

Dichotomy for tree-structured trigraph list homomorphism problems

Tomás Feder^a, Pavol Hell^c, David G. Schell^c, Juraj Stacho^b

^a268 Waverley St., Palo Alto, CA 94301, USA

^bCaesarea Rothschild Institute, University of Haifa, Mt. Carmel, 31905 Haifa, Israel

^cSchool of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, B.C., Canada V5A 1S6

Abstract

Trigraph list homomorphism problems, also known as list matrix partition problems, generalize graph list colouring and digraph list homomorphism problems. While digraph list homomorphism problems enjoy a *dichotomy* (each problem is NP -complete or polynomial time solvable), such dichotomy is not necessarily expected for trigraph list homomorphism problems, and in the few cases where dichotomy has been proved, for small trigraphs, the progress has been slow.

In this paper, we prove dichotomy for trigraph list homomorphism problems where the underlying graph of the trigraph is a tree. In fact, we show that for these trigraphs the trigraph list homomorphism problem is polynomially equivalent to a related *digraph* list homomorphism problem. The result can be extended to a larger class of trigraphs, and we illustrate the extension on trigraph cycles.

Key words: trigraph, list homomorphism, matrix partition, trigraph homomorphism, surjective list homomorphism, dichotomy, trigraph tree

1. Introduction

A *trigraph* H consists of a set $V = V(H)$ of vertices, and two disjoint sets of directed edges on V – the set of *weak edges* $W(H) \subseteq V \times V$, and the set of *strong edges* $S(H) \subseteq V \times V$. If both edge sets $W(H), S(H)$, viewed as relations on V , are symmetric, we have a *symmetric*, or *undirected* trigraph. A weak, respectively strong, edge vv is called a *weak*, respectively *strong*, *loop* at v .

The *adjacency matrix* of a trigraph H , with respect to an enumeration v_1, v_2, \dots, v_n of its vertices, is the $n \times n$ matrix M over $0, 1, *$, in which $M_{i,j} = 0$ if $v_i v_j$ is not an edge, $M_{i,j} = *$ if $v_i v_j$ is a weak edge, and $M_{i,j} = 1$ if $v_i v_j$ is a strong edge. Note that a trigraph H is symmetric if and only if its adjacency matrix is symmetric.

Email addresses: tomas@theory.stanford.edu (Tomás Feder), pavol@cs.sfu.ca (Pavol Hell), dschell@cs.sfu.ca (David G. Schell), stacho@cs.toronto.edu (Juraj Stacho)

¹This research was partially supported by P. Hell's NSERC Discovery Grant. The facilities at IRMACS SFU are gratefully acknowledged.

We consider the class of digraphs to be included in the class of trigraphs, by viewing each digraph H as a trigraph with the same vertex set $V(H)$, and with the weak edge set $W(H) = E(H)$ and strong edge set $S(H) = \emptyset$. Conversely, if H is a trigraph, the *associated digraph* of H is the digraph with the same vertex set $V(H)$, and with the edge set $E(H) = W(H) \cup S(H)$. Moreover, the *underlying graph* of the trigraph H is the underlying graph of the associated digraph, and the *symmetric graph* of the trigraph H is the symmetric graph of the associated digraph. To be specific, xy is an edge of the *underlying graph* of H just if $xy \in W(H) \cup S(H)$ or $yx \in W(H) \cup S(H)$, and xy is an edge of the *symmetric graph* of H just if $xy \in W(H) \cup S(H)$ and $yx \in W(H) \cup S(H)$. These conventions allow us to extend the usual graph and digraph terminology to trigraphs. We speak, for instance, of *adjacent vertices*, *components*, *neighbours*, *cutpoints*, or *bridges* of a trigraph H , meaning the corresponding notions in the associated digraph of H , or in its underlying graph; and we speak of *symmetric edges*, *symmetric neighbours*, etc. in a trigraph H , meaning the edges, neighbours, etc., in the symmetric graph of H . When we speak of a *subgraph* of a trigraph H , we mean a trigraph H' with $V(H') \subseteq V(H)$, $W(H') \subseteq W(H)$, and $S(H') \subseteq S(H)$. If $W(H') = W(H) \cap (V(H') \times V(H'))$ and $S(H') = S(H) \cap (V(H') \times V(H'))$, we say H' is the subgraph of H induced on $V(H')$.

Let G be a digraph and H a trigraph. A *homomorphism* of G to H is a mapping $f : V(G) \rightarrow V(H)$ such that the following two conditions are satisfied for each $u \neq v$:

- if $uv \in E(G)$ then $f(u)f(v) \in W(H) \cup S(H)$
- if $uv \notin E(G)$ then $f(u)f(v) \notin S(H)$.

In other words, edges of G must map to either weak or strong edges of H , and non-edges of G must map to either non-edges or weak edges of H .

If each vertex v of the digraph G has a *list* $L(v) \subseteq V(H)$, then a *list homomorphism* of G to H , with respect to the lists L (or *respecting* the lists L), is a homomorphism f of G to H such that $f(v) \in L(v)$ for all $v \in V(G)$. Following standard practice [20], we also call a homomorphism of G to H an *H -colouring* of G , and a list homomorphism of G to H (with respect to the lists L) a *list H -colouring* of G (with respect to L).

Suppose H is a fixed trigraph. The *H -colouring problem* $\text{HOM}(H)$ has as instances digraphs G , and asks whether or not G admits an H -colouring. The *list H -colouring problem* $\text{L-HOM}(H)$ has as instances digraphs G with lists L , and asks whether or not G admits a list H -colouring with respect to L . As noted earlier, H could be a digraph, viewed as a trigraph (with $W(H) = E(H)$, $S(H) = \emptyset$). Digraph homomorphism and list homomorphism problems have recently been of much interest [20]. If necessary, we will emphasize the distinction between *trigraph* list homomorphism problems and *digraph* list homomorphism problems, depending on whether H is a trigraph or a digraph respectively, i.e., whether H has any strong edges or not. However, note that the input G is always a digraph. We also note that loops in G are not considered in the definition of a trigraph homomorphism, and we assume throughout that G is irreflexive (has no loops).

