Learning Ontologies with Epistemic Reasoning: The $\mathcal{EL}$ Case

Ana Ozaki$^{(\boxtimes)}$ and Nicolas Troquard

KRDB Research Centre, Free University of Bozen-Bolzano, Bolzano, Italy
{ana.ozaki,nicolas.troquard}@unibz.it

Abstract. We investigate the problem of learning description logic ontologies *from entailments* via queries, using epistemic reasoning. We introduce a new learning model consisting of *epistemic membership and example queries* and show that polynomial learnability in this model coincides with polynomial learnability in Angluin's exact learning model with membership and equivalence queries. We then instantiate our learning framework to $\mathcal{EL}$ and show some complexity results for an epistemic extension of $\mathcal{EL}$ where epistemic operators can be applied over the axioms. Finally, we transfer known results for $\mathcal{EL}$ ontologies and its fragments to our learning model based on epistemic reasoning.

Keywords: Exact learning · Epistemic logic · Description logic

1 Introduction

Description logics (DL) balance expressivity and complexity of reasoning, resulting in a family of formalisms which can capture conceptual knowledge in various domains [3].[1] One of the most popular ontology languages, featuring polynomial time complexity of reasoning tasks such as entailment, is $\mathcal{EL}$ [2], which allows conjunctions ($\sqcap$) and existential restrictions ($\exists$) in its concept expressions but disallows negations of concepts. The following example illustrates $\mathcal{EL}$ ontologies (Sect. 4) representing knowledge of experts in different domains.

Example 1. Ana knows about Brazilian music (BM) and Nicolas is an expert in French cuisine (FC). We can represent some parts of their knowledge as follows.

$$\begin{array}{ll}
\mathcal{O}^{\mathsf{BM}}_{\mathsf{Ana}} = \{\mathsf{BrazilianSinger}(\mathsf{Caetano}) & \mathcal{O}^{\mathsf{FC}}_{\mathsf{Nicolas}} = \{\mathsf{FrenchChef}(\mathsf{Soyer}) \\
\mathsf{BossaNova} \sqsubseteq \mathsf{BrazilianMusicStyle} & \mathsf{Crepe} \sqsubseteq \exists \mathsf{contains}.\mathsf{Flour} \\
\mathsf{ViolaBuriti} \sqsubseteq \exists \mathsf{madeFrom}.\mathsf{Buriti}\} & \mathsf{Crepe} \sqcap \exists \mathsf{contains}.\mathsf{Sugar} \sqsubseteq \mathsf{Dessert}\}
\end{array}$$

Naturally, domain experts—humans, or artificial entities with complex neural networks—cannot be expected to be able to easily transfer their knowledge. However, when specific questions about the domain are posed, e.g., 'is Bossa

[1] The technical report [17] is a more complete version of this paper.

F. Calimeri et al. (Eds.): JELIA 2019, LNAI 11468, pp. 418–433, 2019.
https://doi.org/10.1007/978-3-030-19570-0_27

Nova a Brazilian music style?', an expert in the domain of Brazilian music can accurately decide whether such statement holds or not. So the ontology representation of the knowledge of an expert, even though *not directly accessible*, can be learned via a trial and error process in which individuals or machines, generically called *agents*, communicate with each other, in order to learn from the other agents. We assume that the *target* domain of interest to be learned is represented by a logical theory formulated in an ontology language.

In computational learning theory, a classical communication protocol coming from the exact learning model [1] is based on questions of two types: *membership* and *equivalence* queries. In a learning from entailments setting [11], these questions can be described as follows. Membership queries correspond to asking whether a certain statement formulated as a logical sentence follows from the target. Equivalence queries correspond to asking whether a certain logical theory, called *hypothesis*, precisely describes the target. If there are wrong or missing statements in the hypothesis, a statement illustrating the imprecision should be returned to the agent playing the role of the *learner*.

Example 2. Assume Ana wants to learn about $\mathsf{French\ cuisine}$. She asks $\mathsf{Nicolas}$ whether it follows from his knowledge that 'every crepe is a dessert', in symbols, 'does $\mathcal{O}^{\mathsf{FC}}_{\mathsf{Nicolas}} \models \mathsf{Crepe} \sqsubseteq \mathsf{Dessert}$?', which the answer in this case is 'no', since only those which contain sugar are considered desserts. To receive new statements about $\mathsf{French\ cuisine}$ from $\mathsf{Nicolas}$, Ana needs to pose equivalence queries, in symbols 'does $\mathcal{O}^{\mathsf{FC}}_{\mathsf{Ana}} \equiv \mathcal{O}^{\mathsf{FC}}_{\mathsf{Nicolas}}$?'. Each time she poses this type of questions, her best interest is to tell him everything she knows about $\mathsf{French\ cuisine}$.

One of the main difficulties in implementing this protocol in practice [16, p. 297] comes from the putative unreasonableness of equivalence queries. Whenever a learner poses an equivalence query, the expert playing the role of an *oracle* needs to evaluate the whole hypothesis and decide whether or not it is equivalent to the target. If not, then the oracle returns a statement in the logical difference between the hypothesis and the target. One way out of this difficulty is hinted to us by a simple observation: during interactive communication among agents, not only domain knowledge is exchanged and acquired but also second-order knowledge, which is the knowledge of what is known by the other agents.

Example 3. When Ana and $\mathsf{Nicolas}$ communicate, they know what they have already told to each other. If Ana tells $\mathsf{Nicolas}$ that 'Buriti is a Brazilian tree' ($\mathsf{Nicolas}$ now *knows* this statement, in symbols, $\mathbf{K}_{\mathsf{Nicolas}}(\mathsf{Buriti} \sqsubseteq \mathsf{BrazilianTree})$) and that 'Viola de Buriti is made from Buriti' ($\mathbf{K}_{\mathsf{Nicolas}}(\mathsf{ViolaBuriti} \sqsubseteq \exists \mathsf{madeFrom}.\mathsf{Buriti})$) she does not need to tell him that 'Viola de Buriti is made from a Brazilian tree' (as it follows that $\mathbf{K}_{\mathsf{Nicolas}}(\mathsf{ViolaBuriti} \sqsubseteq \exists \mathsf{madeFrom}.\mathsf{BrazilianTree})$, see Sect. 4).

In this paper, we thus propose a new and more realistic learning model. It is based on a protocol which takes into account what is known by the agents, either because a statement was explicitly communicated or because it is a logical consequence of previous statements given during their interaction. Our protocol

$$\text{EPISTEMIC}[\mathsf{MEM},\mathsf{EX}] \overset{\text{Th. 3}}{=} \text{EXACT}[\mathsf{MEM},\mathsf{EQ}] \overset{\text{[16] and [5]}}{\subset} \text{PAC}[\mathsf{MEM}]$$

$$\text{EPISTEMIC}[\mathsf{EX}] \overset{\text{Th. 1 and Th. 2}}{\subset} \text{EXACT}[\mathsf{EQ}] \overset{\text{[1] and [5]}}{\subset} \text{PAC}$$

Fig. 1. Polynomial learnability. Each class denotes the set of frameworks that are polynomial query learnable in the corresponding learning model. MEM, EQ and EX stand for membership, equivalence, and example queries respectively.

is based on queries of two types. The first is an epistemic version of membership queries where the oracle 'remembers' those membership queries whose reply was 'yes'. We call the second type *example queries*. When asked an example query, the oracle answers a statement which follows from its knowledge but does not follow from its knowledge about what the learner knows. The oracle also 'remembers' that the statements given are now known by the learner.

