The Complexity of Homomorphism Factorization

Kevin M. Berg

Faculty of Mathematics and Physics
Charles University

February 21, 2020

This material is based upon work at the University of Colorado supported by the National Science Foundation grant no. DMS 1500254.

The Homomorphism Factorization Problem

We assume throughout that all algebras are finite.
Fix an algebraic language $\mathcal{L}$.

Problem (The Homomorphism Factorization Problem)

Given a homomorphism $f\colon X \to Z$ between $\mathcal{L}$-algebras X and Z and an intermediate $\mathcal{L}$-algebra Y, decide whether there are homomorphisms $g\colon X \to Y$ and $h\colon Y \to Z$ such that $f = hg$.

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, dotted, "\exists g?"'] & & Z \\ & Y \arrow[ur, dotted, "\exists h?"'] & \end{tikzcd}$$

Figure: The general form of the commutative diagram for Homomorphism Factorization Problems.

Variants of the Homomorphism Factorization Problem

Problem (I. The Homomorphism Problem)

When $|Z| = 1$, the homomorphisms f and h from the HFP must be constant, reduces to the problem of deciding whether, given $\mathcal{L}$-algebras X and Y, there is a homomorphism $g\colon X \to Y$.

$$\begin{tikzcd}
X \arrow[rr, "f(x)=\bullet"] \arrow[dr, dotted, "\exists g?"'] & & \bullet \\
& Y \arrow[ur, "h(x)=\bullet"'] &
\end{tikzcd}$$

Figure: The commutative diagram for the Homomorphism Problem.

Variants of the Homomorphism Factorization Problem

Problem (I. The Homomorphism Problem)

When $|Z| = 1$, the homomorphisms f and h from the HFP must be constant, reduces to the problem of deciding whether, given $\mathcal{L}$-algebras X and Y, there is a homomorphism $g\colon X \to Y$.

$$
\begin{tikzcd}
X \arrow[rr, "f(x)=\bullet"] \arrow[dr, dotted, "\exists g?"'] & & \bullet \\
& Y \arrow[ur, "h(x)=\bullet"'] &
\end{tikzcd}
$$

Figure: The commutative diagram for the Homomorphism Problem.

The Homomorphism Factorization Problem also generalizes the Retraction Problem and the Isomorphism Problem.

Variants of the Homomorphism Factorization Problem

Problem (II. The Exists Right-Factor Problem)

Given $\mathcal{L}$-algebras X, Y, and Z, and homomorphisms $f\colon X \to Z$ and $h\colon Y \to Z$, decide whether there is a homomorphism $g\colon X \to Y$ such that $f = hg$.

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, dotted, "\exists g?"'] & & Z \\ & Y \arrow[ur, "h"'] & \end{tikzcd}$$

Figure: The commutative diagram for the Exists Right-Factor Problem.

Remark

The ERFP was proposed in the context of semigroups by J. Van Name in 2017.

Variants of the Homomorphism Factorization Problem

Problem (II. The Exists Right-Factor Problem)

Given $\mathcal{L}$-algebras X, Y, and Z, and homomorphisms $f\colon X \to Z$ and $h\colon Y \to Z$, decide whether there is a homomorphism $g\colon X \to Y$ such that $f = hg$.

$$\begin{tikzcd}
X \arrow[rr, "f"] \arrow[dr, dotted, "\exists g?"'] & & Z \\
& Y \arrow[ur, "h"'] &
\end{tikzcd}$$

Figure: The commutative diagram for the Exists Right-Factor Problem.

Remark

The ERFP was proposed in the context of semigroups by J. Van Name in 2017. There is a corresponding Exists Left-Factor Problem.

Graphs

To investigate the computational complexity of Homomorphism Factorization Problems, we will make use of complexity results from graph theory.

Graphs

To investigate the computational complexity of Homomorphism Factorization Problems, we will make use of complexity results from graph theory.

Definition (Directed Graph, G)

$G = (V_G, E_G)$ is a relational structure consisting of a universe, V_G, of vertices, together with an edge relation, E_G.

Graphs

To investigate the computational complexity of Homomorphism Factorization Problems, we will make use of complexity results from graph theory.

Definition (Directed Graph, G)

$G = (V_G, E_G)$ is a relational structure consisting of a universe, V_G, of vertices, together with an edge relation, E_G.

Definition (Undirected Graph, G)

$G = (V_G, E_G)$ is a relational structure consisting of a universe, V_G, of vertices, together with a symmetric edge relation, E_G.

