List-Homomorphism Problems on Graphs and Arc Consistency

Benoit Larose  
Department of Mathematics and Statistics  
Concordia University  
Montréal, Canada  
larose@mathstat.concordia.ca

Adrien Lemaitre  
Département d’informatique et de recherche opérationnelle  
Université de Montréal  
Montréal, Canada  
lemaitre@iro.umontreal.ca

Abstract—We characterise the graphs (which may contain loops) whose list-homomorphism problem is solvable by arc-consistency, or equivalently, that admit conservative totally symmetric idempotent operations of all arities. We prove that for every bipartite graph $G$, its list-homomorphism problem is tractable if and only if $G$ admits a monochromatic conservative semilattice operation; in particular, its list-homomorphism problem can easily be solved by a combination of two-colouring and arc-consistency.

Keywords—List-homomorphism problems; arc-consistency; totally symmetric operations; symmetric Datalog; graphs.

I. INTRODUCTION

In recent years, the use of universal algebraic methods has become prevalent in the study of the complexity of fixed-target constraint satisfaction problems (CSPs) (see for instance [7], [6]). Such CSPs can be most easily viewed as homomorphism problems: a target structure $H$ is fixed, and one must decide if a given input structure $G$ admits a homomorphism to $H$. Although the general CSP is NP-complete, the tractability of the fixed-target CSP depends on the nature of the target structure; great strides have recently been made to better understand the structures underlying tractable CSPs, motivated in great part by the dichotomy conjecture that states that every CSP is either NP-complete or in P [12]. Recent progress has shown that the algebra of polymorphisms of the target structure controls the complexity of the CSP, and that in all known cases, the nicer the identities satisfied by this algebra, the simpler the decision problem turns out to be. Precise, general conjectures have been proposed [5], [14] which have been verified in various special cases [2], [3], [4].

In their seminal paper [12], Feder and Vardi give a series of reductions of the general problem to various specific kinds of structures. In particular, they show that every fixed target CSP is polynomially equivalent to a retraction problem on a bipartite graph, and to a retraction problem on a reflexive graph. It is thus of interest to understand the complexity of graph retraction problems. Similarly, list-homomorphism problems have attracted a great deal of attention: although Bulatov has settled the dichotomy question there [4], it is still of great interest to understand the refined complexity of these CSPs. The above mentioned conjectures are still wide open in this case, although they have been completely settled recently in the case of list-homomorphism problems for graphs [9]. In this setting, Feder and Hell [10] asked which of these CSPs have tree duality, or equivalently are solvable by arc-consistency methods. It follows from results in [12] and [8] that this boils down to a purely algebraic question, namely on the existence of totally symmetric operations of all arities, see section II-D below. Since no commutative, idempotent operation can preserve a loopless edge, bipartite graphs don’t admit such operations, and their associated CSP cannot have tree duality. On the other hand one may consider a partial form of tree duality by restricting inputs to be bipartite graphs themselves. Viewed differently, this allows one to solve the associated CSP for a bipartite graph by using a simple combination of 2-colouring (to check if the input is bipartite) and then arc-consistency. We introduce the notion of monochromatic polymorphism to characterise graphs with this property (Theorem 3.2). In the list-homomorphism case we show that all tractable bipartite cases are solvable by arc-consistency on bipartite instances (Lemma 4.2). These ideas allow us to settle Feder and Hell’s question (Theorem 4.8): a graph $H$ has its list-homomorphism problem solvable by arc-consistency if and only if it is a bi-arc graph that contains no loopless edge.

II. PRELIMINARIES


A. Relational structures and decision problems.

A signature is a finite set of relation symbols, each with an associated arity. A $\sigma$-structure $T$ consists of a set $T$ (the universe of $T$) and a relation $R(T)$ on $T$ of the corresponding arity for each relation symbol $R$ of $\sigma$. In this paper, we always denote the universe of a structure by its roman equivalent, e.g. the universe of $\mathbb{S}$ is $S$, and so forth.

Let $G$ and $H$ be $\sigma$-structures. A homomorphism from $G$ to $H$ is a function $f : G \to H$ such that for each $n$-ary relation $R$ of $\sigma$ and every tuple $(v_1, \ldots, v_n) \in R(G)$, the tuple $(f(v_1), \ldots, f(v_n))$ is in $R(H)$. We write $G \to H$ to indicate there exists a homomorphism from $G$ to $H$. 

