Exact Learning of Equational Theories

Benjamin Caulfield bcaulfield@berkeley.edu
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1776, USA

Ashish Tiwari tiwari@csl.sri.com
Computer Science Laboratory
SRI, International
Menlo Park, CA 94025, USA

Editor:

Abstract

Equational logic is a formalism used to describe infinite sets of equations between terms (*theories*) using finite sets of equations (*presentations*). Both functional programs and logic programs can be naturally represented as equational logic presentations. Therefore, learning presentations in equational logic can be seen as a form of program synthesis. In general, it is undecidable whether terms are equivalent via a finite presentation. So learning equational presentations is generally intractable.

This paper investigates the exact learning of a restricted class of theories known as *non-collapsing shallow theories*, which can be presented by equations where variables only appear at depth one. The learning algorithms use examples and queries of equations between ground terms, meaning there are no variables in the equations. It is shown that these theories cannot be learned in the limit from only positive examples. A polynomial time algorithm is given which creates a hypothesis presentation consistent with positive and negative examples which will learn in the limit a presentation for the target theory. Finally, an algorithm is given which learns a presentation in polynomial time from a minimally adequate teacher. It is shown that the learned presentations are canonical with respect to an ordering on terms and the presentation is at most polynomially larger than the minimal presentation of the same theory.

1. Introduction

Term rewriting systems are among the many formalisms capable of describing all turing-computable functions. Lambda calculus, combinatory logic, and the theory of turing machines all have very natural representations as term rewriting systems. Often, the order on rewrite rules is ignored, and rewrite rules are treated as sets of unordered equations (identities) between terms. Algorithms such as the Knuth-Bendix algorithm (Knuth and Bendix, 1983) are then applied to such sets of equations to create term rewriting systems with desirable properties. Because of the close connection between these equations and programs (Van Emden and Yukawa, 1987; O'donnell, 1985), applying the Knuth-Bendix algorithm to equations can be seen as a form of program-synthesis (Dershowitz, 1985). Dershowitz and Reddy have also studied the inductive synthesis of programs using ordered-rewriting (Dershowitz and Reddy, 1993). Although there has also been work on learning term-rewriting systems (Arimura et al., 2000; Rao, 2006), there has been much less work on the learning of equational systems, themselves.

The seemingly unrelated field of grammatical inference aims to learn formal languages from information about a target language, such as positive (resp. negative) examples which are in (resp. not in) the target language. Grammatical inference was introduced by Gold in 1967, where he provided very simple classes of languages which are not learnable from positive examples (Gold, 1967) One of the most important results from this field is a polynomial-time algorithm by Angluin for learning regular languages from membership and equivalence queries to an oracle (Angluin, 1987). A membership query presents a string to the oracle and asks "Is this string in the target language?" . An equivalence query presents a hypothesis language to the oracle in the form of a deterministic finite automaton (DFA) and asks "Is this the target language?" . If the answer is "no", the oracle presents a counter-example from the symmetric difference of the hypothesis language and the target language. The algorithm returns the canonical minimal DFA accepting the target language. This result has been generalized to a polynomial-time learning algorithm for rational tree languages (Sakakibara, 1990).

The goal of the present work is to learn finite sets of equations over terms using examples or queries. We focus on exact learning, where the learned equational theory must be equivalent to the target equational theory. Although we do not assume any background in grammatical inference, many of the techniques used in this paper are inspired by this field, and we will draw parallels between our methods and grammatical inference wherever possible. The basic analogy to keep in mind while reading this paper is that equational theories (the equivalence relation between terms) are like formal languages and equations between terms are like strings in the language. Analogous definitions of positive & negative examples and membership & equivalence queries can be applied to learning equational theories.

Of course, if we allow for examples to be any equations, then learning becomes easy. Just use all positive equations as the hypothesis equational presentation. Therefore, we restrict ourselves to studying examples and queries on only ground equations (i.e., equations without variables). We can think of these ground equations as instantiations of non-ground equations in the "real world". This method, however, has the downside that we cannot distinguish between two different equational theories that agree on ground terms. For example, given alphabet $\Sigma := \{f : 1, a : 0\}$ consider the equational theories generated by $E_0 := \{f(f(a)) \approx f(a)\}$ and $E_1 := \{f(x) \approx f(y)\}$ for variables x. Both presentations agree on all ground equations, but the equation $f(x) \approx f(y)$ is provable in E_1 but not E_0. Therefore we only require that the learned equational theory be equal to the target theory on all ground terms.

The paper presents polynomial-time algorithms for the exact learning of *non-collapsing shallow theories*, which are presentable by equations with variables appearing only at depth 1 . Shallow theories, which are presentable by equations with variables appearing at depth 0 or 1, were first introduced by Comon, Haberstrau, and Jouannaud as a non-trivial class of equational theories with a decidable word problem (Comon et al., 1992, 1994). Later work by Niewenhuis showed that the word problem for shallows theories is solvable polynomial time (Nieuwenhuis, 1996). It seems unlikely that there could exist a polynomial-time learning algorithm for a class of theories with no polynomial algorithm for the word problem. Therefore, the set of non-collapsing shallow theories is among the most expressive classes of theories for which an efficient learning algorithm might reasonably exist.

2. Notation and Background

Terms and Substitutions

We follow the book by Baader and Nipkow (Baader and Nipkow, 1999). For clarity, we will use := when defining new objects and = to denote that two objects are equivalent in the normal sense. A *signature* (or *ranked alphabet*) Σ consists of a set of *function* symbols with an associated *arity*, a non-negative number indicating the number of arguments. For example $\Sigma := \{f : 2, a : 0, b : 0\}$ consists of binary function symbol f and constants a and b. For any arity $n \geq 0$, we let $\Sigma^{(n)}$ denote the set of function symbols with arity n (the n-ary symbols). We will refer to the 0-ary function symbols as *constants*.

For any signature Σ and set of *variables* X such that $\Sigma \cap X = \emptyset$, we define the set $T(\Sigma, X)$ of Σ-terms over X inductively as the smallest set satisfying:

- $\Sigma^{(0)}, X \subseteq T(\Sigma, X)$
- For all $n \geq 1$, all $f \in \Sigma^{(n)}$, and all $t_1, \ldots, t_n \in T(\Sigma, X)$, we have $f(t_1, \ldots, t_n) \in T(\Sigma, X)$.

Unless otherwise stated, we will use variations of a, b, and c to denote constants, x, y, and z to denote variables, and f, g, and h to denote non-constant function symbols.

We define the set of *ground terms* of Σ to be the set $T(\Sigma, \emptyset)$, which we will sometimes write $T(\Sigma)$.

The set of *positions* of a term t, denoted $Pos(t)$, is a set of strings over the alphabet of positive integers. It is inductively defined as follows:

- If $t \in X \cup \Sigma^{(0)}$, then $Pos(t) := \{\epsilon\}$
- If $t = f(t_1, \ldots, t_n)$, then $Pos(t) := \{\epsilon\} \cup \bigcup_{i=1}^{n} \{ip \mid p \in Pos(t_i)\}$

Here, ϵ represents the empty string and is called the *root position* of t. For positions p and q, we say $p \leq q$ if there exists a position p' such that $pp' = q$. We say p is parallel to q, denoted $p \| q$, if $p \not\leq q$ and $q \not\leq p$.

For $p \in Pos(t)$, the subterm of t at position p, denoted $t|_p$ is defined by:

- $t|_\epsilon := t$
- $t|_{ip'} := t_i|_{p'}$, if $t = f(t_1, \ldots, t_n)$

For $p \in Pos(t)$, the term $t[s]_p$ is created by replacing the subterm at position p with s. In other words,

- $t[s]_\epsilon := s$
- $f(t_1, \ldots, t_n)[s]_{ip'} := f(t_1, \ldots, t_i[s]_{p'}, \ldots, t_n)$

This can be extended to a set $I \subset Pos(t)$ of parallel positions, so $t[s]_I$ replaces each term at each $i \in I$ with s.

An *equation* is an ordered-pair of terms s and t, written $s \approx t$. Given a set E of equations, new equations can be derived using the rules of *equational logic* as follows:

$$\vdash s \approx s \text{ (reflexive)}$$

$$s \approx t \vdash t \approx s \text{ (symmetric)}$$

$$s \approx t, t \approx u \vdash s \approx u \text{ (transitive)}$$

$$s_1 \approx t_1, \ldots, s_k \approx t_k \vdash f(s_1, \ldots, s_k) \approx f(t_1, \ldots, t_k) \text{ (congruence)}$$

$$s \approx t \vdash s\sigma \approx t\sigma \text{ (substitution)}$$

Let $Th(E)$ be the set of equations, $s \approx t$, such that $E \vdash s \approx t$. Moreover, let $Th_G(E)$ be the set of ground equations in $Th(E)$. We say two sets of equations, E and E', are *ground-equivalent* (written $E \equiv_G E'$) if $Th_G(E) = Th_G(E')$. A presentation of T is any finite set E of equations such that $Th(E) = T$.