Graphs

To investigate the computational complexity of Homomorphism Factorization Problems, we will make use of complexity results from graph theory.

Definition (Directed Graph, G)

$G = (V_G, E_G)$ is a relational structure consisting of a universe, V_G, of vertices, together with an edge relation, E_G.

Definition (Undirected Graph, G)

$G = (V_G, E_G)$ is a relational structure consisting of a universe, V_G, of vertices, together with a symmetric edge relation, E_G.

Unless stated otherwise, we assume all graphs are loop-free and connected.

Graph Homomorphism Problems

The following results hold for both directed and undirected graphs.

Graph Homomorphism Problems

The following results hold for both directed and undirected graphs.

Theorem (Graph Homomorphism Problem)

Given two finite graphs, G and H, the question of whether there exists a relational homomorphism $\phi\colon G \to H$ is NP-complete.

Graph Homomorphism Problems

The following results hold for both directed and undirected graphs.

Theorem (Graph Homomorphism Problem)

Given two finite graphs, G and H, the question of whether there exists a relational homomorphism $\phi\colon G \to H$ is NP-complete.

Theorem (Strong Graph Homomorphism Problem)

Given two finite graphs, G and H, the question of whether there exists a strong relational homomorphism $\phi\colon G \to H$ is NP-complete.

Rich Languages

An algebraic language $\mathcal{L}$ is *rich* if it has at least one operation of arity at least two, or at least two operations of arity one.

Rich Languages

An algebraic language $\mathcal{L}$ is *rich* if it has at least one operation of arity at least two, or at least two operations of arity one.

Theorem (B.)

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

Two Unary Operations

Let $G = (V_G, E_G)$ be a directed graph with at least two vertices.

Two Unary Operations

Let $G = (V_G, E_G)$ be a directed graph with at least two vertices.

Definition ($G^\dagger$)

For every v in V_G, there are two corresponding elements v_1 and v_2 in $G^\dagger$, and for each edge (u, v) in E_G, there are two elements, $a_{(u,v)}$ and $b_{(u,v)}$, in $G^\dagger$. We assign to $G^\dagger$ two unary operations: $f(\cdot)$ and $g(\cdot)$.

Table: The operations of $G^\dagger$.

	$f(\cdot)$	$g(\cdot)$
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	$a_{(u,v)}$

HFPs for Algebras with Two Unary Operations

Theorem (B.)

Let G and H be directed graphs with at least two vertices. There exists a homomorphism $\phi\colon G \to H$ if and only if there exists a homomorphism $\psi\colon G^{\dagger} \to H^{\dagger}$.

HFPs for Algebras with Two Unary Operations

Theorem (B.)

Let G and H be directed graphs with at least two vertices. There exists a homomorphism $\phi\colon G \to H$ if and only if there exists a homomorphism $\psi\colon G^{\dagger} \to H^{\dagger}$.

Corollary

The Homomorphism Problem for finite algebras in a language with two unary operations is NP-complete. Consequently, the Homomorphism Factorization Problem for finite algebras in a language with two unary operations is also NP-complete.

Complications

This approach using the Homomorphism Problem does not work for semigroups.

Complications

This approach using the Homomorphism Problem does not work for semigroups.

Proposition

Every finite semigroup has an idempotent.

Complications

This approach using the Homomorphism Problem does not work for semigroups.

Proposition

Every finite semigroup has an idempotent.

Corollary

The Homomorphism Problem for finite semigroups is in P.

Complications

This approach using the Homomorphism Problem does not work for semigroups.

Proposition

Every finite semigroup has an idempotent.

Corollary

The Homomorphism Problem for finite semigroups is in P.

We instead focus on the Exists Right-Factor Problem for semigroups.

Graph Encoding into Semigroups

Let $G = (V_G, E_G)$ be an undirected graph with at least one vertex.

Graph Encoding into Semigroups

Let $G = (V_G, E_G)$ be an undirected graph with at least one vertex.

Definition (Semigroup X_G)

The universe of X_G consists of an element, v, for each v in V_G; an element, $\chi_{u,v}$, for each u, v in V_G such that (u, v) is not an element of E_G (let $\chi_{u,v} = \chi_{v,u}$); and auxiliary elements b, b^2, c, and 0. We assign to X_G a binary operation, $\cdot$, that encodes whether two vertices are adjacent or not adjacent.

Graph Encoding into Semigroups

Let $G = (V_G, E_G)$ be an undirected graph with at least one vertex.