The first contribution of this work is the introduction of the learning model based on epistemic reasoning, which we call *epistemic learning model*, and an analysis of its 'power' in comparison with the exact learning model (Fig. 1). The second is an instantiation to the $\mathcal{EL}$ ontology language, whose polynomial learnability has been investigated in the exact learning model [10, 14, 15].

In more details, the epistemic learning model is introduced in Sect. 2. We then establish in Sect. 3 that polynomial learnability is strictly harder in the epistemic model without (an epistemic version of) membership queries (Theorems 1 and 2). Nonetheless, it coincides with polynomial learnability in the exact learning model if both types of queries are allowed (Theorem 3). Since it is known that polynomial learnability in the exact learning model with only equivalence queries implies polynomial learnability in the classical probably approximately correct learning model (PAC) [1, 18], it follows that polynomial learnability in the epistemic learning model with only example queries implies polynomial learnability in the PAC learning model. The same relationship holds for the case where we have (an epistemic version of) membership queries in the epistemic model and the PAC model also allows membership queries. We also show in Sect. 4 some complexity results for an epistemic extension of $\mathcal{EL}$, which we call $\mathcal{ELK}$. In particular, we show that satisfiability in $\mathcal{ELK}$, which includes Boolean combinations of $\mathcal{EL}$ axioms, does not increase the NP-completeness of propositional logic (Theorem 4). We then show that a fragment of $\mathcal{ELK}$ features PTime complexity for the satisfiability and entailment problems (Theorem 5), as in $\mathcal{EL}$ [2]. Crucially, it captures the epistemic reasoning that the agent playing the role of the oracle needs to perform. Finally, in Sect. 5 we transfer known results [10, 14] for $\mathcal{EL}$ in the exact learning model to the epistemic learning model.

2 Learning with Epistemic Reasoning

We first define the epistemic extension of a description logic $\mathcal{L}$, which is often a notation variant of a fragment of first-order logic or propositional logic. The epistemic extension of $\mathcal{L}$ allows expressions of the form 'agent i knows some axiom of $\mathcal{L}$'. We then use the epistemic extension of a logic to define a learning framework based on epistemic reasoning.

2.1 The Epistemic Extension of $\mathcal{L}$

In the following, we formalise the epistemic extension $\mathcal{LK}$ of a description logic $\mathcal{L}$. Our notation and definitions can be easily adapted to the case $\mathcal{L}$ is a (fragment of) first-order or propositional logic. Assume symbols of $\mathcal{L}$ are taken from pairwise disjoint and countably infinite sets of concept, role and individual names $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$, respectively. Let $\mathbf{A}$ be a set of agents. An $\mathcal{LK}$ axiom is an expression of the form: $\beta ::= \alpha \mid \mathbf{K}_i\beta$ where α is an $\mathcal{L}$ formula and $i \in \mathbf{A}$. $\mathcal{LK}$ formulas φ, ψ are expressions of the form: $\varphi ::= \beta \mid \neg\varphi \mid \varphi \wedge \psi$ where β is an $\mathcal{LK}$ axiom.

An $\mathcal{L}$ *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ over a non-empty set $\Delta^{\mathcal{I}}$, called the *domain*, defines an *interpretation function* $\cdot^{\mathcal{I}}$ that maps each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each individual name $a \in \mathsf{N_I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The extension of the mapping $\cdot^{\mathcal{I}}$ from concept names to $\mathcal{L}$ complex concept expressions depends on the precise definition of $\mathcal{L}$. We write $\models_{\mathcal{L}}$ and $\equiv_{\mathcal{L}}$ to denote the entailment and equivalence relations for $\mathcal{L}$ formulas, respectively.

An $\mathcal{LK}$ interpretation $\mathfrak{I} = (\mathcal{W}, \{\mathcal{R}_i\}_{i\in\mathbf{A}})$ consists of a set $\mathcal{W}$ of $\mathcal{L}$ interpretations and a set of accessibility relations $\mathcal{R}_i$ on $\mathcal{W}$, one for each agent $i \in \mathbf{A}$. We assume that the relations $\mathcal{R}_i$ are equivalence relations. A pointed $\mathcal{LK}$ interpretation is a pair $(\mathfrak{I}, \mathcal{I})$ where $\mathfrak{I} = (\mathcal{W}, \{\mathcal{R}_i\}_{i\in\mathbf{A}})$ is an $\mathcal{LK}$ interpretation and $\mathcal{I}$ is an element of $\mathcal{W}$. The *entailment* relation $\models_{\mathcal{LK}}$ of an $\mathcal{LK}$ formula φ in $\mathfrak{I} = (\mathcal{W}, \{\mathcal{R}_i\}_{i\in\mathbf{A}})$ pointed at $\mathcal{I} \in \mathcal{W}$ is inductively defined (for simplicity, we may omit the subscript $_{\mathcal{LK}}$ from $\models_{\mathcal{LK}}$):

$$\begin{array}{ll} \mathfrak{I}, \mathcal{I} \models \alpha \text{ iff } \mathcal{I} \models_{\mathcal{L}} \alpha & \mathfrak{I}, \mathcal{I} \models \phi \wedge \psi \text{ iff } \mathfrak{I}, \mathcal{I} \models \phi \text{ and } \mathfrak{I}, \mathcal{I} \models \psi \\ \mathfrak{I}, \mathcal{I} \models \neg\phi \text{ iff not } \mathfrak{I}, \mathcal{I} \models \phi & \mathfrak{I}, \mathcal{I} \models \mathbf{K}_i\beta \text{ iff } \forall(\mathcal{I}, \mathcal{J}) \in \mathcal{R}_i,\ \mathcal{J} \models \beta. \end{array}$$

An $\mathcal{LK}$ formula φ *entails* an $\mathcal{LK}$ formula ψ, written $\varphi \models \psi$, iff for all pointed $\mathcal{LK}$ interpretations $(\mathfrak{I}, \mathcal{I})$, $\mathfrak{I}, \mathcal{I} \models \varphi$ implies $\mathfrak{I}, \mathcal{I} \models \psi$. An $\mathcal{LK}$ formula φ is *equivalent* to an $\mathcal{LK}$ formula ψ, written $\varphi \equiv \psi$ (we may omit $_{\mathcal{LK}}$ from $\equiv_{\mathcal{LK}}$), iff $\varphi \models \psi$ and $\psi \models \varphi$. We use the notion of a set of formulas and the conjunction of its elements interchangeably. The *size* of a formula or an interpretation X, denoted $|X|$, is the length of the string that represents it, where concept, role and individual names and domain elements are considered to be of length 1.

2.2 A Learning Model Based on Epistemic Reasoning

We first adapt the exact learning model with membership and equivalence queries to a multi-agent setting. We then introduce the epistemic learning model in a multi-agent setting and provide complexity notions for these models.

We introduce basic notions for the definition of a learning framework and the learning problem via queries [1], adapted to a *learning from entailments* setting [11] with multiple agents. A *learning (from entailments) framework* $\mathfrak{F}$ is a pair (X, L), where X is a set of *examples* (also called *domain* or *instance space*), and L is a set of *formulas* of a description logic $\mathcal{L}$. We say that $x \in X$ is a *positive example* for $l \in L$ if $l \models_{\mathcal{L}} x$ and a *negative example* for l if $l \not\models_{\mathcal{L}} x$. A *counterexample* x for $l \in L$ and $h \in L$ is either a *positive example* for l such that $h \not\models_{\mathcal{L}} x$ or a *negative example* for l such that $h \models_{\mathcal{L}} x$. A *multi-agent learning framework* $\mathfrak{F}(\mathbf{A})$ is a set $\{\mathfrak{F}_i = (X_i, L_i) \mid i \in \mathbf{A}\}$ of learning frameworks.

