Generating Comprehensible Explanations in Description Logic

Fredrik Engström[1], Abdul Rahim Nizamani[2], and Claes Strannegård[13]

[1] Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
[2] Department of Applied Information Technology
University of Gothenburg
[3] Department of Applied Information Technology
Chalmers University of Technology

Abstract. We propose a method for generating comprehensible explanations in description logic. Such explanations could be of potential use for *e.g.* engineers, doctors, and users of the semantic web. Users commonly need to understand why a logical statement follows from a set of hypotheses. Then, automatically generated explanations that are easily understandable could be of help. A proof system for description logic that can be used for generating comprehensible explanations is proposed. Similar systems have been proposed for propositional logic [30] and first-order logic [28].

1 Introduction

When faced with the problem of explaining a logically valid entailment for a fellow human, mathematicians and logicians tend to give (or at least outline) a proof of the entailment in an informal model-theoretic framework or a proper deductive system. Examples of such formal systems include natural deduction, sequent calculus, and analytic tableaux. However, many such formal systems have an associated notion of deduction that is better suited for machines to check than for humans to understand.

In this paper, a proof formalism for a description logic that is based on a simple cognitive model of a human user is proposed as a formal framework for explanations. By using this formal proof system, complexity measures on entailments in description logics are defined and evaluated against empirical data of a small experiment that was reported in [13].

Description logic has been used as the logical framework of the semantic web and its associated range of ontologies, but also in other knowledge representation systems. Description logic systems tend to be rather expressive, but still computationally feasible. Description logic can be seen as a guarded fragment of first-order logic, reminiscent of multi-modal logic.

1.1 Explanations in Description logic

In many situations, a user of an ontology needs to understand *why* a logical statement φ follows from a set of hypotheses Γ. For instance, an engineer who wants to remove an inconsistency from an ontology Γ needs to understand why $\Gamma \vdash \top \sqsubseteq \bot$ holds. Similarly, a user of a knowledge representation system, such as SNOMED-CT, needs to be sure that an entailment holds not because of an inconsistency or a bug in the ontology.

An example of an entailment that is not so self-evident for non-experts is the following: The set of axioms

$$\begin{gathered}\texttt{Person} \sqsubseteq \neg\texttt{Movie}\\ \texttt{RRated} \sqsubseteq \texttt{Movie}\\ \texttt{RRated} \equiv \texttt{Thriller} \sqcup \forall\texttt{hasViolenceLevel}.\texttt{High}\\ \exists\texttt{hasViolenceLevel}.\top \sqsubseteq \texttt{Movie}\end{gathered}$$

entails that $\texttt{Person} \sqsubseteq \bot$. Written in a more condensed form this becomes

$$C_1 \sqsubseteq \neg C_2,\ C_3 \sqsubseteq C_2,\ C_3 \equiv C_4 \sqcup \forall R.C_5,\ \exists R.\top \sqsubseteq C_2 \quad \vdash \quad C_1 \sqsubseteq \bot \tag{1}$$

Systems like Protégé (a popular ontology editor) only provide the users with a very basic kind of explanation: A *justification* of an entailment $\Gamma \vdash \varphi$ is a minimal subset $\Gamma' \subseteq \Gamma$ such that $\Gamma' \vdash \varphi$. Such explanations may certainly help in pinpointing the part of a set of axioms responsible for a certain entailment, but, as pointed out in [14], it may not be enough for understanding *why* the entailment holds. As in all logic systems, the axioms and rules are important to understandability, but also *how* they are used in the arguments or deductions.

A traditional computer-generated proof (for example in a resolution or tableaux system) is not constructed for humans to understand, rather for machines to check. To machine-generate useful explanations we need to work in a proof system, whose proofs are easier for humans to understand. One such system, *justification oriented proofs*, was suggested in [14], in which an explanation is a tree of formulas. The root of the tree is the conclusion and the leaves are the axioms. Each node in the tree is called a lemma, and the set of children of a certain node is a justification of that node. Also, a node should be *easily deducible* from its children. This notion was made precise by means of a measure of complexity for entailments.

