The complexity of homomorphism factorization

Kevin M. Berg

Abstract. We investigate the computational complexity of the problem of deciding if an algebra homomorphism can be factored through an intermediate algebra. Specifically, we fix an algebraic language, \( L \), and take as input an algebra homomorphism \( f: X \to Z \) between two finite \( L \)-algebras \( X \) and \( Z \), along with an intermediate finite \( L \)-algebra \( Y \). The decision problem asks whether there are homomorphisms \( g: X \to Y \) and \( h: Y \to Z \) such that \( f = hg \). We show that this problem is NP-complete for most languages.

Mathematics Subject Classification. 68Q25, 03C05, 08A05.

Keywords. Homomorphism factorization, NP-complete, Computational complexity, Finite structures.

1. Introduction

In this paper we investigate the complexity of the problem of deciding if an algebra homomorphism can be factored through an intermediate algebra. Specifically, we fix an algebraic language \( L \). The input to our problem is a homomorphism \( f: X \to Z \) between finite \( L \)-algebras \( X \) and \( Z \), along with an intermediate finite \( L \)-algebra \( Y \). The problem is to decide whether there are homomorphisms \( g: X \to Y \) and \( h: Y \to Z \) such that \( f = hg \), as shown in Figure 1. We refer to this as the Homomorphism Factorization Problem.

Problem 1.1 (The Homomorphism Factorization Problem). Given a homomorphism \( f: X \to Z \) between two finite \( L \)-algebras \( X \) and \( Z \), and given an intermediate finite \( L \)-algebra \( Y \), decide whether there are homomorphisms \( g: X \to Y \) and \( h: Y \to Z \) such that \( f = hg \).
There are several interesting special cases of the main problem worth identifying:

I. Consider restricting to the case where $|Z| = 1$. Then the homomorphisms $f$ and $h$ from Problem 1.1 must be constant, so the Homomorphism Factorization Problem reduces to the problem of deciding whether, given $\mathcal{L}$-algebras $X$ and $Y$, there is a homomorphism $g: X \to Y$, as shown in Figure 2. We refer to this special case as the Homomorphism Problem.

II. Consider restricting to the case where the input is $\mathcal{L}$-algebras $X$, $Y$, $Z$, and homomorphisms $f: X \to Z$, and $h: Y \to Z$, so the Homomorphism Factorization Problem reduces to the problem of deciding whether there is a homomorphism $g: X \to Y$, as shown in Figure 3. This special case will be called the Exists Right-Factor Problem.

III. Consider restricting to the case where the input is $\mathcal{L}$-algebras $X$, $Y$, $Z$, and homomorphisms $f: X \to Z$, and $g: X \to Y$, so the Homomorphism Factorization Problem reduces to the problem of deciding whether there is a homomorphism $h: Y \to Z$, as shown in Figure 4. This special case will be called the Exists Left-Factor Problem.

IV. Consider restricting to the case where $f: X \to Z$ is the identity function from $X$ to $Z = X$, as shown in Figure 5. In this case the Homomorphism Factorization Problem reduces to the problem of deciding if, given $X$ and $Y$, the algebra $X$ is a retract of $Y$. This special case will be called the Retraction Problem.

V. Consider restricting the Retraction Problem in the case where $|X| = |Y|$. This special case is the Isomorphism Problem for $\mathcal{L}$-algebras.

Note that the Homomorphism Factorization Problem and its variants are all in complexity class NP.

The problem of deciding the complexity of the Homomorphism Factorization Problem was raised at [11]. There the focus is on algebras in the language of a single fundamental operation, which is binary. The author of the problem expresses interest in the special case where the algebras are semigroups.
We introduce and elaborate on problems of this form, and make note of these special cases.

The questions considered in this paper have been studied for relational structures, so it is important to indicate the differences. Any semigroup can be viewed as a relational structure with a single ternary relation \( \{(x, y, z) \in X^3 \mid z = xy\} \). A function \( f: X \to Z \) between semigroups is an algebra homomorphism when \( X \) and \( Z \) are considered as algebras if and only if it is a relational homomorphism when \( X \) and \( Z \) are considered as relational structures. Therefore, the problem of deciding if a semigroup algebra homomorphism can be factored is the same as the problem of deciding if a semigroup relational homomorphism can be factored.

But the problem of deciding if a semigroup homomorphism can be factored is not the same as the problem of deciding if a homomorphism of relational structures (with one ternary relation) can be factored. The latter problem involves relational structures that are not codings of semigroups. In fact, it is not hard to see, and it is well known, that the homomorphism problem for ternary relational structures is NP-complete. But the homomorphism problem for semigroups always has an affirmative answer. That is, given finite semigroups \( X \) and \( Y \), there is always a semigroup homomorphism \( g: X \to Y \), namely we can take \( g \) to be a constant homomorphism mapping \( X \) to an idempotent of \( Y \).

It is also worth noting that some cases of the Homomorphism Factorization Problem can be easy (e.g., we just noted that the Homomorphism Problem for semigroups always has an affirmative answer), but also can be
hard (e.g., the Group Isomorphism Problem is a special case, and there is no known easy algorithm to decide the Group Isomorphism Problem). In order to characterize these cases, we introduce the notion of a “rich” language. An algebraic language is rich if it has at least two operations of arity one, or at least one operation of arity at least two.

These definitions allow us to state our primary result:

**Theorem 1.2.** The Homomorphism Factorization Problem is NP-complete for rich languages.

### 1.1. Graphs and graph theory

Our analysis of complexity relies on some techniques and complexity results from graph theory, and therefore we briefly describe the relevant concepts. A graph, $G$, is an ordered pair $(V_G, E_G)$ consisting of a set of vertices, $V_G$, together with an edge relation on the vertices, $E_G$—briefly, $G = (V_G, E_G)$, or $(V, E)$ when there is no potential for confusion. We denote an edge between two vertices $u$ and $v$ by the notation $(u, v)$, as usual. If the edge relation is not necessarily symmetric, we will say that $G$ is a directed graph. If the edge relation is symmetric, we will say $G$ is an undirected graph. For undirected graphs, two vertices $u$ and $v$ are adjacent if $(u, v)$ is an edge in $E_G$. For directed graphs, two vertices $u$ and $v$ are adjacent if either $(u, v)$ or $(v, u)$ is an edge in $E_G$. We say that there is a path from $u$ to $v$ if there exists a sequence $u = v_0, v_1, \ldots, v_n = v$ of adjacent vertices, with $n$ a non-negative integer, possibly 0. A graph is connected if, for any two vertices $u$ and $v$, there is a path from $u$ to $v$. A graph is said to be loop-free or loopless if for all vertices $v$, $(v, v)$ is not an edge.

We adopt some notational conventions for special types of graphs. A complete graph on $k$ vertices, $K_k$, is a loop-free connected undirected graph in which every vertex is adjacent to every other vertex. An undirected cycle on $k$ vertices, $C_k$, is a loop-free connected undirected graph consisting of a sequence of $k$ distinct vertices, $v_0, v_1, \ldots, v_{k-1}$ such that for each $v_i$, $(v_i, v_{i+1})$ with addition modulo $k$ is an edge [7]. A directed cycle on $k$ vertices, $\vec{C}_k$, is a loop-free connected directed graph consisting of a sequence of $k$ distinct vertices, $v_0, v_1, \ldots, v_{k-1}$ such that for each $v_i$, $(v_i, v_{i+1})$ with addition modulo $k$ is an edge [7].

