Exact Learning of $\mathcal{EL}$ Ontologies

Ricardo Duarte, Boris Konev, Ana Ozaki

Center for Advancing Electronics Dresden (cfaed), TU Dresden, Germany
University of Liverpool, United Kingdom
Free University of Bozen-Bolzano, Italy

Abstract. We present the learning algorithm underpinning ExactLearner, a tool for exactly learning and teaching $\mathcal{EL}$ terminologies. The algorithm is superpolynomial in the size of the concept expressions and vocabulary of the terminology, but not in the size of the whole terminology (which is optimal). We evaluate ExactLearner's performance on $\mathcal{EL}$ ontologies from the Oxford ontology repository and demonstrate that despite the algorithm being exponential, it successfully terminates for small and medium size ontologies. We investigate the impact of various learner and teacher features and identify those most useful for learning.

1 Introduction

Authoring ontologies is a laborious task that requires a combined expertise of domain experts, who know the vocabulary of terms used in a particular subject area and have an understanding of the conceptual relationships between them, and of knowledge engineers, who can formalise these relations in an appropriate ontology definition language. In [6] the dialogue between an expert and a knowledge engineer is formalised as an instance of Angluin's exact learning framework in which a *learner* tries to exactly identify an ontology by communicating with a teacher, seen as an *oracle*. It is assumed that the vocabulary of terms is known and is communicated directly to the learner; in contrast, the exact ontology composition has to be found through a 'trial and error' learning process.

The learner poses queries of two kinds: membership queries, which ask the oracle to determine whether a given inclusion is entailed by the ontology, and equivalence queries, whether the ontology constructed is complete. If the answer to an equivalence query is negative, the oracle returns a statement which follows from the expert's knowledge but not from the ontology constructed, or the other way around. We are interested in learning algorithms that can identify any target ontology independently of the behaviour of the teacher. For complexity bounds, we consider the worst possible, adversarial teacher, which chooses to reveal as little information as possible.

In this paper we build on results of [6] and give an algorithm for learning $\mathcal{EL}$ terminologies, which is exponential in the size of concept expressions and its vocabulary but not in the size of the whole terminology, thus providing a formal underpinning to the ExactLearner system [5]. This result complements previous results showing that there is no polynomial time algorithm which can

exactly learn (even acyclic) $\mathcal{EL}$ terminologies [6]. We then introduce ExactLearner, a tool for exactly learning and teaching $\mathcal{EL}$ terminologies, which contains an implementation of our learning algorithm as well as a teacher. We evaluate ExactLearner's performance on $\mathcal{EL}$ ontologies from the Oxford ontology repository [8] and demonstrate that despite the algorithm being exponential, it successfully terminates for small and medium size ontologies. We investigate the impact of various learner and teacher features and identify those most useful for learning.

Related work. Most relevant to our work are: the DL-Learner [7], which learns concept expressions (but not ontologies) in various fragments of description logic, using refinement operators; and systems based on the exact learning model, such as: Logan-H [2] for learning function-free first order Horn sentences from interpretations; and EIRENE [1], for learning schema mappings. For a more detailed discussion of related work, see [6].

2 Preliminaries

Description logic. Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite sets of *concept* and *role* names. An $\mathcal{EL}$ concept expression C is formed according to the rule: $C, D := A \mid \top \mid C \sqcap D \mid \exists r.C$, where A ranges over $\mathsf{N_C}$ and r ranges over $\mathsf{N_R}$. A (general) $\mathcal{EL}$ *concept inclusion* has the form $C \sqsubseteq D$, where C and D are $\mathcal{EL}$ concept expressions. An $\mathcal{EL}$ *ontology* is a finite set of $\mathcal{EL}$ concept inclusions [4]. We call an $\mathcal{EL}$ ontology $\mathcal{O}$ a *terminology* if for all $C \sqsubseteq D \in \mathcal{O}$ either C or D is a concept name and $\mathcal{O}$ has at most one[1] inclusion of the form $A \sqsubseteq C$ for every $A \in \mathsf{N_C}$. $\mathcal{EL}_{lhs}$ is the class of $\mathcal{EL}$ terminologies consisting only of inclusions of the form $C \sqsubseteq A$, while $\mathcal{EL}_{rhs}$ only of inclusions of the form $A \sqsubseteq C$.

The *size* $|C|$ of a concept expression C is the length of the string that represents it, where concept names and role names are considered to be of length one. An ontology *vocabulary* is the set of concept and role names occurring in the ontology. The size of a concept inclusion $C \sqsubseteq D$, denoted $|C \sqsubseteq D|$, is $|C| + |D|$ and the size of an ontology $\mathcal{O}$, denoted $|\mathcal{O}|$, is $\sum_{C \sqsubseteq D \in \mathcal{O}} |C \sqsubseteq D|$. The semantics of $\mathcal{EL}$ is defined as usual [3]. We write $\mathcal{I} \models \alpha$ to say that a concept inclusion α is true in $\mathcal{I}$. An interpretation $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{O}$. $\mathcal{O} \models \alpha$ means that $\mathcal{I} \models \alpha$ for all models $\mathcal{I}$ of $\mathcal{O}$; and $\mathcal{O} \equiv \mathcal{O}'$ means that $\mathcal{O} \models \alpha$ if and only if $\mathcal{O}' \models \alpha$ for all concept inclusions α.

We identify each $\mathcal{EL}$ concept expression C with a finite tree T_C where nodes are labelled with sets of concept names and edges are labelled with roles: if A is a concept name, then T_A has a single node d with label $l(d) = \{A\}$; if $C = \exists r.D$, then T_C is obtained from T_D by adding a new root d_0 and an edge from d_0 to the root d of T_D with label $l(d_0, d) = r$ (we then call d an *r-successor* of d_0); if $C = D_1 \sqcap D_2$, then T_C is obtained by identifying the roots of T_{D_1} and

[1] In the literature, the term *terminology* commonly refers to sets of concept inclusions $A \sqsubseteq C$ and concept definitions $A \equiv C$, with no concept name occurring more than once on the left. As $A \equiv C$ can be equivalently rewritten as $A \sqsubseteq C$ and $C \sqsubseteq A$, our definition is a natural extension of this one.