We first provide a formal definition of the exact learning model, based on membership and equivalence queries, and then we introduce the epistemic learning model, with example and epistemic membership queries. Let $\mathfrak{F}(\mathbf{A})$ be a multi-agent learning framework. Each $i \in \mathbf{A}$ aims at learning a *target* formula $l_j \in L_j$ of a description logic $\mathcal{L}$ of each other agent $j \neq i \in \mathbf{A}$ by posing them queries.

Definition 1 (Membership query). *For every $i \in \mathbf{A}$ and every $l_i \in L_i$, let $\mathsf{MEM}_{\mathfrak{F}(\mathbf{A}),l_i}$ be an oracle that takes as input $x \in X_i$ and outputs 'yes' if $l_i \models_{\mathcal{L}} x$ and 'no' otherwise. A* membership query to agent *$i \in \mathbf{A}$ is a call to $\mathsf{MEM}_{\mathfrak{F}(\mathbf{A}),l_i}$.*

Definition 2 (Equivalence query). *For every $i \in \mathbf{A}$ and every $l_i \in L_i$, we denote by $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$ an oracle that takes as input a* hypothesis *formula of a description logic $h \in L_i$ and returns 'yes' if $h \equiv_{\mathcal{L}} l_i$ and a counterexample for l_i and h otherwise. An* equivalence query to agent *$i \in \mathbf{A}$ is a call to $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$. There is no assumption about which counterexample is returned by $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$.*

In this work, we introduce *example* queries, where an agent $i \in \mathbf{A}$ can ask an agent $j \in \mathbf{A}$ to only provide examples which are not logical consequences of what they have already communicated. Intuitively, if agent j returns x to agent i in a language $\mathcal{L}$ and $x \models_{\mathcal{L}} y$ then agent i knows y, in symbols, $\mathbf{K}_i y$. Since agent j returned this example to agent i, the axiom $\mathbf{K}_i y$ is part of the logical theory representing the knowledge of agent j, so agent j acquires knowledge of what is known by agent i as they communicate. We use example queries in combination with an epistemic version of membership queries, called $\mathbf{K}$-*membership* queries. Given $i \in \mathbf{A}$, assume that L_i is a set of formulas of the logic $\mathcal{L}$ and denote by $L_i^{\mathbf{K}}$ the set of all formulas in the epistemic extension of $\mathcal{L}$, which, by definition of $\mathcal{LK}$, includes all $\mathcal{L}$ formulas. The target formula l_i is an element of L_i, however, the oracles for the example and $\mathbf{K}$-membership queries may add $\mathcal{LK}$ formulas to l_i. We denote by l_i^{k+1} the result of updating l_i^k upon receiving the k-th query, where $l_i^1 = l_i$. At all times X_i is a set of examples in $\mathcal{L}$ (not in $\mathcal{LK}$).

Definition 3 (K-membership query). *For every $i \in \mathbf{A}$ and every $l_i^k \in L_i^{\mathbf{K}}$, let $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ be an oracle that takes as input $x \in X_i$ and $j \in \mathbf{A}$, and, if $l_i^1 \models_{\mathcal{L}} x$, it outputs 'yes' and define $l_i^{k+1} := l_i^k \wedge \mathbf{K}_j x$[2].*

Otherwise it returns 'no' and defines $l_i^{k+1} := l_i^k$. The k-th $\mathbf{K}$-membership query to agent *$i \in \mathbf{A}$ is a call to $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F}(\mathbf{A}),l_i^k}$.*

[2] We may write l_i^k for the conjunction of its elements.

Definition 4 (Example query). *For every $i \in \mathbf{A}$ and every $l_i^k \in L_i^{\mathbf{K}}$, let $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ be an oracle that takes as input some $j \in \mathbf{A}$ and outputs $x \in X_i$ such that $l_i^1 \models_{\mathcal{L}} x$ but $l_i^k \not\models_{\mathcal{LK}} \mathbf{K}_j x$ if such x exists; or 'you finished', otherwise.*

Upon returning $x \in X_i$ such that $l_i^1 \models_{\mathcal{L}} x$ the oracle $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ defines $l_i^{k+1} := l_i^k \wedge \mathbf{K}_j x$. The k-th example query to agent *$i \in \mathbf{A}$ is a call to $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$.*

An *exact learning algorithm A_i* for $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ is a deterministic algorithm that takes no input, is allowed to make queries to $\mathsf{MEM}_{\mathfrak{F}(\mathbf{A}),l_i}$ and $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$ (without knowing what the target l_i to be learned is), and eventually halts and outputs some $h \in L_i$ with $h \equiv_{\mathcal{L}} l_i$. An *epistemic learning algorithm* for $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ is a deterministic algorithm that takes no input, is allowed to make queries to $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ and $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ (without knowing what the target l_i^1 to be learned is), and eventually halts after receiving 'you finished' from $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$.

We say that $\mathfrak{F}(\mathbf{A})$ is *exactly learnable* if there is an exact learning algorithm A_i for each $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ and that $\mathfrak{F}(\mathbf{A})$ is *polynomial query exactly learnable* if each $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ is exactly learnable by an algorithm A_i such that at every step the sum of the sizes of the inputs to queries made by A_i up to that step is bounded by a polynomial $p(|l_i|, |x|)$, where l_i is the target and $x \in X_i$ is the largest example seen so far by A_i. $\mathfrak{F}(\mathbf{A})$ is *polynomial time exactly learnable* if each $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ is exactly learnable by an algorithm A_i such that at every step (we count each call to an oracle as one step of computation) of computation the time used by A_i up to that step is bounded by a polynomial $p(|l_i|, |x|)$, where $l_i \in L_i$ is the target and $x \in X$ is the largest counterexample seen so far. We may also say that $\mathfrak{F}(\mathbf{A})$ is learnable in $O(|l_i|, |x|)$ many steps, following the same notion of polynomial time learnability, except that the number of steps is bounded by $O(|l_i|, |x|)$.

We say that $\mathfrak{F}(\mathbf{A})$ is *epistemically learnable* if there is an epistemic learning algorithm for each $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$. Polynomial query/time epistemic learnability is defined analogously, with $p(|l_i^1|, |x|)$ defined in terms of $|l_i^1|$ and $|x|$. Clearly, if a learning framework $\mathfrak{F}(\mathbf{A})$ is polynomial time exactly/epistemically learnable then it is also polynomial query exactly/epistemically learnable.

3 Epistemic and Exact Polynomial Learnability

In this section we confront polynomial query and polynomial time learnability in the exact and epistemic learning models. We start by considering the case where the learner is only allowed to pose one type of query. Clearly, polynomial (query/time) exact learnability with only membership queries coincides with polynomial epistemic learnability with only $\mathbf{K}$-membership queries. We now analyse polynomial learnability with equivalence queries only and example queries only. Our first result is that polynomial (query/time) learnability in the epistemic learning model implies polynomial learnability in exact learning model.

Theorem 1. *If a multi-agent learning framework is polynomial query (resp. time) epistemically learnable with only example queries then it is polynomial query (resp. time) exactly learnable with only equivalence queries.*

Proof. Assume $\mathfrak{F}(\mathbf{A})$ is polynomial query epistemically learnable with only example queries (the case of polynomial time epistemic learnability with only example queries can be similarly proved). For each $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ there is an epistemic learning algorithm A_i for $\mathfrak{F}_i$ with polynomial query complexity which only asks example queries. To construct an exact learning algorithm A'_i for $\mathfrak{F}_i$ which only asks equivalence queries using A_i, we define auxiliary sets $s_i^{\mathcal{K}}(k)$ and $s_i^{\mathcal{L}}(k)$ which will keep the information returned by $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$ up to the k-th query posed by a fixed but arbitrary agent in $\mathbf{A} \setminus \{i\}$ and agent i. We define $s_i^{\mathcal{K}}(1) = \emptyset$ and $s_i^{\mathcal{L}}(1) = \emptyset$.