A graph homomorphism between $G$ and $H$, $\phi: G \to H$, is a function from $V_G$ to $V_H$ that respects the edge relation—that is, if $(u, v)$ is in $E_G$, then $(\phi(u), \phi(v))$ is in $E_H$. A strong graph homomorphism has the additional property that if $(\phi(u), \phi(v))$ is in $E_H$, then $(u, v)$ is in $E_G$. A graph isomorphism is a strong graph homomorphism where the vertex map is bijective as a function. A graph retraction, $\rho: G \to G$, is a graph endomorphism such that $\rho(v) = v$ for every vertex $v$ in $\rho(G)$, and we say that a subgraph $H$ of $G$ is a retract of $G$ if there exists a retraction $\rho: G \to G$ with $H = \rho(G)$. In particular, the core of a graph is a minimal retract.

These notions of homomorphism are, in turn, directly linked to four decision problems which are used throughout this work:
**Problem 1.3 (The Graph Homomorphism Problem).** Given two graphs $G$ and $H$, decide whether there exists a graph homomorphism $\phi: G \to H$.

**Problem 1.4 (The Strong Graph Homomorphism Problem).** Given two graphs $G$ and $H$, decide whether there exists a strong graph homomorphism $\phi: G \to H$.

**Problem 1.5 (The Graph Retraction Problem).** Given two graphs $G$ and $H$, decide whether $H$ is a retract of $G$.

**Problem 1.6 (The Graph Isomorphism Problem).** Given two graphs $G$ and $H$ with $|V_G| = |V_H|$, decide whether there exists a graph isomorphism $\phi: G \to H$.

Problem 1.3 is well-known to be NP-complete [5], and Problem 1.4 is also NP-complete [4]. One variant of the Graph Homomorphism Problem, the $k$-coloring Problem, restricts $H = K_k$ for some fixed $k$. The $k$-coloring Problem is similarly well-known to be NP-complete for any $k \geq 3$ [5]. Problem 1.5 is similarly NP-complete [7]. However, Problem 1.6, a special case of Problem 1.5, is not known to be either NP-complete or solvable in polynomial time [10]. We adopt the convention that problems that can be solved in polynomial time if and only if the Graph Isomorphism Problem can be solved in polynomial time are Graph Isomorphism complete, or GI-complete for short [8]. Similarly, we say a problem is Graph Isomorphism hard, or GI-hard, if there is a polynomial time reduction of the Graph Isomorphism Problem to that problem.

2. Languages with two unary operations

We begin our proof of Theorem 1.2 by considering the Exists Right-Factor Problem for algebras with two or more unary operations. In fact, this follows from a construction in [1], in which the authors, building upon [6], construct an algebra with two unary operations from a loop-free directed graph. As Graph Homomorphism is well-known to be NP-complete [5], this provides the desired result.

**Lemma 2.1.** The Homomorphism Problem for algebras in a language with two unary operations is NP-complete.

**Proof.** The algebras with two unary operations in Lemma 5.3 of [1] can be constructed from an arbitrary loop-free directed graph, $G$, in polynomial time. Therefore, it follows that the problem of finding a homomorphism between two loop-free directed graphs, $G$ and $H$, can be written as the problem of finding a homomorphism between their corresponding algebras in polynomial time. □

This result naturally extends to the case where a language contains more than two unary operations.

**Corollary 2.2.** The Homomorphism Problem for algebras in a language with at least two unary operations is NP-complete.

**Corollary 2.3.** The Homomorphism Factorization Problem for algebras in a language with at least two unary operations is NP-complete.
Table 1. The operations of $G^\dag$

|        | $\alpha(\cdot)$ | $\beta(\cdot)$ |
|--------|-----------------|----------------|
| $u_1$  | $u_1$           | $u_2$          |
| $u_2$  | $u_1$           | $u_2$          |
| $a_{(u,v)}$ | $u_1$   | $b_{(u,v)}$   |
| $b_{(u,v)}$ | $v_2$   | $a_{(u,v)}$   |

In fact, this same construction allows us to completely classify the Homomorphism Factorization Problem for such languages. The observation that the Homomorphism Problem is a special case of the Exists Right-Factor Problem gives the following:

**Corollary 2.4.** The Exists Right-Factor Problem for algebras in a language with at least two unary operations is NP-complete.

To prove the Exists Left-Factor and Retraction Problems, we will introduce a novel construction, $G^\dag$, which has many of the same morphism preserving properties as the construction in [1], but avoids distinguished elements. We believe this construction may also be of use in developing techniques for encodings of more complex algebras, which could arise when exploring the boundaries of complexity—following the addition of more identities—in later work. This construction is also used in the arguments in Section 5.

We may consider any loop-free, finite, connected, directed graph, $G = (V_G, E_G)$, with at least two vertices. We encode this graph into an algebra $G^\dag$, using the following rules to construct the universe: for each vertex $v$ in $V_G$, there are two corresponding elements $v_1$ and $v_2$ in $G^\dag$, and for each edge $(u, v)$ in $E_G$, there are two elements, $a_{(u,v)}$ and $b_{(u,v)}$, in $G^\dag$. We assign to $G^\dag$ two unary operations: $\alpha(\cdot)$ and $\beta(\cdot)$, given by Table 1.

The $G^\dag$ encoding gives us the following results.

**Theorem 2.5.** Let $G$ and $H$ be loop-free, finite, connected, directed graphs. There exists a homomorphism $\phi: G \to H$ if and only if there exists a homomorphism $\psi: G^\dag \to H^\dag$.

**Proof.** Suppose first that there exists a homomorphism $\phi: G \to H$. We construct a function $\psi: G^\dag \to H^\dag$ based on $\phi$—specifically, if for any $v$ in $V_G$ we have $\phi(v)$ in $V_H$, then $\psi(v_1) = \phi(v)_1$, $\psi(v_2) = \phi(v)_2$ and for all $(u, v)$ in $E_G$, $\psi(a_{(u,v)}) = a_{(\phi(u),\phi(v))}$ and $\psi(b_{(u,v)}) = b_{(\phi(u),\phi(v))}$. We claim $\psi$ is well-defined by the well-definition of $\phi$ and the construction of $H^\dag$. Since $u$ maps to a single element of $H$ under $\phi$, that single element, $\phi(u)$, is in turn associated with exactly two distinct elements $\phi(u)_1$ and $\phi(u)_2$ in $H^\dag$. Similarly, for any $(u, v)$ in $E_G$, we must have that $(\phi(u), \phi(v))$ is in $E_H$. In turn, it must be the case that $a_{(\phi(u),\phi(v))}$ and $b_{(\phi(u),\phi(v))}$ are two single elements in $H^\dag$. The claim holds.