T_{D_2}. A mapping h from a tree T_C to an interpretation $\mathcal{I}$ is a *homomorphism* if $A \in l(d)$ implies $h(d) \in A^{\mathcal{I}}$ for every concept name A and $r = l(d, d')$ implies $(h(d), h(d')) \in r^{\mathcal{I}}$ for all role names r. Then $d \in C^{\mathcal{I}}$ if, and only if, there is a homomorphism from T_C to $\mathcal{I}$ mapping the root of T_C to d.

In our proofs, we use notion of a *canonical model*, defined for every $\mathcal{EL}$ ontology $\mathcal{O}$ and $\mathcal{EL}$ concept C. $\mathcal{I}_{C,\emptyset}$, also denoted $\mathcal{I}_C$, is defined as T_C viewed as a tree-shaped interpretation. For $\mathcal{O} \neq \emptyset$, $\mathcal{I}_{C,\mathcal{O}}$ is the limit of the sequence $\mathcal{I}_0, \mathcal{I}_1, \ldots$ of interpretations, where $\mathcal{I}_0 = \mathcal{I}_C$ and $\mathcal{I}_{n+1}$ is obtained from $\mathcal{I}_n$ by applying, in a fair way, the following rule: if $C \sqsubseteq D \in \mathcal{O}$ and $d \in C^{\mathcal{I}_n}$ but $d \notin D^{\mathcal{I}_n}$, then take the interpretation $\mathcal{I}_D$ and add it to $\mathcal{I}_n$ by identifying its root ρ_D with d. $\mathcal{I}_{C,\mathcal{O}} \models \mathcal{O}$ and, for any $\mathcal{EL}$ concept D, we have $\rho_C \in D^{\mathcal{I}_{C,\mathcal{O}}}$ iff $\mathcal{O} \models C \sqsubseteq D$, where ρ_C is the root of $\mathcal{I}_{C,\mathcal{O}}$.

Subsumption learning framework. Given a class of ontologies $\mathcal{L}$ (for example all ontologies in a particular DL, $\mathcal{EL}$ terminologies etc), we are interested in the exact identification of a *target ontology* $\mathcal{O} \in \mathcal{L}$ by posing queries to an oracle. We assume that the vocabulary of the target terminology $\Sigma_{\mathcal{O}}$ is known to the learner. A *membership query* is a call to the oracle to test for an inclusion $C \sqsubseteq D$, where C, D are $\Sigma_{\mathcal{O}}$-concept expressions of the DL under consideration, if $\mathcal{O} \models C \sqsubseteq D$. An inclusion $C \sqsubseteq D$ is a *positive example* w.r.t. a target $\mathcal{O}$ if $\mathcal{O} \models C \sqsubseteq D$ and a *negative example* else. An *equivalence query* is a call to the oracle to check if a *hypothesis* ontology $\mathcal{H}$ is equivalent to the target $\mathcal{O}$. If it is the case, the oracle responds 'yes', otherwise the oracle returns a positive example $C \sqsubseteq D$ with $\mathcal{H} \not\models C \sqsubseteq D$ or a negative example $E \sqsubseteq F$ with $\mathcal{H} \models E \sqsubseteq F$. Such a positive example $C \sqsubseteq D$ (negative example $E \sqsubseteq F$) is called a *positive counterexample* (a *negative counterexample*, resp.) to $\mathcal{H}$ being equivalent to $\mathcal{O}$. For a formal definition of the *subsumption learning framework* and a discussion of how this definition relates to Angluin's exact learning model see [6].

We say that a class of ontologies $\mathcal{L}$ is *exactly learnable* if there is an algorithm, which halts for any target $\mathcal{O} \in \mathcal{L}$ and computes, using membership and equivalence queries, $\mathcal{H} \in \mathcal{L}$ with $\mathcal{H} \equiv \mathcal{O}$ as long as the oracle replies to the queries truthfully. An ontology class is exactly learnable in polynomial time if it is exactly learnable by an algorithm A such that at every step[2] of computation the time used by A up to that step is bounded by a polynomial $p(|\mathcal{O}|, |C \sqsubseteq D|)$, where $\mathcal{O}$ is the target and $C \sqsubseteq D$ is the largest counterexample seen so far. $\mathcal{EL}_{lhs}$ and $\mathcal{EL}_{rhs}$ are known to be exactly learnable in polynomial time, while the class of all $\mathcal{EL}$ ontologies is not learnable in polynomial time [6].

3 Learning $\mathcal{EL}$ ontologies

In this section we present Algorithm 1, which can exactly learn $\mathcal{EL}$ terminologies in time exponential in $|C_{\mathcal{O}}|$, the size of the largest concept expression in $\mathcal{O}$, and $|\Sigma_{\mathcal{O}}|$, the size of the ontology vocabulary, but not in the size of the whole ontology.

[2] We count each call to an oracle as one step.