Let \( H \) be a \( \sigma \)-structure. Our focus will be on decision problems of the following form:

- \( \text{CSP}(H) \)
  
  **Input:** a \( \sigma \)-structure \( G \);
  
  **Question:** is there a homomorphism from \( G \) to \( H \)?

Equivalently, \( \text{CSP}(H) \) denotes the set of all \( \sigma \)-structures \( G \) such that \( G \rightarrow H \). We shall use \( \neg \text{CSP}(H) \) to denote the complement problem, i.e., the set of structures that do not admit a homomorphism to \( H \).

In this paper we consider almost exclusively list-homomorphism problems and so-called retraction problems. Informally, an input to the list-homomorphism problem for the \( \sigma \)-structure \( \mathbb{H} \) consists of a \( \sigma \)-structure \( G \), together with lists \( L_v \) for each vertex \( v \in G \). One must decide whether there exists a homomorphism \( f \) from \( G \) to \( \mathbb{H} \) such that \( f(v) \in L_v \) for all \( v \in G \). The retraction problem, also known as one-or-all list-homomorphism problem, is similar except the lists \( L_v \) can only be all of \( H \) or a one-element set. Equivalently, an input to the retraction problem can be seen as a \( \sigma \)-structure \( G \) with certain vertices coloured by a pre-assigned value from \( H \).\(^1\) We now define formally these problems.

**Definition 2.1:** Let \( H \) be a \( \sigma \)-structure. For each non-empty subset \( A \subseteq H \), let \( S_A \) be a unary relation symbol; when \( A = \{ h \} \) we denote \( S_A \) simply by \( S_h \). Let \( \tau = \sigma \cup \{ S_h : h \in H \} \), and let \( H^C \) denote the \( \tau \)-structure obtained from \( \mathbb{H} \) by adding all relations \( S_h(H^C) = \{ h \} \).

The problem \( \text{CSP}(H^C) \) is called the retraction problem for \( \mathbb{H} \). Let \( \tau' = \sigma \cup \{ S_A : A \subseteq H \} \), and let \( H^{L'} \) denote the \( \tau' \)-structure obtained from \( \mathbb{H} \) by adding all relations \( S_A(H^{L'}) = A \). The problem \( \text{CSP}(H^{L'}) \) is called the list-homomorphism problem for \( \mathbb{H} \).

**B. Graphs and digraphs**

We assume the reader is familiar with the standard terminology of graphs and digraphs (vertex, edge, etc.) and will not define them here. A relational structure \( G \) with a single binary relation \( E \) is called a digraph. If \( E \) is symmetric, i.e., if \( (x, y) \in E \) implies \( (y, x) \in E \) then we say \( G \) is a graph. If \( (x, x) \in E \) for all \( x \in G \) we say the digraph is reflexive, and if \( (x, x) \notin E \) for all \( x \in G \) we say \( G \) is irreflexive. If the vertex set \( G \) of a digraph \( G \) can be partitioned in two sets \( D \) and \( U \) such that \( E \subseteq (D \times U) \cup (U \times D) \) we say \( G \) is bipartite. A sequence of vertices \( v_1, \ldots, v_n \) of \( G \) such that \( (v_i, v_{i+1}) \in E \) for \( (v_{i+1}, v_i) \in E \) for every \( i \in \{ 1, \ldots, n-1 \} \) is called a walk from \( v_1 \) to \( v_n \) in \( G \). If for each pair \( u, v \) there exists a walk in \( G \) from \( u \) to \( v \), \( G \) is called connected.

**C. Algebra**

The product of two \( \sigma \)-structures \( S \) and \( T \), denoted \( S \times T \), is the \( \sigma \)-structure with universe \( S \times T \) such that for every \( n \)-ary \( R \in \sigma \), \( R(S \times T) \) consists of all tuples \( ((u_1, v_1), \ldots, (u_n, v_n)) \) with \( (u_1, \ldots, u_n) \in R(S) \) and \( (v_1, \ldots, v_n) \in R(T) \). The product of \( T \) with itself \( n \) times is denoted \( T^n \).