For a fixed trigraph H , the list H -colouring problem $\text{L-HOM}(H)$ concerns the existence of vertex partitions of the input digraphs G . For instance, if H is the undirected trigraph with $V(H) = \{0, 1\}$, with a strong loop at 1 and a weak edge joining 0 and 1, then an H -colouring of G is precisely a partition of $V(G)$ into a clique and an independent set. Thus G is H -colourable if and only if G is a split graph [17]. Many graph partition problems, especially those arising from the theory of perfect graphs, can be formulated as

trigraph homomorphism (or list homomorphism) problems; this is discussed in detail in [9]. Equivalently, all these problems can be described in terms of the adjacency matrix of the trigraph, in the language of matrix partitions and list partitions (see [2, 9, 11, 13]). In this paper it will be more convenient to emphasize the trigraph (rather than the matrix) terminology, since we are dealing with the structure of the trigraph H .

It is generally believed [15] that for each digraph H the H -colouring problem $\text{HOM}(H)$ is NP -complete or polynomial time solvable. (This is equivalent to the so-called *CSP Dichotomy Conjecture* of Feder and Vardi [15].) One special case when the dichotomy conjecture is known to hold is the case of undirected graphs (i.e., symmetric digraphs) H . In this case, $\text{HOM}(H)$ is polynomial time solvable if H has a loop or is bipartite and is NP -complete otherwise [19]. For the list homomorphism problem $L\text{-HOM}(H)$ for an undirected graph H , it is shown in [8] that $L\text{-HOM}(H)$ is polynomial time solvable if H is a so-called *bi-arc graph*, and is NP -complete otherwise. Bi-arc graphs are a simultaneous generalization of reflexive interval graphs and bipartite graphs whose complements are circular arc graphs [8]. For general constraint satisfaction problems with lists, Bulatov [1] has proved that dichotomy holds. Most relevant for this paper is the following result explicitly classifying the complexity of $L\text{-HOM}(H)$ when H is a digraph.

Theorem 1. [21] *Let H be a digraph.*

If H contains a digraph asteroidal triple, then $L\text{-HOM}(H)$ is NP -complete. Otherwise $L\text{-HOM}(H)$ is polynomial time solvable.

Digraph asteroidal triples are introduced in [21]. Their definition is a bit technical, and is not needed in this paper; they are analogous to asteroidal triples in graphs [22]. The important fact is that testing if a given digraph H contains a digraph asteroidal triple can be performed in polynomial time [21].

By contrast, dichotomy is not known for trigraph list homomorphism problems. In [5], it is however proved that for each trigraph H , the list H -colouring problem is NP -complete or *quasi-polynomial* (of complexity $n^{O(\log^k n)}$). All list H -colouring problems $L\text{-HOM}(H)$ for trigraphs H with three or fewer vertices have been classified as NP -complete or polynomial time solvable in [13]. For symmetric trigraphs with four vertices, this has been accomplished in [2], with the exception of a single trigraph H ; the corresponding problem has earned the name *the stubborn problem*. This problem has only recently been established as polynomial time solvable [3]. Thus progress has been slow, even in showing the (polynomial / NP -complete) dichotomy for trigraph list homomorphism problems for concrete small trigraphs; the general dichotomy for trigraph list homomorphism problems seems quite uncertain.

In this paper, we prove dichotomy for the class of *trigraph trees*, i.e., for trigraphs whose underlying graph is a tree. It turns out that if H is a trigraph tree, then the list H -colouring problem is polynomially equivalent to a certain digraph list colouring problem. Specifically, let H^- be the *digraph* obtained from H by removing each vertex with a strong loop, and each strong edge xy , together with its converse yx , if $yx \in W(H) \cup S(H)$.

Theorem 2. *Let H be a trigraph tree.*

Then $L\text{-HOM}(H)$ is polynomially equivalent to $L\text{-HOM}(H^-)$.

Corollary 3. *Let H be a trigraph tree.*

Then $L\text{-HOM}(H)$ is polynomial time solvable or NP -complete.

By the same techniques we can prove the dichotomy for a larger class of trigraphs. The details of such a proof can be found in [12].

2. Trigraph trees

This and the next sections are devoted to a proof of Theorem 2. The proof is split into several steps. First, we prove Lemma 4 which will allow us to deal with the cases when H contains a strong edge or a pair of adjacent strong loops. Then we prove Lemma 5 which will help us handle the remaining cases of strong loops. Then the proof of the theorem is presented. Below we briefly list other tools that will be used in the proof.

Let H be a trigraph tree, and let G with lists $L(v)$, for $v \in V(G)$, be an instance of L-HOM(H). We say that the instance (G, L) is *arc-consistent* if for each $u, v \in V(G)$ and each $x \in L(u)$ there exists $y \in L(v)$ such that

- (i) $uv \in E(G) \Rightarrow xy \in W(H) \cup S(H)$ and $uv \notin E(G) \Rightarrow xy \notin S(H)$
- (ii) $vu \in E(G) \Rightarrow yx \in W(H) \cup S(H)$ and $vu \notin E(G) \Rightarrow yx \notin S(H)$.

If the graph G is clear from the context, we just say that the lists L are arc-consistent.