Alternatively, we say that $s \approx_e^p t$ if there is a substitution σ, an equation $e := l \approx r$, and a position $p \in Pos(s)$ such that $s|_p = l\sigma$ and $s[r\sigma]_p = t$. The equation $s \approx_e t$ denotes that there is a $p \in Pos(s)$ such that $s \approx_e^p t$. We can think of this as replacing the subterm $l\sigma$ at position p in s with $r\sigma$. We write $s \approx_E t$ if there is a finite sequence of equations and positions $(e_0, p_0), \ldots, (e_k, p_k)$ such that $s = v_0 \approx_{e_0}^{p_0} v_1 \approx_{e_1}^{p_1} \cdots \approx_{e_k}^{p_k} v_{k+1} = t$. It holds that $s \approx_E t$ if and only if $E \vdash s \approx t$.

For example, given the presentation $E := \{e_1 := f(x) \approx f(y), e_2 := g(f(a)) \approx g(b)\}$, we can prove $g(f(b)) \approx g(b)$ by the derivation $g(f(b)) \approx_{e_1}^1 g(f(a)) \approx_{e_2}^\epsilon g(b)$. Given the presentation $E := \{e_1 := f(x, y) \approx f(x, x), e_2 := f(x, y) \approx f(y, x)\}$, we can prove $f(f(a, b), f(b, a)) \approx f(f(b, a), b)$ by the derivation $f(f(a, b), f(b, a)) \approx_{e_2}^1 f(f(b, a), f(b, a)) \approx_{e_1}^\epsilon f(f(b, a), b)$.

Let $EQ(E)$ denote the set of equivalence classes induced by E. We use $[t]_E$ to denote the equivalence class of E containing the term t and $[t]$ when E is implicitly known. A *ground equivalence class* contains at least one ground term, and $EQ_G(E)$ is the set of ground equivalence classes of E.

We use $Vars(\cdot)$ to denote the set of variables occurring in any object, such as a term, equation, or set of equations.

We define the subterms of a term recursively by:

$$\mathit{Subterms}(g(s_1, \ldots, s_k)) := \{g(s_1, \ldots, s_k)\} \cup \bigcup_i \mathit{Subterms}(s_i)$$

We lift the definition to sets S of terms:

$$\mathit{Subterms}(S) := \bigcup_{s \in S} \mathit{Subterms}(s)$$

We say that a set S of terms is *subterm-closed* if $\mathit{Subterms}(S) = S$. We say that s *appears* in t (resp. $t \approx u$) if $s \in \mathit{Subterms}(t)$ (resp. $\mathit{Subterms}(t) \cup \mathit{Subterms}(u)$)

The size of a term is defined by $\| \cdot \|$ so that $\|f(s_1, \ldots, s_k)\| := 1 + \Sigma_i \|s_i\|$ and $\|a\| := 1$ for all symbols f of arity k, constants a, and terms $s_1, \ldots, s_k$. This can be extended to equations, sets of terms, and sets of equations in the natural way.

The *depth* of a term t is the length of the largest $p \in Pos(t)$. A term s (resp. equivalence class c) *occurs* in t at depth d if there is a $p \in Pos(t)$ of length d such that $t|_p = s$ (resp. $s \in C$ such that $t|_p = s$). A term t is *shallow* if no variable occurs at a depth greater than or equal to 2. An equation $s \approx t$ is shallow if s and t are shallow, and a set of equations is shallow if all of its equations are shallow. For example, $f(h(h(a)), x, b) \approx x$ and $g(x, y, b) \approx h(x)$ are shallow, while $f(h(x), a) \approx x$ and $h(h(x)) \approx h(x)$ are not. A theory T is shallow if there exists a shallow presentation E such that $Th(E) = T$.

An equation $s \approx t$ is collapsing if $t \in X$ and $t \in \mathit{Subterms}(s)$. A set of equations is collapsing if it contains at least one collapsing equation, and any theory is collapsing if each of it's presentations is collapsing. Any equation, set of equations, or theory is non-collapsing if it is not collapsing. Non-collapsing shallow theories can be presented by equations where variables only appear at depth 1.

For any equivalence class c, and any term s, we use $D1P_c(s)$ to denote the set of depth 1 positions of terms from c in s. In other words, $D1P_c(s)$ is the largest subset of $\mathbb{N}$ such that for each $i \in D1P_c(s)$, $s|_i \in c$. Define $D1P_x(s)$ analogously for any variable, x. Let $D1(E)$ be the set of E equivalence classes that appear at depth 1 in any presentation E.

3. Properties of Non-Collapsing Shallow Theories

This section investigates some properties of non-collapsing shallow theories that will be useful in the following sections. We introduce a representation for non-collapsing shallow theories known as *maximally-generalized signature equations*. We will show that this representation is canonical for ground-equivalent theories up to a renaming of variables. This representation can be used to determine a canonical presentation for non-collapsing shallow theories. Assuming a fixed signature, we will also show that the size of this representation is polynomial in the size of any ground-equivalent presentation.

Throughout this section, unless otherwise stated, we will assume that every theory is non-trivial. This means that there are at least two ground terms that are equivalent and at least two ground terms that are not equivalent.

The proofs for the statements made in this section are given in an appendix at the end of the paper.

3.1 Signature Equations

Given an equational theory E, a *signature*, *sig* (defined over E), consists of a function symbol $f \in \Sigma_k$ and a set of equivalence classes $C_1, \ldots, C_k \in EQ_G(E)$, represented $\langle f, C_1, \ldots, C_2 \rangle$. We call f the *head* and $C_1, \ldots, C_k$ the *body* of the signature.We write $\mathit{sig}[i]$ for each C_i.

An *instance* of *sig* is a term $f(s_1, \ldots, s_k)$, where for each i, $s_i \in C_i$. We write $\mathit{Inst}(\mathit{sig})$ to represent the set of instances of *sig*. We may also represent *sig* by $\langle f, s_1, \ldots, s_k \rangle$ when $f(s_1, \ldots, s_k) \in \mathit{Inst}(\mathit{sig})$, since each s_i acts as a representative element of C.

An *extended signature*, *sig*′ is defined analogously, but may contain variables in place of equivalence classes. Moreover, instances of extended signatures may replace like variables with like terms. When an instance contains all the same variables as its signature, it is a maximall-generalized instance For example $\langle f, a, x, x \rangle$ and $\langle f, [a]_E, x, x \rangle$ are both representations of the same extended signature, and $f(a, x, x)$ and $f(a, b, b)$ are instances of this signature, though $f(a, x, x)$ is the only maximally-generalized instance. Given two signatures, sig_1 and sig_2, we define the order $\preceq$ such that $\mathit{sig}_1 \preceq \mathit{sig}_2$ if $\mathit{Inst}(()\mathit{sig}_1) \subset \mathit{Inst}(()\mathit{sig}_2)$. We say $\mathit{sig}_1 \prec \mathit{sig}_2$ if $\mathit{sig}_1 \prec \mathit{sig}_2$ but $\mathit{sig}_2 \not\prec \mathit{sig}_1$. For example, $\langle f, a, b, b \rangle \prec \langle f, a, x, x \rangle$. Likewise, $\langle f, a, x, b \rangle \not\prec \langle g, a, x, b \rangle$ since they have different heads, and $\langle f, a, x, y \rangle \not\prec \langle f, x, x, y \rangle$. If $\mathit{sig}_1 \prec \mathit{sig}_2$, we say that sig_2 is more general than sig_1. For a set I of indices, we write $\mathit{sig}[c]_I$ to replace each element at each $i \in I$ with c.

A *signature equation* is an pair of extended signatures, written $\mathit{sig}_1 \approx \mathit{sig}_2$ for signatures sig_1 and sig_2. An *instance* of a signature equation $\mathit{sig}_1 \approx \mathit{sig}_2$ is an equation $s_1 \approx s_2$, where s_1 is an instance of sig_1 and is an instance s_2 of sig_2. The signature equation $sig_1 \approx sig_2$ (defined over E) *holds* on E if for every ground instance $s_1 \approx s_2$ of $sig_1 \approx sig_2$, $s_1 \approx_E s_2$. A *maximally generalized* signature equation (MGSE) is a signature equation $sig_1 \approx sig_2$ defined over E such that for all signatures sig_1' and sig_2' such that $sig_1 \preceq sig_1'$ and $sig_2 \preceq sig_2'$, there is an instance $s_1' \approx s_2'$ of the equation such that $s_1' \not\approx_E s_2'$. The set of maximally generalized signature equations of a theory E is written $MGSE(E)$. Every pair of equivalent ground terms is an instance of some MGSE. Note that for any presentation E', it holds that $E \equiv_G E'$ if and only if $MGSE(E) = MGSE(E')$ (up to a renaming of variables).