Let \( H \) be a relational structure. A \( k \)-ary operation \( f \) on \( H \) is a polymorphism of \( H \) if \( f \) is a homomorphism from \( H^k \) to \( H \); we also say that \( H \) admits \( f \). If \( f(v_1, \ldots, v_k) = v \) for every \( v \in H \), we say \( f \) is idempotent, and if \( f \) satisfies \( f(v_1, \ldots, v_n) \in \{ v_1, \ldots, v_n \} \) for every \( v_1, \ldots, v_n \in H \), we say \( f \) is conservative.

Let \( k \geq 2 \). A \( k \)-ary idempotent operation \( f \) on \( H \) is totally symmetric (TSI) for short if \( f(u_1, \ldots, u_n) = f(v_1, \ldots, v_n) \) whenever \( \{ u_1, \ldots, u_k \} = \{ v_1, \ldots, v_k \} \). A \( k \)-ary idempotent operation \( f \) on \( H \) is a weak near-unanimity (weak NU) operation if

\[
\begin{align*}
    f(u_1, \ldots, u_k) &= f(u_1, \ldots, u_k, u) = \ldots = f(u, u_2, \ldots, u_k) = u
\end{align*}
\]

for all \( u, v \in H \). Notice that a binary idempotent operation is a weak NU if and only if it is commutative. If \( k \geq 3 \) and

\[
\begin{align*}
    f(u_1, \ldots, u_k, v) &= f(u_1, \ldots, u_k, v, u) = \ldots = f(v, u_2, \ldots, u_k, u) = u
\end{align*}
\]

for all \( u, v \in H \), \( f \) is called a near-unanimity (NU) operation. A ternary NU operation is a majority operation. An associative binary TSI operation is called a semilattice operation.

**D. Arc consistency and TSI operations**

We refer the reader to [8], [12] and [6] for details on the following related concepts. Let \( H \) be a digraph, and consider a structure \( G \) consisting of a digraph together with lists \( L_g \subseteq H \) for all \( g \in G \). We say a pair \( (L, L') \) of subsets of \( H \) is consistent if for every \( h \in L \) there exists \( h' \in L' \) such that \( (h, h') \) is an edge of \( \mathbb{H} \) and for every \( h' \in L' \) there exists \( h \in L \) such that \( (h, h') \) is an edge of \( \mathbb{H} \). The structure \( G \) is consistent if for every edge \( (g, g') \) of \( G \), the pair \( (L_g, L_{g'}) \) is consistent. The arc-consistency check algorithm transforms any given structure \( G \) into a consistent structure \( G' \) with the property that there exists a list-preserving homomorphism from \( G \) to \( \mathbb{H} \) if and only if there is one from \( G' \) to \( \mathbb{H} \); the algorithm simply recursively removes offending vertices from non-consistent lists until the lists stabilise. This is a poly-time algorithm, and clearly if the output structure has an empty list then the original structure \( G \) does not admit a list-preserving homomorphism to \( \mathbb{H} \); in fact, if \( f \) is a list-preserving homomorphism from \( G \) to \( \mathbb{H} \) then \( f(g') \in L_{g'} \) for every \( g' \in G' \). On the other hand there is no guarantee in general that if \( G' \) has non-empty lists then there is indeed a homomorphism; if there always is we say that the arc-consistency check decides correctly. One may obviously apply the algorithm to inputs of the problem \( \text{CSP}(\mathbb{H}^L) \), but also to \( \text{CSP}(\mathbb{H}^C) \) by setting \( L_g = H \) for every uncoloured \( g \in G \). If the arc-consistency check decides correctly \( \text{CSP}(\mathbb{H}^L) \) \((\text{CSP}(\mathbb{H}^C)) \) on all inputs we say the problem is solvable by arc-consistency. Various equivalent conditions

\(^1\)The problem gets its name from the fact that it is equivalent (in fact under positive first-order reductions) to the following decision problem: given a structure \( G \) containing a copy of \( \mathbb{H} \), decide if \( G \) retracts to \( \mathbb{H} \).
on a structure $\mathcal{H}$ are known to characterise this property (see [6], Theorem 24), we shall require two of them here. Given a digraph $\mathcal{H}$, consider the digraph $\mathcal{P}(\mathcal{H})$ whose vertices are the non-empty subsets of $H$; there is an edge from $X$ to $Y$ if the pair $(X, Y)$ is consistent. We say a homomorphism $f$ from $\mathcal{P}(\mathcal{H})$ to $\mathcal{H}$ is constant-preserving if $f(\{e\}) = c$ for all $c \in H$, and we say it is conservative if $f(X) \subseteq X$ for all $X \subseteq H$.

