graph on n vertices and let N be an integer with $N \geq \max\{n, 3\}$. If G is an induced subgraph of some graph in $\mathcal{G}_{N+1}$, then there exists a graph H in $\mathcal{G}_N$ such that G is an induced subgraph of H.* □

Finally, we give the proof of our main theorem, providing Algorithm 1.

Proof of Theorem 1.1. Let $n \geq 3$ be the minimum integer such that every graph in L has at most n vertices. By the contrapositive statement to Lemma 4.1, if $\mathcal{G}_n$ has a graph in $\mathrm{Free}(L)$ then $\mathrm{Free}(L)$ must contain a graph from $\mathcal{G}_N$ for every $N \geq n$. Hence $\mathrm{Free}(L)$ has infinitely many non-isomorphic prime graphs.

Now, we may assume that every graph in $\mathcal{G}_n$ is not in $\mathrm{Free}(L)$. By Theorem 2.2, it is enough to decide whether $\mathrm{Free}(L)$ has infinitely many non-isomorphic prime graphs induced by chains. If there exists a string S of length at most 2^n such that $\phi^{-1}(S^{2n-1}) \in \mathrm{Free}(L)$, then by Proposition 2.1 and Lemma 3.3, $\mathrm{Free}(L)$ has infinitely many non-isomorphic prime graphs induced by chains.

On the other hand, if $\phi^{-1}(S^{2n-1}) \notin \mathrm{Free}(L)$ for every string S of length at most 2^n, then by Lemma 3.2 the maximum length of a chain contained in $\mathrm{Free}(L)$ is less than $\lceil \frac{(n-1)4^n+1}{3} \rceil (2^{n-2} + n - 1)$, which implies that $\mathrm{Free}(L)$ has only finitely many non-isomorphic prime graphs. □

Algorithm 1 Does $\mathrm{Free}(L)$ contain infinitely many prime graphs?

```
Let L be the input set of graphs and let n ≥ 3 be the minimum integer such that every
   graph in L has at most n vertices.
if G_n has a graph in Free(L) then
    output YES.
else if there exists a string S of length at most 2^n such that the string S^{2n−1} contains
   no graph in L then
    output YES.
else
    output NO.
end if
```

5 Concluding remarks

Complexity of the procedure. We have not made any particular effort to optimize the procedure described above. The majority of the work lies in determining whether a hereditary class $\mathrm{Free}(L)$ admits arbitrarily long chains or not, and here one may need to exhaust over all $2^{2^n+1} - 1$ chains of length at most 2^n, where $n = \max_{G \in L} |G|$. By contrast, Lemma 4.1 shows that in order to check whether $\mathrm{Free}(L)$ contains arbitrarily large prime graphs of the other types listed in Theorem 2.2, it suffices to check whether each of the 10 graphs in $\mathcal{G}_n$ (each having at most $2n + 1$ vertices) contains some graph in L.

In the analogous problem of deciding whether a hereditary class of permutations contains only finitely many simple permutations, a recent paper due to Bassino, Bouvel, Pierrot and Rossin [2] establishes an algorithm with run time $O(nk\log(nk) + n^{2k})$, where n is the size of the largest forbidden permutation, and k is the number of forbidden permutations. It is quite possible that a similar detailed analysis of chains in graphs could lead to a much more efficient algorithm.

Finding all the prime graphs in a class. If our decision procedure returns YES, then in theory it could provide a 'certificate' of an infinite family of prime graphs that the class contains. On the other hand, if the procedure returns NO, then Lemmas 3.2 and 4.1 give bounds on the number of vertices that the largest prime graph in the class can contain. However, the following result (recently re-discovered by Chudnovsky and Seymour [7]), gives a more practical method that may terminate sooner:

Proposition 5.1 (Schmerl and Trotter [20])**.** *Let $n \geq 3$ be an integer. Every prime graph on n vertices contains a prime induced subgraph on $n-1$ or $n-2$ vertices.*

Furthermore, the only prime graphs that do not contain a prime graph on 1 fewer vertices are the half-graphs of height n, and their complements. Thus, to list all prime graphs in a class, one can successively generate and check for membership the prime graphs of each order, and halt as soon as one finds two consecutive integers where the hereditary class contains no prime graphs of that order.