If the lists L are not arc-consistent, we can in polynomial time modify L to arc-consistent lists L' such that G has a homomorphism to H that respects lists L if and only if G has a homomorphism to H that respects lists L' . This is similar to enforcing arc-consistency in digraphs, cf. [2, 9]. Thus, if convenient, we shall assume that the lists L are arc-consistent. Note that if the lists L are arc-consistent, then either all lists $L(v)$ are empty, or all are non-empty.

Let S be a subset of $V(H)$. We say that the instance (G, L) *contains representatives for S* if for each $x \in S$, there exists $v \in V(G)$ with $L(v) = \{x\}$. We say that the instance (G, L) *contains representatives* if it contains representatives for $V(H)$. (We shall again refer to just lists L containing representatives if G is clear from the context.)

Note that if the instance (G, L) does not contain representatives, we can, in polynomial time, construct a collection of at most $(|V(G)| + 1)^{|V(H)|}$ instances (G, L_i) , for $i = 1 \dots t$, that all contain representatives (for some subsets of $V(H)$), and have the following additional property. G has a homomorphism to H that respects the lists L if and only if G has a homomorphism to H that respects the lists L_i for some i . This is again similar to the case of digraphs [2, 9]. In particular, for each vertex $x \in V(H)$, we either select a vertex of G and set its list to $\{x\}$, or we remove x from all lists in G . The resulting lists L_i contain representatives for some subset S_i of $V(H)$. Moreover, the vertices of $H - S_i$ do not appear in L_i . Therefore, the resulting instance is, in fact, an instance of L-HOM(H'), where H' is the subgraph of H induced on S_i . For the problem L-HOM(H'), the lists L_i contain representatives. Thus, if convenient, we shall assume that the lists L contain representatives.

Now, let x, y be adjacent vertices in H . Let H^{xy} denote the subgraph of H obtained by removing all edges between x and y . Note that since H is a trigraph tree, H^{xy} is a trigraph with precisely two connected components, one containing x and one containing y .

Let G^{xy} denote the subgraph of G obtained from G by removing all edges uv with $x \in L(u)$ and $y \in L(v)$ if $xy \in W(H) \cup S(H)$, and removing all edges uv with $y \in L(u)$ and $x \in L(v)$ if $yx \in W(H) \cup S(H)$.

We say that the instance (G, L) is *subgraph-consistent on H^{xy}* if for all connected components K of H^{xy} and C of G^{xy} , either $L(v) \cap K = \emptyset$ for all $v \in C$, or there is a homomorphism of C to K respecting the lists L . (We shall again say briefly that the lists L are subgraph-consistent if the graph G is clear from the context.)

Again, if convenient, we may assume that the lists L are subgraph-consistent on H^{xy} . If they are not, we can make them subgraph-consistent on H^{xy} by testing for each connected component K of H^{xy} and C of G^{xy} , whether there exists a homomorphism of C to K respecting the lists L . If this test fails for some choice of C and K , we remove the elements of K from the lists of all vertices of C . The resulting instance of L-HOM(H) has a solution if and only if the original instance does. To see this, note that if f is a homomorphism of G to H respecting the lists L , then f is also a homomorphism of G^{xy} to H^{xy} . If f maps a vertex of C to a vertex of K , then every vertex of C is mapped to K by f , since homomorphisms map connected components to connected components. Thus f restricted to C is a homomorphism of C to K respecting the lists L , a contradiction.

Furthermore, if convenient, we may assume that the instance (G, L) is simultaneously arc-consistent, subgraph-consistent on H^{xy} , and contains representatives. To see this, we first ensure that (G, L) contains representatives. Then we alternatively make the instance arc-consistent, and subgraph-consistent on H^{xy} , until the lists stop changing. (Since each step only removes elements from lists, this requires at most $|V(G)||V(H)|$ iterations.) The resulting lists are either arc-consistent, subgraph-consistent on H^{xy} , and contain representatives, or they are all empty. Note that if the list homomorphism problem for H^{xy} is polynomially solvable, then the above procedure is polynomial.

3. Separable instances

The following tool will allow us to handle the cases with strong edges and adjacent strong loops. Let x, y be adjacent vertices of H . Let A, B denote the two connected components of H^{xy} where $x \in A$ and $y \in B$.

We say that the instance (G, L) is (or sometimes briefly just that the lists L are) *separable on x, y* if for all $v \in V(G)$:

(*) if $x \in L(v)$, then $L(v) \cap A = \{x\}$ and if $y \in L(v)$ then $L(v) \cap B = \{y\}$.

(In other words, the instance (G, L) is separable on x, y if no list in G contains x with another element of A , or y with another element of B .)

Lemma 4. *Let H be a trigraph tree. Let x, y be adjacent vertices in H . If L-HOM(H^{xy}) is polynomial time solvable, then L-HOM(H), when restricted to instances (G, L) that are separable on x, y , is also polynomial time solvable.*

Proof. We shall describe a polynomial algorithm for the restricted problem L-HOM(H). Let (G, L) be an instance of L-HOM(H), separable on x, y . Assume that the instance is arc-consistent and subgraph-consistent on H^{xy} . Note that if it is not, then the procedure making it both arc-consistent and subgraph-consistent on H^{xy} cannot violate (*), since it only removes elements from lists.

Let H_1 denote the subgraph of H induced on $\{x, y\}$. Let H_0 denote the trigraph constructed from H_1 by attaching a new vertex a , with a weak loop, adjacent to x by a weak symmetric edge, and a new vertex b , with a weak loop, adjacent to y by a weak

symmetric edge. Let L_0 denote the lists obtained from L by replacing each element of $A \setminus \{x\}$ by a and each element of $B \setminus \{y\}$ by b . Let φ be the mapping of $V(H)$ to $V(H_0)$ that maps x to x , y to y , each vertex of $A \setminus \{x\}$ to a , and each vertex of $B \setminus \{y\}$ to b . Observe that if there is a homomorphism f of G to H respecting the lists L , then there is a homomorphism of G to H_0 respecting the lists L_0 , namely $\varphi \circ f$. We show that the converse is also true.