**Theorem 2.2 ([12],[8]):** Let $\mathcal{H}$ be a digraph. Then the following conditions are equivalent:

1. $\text{CSP}(\mathcal{H}^C)$ (resp. $\text{CSP}(\mathcal{H}^L)$) is solvable by arc-consistency;
2. the structure $\mathcal{P}(\mathcal{H})$ admits a constant-preserving (resp. conservative) homomorphism to $\mathcal{H}$;
3. for every $k \geq 2$, $\mathcal{H}$ admits a $k$-ary (resp. conservative) TSI polymorphism.

**III. MONOCHROMATIC TSI POLYMORPHISMS AND ARC-CONSISTENCY**

If an undirected graph $\mathcal{H}$ admits a TSI operation $f$ and contains at least one edge $(0, 1)$ then $f(0, \ldots, 0, 1)$ and $f(1, \ldots, 1, 0)$ are adjacent and equal, so $\mathcal{H}$ contains a loop. In particular by Theorem 2.2 irreflexive graphs cannot have associated CSPs that are solvable by arc-consistency. However, if we consider partially-defined polymorphisms it is possible to adapt Theorem 2.2 to bipartite graphs. The proof is a simple adaptation of the standard arguments.

**Definition 3.1:** Let $\mathcal{H}$ be a bipartite graph with colour classes $D$ and $U$.

1. A set $X$ of vertices is called monochromatic if $X \subseteq D$ or $X \subseteq U$.
2. A map $f : D^n \cup U^n \to \mathcal{H}$ is called a monochromatic polymorphism of $\mathcal{H}$ (on $D$ and $U$) if it preserves the edge relation of $\mathcal{H}$ whenever it is defined, i.e. if $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_k\}$ are monochromatic and if $(x_i, y_i)$ is an edge of $\mathcal{H}$ for every $i$ then $f(x_1, \ldots, x_k)$ is adjacent to $f(y_1, \ldots, y_k)$.
3. Let $f$ be a monochromatic polymorphism. We say $f$ is a monochromatic semilattice (or NU, TSI, etc.) operation if $f$ satisfies the corresponding identities where defined.

It is easy to see that if $f$ is an idempotent monochromatic polymorphism, then it further satisfies $f(U^k) \subseteq U$ and $f(D^k) \subseteq D$, provided $\mathcal{G}$ is connected: indeed, if $(u_1, \ldots, u_k) \in U^k$ then there exists a path of even length in the subgraph of $\mathcal{H}^k$ induced by $U^k \cup D^k$ from $(u_1, \ldots, u_k)$ to $(u_1, \ldots, u_1)$ and hence there is a path of even length from $f(u_1, \ldots, u_k)$ to $f(u_1, \ldots, u_1) = u_1$ in $\mathcal{H}$.

We say that the problem $\text{CSP}(\mathcal{S})$ is solvable by arc-consistency on bipartite instances to mean that the arc-consistency check correctly decides on bipartite instances. The constants of the graph $\mathcal{P}(\mathcal{H})$ are the one-element subsets of $H$.

**Theorem 3.2:** Let $\mathcal{H}$ be a connected bipartite graph. Then the following are equivalent:

1. $\text{CSP}(\mathcal{H}^C)$ (resp. $\text{CSP}(\mathcal{H}^L)$) is solvable by arc-consistency on bipartite instances;
2. there is a constant-preserving (resp. conservative) partial homomorphism from $\mathcal{P}(\mathcal{H})$ to $\mathcal{H}$ defined on monochromatic sets;
3. there is a constant-preserving (resp. conservative) homomorphism from the connected component of constants in $\mathcal{P}(\mathcal{H})$ to $\mathcal{H}$;
4. for every $k \geq 2$, $\mathcal{H}$ admits a monochromatic $k$-ary (resp. conservative) TSI polymorphism.