Algorithm 1 The learning algorithm for $\mathcal{EL}$

Input: An $\mathcal{EL}$ terminology $\mathcal{O}$ given to the oracle; $\Sigma_\mathcal{O}$ given to the learner
Output: An $\mathcal{EL}$ terminology $\mathcal{H}$ computed by the learner such that $\mathcal{O} \equiv \mathcal{H}$

Set $\mathcal{H} = \{A \sqsubseteq B \mid \mathcal{O} \models A \sqsubseteq B,\ A, B \in \Sigma_\mathcal{O}\}$
while $\mathcal{H} \not\equiv \mathcal{O}$ **do**
  Let $C \sqsubseteq D$ be the returned positive counterexample for $\mathcal{O}$ relative to $\mathcal{H}$
  Compute $C' \sqsubseteq D'$ with C' or D' in $\Sigma_\mathcal{O} \cap \mathsf{N_C}$
  if $C' \in \Sigma_\mathcal{O} \cap \mathsf{N_C}$ **then**
    Compute a right $\mathcal{O}$-essential α from $C' \sqsubseteq D' \sqcap \prod_{C' \sqsubseteq F' \in \mathcal{H}} F'$
  else
    Compute a left $\mathcal{O}$-essential α from $C' \sqsubseteq D'$
  end if
  Add α to $\mathcal{H}$
end while
return $\mathcal{H}$

In the main loop of the algorithm the learner poses an equivalence query to the oracle. If the oracle answers "yes" then the algorithm returns $\mathcal{H}$ equivalent to $\mathcal{O}$. Otherwise, it receives a counterexample $C \sqsubseteq D$. It is easy to see that at all times $\mathcal{O} \models \mathcal{H}$ so the counterexample is always positive. As $\mathcal{O}$ is a terminology, complex C and D in the counterexample can only "connect" via a concept name, which can be identified by asking membership queries.

Lemma 1. *Given a counterexample $C \sqsubseteq D$, one can construct, by posing membership queries, a counterexample $C' \sqsubseteq D'$ such that $|C' \sqsubseteq D'| \leq |C \sqsubseteq D|$ and either C' or D' is a concept name in time polynomial in $|\mathcal{H}|$, $|C|$ and $|\Sigma_\mathcal{O}|$.*

Proof. (Sketch) This lemma relies on the properties of the rooted canonical model $\mathcal{I}_{C,\mathcal{O}}$ of an $\mathcal{EL}$ concept C and an $\mathcal{EL}$ terminology $\mathcal{O}$. It can be proved by induction on the structure of the right-hand side of the counterexample $C \sqsubseteq D$ using the following property proved using the construction of $\mathcal{I}_{C,\mathcal{O}}$:

For any $d \in \Delta^{\mathcal{I}_C}$ and $\mathcal{EL}$ concept expression of the form $\exists r.F$: if there exists a homomorphism h from the labeled tree corresponding to $\exists r.F$ into $\mathcal{I}_{C,\mathcal{O}}$ such that $h(a_{\exists r.F}) = d$ and all nodes in the labeled tree corresponding to F are mapped to $\Delta^{\mathcal{I}_{C,\mathcal{O}}} \setminus \Delta^{\mathcal{I}_C}$, then there exists a concept name E with $d \in E^{\mathcal{I}_C}$ such that $\mathcal{O} \models E \sqsubseteq \exists r.F$. □

Having transformed the counterexample to the case of a concept name on the left or on the right, the algorithm minimises the size of the counterexample and makes it more informative by applying rules. If C' is a concept name then Algorithm 1 merges D' with the right-hand sides of all inclusions in $\mathcal{H}$ with C' on the left (if they exist) and computes a so called *right $\mathcal{O}$-essential* counterexample. Otherwise D' is a concept name, and the algorithm computes a *left $\mathcal{O}$-essential* counterexample. It then adds the resulting $\mathcal{O}$-essential concept inclusion α to $\mathcal{H}$.

Right $\mathcal{O}$-essential concept inclusions We say that an inclusion $A \sqsubseteq C$ is *right $\mathcal{O}$-essential* if none of the following three rules applies to $A \sqsubseteq C$.

Concept saturation for $\mathcal{O}$: If $\mathcal{O} \models A \sqsubseteq C'$ and C' results from C by adding a concept name A' to the label of some node, then replace $A \sqsubseteq C$ by $A \sqsubseteq C'$:

Sibling merging for $\mathcal{O}$: If $\mathcal{O} \models A \sqsubseteq C'$ and C' is the result of identifying in C two r-successors of the same node then replace $A \sqsubseteq C$ by $A \sqsubseteq C'$.

Decomposition on the right for $\mathcal{O}$: If d' is an r-successor of d in C, A' is in the node label of d, and $\mathcal{O} \models A' \sqsubseteq \exists r.C_{d'}$ plus $A' \not\equiv_{\mathcal{O}} A$ if d is the root of C, then replace $A \sqsubseteq C$ by

(a) $A' \sqsubseteq \exists r.C_{d'}$ if $\mathcal{H} \not\models A' \sqsubseteq \exists r.C_{d'}$; or

(b) $A \sqsubseteq C|^{-}_{d'\downarrow}$, otherwise,

where C_d is the concept corresponding to the subtree rooted in d and $C|^{-}_{d\downarrow}$ is the concept corresponding to the result of removing the subtree rooted in d from C.

It follows from [6], Lemma 29 and its proof, that given a counterexample $A \sqsubseteq C$, one can construct a right $\mathcal{O}$-essential counterexample $A' \sqsubseteq C'$ using only polynomially many polynomial size membership queries in $|\mathcal{H}| + |C| + |\Sigma_{\mathcal{O}}|$.

Concept saturation, sibling merging and decomposition on the right are all essential—hence the name—steps of the polynomial learning algorithm for DL-Lite$^{\exists}_{\mathcal{R}}$, which extends $\mathcal{EL}_{rhs}$ with inverse roles and role hierarchies [6].

Example 1. (a) For $\mathcal{H} = \emptyset$ and $\mathcal{O} = \{\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.\mathsf{Human}\}$ the oracle can return arbitrary many arbitrary long $\mathsf{hasParent}$ chains starting at Human as a counterexamples leading to non-termination. For instance, $\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.\exists \mathsf{hasParent}.\top$ is a chain of length two. With concept saturation, this counterexample can be strengthened to $\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.(\mathsf{Human} \sqcap \exists \mathsf{hasParent}.\mathsf{Human})$, which is equivalent to $\mathcal{O}$.