Next, we claim $\psi$ is a homomorphism. Suppose $i$ is an element of $\{1, 2\}$, and $v_i$ is an element of $G^\dag$ coming from any $v$ in $V_G$. We have that $\psi(\alpha(v_i)) =$
ψ(v₁) = φ(v₁) = α(φ(v₁)) = α(ψ(v₁)), and ψ(β(v₁)) = ψ(v₂) = φ(v₂) = β(φ(v₂)) = β(ψ(v₂)). Similarly, if \( a_{(u,v)} \) and \( b_{(u,v)} \) are elements of \( G^\dagger \), then we have that

\[
\psi(\alpha(a_{(u,v)})) = \psi(u₁) = φ(u₁) = α(φ(u₁)) = α(ψ(a_{(u,v)})),
\]

\[
\psi(β(a_{(u,v)})) = \psi(b_{(u,v)}) = b(φ(u),φ(v)) = β(φ(u),φ(v)) = β(ψ(a_{(u,v)})),
\]

\[
\psi(α(b_{(u,v)})) = \psi(v₂) = φ(v₂) = α(b(φ(u),φ(v))) = α(ψ(b_{(u,v)})), \text{ and } \psi(β(b_{(u,v)})) = \psi(ψ(a_{(u,v)})) = a(ψ(φ(u),φ(v))) = β(ψ(b_{(u,v)})).
\]

The claim holds.

Suppose now that there instead exists a homomorphism \( ψ: G^\dagger \rightarrow H^\dagger \).

We claim \( ψ(u₂) = u₂' \) for some \( u₂' \in V_H \). By construction, \( u₂ = βα(u₂) \), and since \( ψ \) is a homomorphism, \( ψ(u₂) = ψ(βα(u₂)) = βα(ψ(u₂)) \). The only \( x \in H^\dagger \) satisfying \( x = βα(x) \) are \( x = u₂' \) for some \( u₂' \in V_H \). The claim holds. Furthermore, since \( ψ(u₂) = u₂' \) for some \( u₂' \in V_H \), it follows that \( ψ(u₁) = u₁' \) for the same \( u₂' \in V_H \)—since \( α(u₂) = u₁ \), \( ψ(u₁) = ψ(α(u₂)) = α(ψ(u₂)) = α(u₂') = u₁' \).

We now claim that \( ψ(b_{(u,v)})) = b_{(u',v')} \) for some \( (u',v') \in E_H \). Since \( β(α(b_{(u,v)})) = v₂ = β(α(b_{(u,v)})), \) \( α(ψ(b_{(u,v)})) = ψ(β(α(b_{(u,v)}))) = β(α(ψ(b_{(u,v)}))) \). Then \( ψ(b_{(u,v)}) \) satisfies the equational condition that \( α(x) = βα(x) \), and the only \( x \) with \( α(x) = βα(x) \) are precisely the elements of the form \( b_{(u',v')} \) for some \( u',v' \in V_H \). The claim holds. Furthermore, since \( ψ(b_{(u,v)}) = b_{(u',v')} \) for some \( u',v' \in V_H \), it follows that \( ψ(α(u,v)) = α(u') \) for the same \( u',v' \in V_H \)—since \( β(b_{(u,v)}) = α(u,v), ψ(α(u,v)) = ψ(β(b_{(u,v)})) = β(ψ(b_{(u,v)})) = β(b_{(u',v')}), \) \( ψ(α(u,v)) = α(u') \).

This provides the basis for our construction of \( φ \)—if \( v \) is in \( V_G \), we let \( φ(v) = w \) in \( V_H \) where \( ψ(v₁) = w₁ \). By the preceding arguments, \( φ \) is a well-defined homomorphism.

We can classify the Exists-Left Factor Problem using a specific input corresponding to a homomorphism with two fixed points.

**Corollary 2.6.** The Exists-Left Factor Problem for algebras in a language with at least two unary operations is NP-complete.

**Proof.** Let \( G = (\{v\},\emptyset) \) be the graph consisting of a single vertex, \( v \), with no edges. We encode \( G \) into an algebra, \( X \), using the standard rules for constructing \( G^\dagger \)—specifically, the universe of \( X \) consists of two elements, \( v₁ \) and \( v₂ \), and we assign to \( X \) two unary operations, \( α(\cdot) \) and \( β(\cdot) \), given by Table 2.

We then set algebras \( Y \) and \( Z \) to be the encodings of two loop-free, finite, connected, directed graphs, \( H \) and \( J \), each together with a new distinguished vertex \( v' \) not connected to any other vertices. In particular, \( v' \) is associated with neither \( a \) nor \( b \) elements in either algebra. Suppose \( f: X \rightarrow Z \) is given by \( f(v) = v' \) and \( g: X \rightarrow Y \) is given by \( g(v) = v' \). Then a homomorphism \( h: Y \rightarrow Z \) with \( f = hg \) exists if and only if there exists a homomorphism between \( H^\dagger \) and \( J^\dagger \). By Corollary 2.2, the determination of the existence of such an \( h \) is NP-complete.

By recalling the NP-completeness of the Graph Retraction Problem [7], this construction also yields a pair of results regarding retracts.

**Theorem 2.7.** Let \( G \) and \( H \) be loop-free, finite, connected, directed graphs. Then \( H \) is a retract of \( G \) if and only if \( H^\dagger \) is a retract of \( G^\dagger \).
Table 2. The operations of $X$

|    | $\alpha(\cdot)$ | $\beta(\cdot)$ |
|----|----------------|--------------|
| $v_1$ | $v_1$ | $v_2$ |
| $v_2$ | $v_1$ | $v_2$ |

Proof. Suppose first that $H$ is a retract of $G$. Then, in particular, there exist graph homomorphisms $\phi_1 : G \to H$ and $\phi_2 : H \to G$ with $\phi_1 \phi_2 = \text{id}_H$, the identity map on $H$. By Theorem 2.5, there must exist homomorphisms $\psi_1 : G^\dagger \to H^\dagger$ and $\psi_2 : H^\dagger \to G^\dagger$. In particular, for any $v$ in $V_G$, we must have $\psi_1(v_1) = \phi_1(v_1)$ and $\psi_1(v_2) = \phi_1(v_2)$. Similarly, for any $(u,v)$ in $E_G$, we must have $\psi_1(a(u,v)) = a(\phi_1(u),\phi_1(v))$ and $\psi_1(b(u,v)) = b(\phi_1(u),\phi_1(v))$. Correspondingly, for any $v$ in $V_H$ or $(u,v)$ in $E_H$, we have $\psi_2(v_1) = \phi_2(v_1)$, $\psi_2(v_2) = \phi_2(v_2)$, $\psi_2(a(u,v)) = a(\phi_2(u),\phi_2(v))$, and $\psi_2(b(u,v)) = b(\phi_2(u),\phi_2(v))$.