- Whenever A_i poses an example query to agent $i \in \mathbf{A}$ (assume it is the k-th query), A'_i calls the oracle $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$ with $s_i^{\mathcal{L}}(k)$ as input. The oracle either returns 'yes' if $s_i^{\mathcal{L}}$ is equivalent to l_i or it returns some counterexample for l_i and $s_i^{\mathcal{L}}(k)$ (we may write $s_i^{\mathcal{L}}(k)$ to denote $\bigwedge_{\beta \in s_i^{\mathcal{L}}(k)} \beta$). Then A'_i adds $\mathbf{K}_j x$ to $s_i^{\mathcal{K}}(k)$ and x to $s_i^{\mathcal{L}}(k)$.

If $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$ returns 'yes' then algorithm A'_i converts it into 'you finished', as expected by algorithm A_i. We now argue that, for all $x \in X_i$ and all $k \geq 0$ such that $l_i \models_{\mathcal{L}} x$, we have that x is a (positive) counterexample for l_i and $s_i^{\mathcal{L}}(k)$ iff $l_i \wedge s_i^{\mathcal{K}}(k) \not\models_{\mathcal{LK}} \mathbf{K}_j\, x$. By definition of $s_i^{\mathcal{K}}(k)$ and $s_i^{\mathcal{L}}(k)$ and since l_i does not contain $\mathcal{LK}$ axioms, for all $x \in X_i$ and all $k \geq 0$, we have that $l_i \wedge s_i^{\mathcal{K}}(k) \models_{\mathcal{LK}} \mathbf{K}_j x$ iff $s_i^{\mathcal{L}}(k) \models_{\mathcal{L}} x$. By definition and construction of $s_i^{\mathcal{L}}(k)$, it follows that $l_i \models_{\mathcal{L}} s_i^{\mathcal{L}}(k)$. So $s_i^{\mathcal{L}}(k) \not\models_{\mathcal{L}} x$ iff $l_i \wedge s_i^{\mathcal{K}}(k) \not\models_{\mathcal{LK}} \mathbf{K}_j\, x$. Hence $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i}$ can simulate $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$, where k represents the number of calls to $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ posed so far by A_i. By definition of A_i, at every step, the sum of the sizes of the inputs to queries made by A_i up to that step is bounded by a polynomial $p(|l_i|, |x|)$, where l_i is the target and $x \in X_i$ is the largest counterexample seen so far by A_i.Then, for all $k \geq 0$, we have that $|s_i^{\mathcal{L}}(k)| \leq |s_i^{\mathcal{K}}(k)| \leq p(|l_i|, |x|)$. Since all responses to queries are as required by A_i, if A_i halts after polynomially many polynomial size queries, the same happens with A'_i, which returns a hypothesis $s_i^{\mathcal{L}}(k)$ equivalent to the target l_i, for some $k \leq p(|l_i|, |x|)$. □

The converse of Theorem 1 does not hold, as we show in the next theorem.

Theorem 2. *There is a multi-agent learning framework $\mathfrak{F}(\mathbf{A})$ such that $\mathfrak{F}(\mathbf{A})$ is polynomial time exactly learnable with only equivalence queries but not polynomial query (so, not polynomial time) epistemically learnable with only example queries.*

Proof. Consider the learning framework $\mathfrak{F} = (X, L)$ where X is the set of propositional formulas over the variables $\mathsf{Prop} = \{q, p, p_1^0, \ldots, p_n^0, p_1^1, \ldots, p_n^1\}$ and $L = \{\varphi \mid \varphi \in X, \varphi \equiv (p \rightarrow q)\}$ (where $\equiv$ denotes logical equivalence in propositional logic). So the target can only be a formula equivalent to $p \rightarrow q$. Now let

$\mathfrak{F}(\mathbf{A})$ be the set $\{\mathfrak{F}_i = (X, L) \mid i \in \mathbf{A}\}$, with all learning frameworks are equal to $\mathfrak{F}$ (this does not mean that the target is the same for all agents but that they are taken from the same set L). If L is a language which only contains propositional formulas equivalent to $p \rightarrow q$, an exact learning algorithm can learn the target with only one equivalence query, passing the hypothesis $\{p \rightarrow q\}$ as input. However, $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),\{p\rightarrow q\}}$ can return any of the exponentially many examples of the form $p \wedge (p_1^{\ell_1} \wedge \ldots \wedge p_n^{\ell_n}) \rightarrow q$, with $\ell_j \in \{0, 1\}$ and $j \in \{1, \ldots, n\}$. The example oracle can always provide an example which does not follow from its knowledge of what is known by the learner by taking a fresh binary sequence. Thus, there is no epistemic algorithm which can learn the target with polynomially many queries. □

Interestingly, if we consider both types of queries then polynomial exact learnability *coincides* with polynomial epistemic learnability.

Theorem 3. *Let $\mathfrak{F}(\mathbf{A})$ be a multi-agent learning framework. $\mathfrak{F}(\mathbf{A})$ is polynomial query (resp. time) exactly learnable if, and only if, $\mathfrak{F}(\mathbf{A})$ is polynomial query (resp. time) epistemically learnable.*

Proof. ($\Rightarrow$) In our proof we use polynomial query exact learnability, the argument for polynomial time exact learnability is analogous. Assume $\mathfrak{F}(\mathbf{A})$ is polynomial query exactly learnable. Then, for each $\mathfrak{F}_i \in \mathfrak{F}(\mathbf{A})$ there is an exact learning algorithm A_i for $\mathfrak{F}_i$. We construct an epistemic learning algorithm A'_i for $\mathfrak{F}_i$ using A_i as follows. Recall that we write l_i^k to denote the target l_i^k after the k-th query (Sect. 2.2).

- Whenever A_i poses a membership query to agent $i \in \mathbf{A}$ with $x \in X_i$ as input, A'_i calls $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ with x as input, where k represents the number of queries posed so far by A'_i.
- Whenever A_i poses an equivalence query to agent $i \in \mathbf{A}$ with a hypothesis h as input, we have that, for each $x \in h$, A'_i calls $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ with x as input (and k is incremented). Then, the algorithm calls the oracle $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$.

$\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F}(\mathbf{A}),l_i}$ behaves as it is required by algorithm A_i to learn $\mathfrak{F}_i$. We show that whenever $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ returns some $x \in X_i$ we have that x is a counterexample for l_i^1 and h, where h is the input of the equivalence query posed by A_i. By definition of A_i, at every step, the sum of the sizes of the inputs to queries made by A_i up to that step is bounded by a polynomial $p(|l_i^1|, |x|)$, where l_i^1 is the target and $x \in X_i$ is the largest example seen so far by A_i. Let h^ℓ denote the input to the ℓ-th equivalence query posed by A_i. For all $\ell > 0$, we have that $|h^\ell| \leq p(|l_i^1|, |x|)$. The fact that x is a counterexample for l_i^1 and h^ℓ follows from the definition of A'_i, which poses membership queries for each $x \in h^\ell$, ensuring that l_i^k is updated with $\mathbf{K}_j x$ after each query. Hence $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^k}$ returns counterexamples for l_i^1 and h^ℓ (if they exist), as $\mathsf{EQ}_{\mathfrak{F}(\mathbf{A}),l_i^1}$. Since A_i poses only polynomially many queries, ℓ is bounded by $p(|l_i^1|, |x|)$. So the sum of the sizes of the inputs to queries made by the epistemic learning algorithm A'_i simulating A_i is quadratic in $p(|l_i^1|, |x|)$.