(b) For $\mathcal{O} = \{\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.(\mathsf{Human} \sqcap \mathsf{Male})\}$ and $\mathcal{H} = \{\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.\mathsf{Human}\}$, upon receiving a counterexample $\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.\mathsf{Male}$, the learner merges its right hand side with the right hand side of the inclusion in $\mathcal{H}$ to form $\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.\mathsf{Male} \sqcap \exists \mathsf{hasParent}.\mathsf{Human}$ and then strengthens it by sibling merging to form the inclusion in $\mathcal{O}$.

(c) For $\mathcal{H} = \emptyset$ and $\mathcal{O} = \{\mathsf{Woman} \sqsubseteq \mathsf{Human}, \mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.\mathsf{Human}\}$, even with concept saturation, there exist infinitely many chain counterexamples; $\mathsf{Woman} \sqsubseteq \mathsf{Human} \sqcap \exists \mathsf{hasParent}.(\mathsf{Human} \sqcap \exists \mathsf{hasParent}.\mathsf{Human})$ is one of them. This inclusion can be decomposed at the root into (a) $\mathsf{Human} \sqsubseteq \mathsf{Woman}$ and (b) $\mathsf{Human} \sqsubseteq \exists \mathsf{hasParent}.(\mathsf{Human} \sqcap \exists \mathsf{hasParent}.\mathsf{Human})$. Picking either of them allows the learner to make progress.

In fact, one can prove that Algorithm 1 polynomially learns $\mathcal{EL}_{rhs}$.

Theorem 1. *The class of $\mathcal{EL}_{rhs}$ ontologies is exactly learnable in polynomial time by Algorithm 1.*

We show that even in the presence of inclusions of the form $D \sqsubseteq B$, right $\mathcal{O}$-essential concept inclusions are bounded in size by the size of the largest concept expression in $\mathcal{O}$.

Lemma 2. *Suppose that $A \sqsubseteq C$ is right $\mathcal{O}$-essential. Then $|C| \leq |C_{\mathcal{O}}| \cdot |\Sigma_{\mathcal{O}}|$, where $C_{\mathcal{O}}$ is the largest concept expression in $\mathcal{O}$.*

Proof. (Sketch) We follow the lines of the proof of Lemma 32 in [6]. First we define

$$A^{\mathcal{O}} = \{A\} \cup \{D \mid \mathcal{O} \models A \sqsubseteq B, B \sqsubseteq D \in \mathcal{O}\}$$

and construct the canonical model $\mathcal{I}_{D_0,\mathcal{O}}$ of $D_0 = \bigsqcap_{D \in A^{\mathcal{O}}} D$ and $\mathcal{O}$. The interpretation $\mathcal{I}_{D_0,\mathcal{O}}$ has the following properties:

1. for every $\mathcal{EL}$ concept expression F: $\rho_{D_0} \in F^{\mathcal{I}_{D_0,\mathcal{O}}}$ iff $\mathcal{O} \models A \sqsubseteq F$;
2. for any $d \in \Delta^{\mathcal{I}_{D_0}}$ and $\mathcal{EL}$ concept expression of the form $F = \exists r.F'$: if there exists a homomorphism h from the labeled tree corresponding to F into $\mathcal{I}_{D_0,\mathcal{O}}$ such that $h(a_F) = d$ and all nodes in the labeled tree corresponding to F' are mapped to $\Delta^{\mathcal{I}_{D_0,\mathcal{O}}} \setminus \Delta^{\mathcal{I}_{D_0}}$, then there exists a concept name E with $d \in E^{\mathcal{I}_{D_0}}$ such that $\mathcal{O} \models E \sqsubseteq F$.

Then, one can show that there is an injective homomorphism h from the labeled tree T_C of C into the restriction $\mathcal{J}$ of $\mathcal{I}_{D_0,\mathcal{O}}$ to $\Delta^{\mathcal{I}_{D_0}}$ which maps a_C to ρ_{D_0}. By definition of $A^{\mathcal{O}}$, there is $B \sqsubseteq D \in \mathcal{O}$ such that $|\Delta^{\mathcal{I}_C}| \leq |\Delta^{\mathcal{I}_D}|$, which means that $|\Delta^{\mathcal{I}_C}| \leq |\Delta^{\mathcal{I}_{C_\mathcal{O}}}|$, where $C_\mathcal{O}$ is the largest concept expression in $\mathcal{O}$. The main observation here is that the argument holds for $\mathcal{EL}$ terminologies, that is, in the presence of concept inclusions of the form $D \sqsubseteq B$. Such inclusions do not interfere in the argument since concept saturation ensures that nodes have all concept names implied by $\mathcal{O}$ in their labels. □

Left $\mathcal{O}$-essential concept inclusions We say that an inclusion $C \sqsubseteq A$ is *left $\mathcal{O}$-essential* if neither of the following rules applies to $C \sqsubseteq A$.

Concept saturation for $\mathcal{H}$: If $\mathcal{H} \models C \sqsubseteq C'$ and C' results from C by adding a concept name A' to the label of some node, then replace $C \sqsubseteq A$ by $C' \sqsubseteq A$.
Decomposition on the left for $\mathcal{O}$: If d is a non-root node such that $\mathcal{O} \models C|^-_{d\downarrow} \sqsubseteq A'$ and $\mathcal{H} \not\models C|^-_{d\downarrow} \sqsubseteq A'$, for some $A' \in \Sigma_\mathcal{O}$, then replace $C \sqsubseteq A$ by $C|^-_{d\downarrow} \sqsubseteq A'$; if $\mathcal{O} \models C_d \sqsubseteq A'$ and $\mathcal{H} \not\models C_d \sqsubseteq A'$, then replace $C \sqsubseteq A$ by $C_d \sqsubseteq A'$.

The applicability of the second rule may depend on the application of the first. For example, for $\mathcal{H} = \{\exists \mathsf{hasParent}.\top \sqsubseteq \mathsf{Human}\}$ and $\mathcal{O} = \mathcal{H} \cup \{\exists \mathsf{hasChild}.\mathsf{Human} \sqsubseteq \mathsf{Human}\}$ a counterexample could be $\exists \mathsf{hasChild}.\exists \mathsf{hasParent}.\top \sqsubseteq \mathsf{Human}$, which can only be decomposed on the left for $\mathcal{O}$ if we apply concept saturation for $\mathcal{H}$ first.