Notice that when the equivalent conditions of the previous result hold, one has a very simple algorithm for the CSPs: on an arbitrary input $\mathcal{G}$, first run the 2-colouring algorithm (which can be done in Logspace), and then apply the arc-consistency check. One may also interpret the results in terms of dualities or obstructions (see [6]): the CSPs admit a duality that consists of trees and odd cycles.

**IV. LIST-HOMOMORPHISM PROBLEMS**

In this section we consider conservative monochromatic polymorphisms on bipartite graphs, which will allow us to refine some results from [9] and [11] on list-homomorphism problems on graphs.

Let $K_2$ denote the complete (irreflexive) graph on $\{0, 1\}$. The complement of a digraph $\mathcal{G}$ is the digraph with set of vertices equal to $G$ and whose set of arcs is $(G \times G) \setminus E(G)$. Let $\mathcal{G}$ be an irreflexive graph, and let $\{A_x : x \in \mathcal{G}\}$ be a family of arcs on the unit circle such that for every $x, y \in \mathcal{G}$ with $x \neq y$, $(x, y) \in E(\mathcal{G})$ if and only if $A_x \cap A_y \neq \emptyset$. Then the family $\{A_0, \ldots, A_n\}$ is called a circular arc representation of $\mathcal{G}$ and $\mathcal{G}$ is called a circular arc graph. Note that a circular arc graph is necessarily irreflexive. If the complement of $\mathcal{G} \times K_2$ is a circular arc graph we say $\mathcal{G}$ is a bi-arc graph [11].

**Lemma 4.1:** Let $\mathcal{G}$ be a graph. If $\mathcal{G}$ is bipartite and the complement of $\mathcal{G}$ is a circular arc graph, then $\mathcal{G}$ admits a conservative monochromatic semilattice polymorphism.

**Proof:** As $\mathcal{G}$ is bipartite, its complement has clique covering number 2. Let $N$ and $S$ be two cliques that partition the vertices of the complement of $\mathcal{G}$. By Lemma 2 of [16], there exist points $n$ and $s$ on the unit circle such that the complement of $\mathcal{G}$ can be represented by a family of circular arcs $A_v$, where every circular arc representing a vertex in $N$ contains $n$ and not $s$ and every circular arc representing a vertex in $S$ contains $s$ and not $n$, with $n$ and $s$ two points of the circle. Revolving clockwise around the circle, call $s(x)$ the starting point of the circular arc representing $x$ and $e(x)$ the ending point of the circular arc representing $x$. Because the graph has a finite number of vertices we can safely assume that there are no $A_u, A_v$ such that $s(u) = s(v)$ or $e(u) = e(v)$. For any two vertices $u, v \in N$, we define $u \leq_N v$ if and only if $e(u) \in A_v$. For any two vertices
A conservative monochromatic semilattice polymorphism $m$. As $m$ is conservative and $\mathbb{H} \times K_2$ contains $\mathbb{H}$ as an induced subgraph (see (2) ⇒ (1) below), then there exists a conservative monochromatic semilattice polymorphism on $\mathbb{H}$.

$(2) \Rightarrow (1)$: suppose that $\mathbb{H}$ admits a conservative monochromatic semilattice $m$ on the colour classes $D$ and $U$. Observe that $\mathbb{H} \times K_2$ is isomorphic to two disjoint copies of $\mathbb{H}$, namely $(D \times \{0\}) \cup (U \times \{1\})$ and $(D \times \{1\}) \cup (U \times \{0\})$. Let $D' = D \times \{0,1\}$ and $U' = U \times \{0,1\}$: these are colour classes for $\mathbb{H} \times K_2$. Define a monochromatic operation $g$ on $H \times K_2$ with respect to $D'$ and $U'$ as follows:

$$g((x,i),(y,j)) = \begin{cases} (m(x,y),i) & \text{if } i = j, \\ (x,0) & \text{if } 0 = i \neq j, x, y \in D, \\ (y,0) & \text{if } 0 = j \neq i, x, y \in D, \\ (x,1) & \text{if } 1 = i \neq j, x, y \in U, \\ (y,1) & \text{if } 1 = j \neq i, x, y \in U. \end{cases}$$