All in all, since all responses to queries are as required by A_i, if A_i halts and outputs some $h \in L_i$ with $h \equiv_{\mathcal{L}} l_i^1$ (with h the input to the last equivalence query) after polynomially many polynomial size queries, we have that $\mathsf{EX}_{\mathfrak{F}(\mathbf{A}),l_i^1}$ is forced to return 'you finished' and so A_i' also halts after polynomially many polynomial size queries. The ($\Leftarrow$) direction is similar to the proof of Theorem 1, except that we now also have (**K**-)membership queries. □

4 The Epistemic $\mathcal{EL}$ Description Logic

To instantiate the multi-agent epistemic learning problem to the $\mathcal{EL}$ case, in Sect. 5, we define and study in this section the epistemic extension of $\mathcal{EL}$, called $\mathcal{ELK}$. We present $\mathcal{EL}$ [3] in Sect. 4.1. $\mathcal{ELK}$ is the instantiation of $\mathcal{LK}$ presented in Sect. 2.1 with the logic $\mathcal{EL}$. We establish the complexity of the satisfiability problem for $\mathcal{ELK}$ in Sect. 4.2 and of one of its fragments in Sect. 4.3.

We showed in Sect. 3 that example queries give strictly less power to the learner than equivalence queries. We also argued, quite informally so far, that example queries are less demanding on the oracle than equivalence queries. Instead of deciding whether two ontologies are equivalent, and then providing a counterexample when it is not the case, the oracle only needs to reason about what they know about the knowledge of the learner. Yet, we did not say anything about the actual complexity of the epistemic reasoning involved in example queries. If reasoning about the knowledge of the learner is harder than evaluating the equivalence of two ontologies, then the advantage of example queries for the oracle would be moot. We show that indeed the epistemic reasoning that the oracle needs to perform is in PTime (Theorem 5). So, the oracle's benefit from example queries over equivalence queries is a net benefit.

4.1 $\mathcal{EL}$: Syntax, Semantics, and Complexity

$\mathcal{EL}$ concepts C, D are expressions of the form: $C, D ::= \top \mid A \mid \exists r.C \mid C \sqcap D$ where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. An *inclusion* is an expression of the form $C \sqsubseteq D$ where C, D are $\mathcal{EL}$ concept expressions; and an *assertion* is of the form $A(a)$ or $r(a, b)$ with $a, b \in \mathsf{N_I}$, $A \in \mathsf{N_C}$, and $r \in \mathsf{N_R}$. An $\mathcal{EL}$ *axiom* is an inclusion or an assertion. An $\mathcal{EL}$ *formula*[3] is an expression of the form $\alpha ::= \mathsf{a} \mid \neg\alpha \mid \alpha \wedge \alpha$ where a is an $\mathcal{EL}$ axiom. An $\mathcal{EL}$ literal is an $\mathcal{EL}$ axiom or its negation. The semantics of $\mathcal{EL}$ is given by $\mathcal{L}$ interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as defined in Sect. 2.1, considering $\mathcal{L} = \mathcal{EL}$. We extend the mapping $\cdot^{\mathcal{I}}$ for $\mathcal{EL}$ complex concept expressions as follows:

$$\top^{\mathcal{I}} := \Delta^{\mathcal{I}}, \qquad (C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}},$$
$$(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I}} : (d, e) \in r^{\mathcal{I}}\}.$$

[3] Typically an $\mathcal{EL}$ *ontology* is a set of $\mathcal{EL}$ axioms [3], and can also be seen as a conjunction of positive $\mathcal{EL}$ axioms. Here we also consider $\mathcal{EL}$ *formulas*, where we allow negations and conjunctions over the axioms.

We now define the entailment relation $\models_{\mathcal{EL}}$ for $\mathcal{EL}$ formulas. Given an $\mathcal{EL}$ interpretation $\mathcal{I}$ and an $\mathcal{EL}$ axiom (which can be an inclusion or an assertion, as above) we define: $\mathcal{I} \models_{\mathcal{EL}} C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; $\mathcal{I} \models_{\mathcal{EL}} A(a)$ iff $a^{\mathcal{I}} \in A^{\mathcal{I}}$; and $\mathcal{I} \models_{\mathcal{EL}} r(a,b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. We inductively extend the relation $\models_{\mathcal{EL}}$ to $\mathcal{EL}$ formulas as in Sect. 2.1: $\mathcal{I} \models_{\mathcal{EL}} \varphi \wedge \psi$ iff $\mathcal{I} \models_{\mathcal{EL}} \varphi$ and $\mathcal{I} \models_{\mathcal{EL}} \psi$; and $\mathcal{I} \models_{\mathcal{EL}} \neg\varphi$ iff not $\mathcal{I} \models_{\mathcal{EL}} \varphi$. In our proofs, we use the following result.

Lemma 1. *Satisfiability of a conjunction of $\mathcal{EL}$ literals is* PTime-*complete* [6].

We establish in Sect. 4.2 that reasoning about $\mathcal{ELK}$ formulas is NP-complete, just like reasoning about $\mathcal{EL}$ formulas. We note that $\mathcal{EL}(\mathcal{K})$ formulas allow arbitrary Boolean combinations of $\mathcal{EL}(\mathcal{K})$ axioms, hence the contrast with the PTime complexity of entailment from an $\mathcal{EL}$ ontology [2]. In Sect. 4.3 we show that reasoning about $\mathcal{ELK}$ restricted to conjunctions of literals is in PTime.

4.2 Reasoning in $\mathcal{ELK}$

Here we study the complexity of the satisfiability problem in $\mathcal{ELK}$. Our combination of epistemic logic and description logic is orthogonal to the work by De Giacomo et al. [8]: while our epistemic operators are over $\mathcal{EL}$ formulas, the epistemic operators of the mentioned work are over concepts and roles. For instance, there, $\mathbf{K}$ $\mathsf{FrenchChef}$ denotes the concept of *known* French chefs. Here, $\mathcal{ELK}$ contains formulas such as $\mathbf{K}_i(\mathsf{FrenchChef}(\mathsf{Soyer})) \wedge \neg\mathbf{K}_i\mathbf{K}_j(\mathsf{Crepe} \sqsubseteq \exists\mathsf{contains}.\mathsf{Egg})$ indicating that agent i knows that Soyer is a French chef, but i does not know that j knows that crepes contain egg.

From the definition of the language of $\mathcal{LK}$ in Sect. 2.1, remember that the language of $\mathcal{ELK}$ does not admit alternating modalities; E.g., $\mathbf{K}_i\neg\mathbf{K}_j A(a)$ is not a formula of $\mathcal{ELK}$. It is rather easy to see that if there were no such syntactic restrictions, the satisfiability problem would turn out to be PSpace-complete. (We could reduce satisfiability and adapt the tableaus method of propositional $S5_n$ [12].) Instead, we establish that satisfiability in $\mathcal{ELK}$ is NP-complete.

The lower bound follows from NP-hardness for propositional logic. The following lemma is central for showing membership in NP. It is a consequence of the fact that $\mathcal{EL}$ and propositional logic have the polynomial size model property and that in $\mathcal{ELK}$ the satisfiability test can be separated into two independent tests: one for the DL dimension and one for the epistemic dimension (see [4,6]).

Lemma 2. *$\mathcal{ELK}$ enjoys the polynomial size model property.*

Since $\mathcal{ELK}$ formulas can be translated into first-order logic, for a fixed $\mathcal{ELK}$ formula φ, checking whether a polynomial size interpretation is a model of φ can be performed in NLogSpace. Thus, membership in NP is by the fact that, by Lemma 2, one can guess a polynomial size model (if one exists) and check that it is a model in NLogSpace $\subseteq$ PTime.