An *essential class* of a presentation E is an equivalence class that appears in the body of some signature in $MGSE(E)$. For example, if $E := \{f(g(a), x) \approx b\}$, then $[g(a)]$ and $[b]$ are essential classes, since $\langle f, g(a), x\rangle \approx \langle b\rangle$ are MGSEs. Then the only MGSE of $E := \{f(a) \approx b\}$ is $\langle f, a\rangle \approx \langle b\rangle$, so its only essential class is $[a]$, since b is in the head, not the body, of the equation. The set $E := \{a \approx b, f(x) \approx g(x)\}$ has MGSEs $\langle f, x\rangle \approx \langle g, x\rangle$ and $\langle a\rangle \approx \langle b\rangle$, and it has no essential classes. Lastly, the set $E := \{f(x, y) \approx f(y, x), f(a, x) \approx b\}$ has MGSEs $\langle f, x, y\rangle \approx \langle f, y, x\rangle$, $\langle f, a, y\rangle \approx \langle b\rangle$, and $\langle f, x, a\rangle \approx \langle b\rangle$.

Let $\mathcal{EC}(E)$ denote the set of of essential classes in E. We will use $\mathcal{EC}$, when E is clear from context.

The following lemma will be useful to identify essential terms.

Lemma 1 *Let $s \approx t \in Th_G(E)$ and let u be a term with $I := D1P_{[u]}(s)$ and $J := D1P_{[u]}(t)$. Then u is an essential term if and only if there is a v such that $s[v]_I \approx t[v]_J \notin Th_G(E)$.*

3.2 The Canonical Representation E_{rep}

Let $<$ be any total ordering on terms such that for any terms s and t, $\|s\| < \|t\|$ implies $s<t$. For a fixed $<$, we will see how to construct a presentation, E_{rep}, of E that is canonical up to a renaming of variables. Only variables and essential terms appear at depth 1 in any equation in E_{rep}.

For each $C \in \mathcal{EC}$, let the *representative term* of C, called rep_C, be minimal ground term in C with respect to $<$. Take any $sig_1 \approx sig_2 \in MGSE(E)$, where $sig_1 := \langle f, u_1, \ldots, u_k\rangle$ and $sig_2 := \langle g, u_{k+1}, \ldots, u_{k+r}\rangle$ and for each i, $u_i \in \mathcal{EC} \cup X$. We define the *representative equation* of $sig_1 \approx sig_2 \in MGSE(E)$ to be the equation $f(u_1', \ldots, u_k') \approx g(u_{k+1}', \ldots, u_{k+r}')$, where $u_i' := u_i$ if $u_i \in X$ and $u_i' := rep_C$ if $u_i \in C \in \mathcal{EC}$. Let E_{rep} be the set of representative equations of $MGSE(E)$. The proof that $E_{rep} \equiv_G E$ is given in the following proposition.

Proposition 2 *Given the presentation E_{rep} constructed from $MGSE(E)$ as above with ordering $<$, $E_{rep} \equiv_G E$.*

3.3 Bounding the Size of E_{rep}

We have seen how to construct a canonical presentation E_{rep} from the essential classes $\mathcal{EC}$ of a presentation E. But what if $\|\mathcal{EC}\|$ is very large? This section gives a polynomial bound for $\|\mathcal{EC}\|$ in terms of the size of any ground-equivalent non-collapsing shallow presentation E. We use this to give polynomial bounds for $|E_{rep}|$ and $\|E_{rep}\|$ in terms of $|E|$ and $\|E\|$, respectively.

Lemma 3 *Let E be any non-collapsing shallow presentation and let s, t be terms in $T(\Sigma, X)$ such that $s \approx_E t$. Assume there is a $u \in T(\Sigma)$ such for every $s' \approx t' \in E$, $[u] \notin D1(s') \cup D1(t')$. Let $I := D1P_{[u]}(s)$ and $J := D1P_{[u]}(t)$. Then $s[x]_I \approx_E t[x]_J$ for some $x \in X \backslash (Vars(s) \cup Vars(t))$.*

Lemma 4 *For any non-collapsing shallow presentation E such that $|EQ_G(E)| \geq |D1(E)| + 2d$, $\mathcal{EC}(E) \subseteq D1(E)$.*

The condition that $|EQ_G(E)| \geq |D1(E)| + 2d$ is in fact necessary for the above lemma, though the inequality isn't necessarily tight. Specifically, if there is a non-collapsing shallow E such that $|D1(E)| + 2d \geq |EQ_G(E)|$, then it is not necessary that $\mathcal{EC}(E) \subseteq D1(E)$. For example, let $\Sigma := \{f : 2, a : 0, b : 0\}$ and let $E := \{f(x, a) \approx a, f(x, x) \approx a, f(x, f(a, b)) \approx a, f(f(a, b), x) \approx a\}$, so $D1(E) = \{[a], [f(a, b)]\}$. A quick check will show that $EQ_G(E) = \{[a], [b], [f(a, b)]\}$. But this leaves $\langle f, b, x \rangle \approx \langle a \rangle$ as an MGSE, though $[b] \notin D1(E)$.

This lemma yields a simple but important corollary.

Corollary 5 *For any non-collapsing shallow presentation E, $|\mathcal{EC}(E)| \leq 2d + |D1(E)|$.*

We can also use the above lemma to prove a bound on how much larger E_{rep} is than any other presentation of the same theory.

Theorem 6 *Let E be any non-collapsing shallow presentation over the alphabet Σ, and let E_{rep} be the representative presentation formed from the essential classes of E. Then $|E_{rep}| \leq |\Sigma|^2(2d|E| + 4d)^{2d}$.*

Note that it is not possible to bound $|E_{rep}|$ in terms of $\|E\|$. For example, for any positive integer k, let $E := \{f^k(a) \approx a\}$, where $f^k(a)$ is the result of applying f to a k times. Then $|E| = |E_{rep}| = 1$, but $\|E\| \geq k$ for any k.

We can also prove a polynomial bound on $\|E_{rep}\|$ depending on $\|E\|$ and $|E|$ for any $E \equiv_G E_{rep}$. In the following, given a complexity class c of E let $minsize_E(c)$ represent the value $\|t\|$ that is minimized for all $t \in c$.

Theorem 7 *Let E be any non-collapsing shallow presentation over the alphabet Σ, and let E_{rep} be the representative presentation formed from the essential classes of E. Then $\|E_{rep}\| \leq |E_{rep}|(\|E\|(2d)^{2d} + 2)$.*

4. Learning in the Limit

This section investigates the possibility of learning non-collapsing shallow theories in the limit from ground examples.

In the *learning in the limit* model, first presented in Gold (1967), the learner is trying to learn a target concept L from a concept class $C \subset 2^X$ for some space X of elements. The learner is given a sequence of examples $e_1, e_2, \ldots$. When learning from *positive examples*, these examples are drawn from the target set (i.e., *language*) L, with the guarantee that every $x \in L$ will eventually be seen in some example. When learning from *negative examples*, each e_i is of the form (x_i, b_i) where the b_i indicates whether x_i is in L, and every $x \in X$ will eventually be seen in some example.

After each new example e_t, the learner is asked to give a hypothesis L_t. It is said to learn in the limit if the learner eventually converges to the target concept, L.

In this paper, X is the set $T(\Sigma)$ of all ground equations, and the concepts L are theories over ground terms.

4.1 Learning from Positive Examples

Learning ground equational presentations in the limit from positive examples is easy. After seeing some set of positive ground examples E, simply use E as the hypothesis presentation.

However, when non-ground equations are allowed, even very simple classes of equational theories are no longer learnable from positive examples. A simple corollary of Gold's theorem shows that the set of equational theories that can be presented by ground equations and (optionally) the equation $f(x, y) \approx f(y, x)$ is not learnable from positive examples. Since these are all non-collapsing shallow theories, this implies that the entire class of non-collapsing shallow theories is not learnable from positive examples.

We restate Gold's theorem below Gold (1967):

Theorem 8 *Let $C' := \{L_\infty, L_0, L_1, \dots\}$ be a set of formal languages such that for all i, $L_i \subset L_{i+1}$ and $L_\infty := \bigcup_i L_i$. Then no class C containing C' is learnable from positive examples.*

We will now construct a set C' of non-collapsing shallow theories that fits the above conditions. Let $\Sigma := \{a : 0, b : 0, f : 2\}$. For each $i \in \mathcal{N}$, let $E_i := \{f(s,t) \approx f(t,s) \mid \mathit{depth}(s), \mathit{depth}(t) \leq i\}$ and let $L_i := \mathit{Th}_G(E_i)$. Let $L_\infty := \mathit{Th}_G(\{f(x, y) \approx f(y, x)\})$. We can see that $C' := \{L_\infty, L_0, L_1, \dots\}$ satisfies the conditions of the above theorem. Therefore, no class whose theories can be presented using only ground equations and the equation $f(x, y) \approx f(y, x)$ can be learned from positive examples.