Similarly to right $\mathcal{O}$-essential counterexamples, given a counterexample $C \sqsubseteq A$, one can construct a left $\mathcal{O}$-essential counterexample $C' \sqsubseteq A'$ using only polynomially many polynomial size membership queries in $|\mathcal{H}| + |C| + |\Sigma_\mathcal{O}|$. We show in Lemma 3 that the size of left $\mathcal{O}$-essential counterexamples is polynomially bounded by $|\Sigma_\mathcal{O}|$ and $|C_\mathcal{O}|$. The intuition behind this lemma is that if neither concept saturation for $\mathcal{H}$ nor decomposition on the left for $\mathcal{O}$ apply to an inclusion $C \sqsubseteq A$ then for some $D \sqsubseteq E \in \mathcal{O}$, with $E \in \mathsf{N_C}$, not only there is a homomorphism h from $\mathcal{I}_D$ to $\mathcal{I}_{C,\mathcal{O}}$ mapping their roots but also all elements of $\mathcal{I}_C$ are in the image of this homomorphism. However, in contrast to Lemma 58 by Konev et. al [6], due to inclusions of the form $B \sqsubseteq F$ with F a complex expression, some nodes of D may be mapped by h outside of $\mathcal{I}_C$. For example, for $\mathcal{H} = \emptyset$,

$\mathcal{O} = \{\exists r.\exists s.B \sqsubseteq E, E \sqsubseteq A, B \sqsubseteq \exists s.B\}$ and a left $\mathcal{O}$-essential counterexample $\exists r.B \sqsubseteq A$, the homomorphism from $\mathcal{I}_{\exists r.\exists s.B}$ to $\mathcal{I}_{\exists r.B,\mathcal{O}}$ covers $\mathcal{I}_{\exists r.B}$ but there is only a *partial homomorphism* (that is, a partial function satisfying properties of a homomorphism) from $\mathcal{I}_{\exists r.\exists s.B}$ to $\mathcal{I}_{\exists r.B}$.

Lemma 3. *Suppose that $C \sqsubseteq A$ is left $\mathcal{O}$-essential. Then $|C| \leq |C_{\mathcal{O}}| \cdot |\Sigma_{\mathcal{O}}|$, where $C_{\mathcal{O}}$ is the largest concept expression in $\mathcal{O}$.*

Proof. To show this lemma, we use the canonical model $\mathcal{I}_{C,\mathcal{O}}$ of C and $\mathcal{O}$. If there is $E \in \Sigma_{\mathcal{O}}$ such that $\mathcal{O} \models C \sqsubseteq E$ (and $C \sqsubseteq E$ is not a tautology) then there is $D \sqsubseteq E \in \mathcal{O}$ such that $\rho_C \in D^{\mathcal{I}_{C,\mathcal{O}}}$. Let

$$(C,\mathcal{O})^{\mathcal{I}_C} = \{D \mid \exists E \in \Sigma_{\mathcal{O}} : \rho_C \in D^{\mathcal{I}_{C,\mathcal{O}}}, \rho_C \notin E^{\mathcal{I}_C}, D \sqsubseteq E \in \mathcal{O}\}.$$

For all $D \in (C,\mathcal{O})^{\mathcal{I}_C}$ there is a homomorphism $h : \mathcal{I}_D \to \mathcal{I}_{C,\mathcal{O}}$ mapping the root of $\mathcal{I}_D$ to the root of $\mathcal{I}_{C,\mathcal{O}}$. Also, by construction of $\mathcal{I}_{C,\mathcal{O}}$, there is $D' \in (C,\mathcal{O})^{\mathcal{I}_C}$ such that:

1. there is a partial homomorphism $h : \mathcal{I}_{D'} \to \mathcal{I}_C$ mapping the root of $\mathcal{I}_{D'}$ to the root of $\mathcal{I}_C$, for $e \in \Delta^{\mathcal{I}_{D'}}$ with $h(e) \in \Delta^{\mathcal{I}_C}$;
2. if D' has a conjunct $\exists r.F$, with $d \in F^{\mathcal{I}_{D'}}$ as the corresponding r-successor of ρ_D, and $h(d) \notin \Delta^{\mathcal{I}_C}$ then there is $A \in \mathsf{N_C}$ such that $\rho_C \in A^{\mathcal{I}_C}$ and $\mathcal{O} \models A \sqsubseteq \exists r.F$.

The fact that $|\Delta^{\mathcal{I}_C}| \leq |\Delta^{\mathcal{I}_{D'}}|$ for such $D' \sqsubseteq E \in \mathcal{O}$ is a result of the Claim.

Claim. For all $d \in \Delta^{\mathcal{I}_C}$, there is $e \in \Delta^{\mathcal{I}_{D'}}$ such that $d = h(e)$.

Suppose to the contrary that there is $d' \in \Delta^{\mathcal{I}_C}$ such that d' is outside the image of $h : \mathcal{I}_{D'} \to \mathcal{I}_{C,\mathcal{O}}$. Let $G = C|^{-}_{d'\downarrow}$ be the concept corresponding to the result of removing the subtree rooted in d'. Since C is concept saturated for $\mathcal{H}$, if $\rho_G \in D'^{\mathcal{I}_{G,\mathcal{O}}}$ and $\rho_G \notin E^{\mathcal{I}_G}$ then this contradicts the fact that decomposition on the left for $\mathcal{O}$ was exhaustively applied. To show the contradiction, we argue that h is a homomorphism from $\mathcal{I}_{D'}$ into the restriction $\mathcal{I}_{G,\mathcal{O}}$ of $\mathcal{I}_{C,\mathcal{O}}$. Our construction has the following properties:

3. for $e \in \Delta^{\mathcal{I}_{D'}}$ with $h(e) \in \Delta^{\mathcal{I}_C}$, we have that $h(e) \in \Delta^{\mathcal{I}_G}$;
4. for all $d \in \Delta^{\mathcal{I}_C} \setminus \{\rho_C\}$, $d \in A^{\mathcal{I}_C}$ iff $d \in A^{\mathcal{I}_{C,\mathcal{O}}}$, for all $A \in \mathsf{N_C}$.