(1) *If there is a homomorphism of G to H_0 respecting the lists L_0 , then there is also a homomorphism of G to H respecting the lists L .*

Suppose that g is a homomorphism of G to H_0 respecting the lists L_0 . First, we observe the following property of g :

(2) *If C is a component of G^{xy} , then either $g(C) \subseteq \{a, x\}$ or $g(C) \subseteq \{y, b\}$.*

By definition of G^{xy} , the homomorphism g cannot map an edge of G^{xy} to xy or yx . The image $g(C)$ of a component C of G^{xy} is connected, and hence cannot contain both a vertex in a, x and a vertex in b, y . This proves (2).

The property (2) ensures that there exists a homomorphism f of G^{xy} to H^{xy} , respecting the lists L , such that if g maps a component C of G^{xy} to $\{a, x\}$, then f maps C to A ; otherwise, f maps C to B . Note that such a homomorphism f is guaranteed by our assumption that the lists L are subgraph-consistent on H^{xy} . We will show that f is, in fact, a homomorphism of G to H which will prove (1).

First, we observe that $g = \varphi \circ f$. Indeed, suppose that $g(v) = x$ for some $v \in V(G)$. Then $x \in L_0(v)$, and $x \in L(v)$, which according to (*) means $L(v) \cap A = \{x\}$. Further, by (2) and the definition of f , we have $f(v) \in A$. (In fact, the entire connected component of G^{xy} containing v maps to A .) Therefore $f(v) = x$. Similarly, from $g(v) = y$, we conclude $f(v) = y$. Now, suppose that $g(v) = a$. It again follows that $f(v) \in A$, and also $a \in L_0(v)$, which implies that $L(v) \cap A \setminus \{x\} \neq \emptyset$. This yields, by (*), that $x \notin L(v)$, and hence, $f(v) \in A \setminus \{x\}$. Similarly, if $g(v) = b$, we obtain $f(v) \in B \setminus \{y\}$. It now follows from the definition of φ that $g = \varphi \circ f$.

Now, suppose that f is not a homomorphism of G to H . This means there is an edge $uv \in E(G)$ with $f(u)f(v) \notin W(H) \cup S(H)$, or there is a non-edge $uv \notin E(G)$ with $f(u)f(v) \in S(H)$. First, consider an edge $uv \in E(G)$ with $f(u)f(v) \notin W(H) \cup S(H)$. Since g is a homomorphism of G to H_0 , we have $g(u)g(v) \in W(H_0) \cup S(H_0)$. Also, f is a homomorphism of G^{xy} to H^{xy} . So, if we had $uv \in E(G^{xy})$, then $f(u)f(v) \in W(H^{xy}) \cup S(H^{xy})$ which implies $f(u)f(v) \in W(H) \cup S(H)$, contrary to our assumption. Thus, $uv \in E(G) \setminus E(G^{xy})$. This implies, by the definition of G^{xy} , that $x \in L(u)$ and $y \in L(v)$, or that $y \in L(u)$ and $x \in L(v)$.

We assume that $x \in L(u)$ and $y \in L(v)$. (The proof when $y \in L(u)$ and $x \in L(v)$ is similar.) From this, we conclude, by (*), that $L(u) \cap A = \{x\}$ and $L(v) \cap B = \{y\}$. In particular, $f(u) \notin A \setminus \{x\}$ and $f(v) \notin B \setminus \{y\}$, since f respects the lists L . If $f(u), f(v) \in \{x, y\}$, then $f(u) = g(u)$ and $f(v) = g(v)$, since $g = \varphi \circ f$. Hence, $f(u)f(v) \in W(H) \cup S(H)$ as $g(u)g(v) \in W(H_0) \cup S(H_0)$. This contradicts our assumption, thus f maps at least one of u, v outside of $\{x, y\}$. Suppose that $f(u) \notin \{x, y\}$. This implies $f(u) \in B \setminus \{y\}$, since $f(u) \notin A \setminus \{x\}$. Hence, $g(u) = b$ because $g = \varphi \circ f$. Thus $g(v) \in \{y, b\}$, since y, b are the only neighbours of b in H_0 and $g(u)g(v) \in W(H_0) \cup S(H_0)$. Hence, $f(v) \in B$ which yields $f(v) = y$, since $f(v) \notin B \setminus \{y\}$. We now show that this

contradicts the arc-consistency of the lists L . That is, we show that there does not exist a $w \in L(v)$ with $f(u)w \in W(H) \cup S(H)$. Otherwise, if such a w exists, we conclude that $w \in B$, because $f(u) \in B \setminus \{y\}$ and so $f(u)$ has only neighbours in B . Therefore, $w \in L(v) \cap B$ and $L(v) \cap B = \{y\}$, and hence $w = y$. This is a contradiction as $w = y = f(v)$ and $f(u)f(v) \notin W(H) \cup S(H)$ by our assumption. So, no such w exists and we must conclude $f(u) \notin B \setminus \{y\}$. By a similar argument, we can also conclude $f(v) \notin A \setminus \{x\}$. Clearly, this is impossible, since we assume that f maps at least one of u, v outside of $\{x, y\}$. This shows that no such edge uv exists.

Now, suppose that some $uv \notin E(G)$ has $f(u)f(v) \in S(H)$. Note that $uv \notin E(G^{xy})$ and hence, $f(u)f(v) \notin S(H^{xy})$. This implies that $f(u), f(v) \in \{x, y\}$, since f is a homomorphism of G^{xy} to H^{xy} and the only edges we removed from H to obtain H^{xy} were between x and y . Since $g = \varphi \circ f$, we conclude that $f(u) = g(u)$ and $f(v) = g(v)$. So, $g(u)g(v) = f(u)f(v) \in S(H)$ contradicting the fact that g is a homomorphism. This proves that f is indeed a homomorphism of G to H which proves (1).