4.2 Learning from Positive and Negative Examples

This subsection presents an algorithm for learning non-collapsing shallow theories in the limit from examples of positive and negative ground-equations. The algorithm takes a set S^+ of positive examples and S^- of negative examples and returns a hypothesis in time polynomial in $\|S^+\| + \|S^-\|$. The hypothesis is consistent with all examples. As more examples are added, the algorithm will eventually converge on the presentation E_{rep} defined with respect to an ordering $<$.

It is fairly easy to create an algorithm that creates a consistent hypothesis in polynomial time and learns the theory in the limit. Say enumerate all non-collapsing shallow presentations, $E_1, E_2, \dots$, in order of increasing presentation size, $\|E_i\|$. Given some examples (S^+, S^-) such that $|S^+| + |S^-| = n$, the algorithm can check if there is an $i < n$ such that E_i is consistent with (S^+, S^-). If so, the algorithm returns the first such E_i. If not, the algorithm returns S^+. To check if E_i is consistent with (S^+, S^-), the learner checks that $s \approx_{E_i} t$ for each $s \approx t \in S^+$ and it checks that $s \not\approx_{E_i} t$ for each $s \approx t \in S^-$. This takes polynomial time, since the size of each E_i for $i < n$ is less than polynomial in n and checking provability of any $s \approx t$ in a non-collapsing theory takes polynomial time. Therefore, finding a consistent hypothesis takes polynomial time in the input size.

However, this algorithm is far from practical. There are at $\Omega(2^n)$ presentations of size less than n. So, this learning process will require exponentially many examples to learn most theories. The rest of this sections presents an algorithm that requires polynomially many good examples before converging on a solution.

Let E be a non-collapsing shallow presentation of the target theory defined over the alphabet Σ.Let $<$ be an ordering on terms such that for all terms s and t, $\|s\| < \|t\|$ implies $s<t$. Given the examples (S^+, S^-), the hypothesis $\hat{E}$ is constructed as follows:

- A set A of essential terms is found. (This is described in more detail in the next subsection)
- The algorithm creates a set B of non-collapsing shallow equations with only terms from A and variables at depth 1.
- For each equation $e \in B$, the algorithm checks whether $\hat{E} \cup \{e\} \cup S^+ \vdash u \approx v$ for any $u \approx v \in S^-$. If not, e is added to $\hat{E}$.
- For all $e, e' \in \hat{E}$ such that $e \prec e'$, e is removed from $\hat{E}$
- The algorithm returns the hypothesis $\hat{E} := \hat{E} \cup \{s \approx t \in S^+ \mid s \not\approx_{\hat{E}} t\}$

This algorithm takes inspiration from the RPNI algorithm for learning regular languages from positive and negative examples (Oncina and Garcia, 1992). In particular, both algorithms try to generalize as much as possible without contradicting the known negative examples.

4.2.1 Identifying Essential Terms

We say that the terms s and t are *provably distinct* if there is a $u \approx v \in S^-$ such that $s \approx_{S^+} u$ and $t \approx_{S^+} v$.

We say a ground signature $f(s_1, \ldots, s_k) \approx g(s_{k+1}, \ldots, s_{k+r})$ is *classified* if for every s_i and s_j, either $s_i \approx_{S^+} s_j$ or s_i is provably distinct from s_j.

Let $s \approx t$ be a classified equation such that $s \approx_{S^+} t$, and let u be a term with $I := D1P_{[u]}(s)$ and $J := D1P_{[u]}(t)$. If there is a v such that $s[v]_I \approx t[v]_J$ is classified and $s[v]_I$ is provably distinct from $t[v]_J$, then u is an essential term by lemma 1.

By finding pairs of equations $s \approx t$ and $s[v]_I \approx t[v]_J$ as above, the algorithm assembles a set A' of essential terms. Since A' might contain multiple terms from the same equivalence class, a new set A is constructed of provably distinct terms is constructed as follows: Set $A := \emptyset$. In increasing order of $<$, take each $s \in A'$. If s is provably distinct from each element of A, then add s to A'.

4.2.2 Characteristic Samples

A *characteristic sample* is a pair of sets (S'^+, S'^-) such that whenever $S'^+ \subseteq S^+$ and $S'^- \subseteq S^-$, the algorithm will yield a correct hypothesis. We will show that the algorithm admits a characteristic sample for every non-collapsing shallow theory. This implies that the algorithm will learn non-collapsing shallow theories in the limit.

Note that the algorithm will always yield the correct solution if there is only one equivalence class.Therefore, we can assume that there are at least two equivalence classes in the target theory.

We will now construct the characteristic sample (S'^{+}, S'^{-}) for the non-collapsing shallow theory $Th_G(E)$ with order $<$. As we describe the construction, we will show that any sample containing the characteristic sample will cause the algorithm to yield the hypothesis E_{rep}.

For each essential class C, recall that rep_C is the minimal element of C with respect to $<$.

For each rep_C, find an MGSE, $sig_1 \approx sig_2$, C in it's body and let I and J be the positions of C in sig_1 and sig_2, respectively. For a $v \notin C$, find a pair of equations $s \approx t$ and $s[v]_I \approx t[v]_J$ such that $s \approx t \in Inst(sig_1 \approx sig_2)$ and $s[v]_I \not\approx_E t[v]_J$.Add $s \approx t$ to S^+ and add $s[v]_I \approx t[v]_J$ to S^-. This guarantees that the algorithm will add rep_C to A'.

Add the set $\{rep_C \approx rep_{C'} \mid C, C' \in \mathcal{EC}(E)\}$ to S'^{-}. This guarantees that the set A will contain all rep_C terms for each $C \in \mathcal{EC}(E)$. No other essential terms will added to A, since they must be in some $C \in \mathcal{EC}(E)$ and thus cannot be provably distinct from rep_C. Therefore, the algorithm will find the set $A := \{rep_C \mid C \in \mathcal{EC}(E)\}$.

For each signature equation $sig_1 \approx sig_2$ that has only variables and essential classes in its body and does not hold for E, find an equation $s \approx t \in Inst(sig_1 \approx sig_2)$ such that $s \not\approx_E t$ and add $s \approx t$ to S^-. Thus, the representative equation for $sig_1 \approx sig_2$ will not be added to $\hat{E}$.

Every representative equation in E_{rep} will be added to $\hat{E}$, since there can be no equation in S^- to contradict it. By the definition of E_{rep}, all other equations that hold for E are instances of E_{rep}. Therefore, the algorithm will return E_{rep} as the final hypothesis.

4.2.3 Examples

Example 1 *Let $\Sigma := \{f : 1, a : 0\}$, $S^+ := \emptyset$, and $S^- := \{f(a) \approx a\}$. There are no classified equations, so no essential terms are identified. The algorithm tries to add $f(x) \approx a$, but fails since it can be used to prove $f(a) \approx a$. It then tries $f(x) \approx f(y)$ and succeeds. The hypothesis presentation is therefore $\hat{E} := \{f(x) \approx f(y)\}$.*

Example 2 *Let $\Sigma := \{f : 1, a : 0, b : 0\}$, $S^+ := \{f(a) \approx f(b)\}$, and $S^- := \{a \approx b, f(f(a)) \approx f(a), f(f(a)) \approx f(b), f(a) \approx b, f(a) \approx a\}$. The equations, $f(a) \approx f(b)$, $f(f(a)) \approx f(b)$, and $f(a) \approx f(f(a))$ are classified. They are used to show that both a and b are essential terms, since $f(a) \not\approx_E a$ and $f(a) \not\approx_E b$. These terms a and b are provably distinct. Thus, the algorithm tries to add the following equations: $f(a) \approx f(b)$ (yes), $f(a) \approx a$ (no), $f(a) \approx b$ (no), $f(a) \approx f(x)$ (no), $f(b) \approx a$ (no), $f(b) \approx b$ (no), $f(b) \approx f(x)$ (no), $f(x) \approx f(y)$ (no). The hypothesis presentation is therefore $\hat{E} := \{f(a) \approx f(b)\}$.*

5. Learning From Queries and Counter-Examples

This section presents the main result from our paper: an efficient algorithm to learn non-collapsing shallow equational theories from ground queries and counter-examples to an oracle. These queries take the following two forms:

- *Membership Query:* The algorithm presents a ground equation $s \approx t$ to the oracle and the oracle states whether or not $s \approx t \in Th_G(E)$,
- *Equivalence Query:* The algorithm presents a hypothesis presentation E' to the oracle and the oracle states whether or not $E' \equiv_G E$. If not, the oracle also returns a counter-example $s \approx t$ from the set $(Th_G(E) \backslash Th_G(E')) \cup (Th_G(E') \backslash Th_G(E))$.