We obtain Point (3) by the fact that since d' is outside the image of $h : \mathcal{I}_{D'} \to \mathcal{I}_{C,\mathcal{O}}$, all descendants of d' are outside the image of $h : \mathcal{I}_{D'} \to \mathcal{I}_{C,\mathcal{O}}$ as well. Point (4) follows from decomposition of the left for $\mathcal{O}$ and concept saturation for $\mathcal{H}$.

For $e \in \Delta^{\mathcal{I}_{D'}}$ with $h(e) \in \Delta^{\mathcal{I}_C}$, by Points (1) and (3), there is a partial homomorphism from $\mathcal{I}_{D'}$ to $\mathcal{I}_G$. For $e \in \Delta^{\mathcal{I}_{D'}}$ with $h(e) \notin \Delta^{\mathcal{I}_C}$, there is $e_1 \in \Delta^{\mathcal{I}_{D'}}$ such that: (i) $h(e_1) \notin \Delta^{\mathcal{I}_C}$; (ii) e is in the subtree rooted in e_1; and (iii) e_1 has minimal distance from $\rho_{D'}$, where 'distance' here is the length of a chain of elements connected via roles from $\rho_{D'}$ to e_1. This means that the parent e_2 of e_1 is such that $h(e_2) \in \Delta^{\mathcal{I}_C}$. Let F be the concept corresponding to the subtree rooted in e_1 and let s be the role between e_2 and e_1. That is, $(e_2, e_1) \in s^{\mathcal{I}_{D'}}$. Since

$h(e_1) \notin \Delta^{\mathcal{I}_C}$ we have that all descendants of e_1 are not mapped to $\Delta^{\mathcal{I}_C}$. Then, by construction of the canonical model $\mathcal{I}_{C,\mathcal{O}}$ of C and the $\mathcal{EL}$ terminology $\mathcal{O}$, one can show that there is a $B \in \mathsf{N_C}$ such that $h(e_2) \in B^{\mathcal{I}_{C,\mathcal{O}}}$ and $\mathcal{O} \models B \sqsubseteq \exists s.F$. So it remains to show that $h(e_2) \in B^{\mathcal{I}_{G,\mathcal{O}}}$, for some $B \in \mathsf{N_C}$ such that $\mathcal{O} \models B \sqsubseteq \exists s.F$. We make a case distinction:

- for $h(e_2) \neq \rho_C$: by Point (4), $h(e_2) \in B^{\mathcal{I}_{C,\mathcal{O}}}$ implies $h(e_2) \in B^{\mathcal{I}_C}$. As $h(e_2) \in \Delta^{\mathcal{I}_C}$, by Point (3), we have that $h(e_2) \in \Delta^{\mathcal{I}_G}$. So $h(e_2) \in B^{\mathcal{I}_G}$, and thus, $h(e_2) \in B^{\mathcal{I}_{G,\mathcal{O}}}$.
- for $h(e_2) = \rho_C$: by Point (2), if $e_1 \in F^{\mathcal{I}_{D'}}$ and $h(e_1) \notin \Delta^{\mathcal{I}_C}$ then there is a concept name B such that $\rho_C \in B^{\mathcal{I}_C}$ and $\mathcal{O} \models B \sqsubseteq \exists s.F$. So $\rho_G \in B^{\mathcal{I}_G}$, which means that $\rho_G = h(e_2) \in B^{\mathcal{I}_{G,\mathcal{O}}}$.

Since we showed for arbitrary $e \in \Delta^{\mathcal{I}_{D'}}$ with $h(e) \in \Delta^{\mathcal{I}_C}$ and $h(e) \notin \Delta^{\mathcal{I}_C}$ that there is $d \in \Delta^{\mathcal{I}_{G,\mathcal{O}}}$ such that $d = h(e)$, we have that $h : \mathcal{I}_{D'} \to \mathcal{I}_{G,\mathcal{O}}$ is a homomorphism. Thus, there is $D' \in (C,\mathcal{O})^{\mathcal{I}_G}$ such that $\rho_G \in D'^{\mathcal{I}_{G,\mathcal{O}}}$ and $\rho_G \notin E^{\mathcal{I}_G}$ for some $D' \sqsubseteq E \in \mathcal{O}$, which contradicts the fact that decomposition on the left for $\mathcal{O}$ was exhaustively applied, so our claim follows.

This bounds the size of the domain of $\mathcal{I}_C$ to the size of the domain of $\mathcal{I}_{D'}$. Since $|\Delta^{\mathcal{I}_{D'}}| \leq |\Delta^{\mathcal{I}_{C_\mathcal{O}}}|$, we have $|\Delta^{\mathcal{I}_C}| \leq |\Delta^{\mathcal{I}_{C_\mathcal{O}}}|$. □

By combining lemmas 2 and 3 we obtain the following result.

Lemma 4. *Given a counterexample $C \sqsubseteq D$ for $\mathcal{O}$ relative to $\mathcal{H}$, one can construct a counterexample $C' \sqsubseteq D'$ such that $|C' \sqsubseteq D'| \leq |C_\mathcal{O}| \cdot |\Sigma_\mathcal{O}| + 1$ in polynomial time in $|C \sqsubseteq D|$, $|\Sigma_\mathcal{O}|$ and $|\mathcal{H}|$.*

Since there are at most $|\Sigma_\mathcal{O}|^{|C_\mathcal{O}|\cdot|\Sigma_\mathcal{O}|+1}$ many inclusions over $\Sigma_\mathcal{O}$ of size $|C_\mathcal{O}| \cdot |\Sigma_\mathcal{O}| + 1$, at most $|\Sigma_\mathcal{O}|^{|C_\mathcal{O}|\cdot|\Sigma_\mathcal{O}|+1}$ counterexamples get added to $\mathcal{H}$ over the run of the algorithm. Thus we obtain the following theorem.