Intuitively, the operation $g$ is built as follows: since it is a conservative semilattice on $D$ and on $U$, the operation $m$ acts as the minimum of two total orderings, one on the vertices of $D$ and the other on $U$. Then $g$ orders $D'$ by placing every element of $D \times \{0\}$ before those from $D \times \{1\}$, and in $U'$ places those in $U \times \{1\}$ before those of $U \times \{0\}$. It is then clear that $g$ is a conservative, monochromatic semilattice operation, and it is easy to verify that it is edge-preserving. Hence $\mathbb{H} \times K_2$ admits a conservative monochromatic semilattice polymorphism, so by Lemma 4.2 $(4) \Rightarrow (1)$ $\mathbb{H}$ admits a conservative weak NU polymorphism.

The following definitions are a special case of those found in [13].

**Definition 4.4.** Let $\mathcal{G}$ be a graph.

1. Let $P = x_0, \ldots, x_n$ and $Q = y_0, \ldots, y_n$ be two paths in $\mathcal{G}$. We say that $P$ avoids $Q$ if for all $i \in \{0, \ldots, n-1\}$, $(x_i, y_{i+1}) \notin E(\mathcal{G})$.

2. A pair of vertices $u, v$ in $\mathcal{G}$ is invertible if
   - there exist paths $P$ from $u$ to $v$ and $Q$ from $v$ to $u$ such that $P$ avoids $Q$, and
   - there exist paths $P'$ from $v$ to $u$ and $Q'$ from $u$ to $v$ such that $P'$ avoids $Q'$.

Observe that if $u_0, \ldots, u_n$ avoids $v_0, \ldots, v_n$ and $u_n, \ldots, u_m$ avoids $v_n, \ldots, v_m$, then $u_0, \ldots, u_n, \ldots, u_m$ avoids $v_0, \ldots, v_n, \ldots, v_m$.

**Lemma 4.5.** Let $\mathcal{G}$ be a graph such that the neighbourhood of every loopless vertex is a clique of loops. If $\mathcal{G}$ doesn’t admit a conservative binary weak NU polymorphism, then there exist paths $P = x_0, \ldots, x_n$, $Q = y_0, \ldots, y_n$ from $u$ to $v$, $Q = y_0, \ldots, y_n$ from $v$ to $u$, $P' = x_0, \ldots, x_m$, $Q' = y_0, \ldots, y_m$ from $v$ to $u$ and $Q' = y_0, \ldots, y_m$ from $u$ to $v$ such that:
   - $P$ avoids $Q$;
   - $Q'$ avoids $P'$;
   - $n$ and $m$ are even.
Proof: If $G$ doesn’t admit a conservative binary weak NU polymorphism then by results in [13] there exist avoiding paths $P, Q, P'$ and $Q'$: it remains to prove that we can find such paths of even length. The trick is to prove that we can always increase the length by inserting avoiding paths of length 3. We show how to do this for $P$ and $Q$, the proof for $P'$ and $Q'$ is identical.

If there exists $i \in \{0, \ldots, n\}$ such that $x_i$ is a loop but $y_i$ isn’t, then $(x_i, y_i) \notin E(G)$ since $(y_i, y_{i+1}) \in E(G)$, $(x_i, y_{i+1}) \notin E(G)$ and the neighbourhood of every loopless vertex in $G$ is a clique of loops. Thus $x_i, x_{i+1}, x_{i+2}$ avoids $y_i, y_{i+1}, y_{i+2}$. If $n$ is odd, we insert these paths in $P$ and $Q$ respectively to obtain avoiding paths of even length.

Similarly, if there exists $i \in \{0, \ldots, n\}$ such that $y_i$ is a loop but $x_i$ isn’t then $(x_i, y_i) \notin E(G)$, and thus $x_i, x_{i-1}, x_{i-2}, x_{i-3}$ avoids $y_i, y_{i-1}, y_{i-2}, y_{i-3}$ (if $i = 0$, we use $a_{m-1}$ instead of $a_{m-2}$).