This algorithm will specifically learn the canonical presentation E_{rep} of a theory E over a fixed alphabet Σ.

Throughout this section, unless otherwise stated, we will assume that every theory is non-trivial. The algorithm can query the presentations $\emptyset$ and $\{x \approx y\}$ to the oracle in order to rule out the trivial cases.

We will first show how to learn using a set of "auxiliary symbols", $C_\alpha := \{\alpha_1, \alpha_2, \ldots, \alpha_{2d}\}$, where d is the maximum arity of any symbol in Σ. The symbols of C_α are all constants. For each $\alpha_i \in C_\alpha$, we have the guarantee that $[\alpha_i]$ is not essential.Later, we will see how to learn non-collapsing shallow theories without assuming such a set C_α exists.

The algorithm runs in iterations, creating a new hypothesis at each iteration. The structure of each iteration is as follows:

1. Use essential classes and C_α to find all MGSEs of E
2. Find set $rep_{\mathcal{EC}}$ of minimal terms from each essential class and find a hypothesis $\hat{E}$ for E_{rep}
3. Pose $\hat{E}$ as an equivalence query to the oracle.
    - If the oracle returns *true* return $\hat{E}$
    - Otherwise, we receive a counter-example $s \approx t$ such that $s \approx_E t$, but $s \not\approx_{\hat{E}} t$.
4. Use the counter-example to find an equation $s' \approx t'$ that is an instance of an unknown MGSE.
5. Use $s' \approx t'$ to find an unknown essential class
6. Start a new iteration from step 1

Those familiar with Angluin's algorithm (Angluin, 1987) will notice similarities with the above algorithm, where states are analogous to essential classes and equations are analogous to transitions.

5.1 Finding the MGSEs

By the assumptions on C_α, we can use membership queries to infer the location of variables in each MGSE. For example, if the query $f(\alpha_1) \approx g(\alpha_1, a)$ holds, then we know that the signature equation $\langle f, x\rangle \approx \langle g, x, a\rangle$ holds, since otherwise $[\alpha_1]$ would be essential. Given a set C of representative essential terms of E, we can query all (polynomially many) pairs of terms from the set $\{f(s_1, \ldots, s_k) \mid f \in \Sigma_k, \forall i, s_i \in C \cup C_\alpha\}$. We can find signature equations (and thus MGSEs) from the queries that return *true*.

5.2 Finding $rep_{\mathcal{EC}}$

We will show how to use the oracle to construct $rep_{\mathcal{EC}}$. For simplicity, we don't assume a fixed ordering $<$ on terms, and just find representative terms of minimal size. The ordering $<$ can be determined by the choices of representative terms.

Let S be a subterm-closed set containing the smallest known term from each essential class. We will proceed iteratively, finding a hypothesized representative $rep^j_{[s]}$ at each step j for each $s \in S$. At each iteration, we will perform the following actions on each $s \in S$ in order of increasing size. Start with $j = 1$. If s is a constant and $rep^j_{[s]}$ is not yet defined, set $rep^j_{[s]} := s$. Now assume $rep^j_{[u]}$ is defined for each $u \in S$ such that $\|u\| < \|s\|$, but $rep^j_{[s]}$ is not yet defined. Let $s := f(s_1, \ldots, s_k)$. Consider each MGSE e of the form $sig_1 \approx sig_2$

such that s is an instance of sig_1. We will construct a term t_e such that $s \approx t_e$ is an instance of $sig_1 \approx sig_2$. Let $sig_1 := \langle f, c_1, \ldots, c_k \rangle$ and $sig_2 := \langle g, d_1, \ldots, d_r \rangle$. Consider each $i \in \{1, \ldots, r\}$. If d_i is a variable and equal to some c_j in sig_1, then set $t_i := rep^j_{[s_j]}$. If d_i is a variable that doesn't appear in sig_1 then set $t_i := a$ for some constant a (choose the same constant each time). Otherwise, d_i is an essential class. If $rep^j_{d_i}$ is not yet defined, then stop constructing t_e. Otherwise, set $t_i := rep^j_{d_i}$. Let $t_e := g(t_1, \ldots, t_r)$. Set $rep^j_{[s]}$ equal to the lowest-depth t_e of all such MGSEs.

Continue this process until $rep^j_{[s]} = rep^{j+1}_{[s]}$ for all $s \in S$. By induction, it is easy to check that after each iteration j, all classes with representative elements of size less than or equal to j are assigned a min-size representative. Therefore, this process completes after $max_{s \in S}\{\|s\|\}$ iterations.

5.3 Assembling $\hat{E}$ and Handling Counter-Examples

So, assuming that we have identified all essential classes, we can efficiently find a presentation $\hat{E}$ equal to E_{rep}. If we are missing an essential class, however, then $\hat{E}$ will not correctly identify E and the oracle will return some counter-example, $s \approx t$. This counter-example will be positive, meaning that $s \approx_E t$, but $s \not\approx_{\hat{E}} t$. This is possible in the following cases, which we will handle differently:

1. $s \approx t$ is an instance of an MGSE $sig_1 \approx sig_2$, but the representative equation for this $MGSE$ cannot be applied to $s \approx t$. Either, there is an essential term u at depth 1 such that $u \not\approx_{\hat{E}} rep_{[u]}$ or there are parallel terms u and v at depth 1 such that $u \approx_E v$ but $u \not\approx_{\hat{E}} v$. In either case, we can recurse, treating $u \approx rep_{[u]}$ or $u \approx v$ as our new counter-example.
2. $s \approx t$ is not an instance of any $MGSE$, and so $s \approx t$ is an instance of a missing MGSE.

Each time a counter-example is processed in case 1, we obtain a smaller counter-example. By the construction of $\hat{E}$, if a counter-example has constants on both sides, it will already be in $\hat{E}$. Therefore, this process will always find an instance $s' \approx t'$ of a missing MGSE.

This can only occur if there is an essential class that is missing from our known set of essential classes. To find the missing essential class, take any u at depth 1 in s' or t' that is not in any known essential class (use oracle queries to confirm this). Let $I := D1P_{[u]}(s')$ and $J := D1P_{[u]}(t')$. Query $s'[\alpha_1]_I \approx t'[\alpha_1]_J$ to the oracle. By lemma 1, u is essential if and only if this query returns false. Otherwise, repeat the same process on another non-essential term.

Once the essential class is found, the algorithm begins the next iteration.

Example 3 *Let $\Sigma := \{g : 1, a : 0, b : 0, c : 0\}$ and assume that the target theory can be presented by $E := \{g(a) \approx g(b), g(b) \approx c\}$. The algorithm queries $\hat{E} := \{x \approx y\}$ to the oracle and the oracle returns false (the counter-example is ignored). The current hypothesis is that $\mathcal{EC} = \emptyset$. The algorithm creates a hypothesis for $MGSE(E)$ by querying $g(\alpha_1) \approx g(\alpha_2)$ (false), $g(\alpha_1) \approx a$ (false), $g(\alpha_1) \approx b$ (false), and $g(\alpha_1) \approx c$ (false). So the hypothesis for $MGSE(E)$ is $\emptyset$ and the hypothesis presentation $\hat{E} := \emptyset$ is passed to the oracle. The oracle returns $g(g(a)) \approx g(g(b))$ as a positive counter-example, meaning $g(g(a)) \approx_E g(g(b))$ but $g(g(a)) \not\approx_{\hat{E}} g(g(b))$. The algorithm queries all depth-1 terms in the counter-example (i.e., $g(a) \approx g(b)$ (true)) to determine that the signature of the equation with respect to E is*

$\langle g, [g(a)]\rangle \approx \langle g, [g(a)]\rangle$. This signature holds in $\hat{E}$, so $\hat{E}$ must fail to prove $g(a) \approx g(b)$. The equation $g(a) \approx g(b)$ is treated as the new positive counter-example and the query $a \approx b$ (false) shows that its signature is $\langle g, [a]\rangle = \langle g, [b]\rangle$. This signature equation must be an instance an unknown MGSE, so there must be an essential class in its body. To determine which classes are essential, the algorithm queries $g(\alpha_1) \approx g(b)$ (false), $g(\alpha_1) \approx g(\alpha_2)$ (false), and $g(a) \approx g(\alpha_1)$ (false). This implies that $\langle g, a\rangle \approx \langle g, b\rangle$ is an MGSE, and that $[a]$ and $[b]$ are essential classes. The hypothesis set of MGSEs is formed by making the following queries: $g(a) \approx g(b)$ (true), $g(a) \approx g(\alpha_1)$ (false), $g(\alpha_1) \approx g(b)$ (false), $g(\alpha_1) \approx g(\alpha_2)$ (false), $g(a) \approx a$ (false), $g(a) \approx b$ (false), $g(a) \approx c$ (true), $g(b) \approx a$ (false), $g(b) \approx b$ (false), and $g(b) \approx c$ (true). This yields the MGSEs $\langle g, a\rangle \approx c$, $\langle g, b\rangle \approx c$, and $\langle g, a\rangle \approx \langle g, b\rangle$. The algorithm chooses a and b as the representative for their respective equivalence classes, yielding $\hat{E} := \{g(a) \approx g(b), g(a) \approx c, g(b) \approx c\}$. This presentation is given to the oracle, which returns true, and the process completes.