We claim that $\text{id}_{H^\dagger}$, the identity map on $H^\dagger$, must be equal to $\psi_1 \psi_2$. Let $v$ be any vertex in $V_H$. Then $\text{id}_{H^\dagger}(v_1) = v_1 = \text{id}_H(v_1) = \phi_1(\phi_2(v_1)) = \psi_1(\phi_2(v_1)) = \psi_1(v_1)$ and $\text{id}_{H^\dagger}(v_2) = v_2 = \text{id}_H(v_2) = \phi_1(\phi_2(v_2)) = \psi_1(\phi_2(v_2)) = \psi_1(v_2)$. Similarly, let $(u,v)$ be any element of $E_H$. Then $\text{id}_{H^\dagger}(a(u,v)) = a(u,v) = a(\phi_1(u),\phi_1(v)) = \phi_2(a(\phi_1(u),\phi_1(v))) = \psi_1(\phi_2(a(u,v))) = \psi_1(\psi_2(a(u,v)))$ and $\text{id}_{H^\dagger}(b(u,v)) = b(u,v) = b(\phi_1(u),\phi_1(v)) = \phi_2(b(\phi_1(u),\phi_1(v))) = \psi_1(b(\phi_2(\phi_1(u),\phi_1(v))) = \psi_1(\psi_2(b(u,v))) = \psi_1(\psi_2(b(u,v)))$. The claim holds.

Suppose now that $H^\dagger$ is a retract of $G^\dagger$. Then, in particular, there exist homomorphisms $\psi_1 : G^\dagger \to H^\dagger$ and $\psi_2 : H^\dagger \to G^\dagger$ with $\psi_1 \psi_2 = \text{id}_{H^\dagger}$, the identity map on $H^\dagger$. By Theorem 2.5, there must exist graph homomorphisms $\phi_1 : G \to H$ and $\phi_2 : H \to G$. In particular, for any $v$ in $V_G$, we must have $\psi_1(v_1) = \phi_1(v_1)$ and $\psi_1(v_2) = \phi_1(v_2)$. Similarly, for any $(u,v)$ in $E_G$, we must have $\psi_1(a(u,v)) = a(\phi_1(u),\phi_1(v))$ and $\psi_1(b(u,v)) = b(\phi_1(u),\phi_1(v))$. Correspondingly, for any $v$ in $V_H$ or $(u,v)$ in $E_H$, we have $\psi_2(v_1) = \phi_2(v_1)$, $\psi_2(v_2) = \phi_2(v_2)$, $\psi_2(a(u,v)) = a(\phi_2(u),\phi_2(v))$, and $\psi_2(b(u,v)) = b(\phi_2(u),\phi_2(v))$. By an argument similar to the preceding one, $\text{id}_H$, the identity map on $H$, must be equal to $\phi_1 \phi_2$.  

Corollary 2.8. The Retraction Problem for algebras in a language with at least two unary operations is NP-complete.

The previous result also immediately demonstrates a corollary about the Isomorphism Problem.

Corollary 2.9. The Isomorphism Problem for algebras in a language with at least two unary operations is GI-hard.

With this, we have classified the computational complexity of the Homomorphism Factorization Problem and its variants for algebras in a language with at least two unary operations. In the next section, we will produce similar results for a language with one binary operation.
### Table 3. The operation $\cdot$ of $G^*$

|   | $a$ | $b$ | $c$ | $d$ | $u_1$ | $v_1$ | $u_2$ | $v_2$ |
|---|-----|-----|-----|-----|-------|-------|-------|-------|
| $a$ | $b$ | $a$ | $a$ | $a$ | $u_1$ | $v_1$ | $u_2$ | $v_2$ |
| $b$ | $a$ | $c$ | $a$ | $a$ | $u_1$ | $v_1$ | $u_2$ | $v_2$ |
| $c$ | $a$ | $a$ | $d$ | $a$ | $u_1$ | $v_1$ | $u_2$ | $v_2$ |
| $d$ | $a$ | $a$ | $a$ | $a$ | $u_1$ | $v_1$ | $u_2$ | $v_2$ |
| $u_1$ | $u_1$ | $u_1$ | $u_1$ | $u_1$ | $d$ | $* c$ | $d$ |
| $v_1$ | $v_1$ | $v_1$ | $v_1$ | $v_1$ | $d$ | $d$ | $c$ |
| $u_2$ | $u_2$ | $u_2$ | $u_2$ | $u_2$ | $c$ | $d$ | $d$ | $b$ |
| $v_2$ | $v_2$ | $v_2$ | $v_2$ | $v_2$ | $d$ | $c$ | $b$ | $d$ |

3. Binary operations

In this section, we demonstrate that the Homomorphism Problem is NP-complete in a language with one binary operation. Specifically, we will show that in this case, any algorithmic solution to the find right-factor problem would necessarily give a solution to the Strong Homomorphism Problem for graphs, and vice versa. Strong Graph Homomorphism is known to be NP-complete [4], and therefore this will provide the desired result.

Any loop-free undirected graph, $G = (V_G, E_G)$, with at least two vertices is encoded into an algebra $G^*$ using the following rules—for every $v$ in $V_G$, there are two elements, $v_1$ and $v_2$ in $G^*$; and there are four distinguished elements, $a$, $b$, $c$, and $d$. We then assign to $G^*$ a single binary operation, $\cdot$, where for any distinct $u, v$ in $V_G$, $\cdot$ is given by Table 3—note that $*$ is either $u_1v_1 = v_1u_1 = a$ if $(u, v)$ is in $E_G$, or else $u_1v_1 = v_1u_1 = d$.

We may intuitively think of $G^*$ as encoding a degenerate coloring on edges for a new graph produced by connecting the vertices of $G$ to their corresponding counterparts in the complete graph on $V_G$. An example of this construction on the four-cycle, $C_4$, is shown in Figure 6.

This encoding allows us to move between the Strong Graph Homomorphism Problem and the Homomorphism Problem for algebras, and gives us the following result.

**Theorem 3.1.** Let $G$ and $H$ be loop-free, finite, undirected graphs with at least two vertices. There exists a strong homomorphism $\phi: G \rightarrow H$ if and only if there exists a homomorphism $\psi: G^* \rightarrow H^*$.

**Proof.** Suppose first that there exists a strong homomorphism $\phi: G \rightarrow H$. We construct a function $\psi: G^* \rightarrow H^*$ based on $\phi$—specifically, if for any $v$ in $V_G$ we have $\phi(v)$ in $V_H$, then $\psi(v_1) = \phi(v)_1$, $\psi(v_2) = \phi(v)_2$ and for all $x$ in \{a, b, c, d\}, $\psi(x) = x$.

We claim $\psi$ is well-defined by the well-definition of $\phi$ and the construction of $H^*$. Since $v$ maps to a single element of $H$ under $\phi$, that single element, $\phi(v)$, is in turn associated with exactly two distinct elements $\phi(v)_1$ and $\phi(v)_2$ in $H^*$. 