Definition (Semigroup X_G)

The universe of X_G consists of an element, v, for each v in V_G; an element, $\chi_{u,v}$, for each u, v in V_G such that (u, v) is not an element of E_G (let $\chi_{u,v} = \chi_{v,u}$); and auxiliary elements b, b^2, c, and 0. We assign to X_G a binary operation, $\cdot$, that encodes whether two vertices are adjacent or not adjacent.

We also define a target semigroup Z by encoding the graph consisting of a single loop on a vertex a.

Homomorphism Factorization for Semigroups

For two undirected graphs G and H, we let $f\colon X_G \to Z$ and $h\colon X_H \to Z$ be defined so as to preserve distinguished elements and so that there is a graph homomorphism to the single loop on the vertex a.

Homomorphism Factorization for Semigroups

For two undirected graphs G and H, we let $f\colon X_G \to Z$ and $h\colon X_H \to Z$ be defined so as to preserve distinguished elements and so that there is a graph homomorphism to the single loop on the vertex a.

Theorem (B.)

There exists a semigroup homomorphism $g\colon X \to Y$ with $f = hg$ if and only if there exists a graph homomorphism $\phi\colon G \to H$.

Homomorphism Factorization for Semigroups

For two undirected graphs G and H, we let $f\colon X_G \to Z$ and $h\colon X_H \to Z$ be defined so as to preserve distinguished elements and so that there is a graph homomorphism to the single loop on the vertex a.

Theorem (B.)

There exists a semigroup homomorphism $g\colon X \to Y$ with $f = hg$ if and only if there exists a graph homomorphism $\phi\colon G \to H$.

Corollary

The Exists Right-Factor Problem for finite semigroups is NP-complete.

Homomorphism Factorization for Semigroups

For two undirected graphs G and H, we let $f\colon X_G \to Z$ and $h\colon X_H \to Z$ be defined so as to preserve distinguished elements and so that there is a graph homomorphism to the single loop on the vertex a.

Theorem (B.)

There exists a semigroup homomorphism $g\colon X \to Y$ with $f = hg$ if and only if there exists a graph homomorphism $\phi\colon G \to H$.

Corollary

The Exists Right-Factor Problem for finite semigroups is NP-complete.

Corollary

The Homomorphism Factorization Problem for finite semigroups is NP-complete.

Rich Languages

Theorem (B.)

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

Rich Languages

Theorem (B.)

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

Proof Idea.

Show that the Exists Right-Factor Problem is NP-complete using either the two unary operations construction or the semigroup construction, depending on the number and arity of the operations. □

Rich Languages

Theorem (B.)

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

Proof Idea.

Show that the Exists Right-Factor Problem is NP-complete using either the two unary operations construction or the semigroup construction, depending on the number and arity of the operations. □

In addition, using these and other encodings it was possible to classify the computational complexity of other variants of the Homomorphism Factorization Problem for rich languages – for example, the Homomorphism Problem for algebras with a single, non-associative binary operation is NP-complete, and the Retraction Problem for semigroups is also NP-complete.

Polynomial Time Cases

Our results suggested two approaches for determining when Homomorphism Factorization Problems are in P:

Polynomial Time Cases

Our results suggested two approaches for determining when Homomorphism Factorization Problems are in P:

1. Classify the complexity of Homomorphism Factorization Problems for algebraic languages that are not rich.

Polynomial Time Cases

Our results suggested two approaches for determining when Homomorphism Factorization Problems are in P:

1. Classify the complexity of Homomorphism Factorization Problems for algebraic languages that are not rich.
2. Determine which specific varieties in rich languages have Homomorphism Factorization Problems in P.

Polynomial Time Cases

Our results suggested two approaches for determining when Homomorphism Factorization Problems are in P:

1. Classify the complexity of Homomorphism Factorization Problems for algebraic languages that are not rich.
2. Determine which specific varieties in rich languages have Homomorphism Factorization Problems in P.

Results in Case 1 have been found in, or can be derived from, prior research into monounary algebras. For this talk, we will focus on Case 2.

Case 2: A Strategy for Additional Polynomial Time Cases

Recall the general commutative diagram for Homomorphism Factorization Problems:

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, dotted, "\exists g?"'] & & Z \\ & Y \arrow[ur, dotted, "\exists h?"'] & \end{tikzcd}$$

Case 2: A Strategy for Additional Polynomial Time Cases

Suppose that there exists a retraction $r\colon X \to X$ with the following property:

Case 2: A Strategy for Additional Polynomial Time Cases

Suppose that there exists a retraction $r\colon X \to X$ with the following property:

Definition (Respects f)

A retraction $r\colon X \to X$ **respects** a homomorphism $f\colon X \to Z$ if $fr = f$.