Now, from (1), we can conclude that to solve L-HOM(H) for the digraph G with lists L in polynomial time, it suffices to find in polynomial time a homomorphism g of G to H_0 that respects lists L_0 . We remark that (*) implies that the lists L_0 are all of size at most two, namely $\{x, y\}$, $\{x, b\}$, $\{y, a\}$, $\{a, b\}$, and their subsets. Thus finding g can be reduced to an instance of 2SAT which can be solved in polynomial time. \square

4. Restricting strong loops

For the cases when H contains strong loops, we shall make use of the following lemma. It explains how to reduce every instance of L-HOM(H) to a set of polynomially many restricted instances which are easier to handle.

We say that the instance (G, L) *contains representatives for strong loops*, if for every vertex x of H with a strong loop, there exists $y \in V(H)$ and $Q_x \subseteq V(G)$ such that

(\star) $L(v) \subseteq \{x, y\}$ for all $v \in Q_x$, and $x \notin L(v)$ for all $v \notin Q_x$.

Note that we allow $y = x$, and as before, if G is clear from the context, we say that the lists L contain representatives for strong loops.

Lemma 5. *Let H be a trigraph tree, and let (G, L) be an instance of L-HOM(H) that is arc-consistent and contains representatives. Then there exists a polynomial collection of instances (G, L_i) , for $i = 1 \dots t$, each containing representatives for strong loops, with the following additional property. G has a homomorphism to H that respects the lists L if and only if G has a homomorphism to H that respects the lists L_i for some i . Moreover, the collection can be computed in polynomial time.*

Proof. Let G' be the symmetric graph of G . (Recall that G' has edge uv just if both uv and vu are in $W(H) \cup S(H)$.) Let G'' be a minimal chordal completion [24] of G' .

Suppose that f is a homomorphism of G to H respecting the lists L . Consider a vertex x of H with a strong loop. Then the set $X = f^{-1}(x)$ induces a symmetric clique in G , that is, a clique in G' . It follows that there exists a maximal clique Q of G'' such that $f(v) = x$ implies $v \in Q$. (In other words, Q is a maximal clique of G'' that contains X .)

Recall that the lists L contain representatives. So, if v_x is a representative for x , i.e., $L(v_x) = \{x\}$, then $v_x \in X$, and, by the arc-consistency of L , all vertices of X are symmetric neighbours of v_x . We let Q_x be the set of vertices consisting of v_x and its symmetric neighbours in Q , and conclude $X \subseteq Q_x$.

If $X = Q_x$, then we let $y = x$, and conclude that $f(v) \in \{x, y\}$ if $v \in Q_x$, and $f(v) \neq x$ if $v \notin Q_x$. Otherwise, if $X \neq Q_x$, we prove that f maps the vertices of $Q_x \setminus X$ to at most one connected component of $H - x$. To see this, suppose otherwise and let $u, v \in Q_x \setminus X$ be such that $f(u)$ and $f(v)$ are in different connected components of $H - x$. If P is a path in $G - X$ between u and v , then the image of P under f is a walk in H , since f is a homomorphism. Thus, P contains a vertex w with $f(w) = x$. So $w \in X$, contradicting the definition of P . Thus, u and v are in different connected components of $G - X$, and hence, in different connected components of $G' - X$. However, this is not possible, since u, v are adjacent in G'' yet separated by the clique X in G' . (No minimal chordal completion of a graph adds edges between vertices that are separated by a clique [24].)

We now let K be the (unique) connected component of $H - x$ to which f maps the vertices of $Q_x \setminus X$. Since H is a trigraph tree, we let y be the unique neighbour of x in K . By the arc-consistency of L , we conclude that y is a symmetric neighbour of x , and that $f(v) = y$ for all $v \in Q_x \setminus X$. Thus, $f(v) \in \{x, y\}$ if $v \in Q_x$, and $f(v) \neq x$ if $v \notin Q_x$.

It now follows that we can restrict the lists $L(v)$ to $\{x, y\}$ for all $v \in Q_x$, remove x from the lists of $v \notin Q_x$, and the homomorphism f still respects the modified lists L . Applying, this argument for all strong loops of H , we eventually obtain lists that contain representatives for strong loops, and the homomorphism f respects these lists.

It remains to show that there are only polynomially many different lists we can obtain this way. We observe that there are only linearly many choices for the clique Q , and thus, only linearly many choices for Q_x . Further, there are only constantly many choices for y and constantly many strong loops in H . Thus, the resulting collection of lists is of polynomial size. Finally, we note that it can also be constructed in polynomial time, since a minimal chordal completion of a graph can be computed in polynomial time [24].

That concludes the proof. \square

This lemma allows us, if convenient, to assume that the lists L contain representatives for strong loops. Moreover, as before, we may assume that the lists L simultaneously contain representatives, contain representatives for strong loops, and are arc-consistent and subgraph consistent on H^{xy} for some edge xy of H . This is possible, since making the lists arc-consistent and subgraph-consistent on H^{xy} cannot violate (\star) or remove representatives (assuming a homomorphism respecting lists L exists). Further, if L-HOM(H^{xy}) is polynomially solvable, this procedure runs in polynomial time.

Now, we are finally ready to present a proof of Theorem 2.

5. Proof of Theorem 2

First, we observe that the connected components of H^- are induced subgraphs of H , since H is a trigraph tree. So, if L-HOM(H) is polynomial time solvable, then so is L-HOM(H^-). To see this, note that if f is a homomorphism of G to H^- , then f maps each connected component C of G to some connected component K of H^- . We consider all possible choices for K and try to find a homomorphism of C to K respecting the lists L .