Theorem 4. *Satisfiability in $\mathcal{ELK}$ is* NP*-complete.*

4.3 Reasoning in Conjunctive $\mathcal{ELK}$

We conclude this section considering the satisfiability problem for *conjunctive* $\mathcal{ELK}$, defined as the fragment of $\mathcal{ELK}$ which only allows negations in front of $\mathcal{EL}$ axioms or in front of $\mathcal{ELK}$ axioms of the form $\mathbb{K}\alpha$, with α a conjunction of $\mathcal{EL}$ literals and $\mathbb{K}$ a non-empty sequence of epistemic operators. Formally, conjunctive $\mathcal{ELK}$ formulas φ are expressions of the form: $\varphi ::= \alpha \mid \beta \mid \neg\beta \mid \varphi \wedge \varphi$ with $\beta ::= \mathbf{K}_i\alpha \mid \mathbf{K}_i\beta$, and $\alpha ::= \mathsf{a} \mid \neg\mathsf{a} \mid \alpha \wedge \alpha$, where a is an $\mathcal{EL}$ axiom.

Algorithm 1. $SAT(\varphi)$, deciding the satisfiability of conjunctive $\mathcal{ELK}$ formulas

```
Input: A conjunctive $\mathcal{ELK}$ formula $\varphi$
Output: TRUE if $\varphi$ is satisfiable, and FALSE otherwise
if $\omega_0 \wedge \bigwedge\{\omega \mid \mathbb{K}_\sigma\omega \in \varphi^\flat\}$ is not $\mathcal{EL}$ satisfiable then
    return FALSE
end if
for $\neg\mathbb{K}_\sigma\omega \in \varphi^\flat$ do
    $\psi = \top \wedge \bigwedge\{\omega' \mid \mathbb{K}_{\sigma'}\omega' \in \varphi^\flat$, and $\sigma$ is a subword of $\sigma'\}$
    $MS = \{\psi \wedge \neg\beta \mid \beta$ is an $\mathcal{EL}$ literal in $\omega\}$
    if all conjunctions of $\mathcal{EL}$ literals in $MS$ are not $\mathcal{EL}$ satisfiable then
        return FALSE
    end if
end for
return TRUE
```

To establish the complexity of reasoning in conjunctive $\mathcal{ELK}$, we use the following notation. For every non-empty sequence $\sigma = a_1 \dots a_k \in \mathbf{A}^+$ of agents, we associate a sequence $\mathbb{K}_\sigma = \mathbf{K}_{a_1} \dots \mathbf{K}_{a_k}$ of epistemic operators. We write $\beta \in \psi$ if β is a conjunct occurring in ψ. We say that $\sigma' \in \mathbf{A}^+$ is a *subword* of $\sigma \in \mathbf{A}^+$ when σ' is the result of removing zero or more elements from σ (at any position of the sequence). Given a conjunctive $\mathcal{ELK}$ formula

$$\varphi = \omega_0 \wedge \mathbb{K}_{\sigma_1}\omega_{\sigma_1} \wedge \dots \wedge \mathbb{K}_{\sigma_n}\omega_{\sigma_n} \wedge \neg\mathbb{K}_{\sigma_{n+1}}\omega_{\sigma_{n+1}} \wedge \dots \wedge \neg\mathbb{K}_{\sigma_m}\omega_{\sigma_m}$$

where $\sigma_i \in \mathbf{A}^+$, for every $1 \leq i \leq m$, and each ω_i, with $0 \leq i \leq m$, is a conjunction of $\mathcal{EL}$ literals, we denote by $\varphi^\flat$ the formula resulting from exhaustively substituting in φ every adjacent repetitions $a \dots a$ of an agent a occurring in σ_i, $1 \leq i \leq m$, with a. (E.g., $a_1a_2a_2a_3a_2$ becomes $a_1a_2a_3a_2$.)

The following proposition is central to the correctness of Algorithm 1.

Proposition 1. *A conjunctive $\mathcal{ELK}$ formula φ is unsatisfiable iff at least one of the following properties holds:*

1. *$\omega_0 \wedge \bigwedge\{\omega \mid \mathbb{K}_\sigma\omega \in \varphi^\flat\}$ is not $\mathcal{EL}$ satisfiable;*
2. *there is $\neg\mathbb{K}_\sigma\omega \in \varphi^\flat$ such that $\neg\omega \wedge \bigwedge\{\omega' \mid \mathbb{K}_{\sigma'}\omega' \in \varphi^\flat$, and σ is a subword of $\sigma'\}$ is not $\mathcal{EL}$ satisfiable.*

Proof. For convenience, we introduce some additional notation. For every $\sigma = a_1 \dots a_k \in \mathbf{A}^*$ we note $\mathcal{R}_\sigma = \mathcal{R}_{a_1} \circ \dots \circ \mathcal{R}_{a_k}$ and $\mathbb{K}_\sigma = \mathbf{K}_{a_1} \dots \mathbf{K}_{a_k}$. The empty sequence is noted ϵ, and we have $\mathcal{R}_\epsilon = Id$, where Id is the identity relation, and $\mathbb{K}_\epsilon \omega = \omega$.

The two following properties, which are instrumental in the proof of the proposition, are simple consequences of well-known properties of the modal system $S5$ [13, p. 58]. We state them without proof.

Claim. The following holds.

i. If $\varphi^\flat$ is an $\mathcal{ELK}$ formula, φ and $\varphi^\flat$ are equivalent.
ii. Let $(\mathcal{W}, \{\mathcal{R}_i\}_{i \in \mathbf{A}})$ be an $\mathcal{ELK}$ interpretation. For all $\sigma \in \mathbf{A}^*$ and $\sigma' \in \mathbf{A}^*$, if σ is a subword of σ' then $\mathcal{R}_\sigma \subseteq \mathcal{R}_{\sigma'}$.

We are ready to prove the proposition. Since, by Point (i) of the Claim, $\varphi^\flat$ and φ are equisatisfiable (in fact equivalent), w.l.o.g., we prove it for $\varphi^\flat$.

($\Leftarrow$) Suppose (1) holds. Since $\mathcal{R}_i$ is reflexive for all $i \in \mathbf{A}$, every model satisfying $\varphi^\flat$ must satisfy $\omega_0 \wedge \bigwedge\{\omega \mid \mathbb{K}_\sigma \omega \in \varphi^\flat\}$. Since it is not $\mathcal{EL}$ satisfiable, there cannot be an $\mathcal{ELK}$ interpretation satisfying $\varphi^\flat$ either. Suppose (2) holds. For some $\neg\mathbb{K}_\sigma \omega \in \varphi^\flat$, we have that $\psi = \neg\omega \wedge \bigwedge\{\omega' \mid \mathbb{K}_{\sigma'}\omega' \in \varphi^\flat$, and σ is a subword of $\sigma'\}$ is not $\mathcal{EL}$ satisfiable. Suppose towards contradiction that there exist an $\mathcal{ELK}$ interpretation $\mathfrak{I} = (\mathcal{W}, \{\mathcal{R}_i\}_{i \in \mathbf{A}})$ and an $\mathcal{EL}$ interpretation $\mathcal{J} \in \mathcal{W}$ such that $\mathfrak{I}, \mathcal{J} \models \varphi^\flat$. It means that $\mathfrak{I}, \mathcal{J} \models \neg\mathbb{K}_\sigma \omega$, that is, there is an $\mathcal{EL}$ interpretation $\mathcal{J}$ such that $(\mathcal{J}, \mathcal{J}') \in \mathcal{R}_\sigma$ and $\mathfrak{I}, \mathcal{J}' \models \neg\omega$. By Point (ii) of the Claim, for every $\mathbb{K}_{\sigma'}\omega' \in \varphi^\flat$, if σ is a subword of σ', then $\mathcal{R}_\sigma \subseteq \mathcal{R}_{\sigma'}$. Hence, $\mathfrak{I}, \mathcal{J}' \models \psi$, which is a contradiction as ψ is not $\mathcal{EL}$ satisfiable.