Next, we claim $\psi$ is a homomorphism. Recall that $G^*$ and $H^*$ are commutative. Suppose $u$ and $v$ are distinct elements of $V_G$. If $(u, v)$ is in $E_G$, then $u_1v_1 = a$, and since $(\phi(u), \phi(v))$ is in $E_H$, $\psi(u_1)\psi(v_1) = \phi(u)\phi(v)_1 = a = \psi(a) = \psi(u_1v_1)$. If $(u, v)$ is not in $E_G$, then $u_1v_1 = d$, and since $(\phi(u), \phi(v))$ is also not in $E_H$, $\psi(u_1)\psi(v_1) = \phi(u)\phi(v)_1 = d = \psi(d) = \psi(u_1v_1)$. Similarly, $\psi(u_1)\psi(u_1) = \phi(u)\phi(u)_1 = d = \psi(d) = \psi(u_1u_1)$, and $\psi(u_1)\psi(v_2) = \phi(u)\phi(v)_2 = d = \psi(d) = \psi(u_1v_2)$. In addition, $\psi(u_2)\psi(v_2) = \phi(u_2)\phi(v)_2 = b = \psi(b) = \psi(u_2v_2)$, and $\psi(u_1)\psi(u_2) = \phi(u)\phi(u)_2 = c = \psi(c) = \psi(u_1u_2)$. Furthermore, for all $x$ in $\{a, b, c, d\}$, $\psi(u_1)\psi(x) = \psi(u_1) = \psi(u_1x)$ and $\psi(u_2)\psi(x) = \psi(u_2) = \psi(u_2x)$.

Suppose now that there exists a homomorphism $\psi: G^* \to H^*$. We begin by arguing that for any $x$ in $\{a, b, c, d\}$, it must be the case that $\psi(x) = x$. Since $d^2 = a$, $\psi(d^2) = \psi(a) = \psi(d)^2$. Since $\psi(d)^2$ is a square, it must be the case that $\psi(a)$ is in $\{a, b, c, d\}$. Suppose by way of contradiction that $\psi(a) = b$. Then $\psi(b) = \psi(a^2) = \psi(a)\psi(a) = b^2 = c$, and similarly $\psi(c) = d$ and $\psi(d) = a$. But then $\psi(ad) = \psi(a)\psi(d) = a$. By similar arguments showing the inconsistency of $\psi(a) = c$ and $\psi(a) = d$, it must be the case that $\psi(a) = a$. It is then immediate that, in fact, $\psi(x) = x$ for any $x$ in $\{a, b, c, d\}$.

Next, suppose $v$ in $V_G$. We claim that $\psi(v_1) = u_1$ and $\psi(v_2) = u_2$ for some $u$ in $V_H$. We first argue that $\psi(v_1)$ is $u_1$ or $u_2$: suppose by way of contradiction that $\psi(v_1)$ were in $\{a, b, c, d\}$. Since $G$ is loop-free, $\psi(v_1)^2 = \psi(v_1^2) = \psi(d)$, so by the previous argument, $\psi(v_1)^2 = d$. It follows that $\psi(v_1) = c$, and thus $\psi(v_1)(\psi(v_1)^2) = a$. But since $\psi$ is a homomorphism, $\psi(v_1)(\psi(v_1)^2) = \psi(v_1v_1^2) = \psi(v_1d) = \psi(v_1) = c$, a contradiction. So it must be the case that $\psi(v_1) = u_1$ or $\psi(v_1) = u_2$. Since $G$ has at least two distinct elements, there exists a $w$ in $V_G$ with $v_2w_2 = b$. In particular, $\psi(v_2)\psi(w_2) = \psi(v_2w_2) = \psi(b) = b$, so $\psi(v_2) = u_2$ for some $u$ in $V_H$. Since $v_1v_2 = c$, $\psi(v_1)\psi(v_2) = \psi(v_1v_2) = \psi(c) = c$. So it must be the case that $\psi(v_1)u_2 = c$, so $\psi(v_1) = u_1$. This completes the proof of the claim.

The preceding argument allows us to define $\phi$ based on $\psi$—specifically, let $\phi(v)$ be the element in $V_H$ corresponding to $\psi(v_1)$ and $\psi(v_2)$.
We claim $\phi$ is well-defined by the well-definition of $\psi$ and the construction of $H^*$. Let $v$ be an element in $V_G$. That single element is in turn associated with exactly two distinct elements $v_1$ and $v_2$ in $H^*$. By the preceding arguments, $\psi(v_1) = u_1$ and $\psi(v_2) = u_2$ for some $u$ in $V_H$. By construction, $\phi(v) = u$. If $\phi(v) = u'$, then by the definition of $\phi$, $\psi(v_1) = u'$ and $\psi(v_2) = u'_2$, and since $\psi$ is well-defined, $u'_1 = u_1$ and $u'_2 = u_2$. Thus, $u' = u$. The claim holds.

We now claim that $\phi$ is a strong homomorphism. Let $(u, v)$ be in $E_G$—then $u_1v_1 = a$. By construction, $\phi(u)_1 = \psi(u_1)$ and $\phi(v_1) = \psi(v_1)$, and $\phi(u)_1\phi(v)_1 = \psi(u_1)\psi(v_1) = \psi(u_1v_1) = \psi(a) = a$. So $(\phi(u), \phi(v))$ is in $E_H$. Suppose instead that $(u, v)$ is not in $E_G$—then $u_1v_1 = d$. By construction, $\phi(u)_1 = \psi(u_1)$ and $\phi(v)_1 = \psi(v_1)$, and $\phi(u)_1\phi(v)_1 = \psi(u_1)\psi(v_1) = \psi(u_1v_1) = \psi(d) = d$. So $(\phi(u), \phi(v))$ is not in $E_H$. Thus, $\phi$ preserves edges and non-edges, and the claim holds. □

**Corollary 3.2.** The Homomorphism Problem is NP-complete for algebras in a language with one binary operation.

Using arguments similar to those in Section 2, the preceding result can be extended to classify all major variants of the Homomorphism Factorization Problem.

**Corollary 3.3.** The Homomorphism Factorization Problem is NP-complete for algebras in a language with one binary operation, as are the Exists Right-Factor, Exists Left-Factor, and Retraction Problems. The Isomorphism Problem is GI-hard for algebras in a language with one binary operation.

### 4. Associative binary operations

We will now classify the computational complexity of Homomorphism Factorization Problems for semigroups. Specifically, we will show that the Exists Right-Factor Problem is NP-complete. This will both eventually allow us to complete the proof of Theorem 1.2, and provides a negative answer to the problem regarding semigroups [11] that initially motivated this investigation. We then modify this encoding to show that the Exists Left-Factor Problem for semigroups is also NP-complete.

In Section 5 of [9], for any loop-free, finite, undirected graph $G$, the authors define a commutative 3-nilpotent semigroup, $T(G)$, that can be constructed in polynomial time. The following arguments about Homomorphism Factorization Problem complexity could be rearranged to make use of this construction. In fact, we will begin by noting an easy corollary of the proof of Lemma 5.1 of [9].

**Corollary 4.1.** The Retraction Problem for commutative semigroups is NP-complete.

*Proof.* It is sufficient to consider the case where $X = T(G)$ for some loop-free finite undirected graph $G$ and $Y = T(H)$ for some loop-free finite undirected graph $H$. It is clear from the graph homomorphism extensions in the proof of
Lemma 5.1 of [9] that \( T(G) \) is a retract of \( T(H) \) if and only if \( G \) is a retract of \( H \). The NP-completeness of the Retraction Problem for commutative semigroups then follows from the NP-completeness of the Graph Retraction Problem [7].