We do this by restricting the lists of the vertices in C to K and solving L-HOM(H) for this instance. This results in at most $|V(G)||V(H)|$ instances of L-HOM(H).

Now, for the converse, we shall assume, by Theorem 1, that L-HOM(H^-) is polynomial time solvable, and show how to solve L-HOM(H) in polynomial time.

We proceed by induction on the number of strong edges and strong loops in H . If H contains no strong edge or strong loop, then $H = H^-$ and there is nothing to prove. So, we assume that H contains at least one strong edge or strong loop.

First, assume that H contains a strong edge xy . Consider an instance of L-HOM(H), that is, a digraph G with lists $L(v)$, for $v \in V(G)$. Note that $H^- = (H^{xy})^-$. This implies that L-HOM(H^{xy}) is polynomially equivalent to L-HOM(H^-) by induction, and consequently, L-HOM(H^{xy}) is polynomially solvable. This allows us to assume that the lists L are arc-consistent, subgraph-consistent on H^{xy} , and contain representatives.

If for some (and hence all) $v \in V(G)$, the list $L(v)$ is empty, the instance has no list homomorphism to H . Otherwise, we shall use Lemma 4 to solve L-HOM(H) for G with lists L . To do that, we need two observations. Firstly, we recall that L-HOM(H^{xy}) is polynomially solvable. Secondly, we show that the lists L are separable on x, y . Let A and B be the two components of H^{xy} where $x \in A$ and $y \in B$. Suppose otherwise, and first let v be a vertex of G with $\{a, x\} \subseteq L(v)$ where $a \in A \setminus \{x\}$. Recall that the lists L are arc-consistent and contain representatives. Let v_y be a representative for y , i.e., $L(v_y) = \{y\}$. We note that the arc-consistency of L implies that $vv_y \in E(G)$, since $x \in L(v)$ and $xy \in S(H)$. But then, again by arc-consistency, and the fact that $ay \notin W(H) \cup S(H)$, we conclude that $a \notin L(v)$, a contradiction. The same argument proves the case when $y \in L(v)$. This shows that the lists L are separable on x, y . Therefore, using Lemma 4, we solve L-HOM(H) in polynomial time.

Next, suppose that H has a strong loop, say on a vertex x , but has no strong edges. Consider an instance (G, L) of L-HOM(H), and assume that it is arc-consistent, contains representatives, and contains representatives for strong loops (see Lemma 5).

Then there exists $y \in V(H)$ and a set $Q_x \subseteq V(G)$ such that

(\star) $L(v) \subseteq \{x, y\}$ for all $v \in Q_x$, and $x \notin L(v)$ for all $v \notin Q_x$.

In fact, by the proof of Lemma 5, we may assume that y is a symmetric neighbour of x , i.e., that $xy, yx \in W(H) \cup S(H)$. Suppose that $y = x$. Then (\star) implies that $L(v) = \{x\}$ for all $v \in Q_x$, and $x \notin L(v)$ for all $v \notin Q_x$. Thus G has a homomorphism to H respecting the lists L if and only if $G - Q_x$ has a homomorphism to $H - x$ respecting the lists L . Such a homomorphism can be found in polynomial time by induction.

Thus, we may assume that $y \neq x$. So, since we assume that H contains no strong edges, we have $xy, yx \in W(H)$. Further, since $y \neq x$, we may assume that the lists L are, in addition, subgraph-consistent on H^{xy} . (See the discussion after Lemma 5.) For this, note that L-HOM(H^{xy}) is polynomially solvable by induction, since $(H^{xy})^- = H^-$.

There are three remaining cases:

Case 1: Suppose that y has no loop. If f is a homomorphism of G to H respecting the lists L , then, by (\star), Q_x partitions into a symmetric clique and an independent set. Namely, the symmetric clique is formed by the vertices $v \in Q_x$ with $f(v) = x$ and the independent set are the vertices $v \in Q_x$ with $f(v) = y$. In other words, the subgraph of the underlying graph of G induced on Q_x is a split graph [17]. Note that one can test if a

graph is a split graph in linear time, and there are only linearly many different partitions of a split graph into a clique and an independent set [17]. Thus, by trying all possibilities, we consider each a partition of Q_x into $Q_1 \cup Q_2$ where Q_1 induces a symmetric clique in G and Q_2 is an independent set in G . For this partition, we construct lists L' by setting $L'(v) = \{x\}$ for each $v \in Q_1$, $L'(v) = \{y\}$ for each $v \in Q_2$, and $L'(v) = L(v)$ for $v \notin Q_x$. Afterwards, we make the lists L' arc-consistent. It follows that G has a homomorphism to H respecting the lists L if and only if $G - Q_x$ has a homomorphism to $H - x$ respecting the lists L' for some choice of the partition $Q_1 \cup Q_2$. The existence of such a homomorphism can be again decided in polynomial time by induction.

Case 2: Suppose that y has a weak loop. In this case, we proceed similarly to Lemma 4 and construct a homomorphism directly. Recall that A, B are the two connected components of H^{xy} where $x \in A$ and $y \in B$. We let f be a homomorphism of G^{xy} to H^{xy} such that if C is a connected component of G^{xy} then f maps C to B if possible, otherwise, f maps C to A . We emphasize that f only maps C to A if it is not possible to map it to B . Such a homomorphism is guaranteed by the subgraph-consistency of L .