So we may now suppose that there exist no $i \in \{0, \ldots, n\}$ such that exactly one of $x_i$ and $y_i$ is a loop. Then either there exist $i, j \in \{0, \ldots, n\}$ such that $x_i, y_i, x_{i+1}, y_{i+1}$ are loops and then $x_{i+1}, x_{i+2}, x_{i+3}$ avoids $y_{i+1}, y_{i+2}, y_{i+3}$; or otherwise $n$ must be even. Indeed, by hypothesis neighbours of non-loops are loops, and by the last observation the neighbours of loops along the paths are non-loops, thus as we go from $u$ to $v$ we must traverse an even number of edges.

We say that an edge $(u, v)$ of a graph $G$ is loopless if neither $u$ nor $v$ is a loop.

**Lemma 4.6** Let $G$ be a graph with no loopless edge. Then the following are equivalent:

1) $G$ admits a conservative binary weak NU polymorphism;
2) $G \times K_2$ admits a conservative binary monochromatic weak NU polymorphism.

Proof: Let $D$ and $U$ be the colour classes of $G \times K_2$, $D = G \times \{0\}$ and $U = G \times \{1\}$. Suppose that $G$ admits a binary, conservative, commutative polymorphism $f$. Let $g: D^2 \cup U^2 \rightarrow H \times K_2$ be the function defined by:

$$g((a, x), (b, v)) = (f(a, b), x).$$

It is immediate that $g$ is a binary, conservative, commutative, monochromatic polymorphism of $G \times K_2$.

Conversely, suppose that $G$ doesn’t admit a binary, conservative, commutative polymorphism. Suppose first that some loopless vertex has a neighbourhood which is not a clique of loops. Then there exist vertices $u, v, w \in G$ such that $u$ and $w$ are loops but $v$ isn’t, $u$ and $w$ are neighbours of $v$ but not adjacent to each other. Then consider the following paths in $G \times K_2$:

- $(u, 0), (v, 1), (w, 0), (w, 1), (v, 0), (u, 1), (u, 0), (v, 1), (u, 0)$
- $(v, 0), (w, 1), (v, 0), (u, 1), (u, 0), (v, 1), (w, 0), (w, 1), (v, 0)$.

These are avoiding paths. Let $f$ be a conservative, binary monochromatic polymorphism of $G \times K_2$. If $f((u, 0), (v, 0)) = (u, 0)$, then applying $f$ on the pairs in the paths from left to right we see that $f((v, 1), (w, 1)) = (v, 1)$; moving along this way we get after 3 more steps that $f((v, 0), (u, 0)) = (v, 0)$ and hence $f$ cannot be commutative. Similarly, if $f((v, 0), (u, 0)) = (v, 0)$, starting on the middle pair of the paths and moving right, we see that we end up with $f((u, 0), (v, 0)) = (u, 0)$. Hence $G \times K_2$ admits no binary monochromatic commutative conservative polymorphism.

We may now assume that the hypotheses of Lemma 4.5 are satisfied. Then there exist $u, v \in G$ and paths $P = x_0, \ldots, x_n$ from $u$ to $v$, $Q = y_0, \ldots, y_n$ from $v$ to $u$, $P' = x_n, \ldots, x_m$ from $v$ to $u$ and $Q'$ from $u$ to $v$ such that $P$ avoids $Q$ and $P'$ avoids $Q'$. Thus there exist paths $R = (x_0, c_0), \ldots, (x_m, c_m)$ from $u$ to $(v, 0)$, $S = (y_0, c_0), \ldots, (y_m, c_m)$ from $(v, 0)$ to $(u, 0)$, $R' = (x_n, c_n), \ldots, (x_m, c_m)$ from $(v, 0)$ to $(u, 0)$ and $S' = (y_n, 0), \ldots, (y_m, 0)$ from $(u, 0)$ to $(v, 0)$ such that $c_i = 0$ if $i$ is even, $c_i = 1$ if $i$ is odd, $R$ avoids $S$ and $R'$ avoids $S'$. An argument similar to the previous case shows that $H \times K_2$ doesn’t admit any binary, conservative, commutative monochromatic polymorphism.

**Lemma 4.7** Let $H$ be a graph. Then the following are equivalent:

1) $H$ admits a conservative binary weak NU polymorphism;
2) $H$ admits conservative TSI polymorphisms of every arity $k \geq 2$.