Other relevant work includes [3] and [19] in which formal proofs are proposed as explanations, but in which cognitive considerations are not taken seriously.

In this paper we take a rather different approach, designing a new proof system for description logic, reminiscent of proof systems previously designed for propositional logic [30] and model-checking in first-order logic [28]. The proof system uses bounded cognitive resources and thus an explicit connection to human cognition is provided. The proof system was "optimized" for producing understandable proofs/explanations mechanically. Using this proof system we use the minimum length of a proof as a simple cognitive complexity measure on entailments, thus making it possible to evaluate the approach.

1.2 Logical reasoning

Cognitive architectures such as Soar [18], ACT-R [2], CHREST [10], and NARS [32] have been used for modeling human reasoning in a wide range of domains. They typically include models of cognitive resources such as working memory, sensory memory, declarative memory, and procedural memory. These cognitive resources are all bounded in various ways, e.g., with respect to capacity, duration, and access time [17]. Notably, the working memory can typically only hold a small number of items, or chunks, and is a well-known bottleneck in many types of human problem solving [31].

In cognitive psychology, several models of logical reasoning have been presented, particularly in the mental logic tradition [4] and the mental models tradition [16]. Computational models in the mental logic tradition are commonly based on natural deduction systems [22]; the PSYCOP system [23] is one example. The focus of the mental models tradition is on particular examples of reasoning rather than general computational models.

Human reasoning in the domain of logic is considered from several perspectives in [1] and [12]. Stenning and van Lambalgen consider logical reasoning both "in the lab" and "in the wild" and investigate how reasoning problems formulated in natural language and situated in the real world are interpreted (reasoning to an interpretation) and then solved (reasoning from an interpretation) [27].

Formalisms of logical reasoning include natural deduction [22, 15, 7], sequent calculus [21], natural deduction in sequent calculus style [21], natural deduction in linear style [6, 8], and analytic tableaux [26]. Several formalisms have also emerged in the context of automated reasoning, e.g., Robinson's resolution system [24] and Stålmarck's system [25]. None of these formalisms represent working memory explicitly, and most of them were constructed for purposes other than modeling human reasoning.

We suspect that working memory is a critical cognitive resource also in the special case of logical reasoning [9, 11, 31]. Therefore, we define a proof system that includes an explicit bound on the working memory capacity.

For the sake of concreteness, we will use a particular description logic, a particular set of rules for this logic, a particular set of cognitive resources, and a particular choice of bounds on those cognitive resources. By no means do we claim those choices to be optimal in any way. We merely use them for illustrating the general idea of using machine generated proofs with bounded cognitive resources in order to produce explanations in description logic that are designed for understandability.

2 Description logic

We have chosen to focus on the description logic ALC in this paper, but hope to be able to extend our results to more expressive logics in future works. Concepts in ALC are built up from atomic concepts A and roles R in the following manner:

$$C ::= A \mid \top \mid \bot \mid \neg C \mid C \sqcup C \mid C \sqcap C \mid \exists R.C \mid \forall R.C$$

where A is an atomic concept and R a role. A formula φ is either of the form $C \equiv D$ or $C \sqsubseteq D$, where C and D are concepts.

A model $\mathcal{M}$ consists of a domain M and for each atomic concept C an interpretation $C^{\mathcal{M}} \subseteq M$, and for each role R an interpretation $R^{\mathcal{M}} \subseteq M \times M$. Each concepts gets interpreted as a subset of the domain M using the following standard recursive definition:

- $\top^{\mathcal{M}} = M$ and $\bot^{\mathcal{M}} = \emptyset$
- $(C \sqcup D)^{\mathcal{M}} = C^{\mathcal{M}} \cup D^{\mathcal{M}}$.
- $(C \sqcap D)^{\mathcal{M}} = C^{\mathcal{M}} \cap D^{\mathcal{M}}$.
- $(\neg C)^{\mathcal{M}} = M \setminus C^{\mathcal{M}}$.
- $(\forall R.C)^{\mathcal{M}} = \{ a \in M \mid \forall x{\in}M(a\ R^{\mathcal{M}}\ x \to C^{\mathcal{M}}x) \}$
- $(\exists R.C)^{\mathcal{M}} = \{ a \in M \mid \exists x{\in}M(a\ R^{\mathcal{M}}\ x \wedge C^{\mathcal{M}}x) \}$

Furthermore, $\mathcal{M} \models \varphi$ and $\Gamma \models \varphi$ are defined in the standard way; for example,

- $\mathcal{M} \models C \sqsubseteq D$ iff $C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$.