We show that f is also a homomorphism of G to H . Suppose otherwise, and first, let uv be an edge of G with $f(u)f(v) \notin W(H) \cup S(H)$. Clearly, $uv \notin E(G^{xy})$ because f is a homomorphism of G^{xy} to H^{xy} . So, by the definition of G^{xy} , we conclude that $x \in L(u)$ and $y \in L(v)$, or that $y \in L(u)$ and $x \in L(v)$.

We shall assume $x \in L(u)$ and $y \in L(v)$. (The proof in the latter case is similar.) First, we conclude $f(v) \in B$ from the subgraph-consistency of L , the definition of f , and the fact that $y \in L(v)$. (In fact, f maps to B the whole component of G^{xy} that contains v .) Also, since $x \in L(u)$, we conclude $u \in Q_x$, and hence, $L(u) \subseteq \{x, y\}$ by (\star) . Note that no vertex of $B \setminus \{y\}$ is adjacent to x . So, since $f(v) \in B$ and y has a weak loop, we conclude, by arc-consistency, that $yf(v) \in W(H) \cup S(H)$. This yields $f(u) \neq y$, and hence, $f(u) = x$ since f respects the lists L . But now $y \notin L(u)$, since otherwise $f(u) \in B$ by the definition of f . So, $L(u) = \{x\}$ and the arc-consistency of L implies that $f(v) = y$. (Recall that $f(v) \in B$ and x is not adjacent to any vertex in $B \setminus \{y\}$.) This, however, is a contradiction, since $f(u)f(v) = xy$ and $xy \in W(H)$.

So, no such edge uv exists. Therefore, there is $uv \notin E(G)$ with $f(u)f(v) \in S(H)$. In particular, $uv \notin E(G^{xy})$ since G^{xy} is a subgraph of G . So, $f(u)f(v) \notin S(H^{xy})$ since f is a homomorphism of G^{xy} to H^{xy} . However, $S(H^{xy}) = S(H)$, and hence, $f(u)f(v) \notin S(H)$, a contradiction.

This proves that f is indeed a homomorphism of G to H as claimed earlier, and we constructed it in polynomial time.

Case 3: Suppose that y has a strong loop. In this case, since the lists L contain representatives for strong loops, we have a set $Q_y \subseteq V(G)$ and a vertex $y' \in V(H)$ such that $L(v) \subseteq \{y, y'\}$ for $v \in Q_y$, and $y' \notin L(v)$ for $v \notin Q_y$.

If $y' \neq x$, then we have $L(v) \subseteq \{x, y\} \cap \{y, y'\} = \{y\}$ for all $v \in Q_x \cap Q_y$, and $L(v) = \{x\}$ for all $v \in Q_x \setminus Q_y$. Thus, by arc-consistency of L , we may conclude that G has a homomorphism to H respecting the lists L if and only if $G - Q_x$ has a homomorphism to $H - x$ respecting the lists L . By induction, the existence of such a homomorphism can again be decided in polynomial time.

Therefore, we may assume that $y' = x$. We now again use Lemma 4. To do that, we observe the vertices $Q_x \cup Q_y$ are the only vertices with x or y on their lists, and $L(v) \subseteq \{x, y\}$ for each $v \in Q_x \cup Q_y$. Recall that A, B are the two connected components

of H^{xy} where $x \in A$ and $y \in B$. So, if $x \in L(v)$, then $L(v) \cap A = \{x\}$ since $L(v) \subseteq \{x, y\}$. Similarly, $y \in L(v)$ implies $L(v) \cap B = \{y\}$. This shows that the lists L are separable on x, y . Further, $L\text{-HOM}(H^{xy})$ is decidable in polynomial time by induction. Thus, using Lemma 4, we can solve $L\text{-HOM}(H)$ for G with lists L in polynomial time.

That concludes the proof. \square

6. Extensions

In the previous sections, we have focused on the list homomorphism problems for trigraphs H whose underlying graph is a tree. It turns out that the tools we used for this characterization work in a more general context. In particular, we can describe a more general class of trigraphs H for which an analogue of Theorem 2 is true. Unfortunately, the description of these trigraphs is quite technical; the interested reader can find the definition and the proofs in our preprint on arxiv.org [12].

Here, we only illustrate how this result applies to *trigraph cycles*, i.e., trigraphs H whose underlying graph is a cycle. We assume that H contains at least one strong edge or loop, otherwise, it is covered by Theorem 1.

We say that a trigraph cycle H is a *good cycle* if H has at least one of the following:

- (i) two strong edges, or
- (ii) three consecutive strong loops, or
- (iii) two pairs of consecutive strong loops, or
- (iv) a strong edge and a distinct pair of consecutive strong loops, or
- (v) two strong loops joined by a nonsymmetric edge, or
- (vi) a strong loop whose both neighbours have no loops, or
- (vii) a strong loop having non-symmetric edges to its both neighbours, or
- (viii) a strong edge whose (at least) one endpoint has no loop.

Theorem 6. [12] *Let H be a good cycle.*

Then the problem $L\text{-HOM}(H)$ is polynomial time solvable or NP-complete.

The theorem covers almost all cases of trigraph cycles. In Figure 1, we list all the remaining cases of trigraph cycles not covered by the above theorem. These are trigraph cycles H that contain vertices x, y such that either $xy \in S(H)$ and $xx, yy \in W(H)$, or all of xx, xy, yx , and yy are edges (weak or strong) but at least one is strong. (Only three typical cases of this are shown in the figure.) All other edges and loops not shown are weak. As far as we see, the complexity of the corresponding problems $L\text{-HOM}(H)$ in these cases is open; but we have not explicitly considered these problems.

7. Conclusions

We have shown that dichotomy holds for trigraph trees. By combining Theorem 1 and Theorem 2, we obtain an algorithm that decides, in time polynomial in the size of the trigraph H , whether $L\text{-HOM}(H)$ is NP-complete or polynomial time solvable.