($\Rightarrow$) Assume that none of (1) and (2) hold. We must show that $\varphi^\flat$ is satisfiable. It suffices to build an $\mathcal{ELK}$ interpretation $\mathfrak{I} = (\mathcal{W}, \{\mathcal{R}_i\}_{i \in \mathbf{A}})$ for $\varphi^\flat$. $\mathcal{W}$ contains an $\mathcal{EL}$ interpretation $\mathcal{J}_0$ satisfying $\omega_0 \wedge \bigwedge\{\omega \mid \mathbb{K}_\sigma \omega \in \varphi^\flat\}$. Such an interpretation exists because (1) does not hold. For each $\neg\mathbb{K}_\sigma \omega \in \varphi^\flat$, where $\sigma = a_1 \dots a_k$, $\mathcal{W}$ also contains an interpretation $\mathcal{J}_k^\sigma$ satisfying $\neg\omega \wedge \bigwedge\{\omega' \mid \mathbb{K}_{\sigma'}\omega' \in \varphi^\flat$, and σ is a subword of $\sigma'\}$. Such an interpretation exists because (2) does not hold. Still for each $\neg\mathbb{K}_\sigma \omega \in \varphi^\flat$, $\mathcal{W}$, where $\sigma = a_1 \ \dots a_k$, for each $1 \leq i < k$, $\mathcal{W}$ also contains an interpretation $\mathcal{J}_i^\sigma$ satisfying $\bigwedge\{\omega' \mid \mathbb{K}_{\sigma'}\omega' \in \varphi^\flat$, and $a_1 \dots a_i$ is a subword of $\sigma'\}$. Such interpretations exist because (1) does not hold. $\mathcal{W}$ does not contain any more $\mathcal{EL}$ interpretations. We turn to the construction of $\mathcal{R}$. For every $\neg\mathbb{K}_\sigma \omega \in \varphi^\flat$, where $\sigma = a_1 \dots a_k$, for every $1 \leq i \leq k$, let $(\mathcal{J}_{i-1}^\sigma, \mathcal{J}_i^\sigma) \in \mathcal{R}'_{a_i}$, where $\mathcal{J}_0^\sigma = \mathcal{J}_0$. For every $i \in \mathbf{A}$, let $\mathcal{R}_i$ be the equivalence closure of $\mathcal{R}'_i$. It is routine to check that $\mathfrak{I}, \mathcal{J}_0 \models \varphi^\flat$. □

Proposition 1 suggests that the satisfiability of conjunctive $\mathcal{ELK}$ formulas can be reduced to checking the satisfiability of a few conjunctions of $\mathcal{EL}$ literals. We are finally ready to prove the complexity of deciding whether a conjunctive $\mathcal{ELK}$ formula is satisfiable.

Theorem 5. *Satisfiability in conjunctive $\mathcal{ELK}$ is* PTIME*-complete.*

Proof. Consider Algorithm 1. The conjunctive $\mathcal{ELK}$ formula φ is satisfiable iff $SAT(\varphi)$ returns TRUE. The correctness of the algorithm follows immediately from Proposition 1. It suffices to observe that Lines 5–9 check the unsatisfiability of an $\mathcal{EL}$ formula $\neg\omega \wedge \psi$ where ω and ψ are two of conjunctions of $\mathcal{EL}$ literals ($\neg\omega \wedge \psi$ is *not* a conjunction of $\mathcal{EL}$ literals, unless ω contains only one literal) by checking the unsatisfiability of as many conjunctions of $\mathcal{EL}$ literals $\neg\beta \wedge \psi$ as there are literals β in ω. A simple analysis shows that the algorithm runs in time polynomial in the size of φ, with a polynomial number of calls to a procedure for checking the unsatisfiability of conjunctions of $\mathcal{EL}$ literals. By Lemma 1, each of these checks can be done in polynomial time. Membership in PTIME follows. □

Algorithm 2. Adaptation of the learning algorithm for $\mathcal{EL}_{\mathsf{lhs,rhs}}$ [10]

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 $\mathcal{O} \equiv_{\mathcal{EL}} \mathcal{H}$

```
 1: $\mathcal{H} := \{\mathsf{a} \mid \mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F},\mathcal{O}^k}(\mathsf{a})=$‘yes’, $\mathsf{a}$ is a $\Sigma_\mathcal{O}$-assertion or $\mathsf{a} = A \sqsubseteq B,\ A, B \in \Sigma_\mathcal{O}\}$
 2: while $\mathsf{EX}_{\mathfrak{F},\mathcal{O}^k} \neq$ ‘you finished’ do
 3:     Let $C \sqsubseteq D$ be the returned positive example for $\mathcal{O}$
 4:     Compute, with $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F},\mathcal{O}^k}$, $C' \sqsubseteq D'$ such that $C'$ or $D'$ in $\Sigma_\mathcal{O} \cap \mathsf{N_C}$
 5:     if $C' \in \Sigma_\mathcal{O} \cap \mathsf{N_C}$ then
 6:         Compute with $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F},\mathcal{O}^k}$ a right $\mathcal{O}$-essential $\mathsf{a}$ from $C' \sqsubseteq D' \sqcap \bigsqcap_{C' \sqsubseteq F' \in \mathcal{H}} F'$
 7:     else
 8:         Compute with $\mathsf{MEM}^{\mathbf{K}}_{\mathfrak{F},\mathcal{O}^k}$ a left $\mathcal{O}$-essential $\mathsf{a}$ from $C' \sqsubseteq D'$
 9:     end if
10:     Add $\mathsf{a}$ to $\mathcal{H}$
11: end while
12: return $\mathcal{H}$
```

5 Learning $\mathcal{EL}$ with Epistemic Reasoning

It is known that $\mathcal{EL}$ ontologies are not polynomial query exactly learnable, while the fragments of $\mathcal{EL}$ restricting one of the sides of inclusions to be a concept name, namely $\mathcal{EL}_{\mathsf{lhs}}$ and $\mathcal{EL}_{\mathsf{rhs}}$, are exactly learnable in polynomial time [14]. In this section, we transfer results known for $\mathcal{EL}$ and its fragments to our learning model. Our results are for learning frameworks where the learning language is the same for all agents. That is, we deal with the special case of a multi-agent learning framework $\mathfrak{F}(\mathbf{A}) = \{\mathfrak{F}_i = (X_i, L_i) \mid i \in \mathbf{A}\}$ where all formulas in all L_i are from a DL $\mathcal{L}$, denoted $\mathfrak{F}(\mathcal{L}, \mathbf{A})$. Theorem 6 is a consequence of Theorem 3 and complexity results for $\mathcal{EL}$ and its fragments in the exact learning model [14].

Theorem 6. *The learning framework $\mathfrak{F}(\mathcal{EL}, \mathbf{A})$ is not polynomial query epistemically learnable. The learning frameworks $\mathfrak{F}(\mathcal{EL}_{\mathsf{lhs}}, \mathbf{A})$ and $\mathfrak{F}(\mathcal{EL}_{\mathsf{rhs}}, \mathbf{A})$ are polynomial time epistemically learnable.*

The hardness result for $\mathcal{EL}$ holds even for the fragment of $\mathcal{EL}$ ontologies defined as the union of $\mathcal{EL}_{\mathsf{lhs}}$ and $\mathcal{EL}_{\mathsf{rhs}}$, that is, in a named form where at least one of the sides of concept inclusions is a concept name, which we call $\mathcal{EL}_{\mathsf{lhs,rhs}}$.