Again as before, this result implies that the Isomorphism Problem for semigroups is GI-hard. In fact, it has been shown in [2] that the Isomorphism Problem for semigroups is GI-complete. Furthermore, that the Homomorphism Problem for semigroups can be solved in polynomial time, by choosing a homomorphism mapping everything to an idempotent, is readily apparent.

Indeed, our broad classification of the complexity of the Homomorphism Factorization Problem for semigroups follows immediately.

**Corollary 4.2.** The Homomorphism Factorization Problem for semigroups is NP-complete.

As was previously mentioned, the construction of \( T(G) \), together with modifications to accommodate specific graphs, is likely sufficient to prove our remaining results. However, to prove the Exists Right-Factor and Exists Left-Factor Problems, and in anticipation of our arguments in Section 5, we instead introduce a new construction that has a graph coloring interpretation and can encode graphs with loops.

Consider an arbitrary undirected graph, \( G = (V_G, E_G) \), which will be encoded into a semigroup, \( X_G \). The universe is \( X_G = V_G \cup \{ \chi(u,v) \mid (u,v) \in E_G \} \cup \{ b, b^2, c, 0 \} \)—note that we adopt the convention \( \chi(u,v) = \chi(v,u) \), unlike in the unary example. We assign to \( X_G \) the single binary operation, \( \cdot \), given by Table 4, where for any \( u \) and \( v \) in \( V_G \), \( * \) is either \( uv = vu = c \) if \( (u,v) \) is in \( E_G \), or else \( uv = vu = \chi(u,v) \); and \( \chi \) is a placeholder for any \( \chi(u,v) \) in the semigroup.

Since all triple products in \( X_G \) are 0, \( X_G \) is indeed a semigroup—in fact, \( X_G \) is a commutative nilpotent semigroup of degree three [3]. Intuitively, \( X_G \) is a description of the graph, \( G \), together with a new distinguished vertex, \( b \), which is connected to all vertices of \( G \). An example of this construction on the four-cycle, \( C_4 \), is shown in Figure 7.

We now define \( Z \) to be the semigroup with a single binary operation, \( \cdot \), given by Table 5.

---

**Table 4.** The operation \( \cdot \) of \( X_G \)

| \( \cdot \) | 0   | \( b \) | \( b^2 \) | \( c \) | \( u \) | \( v \) | \( \chi \) |
|--------|-----|--------|---------|-----|-----|-----|-------|
| 0      | 0   | 0      | 0       | 0   | 0   | 0   | 0     |
| \( b \)| 0   | \( b^2 \) | 0       | 0   | \( c \) | \( c \) | 0     |
| \( b^2 \)| 0   | 0      | 0       | 0   | 0   | 0   | 0     |
| \( c \)| 0   | 0      | 0       | 0   | 0   | 0   | 0     |
| \( u \)| 0   | \( c \) | 0       | \( * \) | \( * \) | 0   | \( \chi \) |
| \( v \)| 0   | \( c \) | 0       | \( * \) | \( * \) | 0   | \( \chi \) |
| \( \chi \)| 0   | 0      | 0       | 0   | 0   | 0   | 0     |
Note that $Z$ is equivalent to the encoding of the graph consisting of a single loop on a vertex $a$.

We take as our input two loop-free undirected graphs, $G = (V_G, E_G)$ and $H = (V_H, E_H)$. We encode $G$ and $H$ into semigroups, $X_G$ and $X_H$, using the methods previously discussed. We then construct surjective homomorphisms $f: X_G \to Z$ and $h: X_H \to Z$ by taking $f(0) = h(0) = 0$, $f(b) = h(b) = b$, $f(b^2) = h(b^2) = b^2$, and for any $u$ in $V_G$ or $v$ in $V_H$, $f(u) = h(v) = a$, with all other elements going to $c$. This construction is then used in the following theorem.

**Theorem 4.3.** There exists a homomorphism $g: X_G \to X_H$ with $f = hg$ if and only if there exists a homomorphism $\phi: G \to H$.

**Proof.** Suppose first that there exists a homomorphism $\phi: G \to H$. We construct a function $g: X_G \to X_H$ based on $\phi$—specifically, for any $v$ in $V_G$, we set $g(v) = \phi(v)$. We then set $g(0) = 0$, $g(b) = b$, $g(b^2) = b^2$, and $g(c) = c$. For any $u$, $v$ in $V_G$ with $(u, v)$ not in $E_G$, we map $\chi_{(u,v)}$ to $c$ if $(\phi(u), \phi(v))$ is in $E_H$ or $\chi(\phi(u), \phi(v))$ if it is not.

Such a $g$ is well-defined by construction. We claim that it is also a homomorphism. Suppose that $u$ and $v$ are elements of $X_G$ corresponding to $u$ and $v$ in $V_G$. It is either the case that $uv = vu = c$ or $uv = vu = \chi_{(u,v)}$.

Case 1: Suppose $uv = vu = c$. Since $g(c) = c$, we have $g(u)g(v) = \phi(u)\phi(v) = c = g(uv)$.
Case 2: Suppose $uv = vu = \chi_{(u,v)}$. If $(\phi(u), \phi(v))$ is in $E_H$, then by construction $g(uv) = g(\chi_{(u,v)}) = c$, and $g(u)g(v) = \phi(u)\phi(v) = c$. If $(\phi(u), \phi(v))$ is in $V^2_H \setminus E_H$, then by construction $g(uv) = g(\chi_{(u,v)}) = \chi(\phi(u), \phi(v))$ and $g(u)g(v) = \phi(u)\phi(v) = \chi(\phi(u), \phi(v))$.

Since $ub = bu = c$, we have that $g(u)g(b) = \phi(u)b = c = g(c) = g(ub)$. By construction, we also have that $g(u)g(c) = \phi(u)c = 0 = g(0) = g(uc)$ and $g(u)g(b^2) = \phi(u)b^2 = 0 = g(0) = g(ub^2)$. Since $g(\chi) = c$ or $c$ for any $\chi$ in $X_G$, $g(u)g(\chi) = \phi(u)g(\chi) = 0 = g(0) = g(u\chi)$. We now consider all other possible forms of $x$ and $y$ in $X_G$—in the special case where $x = y = b$, we have that $g(b)g(b) = bb = b^2 = g(b^2) = g(bb)$. In all other cases we have that $g(x)g(y) = 0 = g(0) = g(xy)$. The claim holds.

g also has the property that $f = hg$—this is clear from construction for all elements of $X_G$ aside from those coming from $V_G$. Suppose then that $v$ is in $V_G$. Then $g(v) = \phi(v)$, and thus $h(\phi(v)) = a$. Therefore, $hg(v) = a = f(v)$.

Suppose now that there exists a homomorphism $g: X_G \rightarrow X_H$ with $f = hg$. We claim that for any $v$ in $V_G$, $g(v)$ is an element of $X_H$ corresponding to some vertex in $V_H$. Since $v$ is in $V_G$, $f(v) = a$, and since $hg = f$, $h(g(v)) = a$. But the only elements which map to $a$ under $h$ are precisely those elements corresponding to vertices in $V_H$; therefore, the claim holds. We then construct a function, $\phi: G \rightarrow H$, by restricting $g$ to $V_G$—that is, $\phi(v) = g(v)$ for any $v$ in $V_G$.