Certain generalizations are similarly handled in [12]; this includes most trigraph cycles, with the few exceptions illustrated in the Figure 1. For small trigraphs, all list homomorphism problems for symmetric trigraphs with up to four vertices have now been classified as polynomial time solvable or NP-complete by a combined effort of [2, 3, 4, 9].

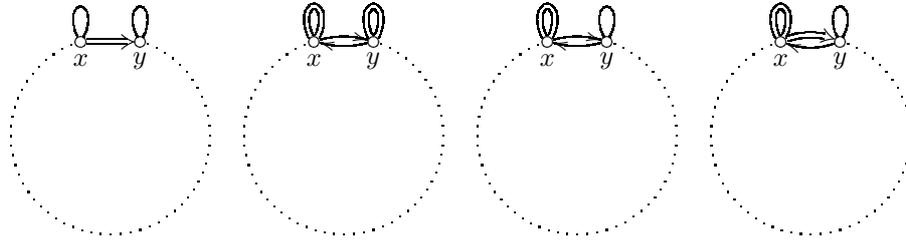


Figure 1: Example trigraph cycles H for which the complexity of $L\text{-HOM}(H)$ is left open.

References

- [1] A. A. BULATOV, *Complexity of conservative constraint satisfaction problems*, ACM Transactions on Computational Logic, (2011). to appear.
- [2] K. CAMERON, E. M. ESCHEN, C. T. HOÀNG, AND R. SRITHARAN, *The complexity of the list partition problem for graphs*, SIAM Journal on Discrete Mathematics, 21 (2007), pp. 900–929.
- [3] M. CYGAN, M. PILIPCZUK, M. PILIPCZUK, AND J. O. WOJTASZCZYK, *The stubborn problem is stubborn no more (a polynomial algorithm for 3-compatible colouring and the stubborn list partition problem)*, in Proceedings of the twenty second annual ACM-SIAM symposium on Discrete algorithms (SODA), 2011, pp. 1666–1674.
- [4] C. M. H. DE FIGUEIREDO AND S. KLEIN, *Finding skew partitions efficiently*, Journal of Algorithms, 37 (2000), pp. 505–521.
- [5] T. FEDER AND P. HELL, *Full constraint satisfaction problems*, SIAM Journal on Computing, 36 (2006), pp. 230–246.
- [6] T. FEDER AND P. HELL, *Matrix partitions of perfect graphs*, Discrete Mathematics, 306 (2006), pp. 2450–2460.
- [7] T. FEDER, P. HELL, AND W. HOCHSTÄTTLER, *Generalized colourings of cographs*, in Graph Theory in Paris, Birkhauser Verlag, 2006, pp. 149–167. invited chapter.
- [8] T. FEDER, P. HELL, AND J. HUANG, *Bi-arc graphs and the complexity of list homomorphisms*, Journal of Graph Theory, 42 (2003), pp. 61–80.
- [9] T. FEDER, P. HELL, S. KLEIN, AND R. MOTWANI, *List partitions*, SIAM Journal on Discrete Mathematics, 16 (2003), pp. 449–478.
- [10] T. FEDER, P. HELL, S. KLEIN, L. T. NOGUEIRA, AND F. PROTTI, *List matrix partitions of chordal graphs*, Theoretical Computer Science, 349 (2005), pp. 52–66.
- [11] T. FEDER, P. HELL, D. KRÁL’, AND J. SGALL, *Two algorithms for general list matrix partitions*, in Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA), 2005, pp. 870–876.
- [12] T. FEDER, P. HELL, D. G. SCHELL, AND J. STACHO, *Dichotomy for tree-structured trigraph list homomorphism problems*. Preprint <http://arxiv.org/abs/1009.0358>.
- [13] T. FEDER, P. HELL, AND K. TUCKER-NALLY, *Digraph matrix partitions and trigraph homomorphisms*, Discrete Applied Mathematics, 154 (2006), pp. 2458–2469.
- [14] T. FEDER, P. HELL, AND W. XIE, *Matrix partitions with finitely many obstructions*, Electronic Journal of Combinatorics R58, 14 (2007).
- [15] T. FEDER AND M. Y. VARDI, *The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory*, SIAM Journal on Computing, 28 (1999), pp. 57–104.
- [16] D. R. FULKERSON AND O. A. GROSS, *Incidence matrices and interval graphs*, Pacific Journal of Mathematics, 15 (1965), pp. 835–855.
- [17] P. L. HAMMER AND B. SIMEONE, *The splittance of a graph*, Combinatorica, 1 (1981), pp. 275–284.
- [18] P. HELL, S. KLEIN, L. T. NOGUEIRA, AND F. PROTTI, *Partitioning chordal graphs into independent sets and cliques*, Discrete Applied Mathematics, 141 (2004), pp. 185–194.
- [19] P. HELL AND J. NEŠETŘIL, *On the complexity of H -colouring*, Journal of Combinatorial Theory B, 48 (1990), pp. 92–110.
- [20] ———, *Graphs and Homomorphisms*, Oxford University Press, 2004.
- [21] P. HELL AND A. RAFIEY, *The dichotomy of list homomorphisms for digraphs*, in Proceedings of the twenty second annual ACM-SIAM symposium on Discrete algorithms (SODA), 2011, pp. 1703–1713.

- [22] C. G. LEKKERKERKER AND J. C. BOLAND, *Representation of a finite graph by a set of intervals on the real line*, *Fund. Math.*, 51 (1962), pp. 45–64.
- [23] D. G. SCHELL, *Matrix partitions of digraphs*, Master's thesis, Simon Fraser University, 2008.
- [24] R. E. TARJAN, *Decomposition by clique separators*, *Discrete Mathematics*, 55 (1985), pp. 221–232.