An implementation of a learning algorithm for $\mathcal{EL}$ ontologies in this named form was presented by Duarte et al. [9,10]. The algorithm is exponential in the size of the vocabulary $\Sigma_\mathcal{O}$ of the ontology $\mathcal{O}$ (which is the set of concept/role and individual names occurring in $\mathcal{O}$) and the largest concept expression $C_\mathcal{O}$[4], but it is not exponential in the size of the whole ontology.

Theorem 3 is not directly applicable in this case, however, we observe that if the exact learning algorithm uses the epistemic learning model, then the outcome of each example query will be a counterexample, and so, the complexity result obtained with that algorithm is transferable to the epistemic setting. To see this, consider Algorithm 2, which is an adaptation of the exact learning algorithm for $\mathcal{EL}_{\mathsf{lhs,rhs}}$ [9,10]. Assume $\mathfrak{F}$ in Algorithm 2 is $\mathfrak{F}(\mathcal{EL}_{\mathsf{lhs,rhs}}, \mathbf{A})$. The number of $\Sigma_\mathcal{O}$-assertions (defined as assertions with only symbols from $\Sigma_\mathcal{O}$) is polynomial in the size of $\mathcal{O}$, so, in Line 1, Algorithm 2 identifies those that occur in $\mathcal{O}$ using $\mathbf{K}$-membership queries. It follows that all examples returned by the oracle in the 'while' loop are concept inclusions. In each iteration of the 'while' loop, the algorithm uses the examples returned by the $\mathsf{EX}_{\mathfrak{F}(\mathcal{EL}_{\mathsf{lhs,rhs}},\mathbf{A}),\mathcal{O}^k}$ oracle to compute what is called 'left $\mathcal{O}$-essential' and 'right $\mathcal{O}$-essential' concept inclusions using $\mathbf{K}$-membership queries, and then updates the hypothesis with such inclusions. We do not go into details of the algorithm, which is fully presented in the mentioned reference, but point out that it only adds to its hypothesis concept inclusions that follow from the target ontology $\mathcal{O}$.

Since we use $\mathbf{K}$-membership queries, the oracle is aware of the knowledge obtained by the learner in this way and does not return examples which follow from such entailments. With an inductive argument on the number of iterations of the main loop of the algorithm one can show that, at each iteration, if the learner asks for an example query instead of an equivalence query, the outcome will indeed be a counterexample for $\mathcal{O}$ and $\mathcal{H}$. So the number of membership and equivalence queries is the same as the number of $\mathbf{K}$-membership and example queries. Moreover, the hypothesis $\mathcal{H}$ computed by Algorithm 2 is equivalent to the target $\mathcal{O}$ (where $\mathcal{O} = \mathcal{O}^1$, so without epistemic axioms). Our next theorem formalises the fact that the number of queries performed by the exact learning algorithm has the same bound in the epistemic learning framework.

Theorem 7. $\mathfrak{F}(\mathcal{EL}_{\mathsf{lhs,rhs}}, \mathbf{A})$ *is epistemically learnable in* $O(|\Sigma_\mathcal{O}|^{\sharp_\mathcal{O}} \cdot (|C \sqsubseteq D|)^2)$ *many steps, where* $\sharp_\mathcal{O} = 2 \cdot |C_\mathcal{O}| \cdot |\Sigma_\mathcal{O}| + 2$*,* $C_\mathcal{O}$ *is the largest concept expression in* $\mathcal{O}$ *and* $C \sqsubseteq D$ *is the largest counterexample given by the oracle.*

6 Discussion

We introduced the epistemic learning model and investigated polynomial learnability in our model, establishing that it coincides with polynomial learnability in the exact learning model, and as a consequence, we can also transfer results in our model to the PAC learning model extended with membership queries.

[4] 'The largest' concept expression (and, later, counterexample) refers to the maximum of the sizes of counterexamples/concept expressions.

When the learner is only allowed to pose example queries, we showed that polynomial learnability in our model in strictly harder than in the exact learning model with only equivalence queries. This suggests that example queries are less demanding for the oracle than equivalence queries. We showed that, in the $\mathcal{EL}$ case, the epistemic reasoning that the oracle needs to perform features PTime complexity. Our results complement previous research on polynomial learnability in the exact and PAC learning models [7], where the authors analyse models between the exact and PAC learning models, in a learning from interpretations setting. As future work, we plan to investigate whether the implementation for $\mathcal{EL}_{\mathsf{lhs,rhs}}$ [10] could benefit from our approach, where the oracle keeps track of the knowledge passed to the learner, instead of processing the hypothesis at each iteration.

References

1. Angluin, D.: Queries and concept learning. Mach. Learn. **2**(4), 319–342 (1988)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Kaelbling, L., Saffiotti, A. (eds.) Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp. 364–369. Professional Book Center (2005)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications, 2nd edn. Cambridge University Press, Cambridge (2007)
4. Baader, F., Ghilardi, S., Lutz, C.: LTL over description logic axioms. ACM Trans. Comput. Logic **13**(3), 21:1–21:32 (2012)
5. Blum, A.L.: Separating distribution-free and mistake-bound learning models over the Boolean domain. SIAM J. Comput. **23**(5), 900–1000 (1994)
6. Borgwardt, S., Thost, V.: Temporal query answering in the description logic EL. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, pp. 2819–2825 (2015)
7. Bshouty, N.H., Jackson, J.C., Tamon, C.: Exploring learnability between exact and PAC. J. Comput. Syst. Sci. **70**(4), 471–484 (2005)
8. De Giacomo, G., Iocchi, L., Nardi, D., Rosati, R.: Moving a robot: the KR&R approach at work. In: Aiello, L.C., Doyle, J., Shapiro, S.C. (eds.) Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 198–209. Morgan Kaufmann (1996)
9. Duarte, M.R.C., Konev, B., Ozaki, A.: Exact learning of EL ontologies. In: Proceedings of the 31st International Workshop on Description Logics co-located with 16th International Conference on Principles of Knowledge Representation and Reasoning (KR) (2018)
10. Duarte, M.R.C., Konev, B., Ozaki, A.: Exactlearner: a tool for exact learning of EL ontologies. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR, pp. 409–414 (2018)
11. Frazier, M., Pitt, L.: Learning from entailment: an application to propositional Horn sentences. In: International Conference on Machine Learning, ICML, pp. 120–127 (1993)
12. Halpern, J.Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artif. Intell. **54**(3), 319–379 (1992). https://doi.org/10.1016/0004-3702(92)90049-4

13. Hughes, G., Cresswell, M.: A New Introduction to Modal Logic. Routledge, London and New York (1996)
14. Konev, B., Lutz, C., Ozaki, A., Wolter, F.: Exact learning of lightweight description logic ontologies. J. Mach. Learn. Res. **18**(201), 1–63 (2018)
15. Konev, B., Ozaki, A., Wolter, F.: A model for learning description logic ontologies based on exact learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 1008–1015 (2016)
16. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. The MIT Press, Cambridge (2012)
17. Ozaki, A., Troquard, N.: Learning ontologies with epistemic reasoning: the EL case. CoRR (2019). https://arxiv.org/abs/1902.03273
18. Valiant, L.G.: A theory of the learnable. Commun. ACM **27**(11), 1134–1142 (1984)