Since $g$ is a homomorphism, $\phi$ is well-defined by construction. It will therefore suffice to show that if $u$ and $v$ are vertices in $V_G$ and $(u, v)$ is an edge in $E_G$, then $(\phi(u), \phi(v)) = (g(u), g(v))$ is an edge in $E_H$. Since $(u, v)$ is an edge in $E_G$, $uv = c$ in $X_G$. Consequently, $f(uv) = f(c) = c$, and similarly $h(g(uv)) = h(g(c)) = c$. It must then be either the case that $g(c) = c$ or $g(c) = \chi_{(u', v')}$ for some $u'$ and $v'$ in $V_H$. However, $g$ is a homomorphism, and $g(ub) = g(u)g(b) = g(c)$. Since $f(b) = h(b) = b$, and $b$ is the only element of $X_G$ that maps to $b$ under $f$ and the only element of $X_H$ that maps to $b$ under $h$, it must be the case that $g(b) = b$. Consequently, $g(ub) = g(u)b = c$, since $g(u)$ corresponds to a vertex in $V_H$ by previous argument. Therefore, $g(c) = c$, and it follows that $g(uv) = g(u)g(v) = c$. Necessarily, then, $(\phi(u), \phi(v)) = (g(u), g(v))$ is an edge in $E_H$. Since $\phi$ is well-defined and preserves the edge relation, it is indeed a homomorphism. 

**Corollary 4.4.** The Exists Right-Factor Problem for commutative semigroups is NP-complete.

This leaves the Exists Left-Factor Problem. Unlike in our previous arguments, the Exists Left-Factor Problem will require a modified construction, and requires the additional assumption that the graphs to be encoded are connected with at least two elements. Therefore, consider a connected, undirected graph, $G = (V_G, E_G)$. Create a new undirected graph $G'$ by taking $V_{G'} = V_G \cup \{w\}$ for some new distinguished vertex $w$, with $w$ not connected to any vertex in $G$. Then construct $X_{G'}$ as per the previous instructions.
then construct homomorphisms $X$ into semigroups, $h$ least two elements, a single vertex with no edges.

We now take as our input two connected, undirected graphs with at least two elements, $G = (V_G, E_G)$ and $H = (V_H, E_H)$. We encode $G$ and $H$ into semigroups, $X_{G'}$ and $X_{H'}$, using the methods previously discussed. We then construct homomorphisms $f: Z' \to X_{H'}$ and $g: Z' \to X_{G'}$ by taking $f(0) = g(0) = 0$, $f(b) = g(b) = b$, $f(b^2) = h(b^2) = b^2$, $f(a) = w$ for the $w$ in $H$, $g(a) = w$ for the $w$ in $G'$, $f(a^2) = \chi_{(w,w)}$ in $X_{H'}$, $g(a^2) = \chi_{(w,w)}$ in $X_{G'}$, and $f(c) = g(c) = c$. This construction is then used in the following theorem.

**Theorem 4.5.** There exists a homomorphism $h: X_{G'} \to X_{H'}$ with $f = hg$ if and only if there exists a homomorphism $\phi: G \to H$.

**Proof.** Suppose first that there exists a homomorphism $\phi: G \to H$. We construct a function $h: X_{G'} \to X_{H'}$ based on $\phi$ – specifically, for any $v$ in $V_G$, we set $h(v) = \phi(v)$. We then set $h(0) = 0$, $h(b) = b$, $h(b^2) = b^2$, $h(w) = w$, $h(\chi_{(w,w)}) = \chi_{(w,w)}$ and $h(c) = c$. For any $u$, $v$ in $V_G$ with $(u, v)$ in $V_G^2 \setminus E_G$, we map $\chi_{(u,v)}$ to $c$ if $(\phi(u), \phi(v))$ is in $E_H$ or $\chi_{(\phi(u), \phi(v))}$ if it is not.

Such an $h$ is well-defined by construction. We claim that it is also a homomorphism. Suppose that $u$ and $v$ are elements of $X_{G'}$ corresponding to $u$ and $v$ in $V_{G'}$. It is either the case that $uv = vu = c$ or $uv = vu = \chi_{(u,v)}$.

Case 1: Suppose $uv = vu = c$. Since $h(c) = c$, we have $h(u)h(v) = \phi(u)\phi(v) = c = h(uv)$.

Case 2: Suppose $uv = vu = \chi_{(u,v)}$. If $(\phi(u), \phi(v))$ is in $E_H$, then by construction $h(uv) = h(\chi_{(u,v)}) = c$, and $h(u)h(v) = \phi(u)\phi(v) = c$. If $(\phi(u), \phi(v))$ is in $V_{H'}^2 \setminus E_H$, then by construction $h(uv) = h(\chi_{(u,v)}) = \chi_{(\phi(u), \phi(v))}$ and $h(u)h(v) = \phi(u)\phi(v) = \chi_{(\phi(u), \phi(v))}$.

Since $ub = bu = c$, we have that $h(u)h(b) = \phi(u)b = c = h(c) = h(ub)$. By construction, we also have that $h(u)h(c) = \phi(u)c = 0 = h(0) = h(uc)$ and $h(u)h(b^2) = \phi(u)b^2 = 0 = h(0) = h(ub^2)$. Since $h(\chi) = c$ or $\chi$ for any $\chi$ in $X_G$, $h(u)h(\chi) = \phi(u)h(\chi) = 0 = h(0) = h(u\chi)$. We now consider all other possible forms of $x$ and $y$ in $X_G$—in the special case where $x = y = b$, we have that $h(b)h(b) = bb = b^2 = h(b^2) = h(bb)$. In all other cases we have that $h(x)h(y) = 0 = h(0) = h(xy)$. The claim holds.
Suppose now that there exists a homomorphism $h : X_{G'} \to X_H$ with $f = hg$. We claim that the restriction of $h$ to $V_G$ is a homomorphism $\phi : G \to H$. Since $h$ is a homomorphism, $\phi$ is well-defined by construction.

We first show that this restriction produces a map from $V_G$ to $V_H$. By the composition, it must be the case that $h(0) = 0$, $h(b) = b$, $h(b^2) = b^2$, $h(w) = w$, $h(\chi_{(w,w)}) = \chi_{(u,w)}$ and $h(c) = c$. It will therefore suffice to show that for any $v$ in $V_G$, $h(v)$ is an element of $X_H$ corresponding to some vertex in $V_H$. Since $v$ is in $V_G$, and $G$ is connected with at least two elements, there exists a $u$ in $V_G$ with $(u,v)$ in $E_G$. So $uv = c$, and therefore $h(uv) = h(c) = c = h(u)h(v)$. It must either be that $h(v)$ is in $V_H$ or $h(v) = b$. However, the second case cannot occur: since $h(vb) = h(c) = c = h(v)h(b) = h(v)b$, then if $h(v) = b$ it follows that $h(v)b - b^2 \neq c$. We indeed have a map from $V_G$ to $V_H$.