Example 4 *Let $\Sigma := \{f : 1, a : 0\}$ and assume that the target theory can be presented by $E := \{f(x) \approx f(y)\}$. The algorithm queries $\hat{E} := \{x \approx y\}$ to the oracle and the oracle returns false (the counter-example is ignored). The current hypothesis is that $\mathcal{EC} = \emptyset$. The algorithm creates a hypothesis for $MGSE(E)$ by querying $f(\alpha_1) \approx f(\alpha_2)$ (true) and $f(\alpha_1) \approx a$ (false). This yields the MGSE $\langle f, x\rangle \approx \langle f, y\rangle$ and the hypothesis presentation $\hat{E} := \{f(x) \approx f(y)\}$. This is given to the oracle, the oracle returns true, and the process completes.*

6. Learning From Queries Without Auxiliary Symbols

The algorithm from the previous section requires the existence of $2d$ distinct "auxiliary" symbols. However, it is likely that an oracle will only be able to answer queries over a given alphabet. Therefore, it is worthwhile to try to run the above algorithm without the use of these symbols. In this section, we accomplish this by maintaining a set T_α containing terms $t_\alpha^1, \ldots, t_\alpha^{2d}$ from distinct equivalents classes which are believed to be non-essential in E. We will first show how to update the algorithm, then show how to find new t_α terms.

6.0.1 Updating the algorithm

The new algorithm works the same as the old one, using the t_α terms in place of the C_α symbols to infer the location of variables and essential terms. Each time an MGSE is found, the *representative query* for that MGSE is stored. The representative query for an MGSE e is a pair $(s \approx t, \sigma)$, where σ maps variables to terms in $T\alpha$ and $s \approx t$ is the representative equation of e. Note that there is only one σ such that $s\sigma \approx t\sigma$ is queried to find e, so the representative query is unique.

Since the t_α terms may actually be essential, the learned MGSEs may be more general than the actual MGSEs. For example, let $E := \{f(a, x) = g(c)\}$, $t_\alpha^1 := a$, $t_\alpha^2 := b$, and $t_\alpha^3 := d$. Using membership queries, the algorithm can determine that $f(t_\alpha^1, t_\alpha^2) \not\approx_E g(t_\alpha^3)$ (i.e., $f(a, b) \not\approx_E g(d)$), but $f(t_\alpha^1, t_\alpha^2) \approx_E g(c)$. This leads the algorithm to conclude that $\langle f, x, y\rangle \approx \langle g, [c]\rangle$ is an MGSE of E, since t_α^1 is believed to mark the place of a variable. However, t_α^1 is actually an essential term, and the learned MGSE is a generalization of the true MGSE, $\langle f, [a], y\rangle \approx \langle g, [c]\rangle$.

Therefore, when a hypothesis $\hat{E}$ is given to an oracle, the oracle may return a *negative* counter-example $s \approx t$, meaning $s \approx_{\hat{E}} t$ but $s \not\approx_E t$.

Assume such a negative counter-example $s \approx t$ is given. Using the algorithm described below to find a new term t'_α. Using Proposition 2, we can construct a proof that $s \approx_{\hat{E}} t$ of the form $s = u_0 \approx_{e_0}^{p_0} u_1 \cdots \approx_{e_{r-1}}^{p_{r-1}} u_r = t$. For each $i \in \{0, \dots, r-1\}$, query $u_i \approx u_{i+1}$. Let i' be the smallest index such that $u_{i'} \not\approx_E u_{i'+1}$. Such an i' must exist since otherwise, $s \approx_E t$ would hold. By the construction of $\hat{E}$ in Proposition 2, the equation $e_{i'}$ used at step i' in the derivation must be the representative of some MGSE, $sig_1 \approx sig_2$. Let $(u \approx v, \sigma)$ be the representative query of $sig_1 \approx sig_2$. Since $u_i \not\approx_E u_{i+1}$, but $u_i \approx_{u \approx v} u_{i+1}$, we know $sig_1 \approx sig_2$ must be an over-generalization. Therefore, there is a variable in $sig_1 \approx sig_2$ that should be replaced at some positions with a ground term. For each variable $y \in Vars(sig_1) \cup Vars(sig_2)$, let $x\sigma_y := x\sigma$ for all $x \neq y$ and $y\sigma_y := t'_\alpha$. Query $s\sigma_y \approx t\sigma_y$. If the query returns *false*, then $y\sigma$ must be an essential term. So set $T_\alpha := (T_\alpha \backslash \{y\sigma\}) \cup \{t'_\alpha\}$ and set $\mathcal{EC} := \mathcal{EC} \cup \{[y\sigma]\}$. If no such variable is found, then t'_α must be essential. So set $\mathcal{EC} := \mathcal{EC} \cup \{t'_\alpha\}$, find a new t_α term, and repeat the above process.

6.0.2 Finding new t_α terms

Throughout the run of this algorithm, we maintain a set $C := \{C_1, \dots, C_r\}$. Each C_i contains a set of terms that are equivalent in E, and $\bigcup_i C_i$ is subterm-closed. Each new t_α term is chosen as follows

1. If there is a constant c not in C, then for each i choose a $c_i \in C_i$. Query $c_i \approx c$.
    - If there is an i such that $c_i \approx_E c$, then add c to C_i and restart.
    - Otherwise, set $C := C \cup \{\{c\}\}$ and return $t_\alpha := c$
2. For each symbol f of arity k and each $(i_1, \dots, i_{k+1}) \in \{1, \dots, r\}^{k+1}$, query $f(c_{i_1}, \dots, c_{i_k}) \approx c_{i_{k+1}}$, where each c_i is in C_i
    - If $f(c_{i_1}, \dots, c_{i_k}) \approx_E c_{i_{k+1}}$, then add $f(c_{i_1}, \dots, c_{i_k})$ to $C_{i_{k+1}}$.
    - If there is no i_{k+1} such that $f(c_{i_1}, \dots, c_{i_k}) \approx_E c_{i_{k+1}}$, then set $C := C \cup \{f(c_{i_1}, \dots, c_{i_k})\}$ and return $t_\alpha := f(c_{i_1}, \dots, c_{i_k})$

As described above, the learning algorithm first finds a set of $2d$ different t_α terms, then adds at most one new t_α terms for each class in C. Therefore, $|C|$ never exceeds $2d + |\mathcal{EC}|$.

If the algorithm does not return any new t_α, then the set of classes represented in C_i is closed under each f application. Thus, every equivalence class is represented in C, and a presentation of E of size $Poly(2d + |\mathcal{EC}|)$ can be easily inferred.

Each call to this algorithm takes polynomial time assuming fixed Σ.

Example 5 *Let $\Sigma := \{g : 1, a : 0, b : 0, c : 0\}$ and assume the target theory can be presented by $E := \{g(a) \approx c\}$. The algorithm queries $\hat{E} := \{x \approx y\}$ to the oracle and the oracle returns false. It then sets $t_\alpha^1 := a$ and $t_\alpha^2 := b$. To find the MGSEs it queries $g(t_\alpha^1) \approx g(t_\alpha^2)$ (false), $g(t_\alpha^1) \approx a$ (false), $g(t_\alpha^1) \approx b$ (false), and $g(t_\alpha^1) \approx c$ (true). Since the algorithm believes that t_α^1 (i.e., a) is not essential, the fact that $g(t_\alpha^1) \approx_E c$ leads it to conclude that $\langle g, x \rangle \approx \langle c \rangle$ is an MGSE. It passes the hypothesis $\hat{E} := \{g(x) \approx c\}$ to the oracle. The oracle returns false and gives the negative counter-example $g(g(b)) \approx c$, meaning $g(g(b)) \not\approx_E c$ but $g(g(b)) \approx_{\hat{E}} c$. This equation is provable in $\hat{E}$ by the derivation $g(g(b)) \approx_{g(x) \approx c} c$, implying that the equation $g(x) \approx c$ should not be in $\hat{E}$. Therefore, $e := \langle g, x \rangle \approx \langle c \rangle$ is an over-generalization of an actual MGSE of E. Since e was added because of the representative* *query $e' := g(t_\alpha^1) \approx c$, one of the t_α terms used in e' must be an essential term. The algorithm finds a new term $t'_\alpha := c$ and queries $g(t'_\alpha) \approx c$ (false). Since $g(t'_\alpha) \not\approx_E c$ and $g(t_\alpha^1) \not\approx_E c$, t_α^1 must be an essential term. So the algorithm adds $[a]$ to $\mathcal{EC}$ and sets $t_\alpha^1 := c$. To find the MGSEs, the algorithm queries $g(t_\alpha^1) \approx g(t_\alpha^2)$ (false), $g(t_\alpha^1) \approx a$ (false), $g(t_\alpha^1) \approx b$ (false), $g(t_\alpha^1) \approx c$ (false), $g(a) \approx a$ (false), $g(a) \approx b$ (false), and $g(a) \approx c$ (true). This yields the MGSE $\langle g, [a]\rangle \approx \langle c\rangle$. The algorithm sets a to be the representative element of its equivalence class and queries $\hat{E} := \{g(a) \approx c\}$ to the oracle. The oracle returns true and the process completes.*