Proof: Let $\phi$ be a binary, conservative, commutative polymorphism of $H$. Since $\phi$ is a weak NU operation, by Lemma 4.2 the graph $H \times K_2$ admits a conservative monochromatic semilattice polymorphism $m$. We may extend this binary semilattice to a $k$-ary one in the obvious way, by defining

$$m(u_1, \ldots, u_k) = m(m(u_1, u_2), \ldots, u_k).$$

We define a $k$-ary operation $f$ on $H$ as follows: for all $x_i \in H$, define

$$f(x_1, \ldots, x_k) = \phi(\pi_H(g_0(x_1, \ldots, x_k)), \pi_H(g_1(x_1, \ldots, x_k)))$$

where $\pi_H : H \times K_2 \rightarrow H$ is simply the projection on the $H$ factor, and $g_0(x_1, \ldots, x_k) = m((x_1, 0), \ldots, (x_k, 0))$ and $g_1(x_1, \ldots, x_k) = m((x_1, 1), \ldots, (x_k, 1))$.

It is easy to verify that $f$ is conservative and that it is a TSI operation. It remains to prove that it is a polymorphism. Suppose that $(x_i, y_i)$ is an edge of $H$ for all $i = 1, \ldots, k$. Then

- $u = g_0(x_1, \ldots, x_k)$ is adjacent to $v = g_1(y_1, \ldots, y_k)$
- $v' = g_1(x_1, \ldots, x_k)$ is adjacent to $v' = g_0(y_1, \ldots, y_k)$
and hence
\[(\pi_H(u), \pi_H(u')) \text{ is adjacent to } (\pi_H(v), \pi_H(v'))\]

then
\[f(x_1, \ldots, x_k) = \phi(\pi_H(u), \pi_H(u'))\]
is adjacent to
\[\phi(\pi_H(v), \pi_H(v')) = \phi(\pi_H(v'), \pi_H(v)) = f(y_1, \ldots, y_k)\]
and we are done.

**Theorem 4.8:** Let \( H \) be a graph. Then the following are equivalent:

1) \( CSP(H^L) \) is solvable by arc-consistency;
2) \( H \) is a bi-arc graph with a conservative binary, commutative idempotent polymorphism. As remarked at the beginning of section III, this implies immediately that every edge of \( H \) is incident to at least one loop.
3) \( H \) admits a conservative binary weak NU polymorphism.

**Proof:** (1) \(\Rightarrow\) (2): by Theorem 2.2 and Lemma 4.2 \( H \) is a bi-arc graph with a conservative binary, commutative idempotent polymorphism. As remarked at the beginning of section III, this implies immediately that every edge of \( H \) is incident to at least one loop.

(2) \(\Rightarrow\) (3): If \( H \) is a bi-arc graph, then by Lemma 4.2 \( H \times K_2 \) admits a conservative monochromatic semilattice polymorphism. We can now conclude using Lemma 4.6.

(3) \(\Rightarrow\) (1): immediate by Lemma 4.7 and Theorem 2.2.

**V. Conclusion**

We have characterised the graphs whose list-homomorphism problem is solvable by arc consistency. We proved that for every bipartite graph \( G \), its list-homomorphism problem is tractable if \( G \) admits a monochromatic conservative semilattice operation, and \( NP \)-complete otherwise. In the long version of this paper we also consider the analogous problem for the retraction problem for graphs. This question seems more involved, and we present some partial results in this direction. It is known that if a reflexive graph admits a near-unanimity (NU) operation then its retraction problem is actually first-order definable, and in particular is solvable by arc-consistency. In the irreflexive case, the tractable retraction problems have underlying graphs that are bipartite and hence we consider only monochromatic TSI polymorphisms. We prove that if a bipartite graph admits a near-unanimity polymorphism, then it admits monochromatic TSI polymorphisms of all arities, and its retraction problem is solvable in Logspace via the query language symmetric Datalog; we also provide an example of a graph with retraction problem solvable in Nondeterministic Logspace (via linear Datalog), but does not admit monochromatic TSI polymorphisms (and hence no NU polymorphisms either.)

**ACKNOWLEDGMENT**

This paper is dedicated to Ivo G. Rosenberg, from mathematical son and grandson. We are indebted to Victor Dalmau and Laci Egri for their helpful comments. The first author’s research is supported by grants from FQRNT and NSERC.

**REFERENCES**