It will now suffice to show that if $u$ and $v$ are in $V_G$ and $(u,v)$ is in $E_G$, then $(\phi(u), \phi(v)) = (h(u), h(v))$ is in $E_H$. Since $(u,v)$ is in $E_G$, $uv = c$ in $X_{G'}$. Consequently, $h(u)h(v) = h(uv) = h(c) = c$. Necessarily, then, $(\phi(u), \phi(v)) = (h(u), h(v))$ is in $E_H$. Therefore, $\phi$ preserves edges. The claim holds.

Corollary 4.6. The Exists Left-Factor Problem for commutative semigroups is NP-complete.

5. The proof of Theorem 1.2

To complete the proof of Theorem 1.2, it will suffice to show the following:

Theorem 5.1. The Exists Right-Factor Problem is NP-complete for algebras in a rich language.

Proof. Suppose our language contains two or more operations of arity one but no operations of larger arity. We use two operations to define the operations of $G^1$ as shown in Table 1, and allow our remaining operations to be unary projections. If there are no constants, we choose $Z$ to be the single element algebra, and let $X$ and $Y$ encode two loop-free, finite, connected, directed graphs. We then set $f : X \to Z$ and $h : Y \to Z$ to be the constant mappings. Then by Theorem 2.5, there exists a $g : X \to Y$ with $f = hg$ if and only if there exists a homomorphism between the two graphs encoded by $X$ and $Y$. If our language contains constants, we choose $Z$ to be the two element algebra $\{0, 1\}$ with all operations acting as the identity on both elements, and mapping all constants to 0. Let $X$ and $Y$ encode two loop-free, finite, connected, directed graphs with the operations acting as the identity on constants. We then set $f : X \to Z$ and $h : Y \to Z$ to be the homomorphisms that map all non-constant elements to 1 and all constants to 0. Then by Theorem 2.5, there exists a $g : X \to Y$ with $f = hg$ if and only if there exists a homomorphism between the two graphs encoded by $X$ and $Y$.

Suppose our language contains an operation of arity $n \geq 2$, $o$. We define a modified version of the operation of the semigroup $X_G$, as shown in Table 4, based on the first two components of our operation, and ignore all other components. Specifically, we set $o(x_1, x_2, \ldots, x_n) = x_1 \cdot x_2$. We set the
remaining operations to be projections. We choose $Z$ to be the distinguished algebra based on Table 5 constructed using this operation, setting $x_1 \cdot x_2 = 0$ if either $x_1$ or $x_2$ is a constant in our language. Similarly, we let $X$ and $Y$ encode two loop-free, finite, undirected graphs, again setting $x_1 \cdot x_2 = 0$ if either $x_1$ or $x_2$ is a constant in our language. We then set $f: X \to Z$ and $h: Y \to Z$ by $f(d) = h(d) = 0$ for any constant $d$, $f(0) = h(0) = 0$, $f(b) = h(b) = b$, $f(b^2) = h(b^2) = b^2$, and for any $u$ in $V_G$ or $v$ in $V_H$, $f(u) = h(v) = a$, with all other elements going to $c$. Then by Theorem 4.3, there exists a $g: X \to Y$ with $f = hg$ if and only if there exists a homomorphism between the two graphs encoded by $X$ and $Y$.

By similar constructions, we also observe:

**Corollary 5.2.** *The Exists Left-Factor and Retraction Problems are NP-complete for algebras in a rich language. The Isomorphism Problem is GI-hard for algebras in a rich language.*

The presence of constants can affect the complexity of the Homomorphism Problem, and our constructions do not point towards a general result when they are permitted. By contrast, our constructions can be used to classify the complexity of the Homomorphism Problem when we forbid constants.

**Theorem 5.3.** *The Homomorphism Problem is NP-complete for algebras in a rich language with no constants.*

*Proof.* Suppose our language contains two or more operations of arity one but no operations of larger arity. We use two operations to define the operations of $G^t$ as shown in Table 1, and allow our remaining operations to be unary projections. Let $X$ and $Y$ encode two loop-free, finite, connected, directed graphs. By Theorem 2.5, there exists a homomorphism $f: X \to Y$ if and only if there exists a homomorphism between the two graphs encoded by $X$ and $Y$.

Suppose our language contains an operation of arity $n \geq 2$, $o$. We define the operation of the algebra $G^*$, as shown in Table 3, based on the first two components of our operation, and ignore all other components. Specifically, we set $o(x_1, x_2, \ldots, x_n) = x_1 \cdot x_2$. We set the remaining operations to be projections. We let $X$ and $Y$ encode two loop-free, finite, undirected graphs with at least two vertices. By Theorem 3.1, there exists a homomorphism $f: X \to Y$ if and only if there exists a strong homomorphism between the two graphs encoded by $X$ and $Y$.

By similar constructions, we also observe:

**Corollary 5.4.** *The Homomorphism Factorization, Exists Right-Factor, Exists Left-Factor, and Retraction Problems are NP-complete for algebras in a rich language with no constants. The Isomorphism Problem is GI-hard for algebras in a rich language with no constants.*
6. Future work

Our results suggest two possibilities for investigating when homomorphism factorization can be performed in polynomial time: either the algebraic language must not be rich, or additional requirements must be imposed upon a variety in a rich language. We intend to address these special cases in a work to appear at a later date.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Bergman, C., Slutzki, G.: Complexity of some problems concerning varieties and quasi-varieties of algebras. SIAM J. Comput. 30(2), 359–382 (2000)
[2] Booth, K.S.: Isomorphism testing for graphs, semigroups, and finite automata are polynomially equivalent problems. SIAM J. Comput. 7(3), 273–279 (1978)
[3] Distler, A., Mitchell, J.D.: The number of nilpotent semigroups of degree 3. Electron. J. Comb. 19(2), P51 (2012)
[4] Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. Springer, Berlin (2006)
[5] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. A Series of Books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco (1979)
[6] Hedrlín, Z., Pultr, A.: On full embeddings of categories of algebras. Ill. J. Math. 10(3), 392–406 (1966)
[7] Hell, P., Nešetřil, J.: Graphs and Homomorphisms. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, Oxford (2004)
[8] Hoffmann, C.M.: Group-Theoretic Algorithms and Graph Isomorphism. Lecture Notes in Computer Science, vol. 136. Springer, Berlin (1982)
[9] Jackson, M., McKenzie, R.: Interpreting graph colorability in finite semigroups. Int. J. Algebra Comput. 16(01), 119–140 (2006)
[10] Torán, J.: Reductions to graph isomorphism. Theory Comput. Syst. 47(1), 288–299 (2010)
[11] Van Name, J.: Is there a good computer program for searching for endomorphisms between finite algebras which make diagrams commute? Is this problem NP–complete? https://mathoverflow.net/questions/261966/. Accessed 14 Feb 2018
Kevin M. Berg
Department of Mathematics
University of Colorado
Boulder CO
USA

*Present Address*
Kevin M. Berg
Faculty of Mathematics and Physics
Charles University
Prague
Czech Republic
e-mail: berg.kevinm@gmail.com

Received: 8 January 2020.
Accepted: 6 June 2021.