Classes with infinitely many minimal forbidden graphs One may ask whether it is possible for a hereditary class $\mathcal{C} = \mathrm{Free}(L)$ to contain only finitely many prime graphs when L is an *infinite* minimal set of forbidden graphs. The answer to this is no: any hereditary class containing only finitely many prime graphs possesses the property of being *labelled* well-quasi-ordered (see [1, Theorem 2]), and any such class is defined by a finite set of minimal forbidden graphs (this latter observation is essentially due to Pouzet [19]).

The same observation (that a hereditary class with only finitely many prime graphs is defined by finitely many minimal forbidden graphs) also leads to a quick proof of a special case of the results concerning 'prime extensions': namely that a finite set of prime graphs necessarily only has finitely many prime extensions (see Giakoumakis and Olariu [16]).

Acknowledgment The authors would like to thank the reviewer for providing a careful review and an alternative proof of the theorem in terms of monadic second-order logic.

References

[1] A. Atminas and V. V. Lozin. Labelled Induced Subgraphs and Well-Quasi-Ordering. *Order*, 32(3):313–328, 2015.

[2] F. Bassino, M. Bouvel, A. Pierrot, and D. Rossin. An algorithm for deciding the finiteness of the number of simple permutations in permutation classes. *Adv. in Appl. Math.*, 64:124–200, 2015.

[3] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: a survey.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[4] R. Brignall, S. Huczynska, and V. Vatter. Decomposing simple permutations, with enumerative consequences. *Combinatorica*, 28:385–400, 2008.

[5] R. Brignall, N. Ruškuc, and V. Vatter. Simple permutations: decidability and unavoidable substructures. *Theoret. Comput. Sci.*, 391(1-2):150–163, 2008.

[6] M. Chudnovsky, R. Kim, S.-i. Oum, and P. Seymour. Unavoidable induced subgraphs in large graphs with no homogeneous sets. *J. Combin. Theory Ser. B*, 118:1–12, 2016.

[7] M. Chudnovsky and P. Seymour. Growing without cloning. *SIAM J. Discrete Math.*, 26(2):860–880, 2012.

[8] D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3(3):163–174, 1981.

[9] B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic*, volume 138 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2012. A language-theoretic approach, With a foreword by Maurice Nivat.

[10] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[11] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Appl. Math.*, 101(1-3):77–114, 2000.

[12] A. Ehrenfeucht, T. Harju, and G. Rozenberg. 2-structures—a framework for decomposition and transformation of graphs. In *Handbook of graph grammars and computing by graph transformation, Vol. 1*, pages 401–478. World Sci. Publ., River Edge, NJ, 1997.

[13] R. Fraïssé. On a decomposition of relations which generalizes the sum of ordering relations. *Bull. Amer. Math. Soc.*, 59:389, 1953.

[14] T. Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hungar.*, 18:25–66, 1967.

[15] V. Giakoumakis. On the closure of graphs under substitution. *Discrete Math.*, 177(1-3):83–97, 1997.

[16] V. Giakoumakis and S. Olariu. All minimal prime extensions of hereditary classes of graphs. *Theoret. Comput. Sci.*, 370(1-3):74–93, 2007.

[17] P. Ille. Indecomposable graphs. *Discrete Math.*, 173(1-3):71–78, 1997.

[18] N. Korpelainen and V. Lozin. Two forbidden induced subgraphs and well-quasi-ordering. *Discrete Math.*, 311(16):1813–1822, 2011.

[19] M. Pouzet. Un bel ordre d'abritement et ses rapports avec les bornes d'une multirelation. *C. R. Acad. Sci. Paris Sér. A-B*, 274:A1677–A1680, 1972.

[20] J. H. Schmerl and W. T. Trotter. Critically indecomposable partially ordered sets, graphs, tournaments and other binary relational structures. *Discrete Math.*, 113(1-3):191–205, 1993.

[21] J. Spinrad. P_4-trees and substitution decomposition. *Discrete Appl. Math.*, 39(3):263–291, 1992.