7. Conclusion & Future Work

In this paper, we have discussed the problem of learning presentations of equational theories through ground examples and queries. We presented polynomial-time algorithms for learning presentations of non-collapsing shallow equational theories. It remains open whether these techniques can be extended to learning shallow theories that include collapsing equations. The definition of MGSEs can be extended to include signatures of a single variable (e.g., $\langle f, x\rangle \approx \langle x\rangle$). This allows for many of the results of this paper to be extended to all shallow theories. However, it is not clear whether a bound can be established on the number of essential classes that must appear at depth 1 of any presentation of a shallow theory. This may be investigated in future work.

References

Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

Hiroki Arimura, Hiroshi Sakamoto, and Setsuo Arikawa. Learning term rewriting systems from entailment. In *ILP Work-in-progress reports*, 2000.

Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge university press, 1999.

Hubert Comon, Marianne Haberstrau, and J-P Jouannaud. Decidable problems in shallow equational theories. In *Logic in Computer Science, 1992. LICS'92., Proceedings of the Seventh Annual IEEE Symposium on*, pages 255–265. IEEE, 1992.

Hubert Comon, Marianne Haberstrau, and Jean-Pierre Jouannaud. Syntacticness, cycle-syntacticness, and shallow theories. *Information and Computation*, 111(1):154–191, 1994.

Nachum Dershowitz. Synthesis by completion. *Urbana*, 51:61801, 1985.

Nachum Dershowitz and Uday S Reddy. Deductive and inductive synthesis of equational programs. *Journal of Symbolic Computation*, 15(5-6):467–494, 1993.

E Mark Gold. Language identification in the limit. *Information and control*, 10(5):447–474, 1967.

Donald E Knuth and Peter B Bendix. Simple word problems in universal algebras. In *Automation of Reasoning*, pages 342–376. Springer, 1983.

Robert Nieuwenhuis. Basic paramodulation and decidable theories. In *Logic in Computer Science, 1996. LICS'96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 473–482. IEEE, 1996.

Michael J O'donnell. Equational logic as a programming language. 1985.

José Oncina and Pedro Garcia. Inferring regular languages in polynomial update time. 1992.

MRK Rao. Learnability of term rewrite systems from positive examples. In *Proceedings of the 12th Computing: The Australasian Theroy Symposium-Volume 51*, pages 133–137. Australian Computer Society, Inc., 2006.

Yasubumi Sakakibara. Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, 76(2-3):223–242, 1990.

Maarten H Van Emden and Keitaro Yukawa. Logic programming with equations. *The Journal of Logic Programming*, 4(4):265–288, 1987.

Appendix

Lemma 1 Let $s \approx t \in Th_G(E)$ and let u be a term with $I := D1P_{[u]}(s)$ and $J := D1P_{[u]}(t)$. Then u is an essential term if and only if there is a v such that $s[v]_I \approx t[v]_J \notin Th_G(E)$.
Proof Assume there is a v such that $s[v]_I \approx t[v]_J \notin Th_G(E)$. Let $sig_1 \approx sig_2$ be a signature equation with $s \approx t$ as an instance. Assume for contradiction that $[u]$ is not in the body of $sig_1 \approx sig_2$. So there must only be variables at position I in sig_1 and position J in sig_2. Let x be any such variable. Assume for contradiction w.l.o.g. that x also appears in sig_1 at some position i not in I. Since $i \notin I$, $u \not\approx_E s|_i$, so $s \approx t \notin Inst(sig_1 \approx sig_2)$. Therefore $D1P_x(s) \subseteq I$ and $D1P_x(t) \subseteq J$, by contradiction. Therefore, $s[v]_I \approx t[v]_J \in Inst(sig_1 \approx sig_2)$. By contradiction, u must be in the body of $sig_1 \approx sig_2$ and so u must be an essential term.

Now assume u is an essential term, and let $sig_1 \approx sig_2$ be an MGSE with $s \approx t$ as an instance and let x be a fresh variable. Assume for contradiction that for all v, $s[v]_I \approx_E t[v]_J$. Then every ground instance of $s[x]_I \approx t[x]_J$ holds for E. Since the choice of $s \approx t$ was arbitrary, we can replace each $[u]$ in $sig_1 \approx sig_2$ with an x to get a new signature that holds for E. Thus $sig_1 \approx sig_2$ is not an MGSE. By contradiction, there must be a v such that $s[v]_I \approx t[v]_J \notin E$. ■

Proposition 2 Given the presentation E_{rep} constructed from $MGSE(E)$ as above with ordering $<$, $E_{rep} \equiv_G E$.
Proof It is easy to check that all ground terms that are equivalent in E_{rep} are equivalent in E, by the definition of $MGSE(E)$. To show the other direction, we will show by induction on k that for all ground terms s and t such that $\|s\|, \|t\| \leq k$, $s \approx_{E_{rep}} t$ if $s \approx_E t$. Base, $k = 2$: Let $a, b \in \Sigma_0$ be distinct constants. If $a \approx_E b$, then $\langle a \rangle \approx \langle b \rangle \in MGSE(E)$, so $a \approx b \in E_{rep}$. Inductive step: Assume for all ground terms s and t such that $\|s\|, \|t\| \leq k$, $s \approx_E t$ if and only if $s \approx_{E_{rep}} t$. Let $s := f(s_1, \ldots, s_l)$ and $t := g(t_1, \ldots, t_r)$ be terms such

that $\|s\|, \|t\| \leq k+1$. If s and t have the same signature in E, then $f = g$, $l = r$, and for all i, $s_i \approx_E t_i$, so $s \approx_{E_{rep}} t$ by the inductive hypothesis. Otherwise, $s \approx t$ is an instance of an $MGSE$, $sig_1 \approx sig_2$, with representative equation $u \approx v$ in E_{rep}. For each class C such that $C = sig_1[i]$ (resp. $C = sig_2[i]$), the inductive hypothesis implies that $s|_i \approx_{E_{rep}} rep_C$ (resp. $t|_i \approx_{E_{rep}} rep_C$), since $\|rep_C\| \leq \|s|_i\| \leq k+1$ (resp. $\|rep_C\| \leq \|t|_i\| \leq k+1$) by the definition of rep_C and $<$. For each variable x appearing at positions $I \subseteq \mathbb{N}$ in sig_1 and $I' \subseteq \mathbb{N}$ in sig_2, we can choose some $i \in I$ (resp. $i' \in I'$) and let $rep_x := s_i$ (resp. $rep_x := t_{i'}$). Using the inductive hypothesis, we can see that $\forall j \in I$, $s_j \approx_{E_{rep}} rep_x$ and $\forall j \in I'$, $t_j \approx_{E_{rep}} rep_x$. Therefore, we can construct terms $s' := f(s'_1, \ldots, s'_l)$ and $t' := g(t'_1, \ldots, t'_r)$ such that for all j, if $sig_1[j]$ (resp. $sig_2[j]$) is a class C, then $s'_i := rep_C$ (resp. $t'_i := rep_C$) and if $sig_1[j]$ (resp. $sig_2[j]$) is a variable x, then $s'_i := rep_x$ (resp. $t'_i := rep_x$). By this construction, we can see that $s \approx_{E_{rep}} s'$, $t \approx_{E_{rep}} t'$, and $s' \approx_{u \approx v} t'$. Since $u \approx v \in E_{rep}$, this means that $s \approx_{E_{rep}} t$. Therefore, $E_{rep} \equiv_G E$. ■

Lemma 3 Let E be any non-collapsing shallow presentation and let s, t be terms in $T(\Sigma, X)$ such that $s \approx_E t$. Assume there is a $u \in T(\Sigma)$ such for every $s' \approx t' \in E$, $[u] \notin D1(s') \cup D1(t')$. Let $I := D1P_{[u]}(s)$ and $J := D1P_{[u]}(t)$. Then $s[x]_I \approx_E t[x]_J$ for some $x \in X \backslash (Vars(s) \cup Vars(t))$.