Theorem 2. *The class of $\mathcal{EL}$ terminologies is exactly learnable by Algorithm 1 in $O(|\Sigma_\mathcal{O}|^{2|C_\mathcal{O}|\cdot|\Sigma_\mathcal{O}|+2} \cdot |C \sqsubseteq D|^2)$ steps, where $\Sigma_\mathcal{O}$ is the vocabulary of the target terminology $\mathcal{O}$, $C_\mathcal{O}$ is the largest concept expression in $\mathcal{O}$ and $C \sqsubseteq D$ is the largest counterexample seen by Algorithm 1.*

4 Evaluation

The ExactLearner system [5] has two main components: a learner and a teacher. The learner supports (1) "Concept Saturation", (2) "Sibling Merging", (3) "Decomposition", applied on the right side of inclusions, and (4) "Concept Desaturation", (5) "Sibling Branching" and (6) "Decomposition", applied on the left. Operations (1), (2), (3) and (6) have already been described. In addition, we have also implemented (4) and (5), which act as heuristics to construct smaller, more informative counterexamples. Concept desaturation tries to remove concept names from nodes in the left of counterexamples to make them logically stronger. Sibling branching tries to strengthen a counterexample by splitting paths on

	Test 2			Test 3		
p	# timeouts	avg CE	avg max C	# timeouts	avg CE	avg max C
0.01	3	17.2	27.7	2	5.6	31.7
0.5	25	107.8	26.6	3	6.1	31.6
1.0	26	190.4	19.5	3	6.3	31.9

Table 1. Learner against the adversarial teacher.

the left. For example, for $\mathcal{O} = \{\exists \mathsf{hasDegree}.\mathsf{BSc} \sqcap \exists \mathsf{hasDegree}.\mathsf{MSc} \sqsubseteq \mathsf{PG}\}$ and $\mathcal{H} = \emptyset$, the inclusion $\exists \mathsf{hasDegree}.(\mathsf{BSc} \sqcap \mathsf{MSc} \sqcap \mathsf{PhD}) \sqsubseteq \mathsf{PG}$ is a counterexample, from which desaturation removes the irrelevant PhD and then sibling branching strengthens it to the one in $\mathcal{O}$. Concept saturation for $\mathcal{H}$ was implemented as part of Rule (6), "Decomposition" applied to the left, as the applicability of Rule (6) may depend on the exhaustive application of this rule.

We have evaluated ExactLearner's performance on $\mathcal{EL}$ ontologies from the Oxford ontology repository [8]. As a first experiment we ran the learner against a naïve teacher, which presents the target ontology inclusions one by one without modification. This experiment aims at estimating the overheads of the learning process under the best possible conditions. In this first experiment, for 50 out of 174 $\mathcal{EL}$ ontologies computations concluded within 1 hour. We selected these ontologies for further experiments.

The selected ontologies range in size from 9 to 11 177 inclusions with signature sizes ranging from 23 to 9334 concept names and from 2 to 25 role names. The average size of counterexamples produced by the teacher was 5.48 while the average size of the largest concept in $\mathcal{O}$ was 2.7. The average size of the largest concept in $\mathcal{H}$ was 31.3, an increase caused by concept saturation on the right side of inclusions. The performance bottlenecks in our system are checking if the presented inclusion is a counterexample w.r.t. the current hypothesis ontology at the server side and entailment checks in the learner.

We have also introduced an adversarial teacher, which forces the learner to apply particular operations from (1)–(6) above by manipulating the counterexamples. For instance, to force the learner to perform concept saturation on the right of $A \sqsubseteq C$, the teacher exhaustively tries to remove concept names from every node in the tree representation of C, while ensuring that the modified inclusion is still a counterexample. The adversarial teacher can apply: (7) "Concept Desaturation" on the right, which we have just described; (8) "Sibling Branching" on the right, which weakens counterexamples of the form $A \sqsubseteq \exists r.(C \sqcap D)$ into $A \sqsubseteq \exists r.C \sqcap \exists r.D$ (provided the latter is still a counterexample); (9) "Concept Saturation" on the left; and (10) "Sibling Merging" on the left, which are the opposite of learner's concept desaturation and sibling branching. We also substitute concept definitions into counterexamples, for instance, if $A \sqsubseteq \exists r.B$ is a counterexample and $B \sqsubseteq C \in \mathcal{O}$ we test $A \sqsubseteq \exists r.C$ for being a counterexample as well. We call this operation (11) "Composition on the right". (12) "Composition on the left" is its