Case 2: A Strategy for Additional Polynomial Time Cases

Suppose that there exists a retraction $r\colon X \to X$ with the following property:

Definition (Respects f)

A retraction $r\colon X \to X$ **respects** a homomorphism $f\colon X \to Z$ if $fr = f$.

$$\begin{tikzcd}
X \arrow[loop left, "r"] \arrow[rr, "f"] \arrow[dr, "g"'] & & Z \\
& Y \arrow[ur, "h"'] &
\end{tikzcd}$$

Case 2: A Strategy for Additional Polynomial Time Cases

Let $X' = r(X)$. We have:

Case 2: A Strategy for Additional Polynomial Time Cases

Let $X' = r(X)$. We have:

$$
\begin{tikzcd}
X \arrow[r, bend right, "r"'] \arrow[rrr, bend left, "f"] \arrow[drr, bend right, "g"'] & X' \arrow[l, bend right, "\supseteq"'] & & Z \\
& & Y \arrow[ur, "h"'] &
\end{tikzcd}
$$

Case 2: A Strategy for Additional Polynomial Time Cases

Let $X' = r(X)$. We have:

$$
\begin{tikzcd}
X \arrow[rrr, bend left, "f"] \arrow[r, bend right, "r"'] \arrow[drr, bend right, "g"'] & X' \arrow[l, bend right, "\supseteq"] \arrow[rr, "f|_{X'}"] \arrow[dr, "g'"'] & & Z \\
& & Y \arrow[ur, "h"'] &
\end{tikzcd}
$$

Case 2: A Strategy for Additional Polynomial Time Cases

Let $X' = r(X)$. We have:

$$\begin{tikzcd}
X \arrow[r, bend right, "r"] \arrow[rrr, bend left=40, "f"] \arrow[drr, bend right, "g"'] & X' \arrow[l, bend right, "\supseteq"] \arrow[rr, "f|_{X'}"] \arrow[dr, "g'"'] & & Z \\
& & Y \arrow[ur, "h"'] &
\end{tikzcd}$$

If r respects f, then $fr = f$. Consequently,

Case 2: A Strategy for Additional Polynomial Time Cases

Let $X' = r(X)$. We have:

$$\begin{tikzcd}
X \arrow[r, bend right, "r"'] \arrow[rrr, bend left=40, "f"] \arrow[drr, bend right, "g"'] & X' \arrow[l, bend right, "\supseteq"'] \arrow[rr, "f|_{X'}"] \arrow[dr, "g'"] & & Z \\
& & Y \arrow[ur, "h"'] &
\end{tikzcd}$$

If r respects f, then $fr = f$. Consequently,

Proposition

f factors through Y if and only if $f|_{X'}$ factors through Y.

Case 2: A Strategy for Additional Polynomial Time Cases

Definition (f-Core)

A is an **f-core of X** if A is minimal with respect to the existence of an onto, f-respecting retraction, $r\colon X \to A$. If X is its own f-core, X is an **f-core**.

Case 2: A Strategy for Additional Polynomial Time Cases

Definition (f-Core)

A is an f-**core of** X if A is minimal with respect to the existence of an onto, f-respecting retraction, $r\colon X \to A$. If X is its own f-core, X is an f-**core**.

Definition (Bounded f-Core)

Variety $\mathcal{V}$ has *bounded f-cores* if, for any finite algebra Z, there exists a function s such that for any finite algebra X in $\mathcal{V}$ and any surjective homomorphism $f\colon X \to Z$ for which X is an f-core, $|X| \leq s(|Z|)$.