Proof Assume $s = v_0 \approx_{e_0}^{p_0} v_1 \approx_{e_1}^{p_1} \ldots v_{r-1} \approx_{e_{r-1}}^{p_{r-1}} v_r = t$. We will prove the lemma by induction on r. Base $r = 0$: $s = t$, so $s[x]_I = t[x]_J$. Induction step: Assume $s[x]_I \approx_E v_{r-1}[x]_K$, where $K := D1P_{[u]}(v_{r-1})$ We will show that $v_{r-1}[x]_K \approx_E t[x]_J$ for all possible cases of p_{r-1}: I) If $p_{r-1} \geq k$ for some $k \in K$, then $v_{r-1}|_k \approx_E v_r|_k \in [u]$. So $K = J$ and $v_{r-1}[x]_K = t[x]_J$. II) If $p_{r-1} \geq n \in \mathbb{N}/K$, then $v_{r-1}[x]_I \approx_{e_{r-1}}^{p_{r-1}} v_r t$. III) If $p = \epsilon$, then let $s' \approx t'$ equal e_{r-1}. There is a $Y \subset X$ such that for each $k \in K$ and each $j \in J$, $s'|_k, t'|_j \in Y$ (otherwise $s'|_k$ or $t'|_j$ is in $[u]$, violating our hypothesis). By the definition of $\approx_E$, there is a substitution $sigma$ such that $v_{r-1} = s'\sigma$ and $t = t'\sigma$. Note that this implies that no variable in Y appears anywhere other than K in v_{r-1} and J in t. We can define σ' such that $\sigma'(y) := x$ for each $y \in Y$ and $\sigma'(y) := \sigma(y)$ for all $y \in X \backslash Y$. We then get that $s'\sigma' = v_{r-1}[x]_K$ and $t'\sigma' = t[x]_J$, so $v_{r-1}[x]_K \approx_E t[x]_J$. Thus $s[x]_I \approx_E t[x]_J$. ■

Lemma 4 For any non-collapsing shallow presentation E such that $|EQ_G(E)| \geq |D1(E)| + 2d$, $\mathcal{EC}(E) \subseteq D1(E)$.

Proof Assume for contradiction that the lemma isn't true. Let $c \in \mathcal{EC}(E)$ be a class that doesn't appear at depth 1 in E. Since c is essential, is must appear in the body of some $MGSE$ of E, $sig_1 \approx sig_2$. Choose a variable x' not in $Vars(sig_1 \approx sig_2)$ and form the signature $sig'_1 \approx sig'_2$ by replacing each occurrence of c in $sig_1 \approx sig_2$ with x'. We will show that $sig'_1 \approx sig'_2$ holds for E, which implies that c is not essential. For each variable $x_i \notin Vars(sig_1 \approx sig_2)$, choose a new $c_i \in EQ_G(E) \backslash D1(E)$ and set $\sigma(x_i) := t_i$, where $t_i \in c_i$. We can choose these distinct c_i classes since $|EQ_G(E)| \geq |D1(E)| + 2d$. Take any instance $s' \approx t'$ of $sig_1\sigma \approx sig_2\sigma$. Since $s' \approx t'$ is ground and $sig_1 \approx sig_2$ holds for E, $s' \approx_E t'$. Let $I_0 := D1P_{c'}(s')$ and $J_0 := D1P_{c'}(t')$ and set $s_0 := s'[x']_{I_0}$ and $t_0 := t'[x']_{J_0}$. Since $c' \notin D1(E)$, we can apply lemma 3 to see that $s_0 \approx_E t_0$. Now for each $x_i \in Vars(sig_1 \approx sig_2)$ set $I_i := D1P_\sigma x_i(s)$m $J_i := D1P_\sigma x_i(s)$, $s_i := s_{i-1}[x_i]_{I_i}$, and $t_i := t_{i-1}[x_i]_{J_i}$. Since σx_i is not in $D1(E)$, we can apply lemma 3 to see that $s_i \approx_E t_i$

for each i. If there are r variables in $sig_1 \approx sig_2$, then it is easy to check that $s_r \approx t_r$ is a maximally-generalized instance of $sig'_1 \approx sig'_2$, and $s_r \approx_E t_r$. Thus, $sig'_1 \approx sig'_2$ holds for E, and c is not essential by contradiction (see below this proof for an example). So $\mathcal{EC}(E) \subseteq D1(E)$. ■

To understand the above lemma, take for example a presentation E over the alphabet $\Sigma := \{f : 3, g : 2, a : 1, b : 1, c : 1\}$. Assume that E has $\langle f, y, a, z\rangle \approx \langle g, a, y\rangle$ as an MGSE and that $[a], [b], [c] \notin D1(E)$. We can generalize the signature equation to $\langle f, y, x', z\rangle \approx \langle g, x', y\rangle$ and consider the instance $f(b, a, c) \approx g(a, b) \in Th_G(E)$. We can then apply lemma 3 to see that $f(b, x', c) \approx_E g(x', b)$, $f(x_1, x', c) \approx_E g(x', x_1)$, and $f(x_1, x', x_2) \approx_E g(x', x_1)$. This is a maximally-generalized instance of $\langle f, y, x', z\rangle \approx \langle g, x', y\rangle$, which must hold on E. Thus $\langle f, y, a, z\rangle \approx \langle g, a, y\rangle$ must not be an MGSE, and we have a contradiction.

Theorem 6 Let E be any non-collapsing shallow presentation over the alphabet Σ, and let E_{rep} be the representative presentation formed from the essential classes of E. Then $|E_{rep}| \leq |\Sigma|^2(2d|E| + 4d)^{2d}$.

Proof Only essential terms and variables can appear at depth one in E_{rep}. Therefore, since there are at most $|2d|$ variables in any equation in E_{rep} and $|\Sigma|^2$ possible pairs of root symbols, we get that $|E_{rep}| \leq |\Sigma|^2(|\mathcal{EC}(E)| + 2d)^{2d}$. It remains to be shown that $|\mathcal{EC}(E)| \leq (2d|E| + 2d)$. This proof proceeds by cases depending on the size of $EQ_G(E)$, the equivalence classes of E that contain at least one ground term. Case 1: $|EQ_G(E)| < |D1(E)| + 2d$. Since all essential classes contain at least one ground term, $\mathcal{EC}(E) \subseteq EQ_G(E)$. Since at most $2d$ classes can appear at depth 1 per equation in E, $|D1(E)| \leq 2d|E|$. Thus $|\mathcal{EC}(E)| \leq |EQ_G(E)| < |D1(E)| + 2d \leq 2d|E| + 2d$. Case 2: $|EQ_G(E)| \geq |D1(E)| + 2d$. By lemma 4, $\mathcal{EC}(E) \subseteq D1(E)$, so $|\mathcal{EC}(E)| \leq |D1(E)|$. Since $|D1(E)| \leq 2d|E|$, we get that $|\mathcal{EC}(E)| \leq 2d|E|$. ■

Theorem 7 Let E be any non-collapsing shallow presentation over the alphabet Σ, and let E_{rep} be the representative presentation formed from the essential classes of E. Then $\|E_{rep}\| \leq |E_{rep}|(\|E\|(2d)^{2d} + 2)$.

Proof Assume $|D1(E)| + 2d \leq EQ_G(E)$. By lemma 4, $\mathcal{EC}(E) \subseteq D1(E)$. Let $m' := max_{c \in \mathcal{EC}(E)}\{minsize_E(c)\}$. Since a term from every essential class must appear in E, we get that $m' \leq \|E\|$. Each equation in E_{rep} contains only variables and essential terms at depth 1, and all essential terms are the smallest in their equivalence class. Therefore, for each $s \approx t \in E_{rep}$, $\|s \approx t\| \leq 2dm' + 2$. So $\|E_{rep}\| \leq |E_{rep}|(2dm' + 2) \leq |E_{rep}|(\|E\|2d + 2)$

Assume $|D1(E)|+2d > EQ_G(E)$ and let $\hat{C} := EQ_G(E)\backslash D1(E)$. If $maxarg_{c \in \mathcal{EC}(E)}\{minsize_E(c)\} \in D1(E)$, then we can apply the same reasoning above to bound $\|E_{rep}\|$.Otherwise, number the c_i in $\hat{C}$ such that $minsize_E(c_1) \leq minsize_E(c_2) \leq \dots$, and let $m := max_{c \in D1(E)}\{minsize_E(c)\}$ (note $m \leq \|E\|$). The value of $minsize_E(c_1)$ is maximized if its minimal size term contains only terms of size m and has maximum arity. Therefore, $minsize_E(c_1) \leq dm$. Likewise, for each $1 \leq i \leq |\hat{C}|$, $minsize_E(c_i) \leq d \cdot minsize_E(c_{i-1})$. So $minsize_E(c_{|\hat{C}|}) \leq d^{|\hat{C}|}m \leq d^{2d-1}\|E\|$. So $\|E_{rep}\| \leq |E_{rep}|(2d^{2d}\|E\| + 2)$. ■