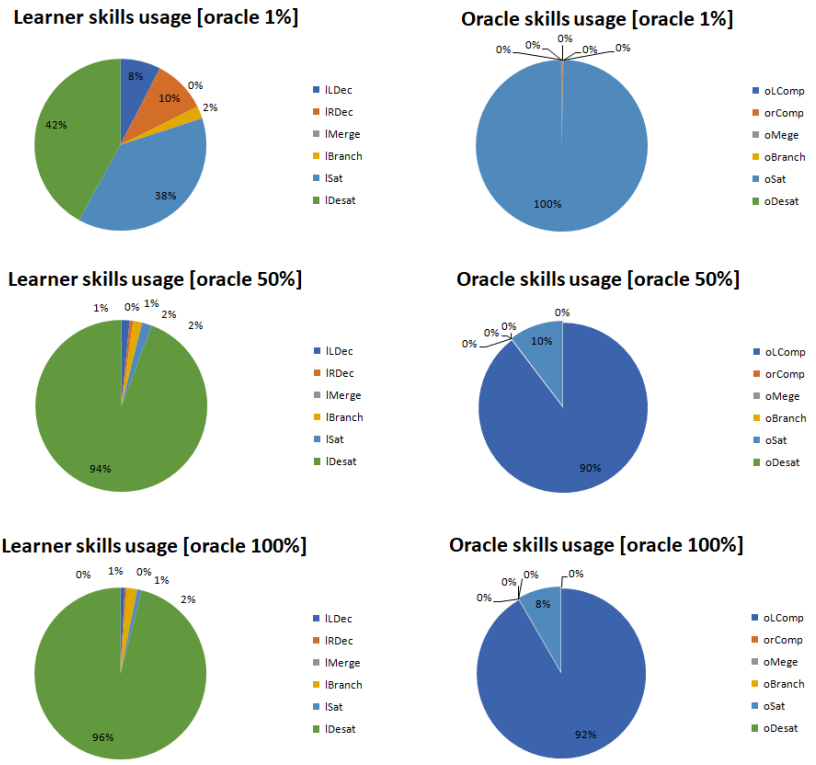

Fig. 1. Oracle and learner skills usage in Test 2.

counterpart. Operations (7)–(12) are applied at random with set probabilities so that the level of difficulty could be controlled (whenever the oracle applies an operation, the result is a 'less informative' counterexample, so learning from those counterexamples may require more queries, and thus, increase the difficulty).

Table 1 presents statistics of running the learner against the adversarial teacher. In Test 2 the teacher was applying transformations (7)–(12) with probability p of 0.01, 0.5 and 1.0. The learner can cope with a small distortion of examples ($p = 0.01$) but a significant distortion leads to a big increase in the number of time-outs. Figure 1 shows the percentage of the rules applied by the learner and the oracle in Test 2. As the chart indicates, the most frequently applied rule (42%) for the learner with probability $p = 0.01$ is desaturation in the left. This number grows to 94% when $p = 0.5$ and to 96% when $p = 1.0$. For the oracle, the most frequently applied rule with $p = 0.01$ is saturation on the left. Then, we see an increase on the application of composition on the left, which grows to 90% when $p = 0.5$ and to 92% when $p = 1.0$.

As the oracle applies saturation on the left, there are more concept names on the left which can be used for composition on the left. The discrepancy between the number of compositions on the left by the oracle and decompositions on the

left by the learner is due to the fact that the oracle applies the rule repeatedly while the learner finds a minimal subtree in one rule application.

In Test 3 we have disabled rule (9) at the oracle side as well as rules (8) and (10) as they almost never applied, see Figure 1, yet took up a significant time in our tests. This led to a significant drop in the failure rate even though other adversarial teacher operations were applied with a high probability. This suggests that the main cause of time-outs is the exponential explosion in the size of the signature rather than the size of concepts in the target ontology. We also

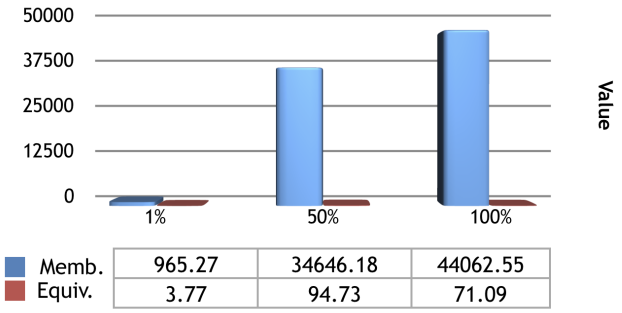

Fig. 2. Membership and equivalence queries.

measured the increase on the number of queries in Test 2 when the probability for the oracle to apply a certain rule increases. In Test 2, 22 ontology computations concluded within 1 hour with the probability of the oracle to apply a certain rule set to 1.0 (the oracle always apply all rules exhaustively). Figure 2 shows the delta between averages of the number of queries when the probability for the oracle to apply its rules are set to 0.0 and the other values: 0.01, 0.5 and 1.0. The number of membership queries visibly increases, while the number of equivalence queries remains nearly the same. This is expected, since computing $\mathcal{O}$-essential inclusions from less informative counterexamples need more membership queries. Though, since the learner indeed computes $\mathcal{O}$-essential counterexamples, there is only a small impact on the number of equivalence queries.

5 Conclusion

We presented the learning algorithm underpinning ExactLearner. As future work, we plan to extend our algorithm in an ontology-based data access setting and adopt the Probably Approximately Correct learning model extended with membership queries, so that our algorithm can also run without the teacher.

Acknowledgements. Konev was supported by the EPSRC project EP/H043594/1. We would like to thank Frank Wolter for fruitful discussions and Liyi Zhao for her contribution to an earlier version of ExactLearner.

References

1. Bogdan Alexe, Balder ten Cate, Phokion G. Kolaitis, and Wang Chiew Tan. EIRENE: interactive design and refinement of schema mappings via data examples. *PVLDB*, 4(12):1414–1417, 2011.
2. Marta Arias, Roni Khardon, and Jérôme Maloberti. Learning horn expressions with LOGAN-H. *Journal of Machine Learning Research*, 8:549–587, 2007.
3. F. Baader, D. Calvanese, D. McGuiness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2003.
4. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, pages 364–369. Professional Book Center, 2005.
5. Ricardo Duarte, Boris Konev, and Ana Ozaki. Exact learning of $\mathcal{EL}$ ontologies. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 16th International Conference, KR 2018, Tempe, USA, October 30-November 2, 2018*, 2018.
6. Boris Konev, Carsten Lutz, Ana Ozaki, and Frank Wolter. Exact learning of lightweight description logic ontologies. *Journal of Machine Learning Research*, 18(201):1–63, 2018.
7. Jens Lehmann. DL-Learner: Learning concepts in description logics. *Journal of Machine Learning Research*, 10:2639–2642, 2009.
8. Oxford. Information systems group ontologies. Retrieved from `https://www.cs.ox.ac.uk/isg/ontologies/`. Accessed: 18 May 2018.