Bounded f-Cores and the Exists Right-Factor Problem

When the target algebra Z is fixed, with bounded f-cores:

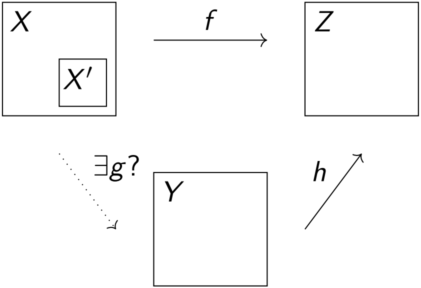
Bounded f-Cores and the Exists Right-Factor Problem

When the target algebra Z is fixed, with bounded f-cores:

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, dotted, "\exists g?"] & & Z \\ & Y \arrow[ur, "h"'] & \end{tikzcd}$$

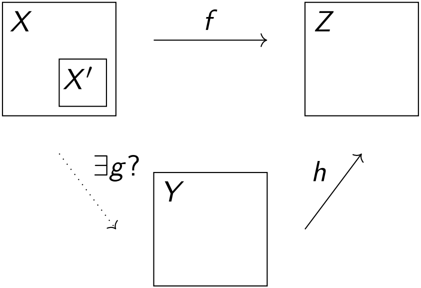
Bounded f-Cores and the Exists Right-Factor Problem

When the target algebra Z is fixed, with bounded f-cores:



Bounded f-Cores and the Exists Right-Factor Problem

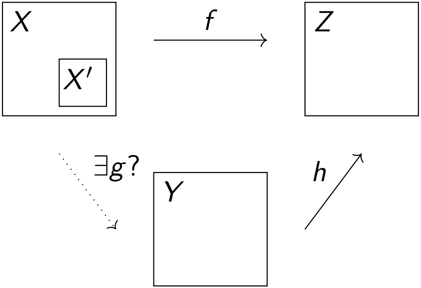
When the target algebra Z is fixed, with bounded f-cores:



There are $|Y|^{|X'|} \leq |Y|^{s(|Z|)}$ many choices for g;

Bounded f-Cores and the Exists Right-Factor Problem

When the target algebra Z is fixed, with bounded f-cores:



There are $|Y|^{|X'|} \leq |Y|^{s(|Z|)}$ many choices for g; the number of possible solutions to the Exists Right-Factor Problem with fixed Z is bounded above by a polynomial in $|Y|$.

Bounded f-Cores and P

Theorem (B.)

The Exists Right-Factor Problem with fixed Z for a variety $\mathcal{V}$ is in P if the following conditions hold:

Bounded f-Cores and P

Theorem (B.)

The Exists Right-Factor Problem with fixed Z for a variety $\mathcal{V}$ is in P if the following conditions hold:

1. *$\mathcal{V}$ has bounded f-cores.*

Bounded f-Cores and P

Theorem (B.)

The Exists Right-Factor Problem with fixed Z for a variety $\mathcal{V}$ is in P if the following conditions hold:

1. $\mathcal{V}$ *has bounded* f*-cores.*
2. *The* f*-cores of finite algebras in* $\mathcal{V}$ *can be found in polynomial time.*

Bounded f-Cores and P

Theorem (B.)

The Exists Right-Factor Problem with fixed Z for a variety $\mathcal{V}$ is in P if the following conditions hold:

1. $\mathcal{V}$ *has bounded* f*-cores.*
2. *The* f*-cores of finite algebras in* $\mathcal{V}$ *can be found in polynomial time.*
3. *Given a finite algebra* X *in* $\mathcal{V}$*, a retraction from* X *to its* f*-core can be found in polynomial time.*

Varieties with Bounded f-Cores

Let f be a homomorphism appropriately defined for a given variety.

Varieties with Bounded f-Cores

Let f be a homomorphism appropriately defined for a given variety.

Theorem (Boolean Algebras)

Boolean algebras have bounded f-cores, and the Exists Right-Factor Problem with fixed Z is in P.

Varieties with Bounded f-Cores

Let f be a homomorphism appropriately defined for a given variety.

Theorem (Boolean Algebras)

Boolean algebras have bounded f-cores, and the Exists Right-Factor Problem with fixed Z is in P.

Theorem (Vector Spaces)

Let F be a field. The variety of vector spaces over F has bounded f-cores, and the Exists Right-Factor Problem with fixed Z is in P.

Can We Do Better?

The arguments for both Boolean algebras and vector spaces can be extended to show that the Homomorphism Factorization Problem for these languages is also in P. Could the presence of bounded f-cores be a sufficient condition for this to hold in general?

Can We Do Better?

The arguments for both Boolean algebras and vector spaces can be extended to show that the Homomorphism Factorization Problem for these languages is also in P. Could the presence of bounded f-cores be a sufficient condition for this to hold in general? Maybe.

Can We Do Better?

The arguments for both Boolean algebras and vector spaces can be extended to show that the Homomorphism Factorization Problem for these languages is also in P. Could the presence of bounded f-cores be a sufficient condition for this to hold in general? Maybe.

Thank you for your time.