A *TBox* is a set Γ of formulas of the form $A \equiv D$ or $A \sqsubseteq D$ where A is an atomic concept and each atomic concept occurs at most once on the left hand side of a formula in Γ. Furthermore, we restrict our attention to the case of *acyclic* TBoxes: An atomic concept A *directly uses* an atomic concept B in Γ if B occurs in the right-hand side of any axiom with A on the left-hand side. The relation *uses* is the transitive closure of the *directly uses* relation. Γ is *acyclic* if no atomic concept A uses A.

3 Proof system

Several proof systems for description logics exist, including systems originally designed for modal and first-order logic. All of them, as far as we know, are *shallow*, meaning that the rules can only handle the main logical constants of the formulas. Such systems are not suited for studying human reasoning, since humans tend to use deep rules, such as substitution, when reasoning. We construct a proof system that is deep and linear, and which allows us later to put bounds on the length of formulas corresponding to bounds on the working memory of humans. The system is based on similar systems for propositional logic [30] and first-order logic [28], on a system for inductive reasoning [29], and also on introspection.

The proof system operates with sets of formulas. We let $\Gamma, \varphi = \Gamma \cup \{ \varphi \}$ and identify φ with the singleton set $\{ \varphi \}$. It should be noted that the system is not constructed as a minimal system, many of the rules and axioms are admissible in the strict formal sense. However, we are are interested in complexity measures that correspond to cognitive complexity, and those are sensitive to the exact formulation of the proof system.

We define the relation $\Gamma \vdash \varphi$, where Γ is a TBox (or more generally any set of formulas) and φ a formula of the form $C \sqsubseteq D$, meaning that there is a derivation from φ ending with $\emptyset$ in the proof system described below using the set Γ as the non-logical axioms.

$\Gamma \vdash C \equiv D$ means that $\Gamma \vdash C \sqsubseteq D$ *and* $\Gamma \vdash D \sqsubseteq C$.

3.1 Axioms

The axioms consist of two kinds:

- the non-logical axioms from the set Γ:
  A1. All formulas in Γ.
  A2. $A \equiv \bar{A} \sqcap C$ if $A \sqsubseteq C$ is in Γ. Here $\bar{A}$ is a new atomic concept.
- the logical axiom schemas (see [5] for a list of the axioms).

3.2 Rules

- The axiom rule:

$$\frac{\Delta,\ \varphi}{\Delta}\ \mathrm{Ax}$$

  Given that φ is an axiom.
- The substitution rule:

$$\frac{\Delta,\ \varphi}{\Delta,\ \varphi[C/D]}\ \mathrm{Subst}$$

  Given that $C \equiv D$ is an axiom. Here $\varphi[C/D]$ is the result of substituting *one* of the occurrences of D in φ with C.
- The strengthening rule:

$$\frac{\Delta,\ A \sqsubseteq B}{\Delta,\ A[D/C]^{+} \sqsubseteq B}\ \mathrm{Str}$$

$$\frac{\Delta,\ A \sqsubseteq B}{\Delta,\ A \sqsubseteq B[C/D]^{+}}\ \mathrm{Str}$$

  Both rules having the side condition that $C \sqsubseteq D$ is an axiom. Here $A[C/D]^{+}$ is the result of substituting *one* of the *negation free* occurrences of D in A with C. An occurrence is negation free if it is not in the scope of a negation.
- The first split rules:

$$\frac{\Delta,\ C \sqsubseteq A \sqcap B}{\Delta,\ C \sqsubseteq A,\ C \sqsubseteq B}\ \mathrm{Split1}$$

$$\frac{\Delta,\ A \sqcup B \sqsubseteq C}{\Delta,\ A \sqsubseteq C,\ B \sqsubseteq C}\ \mathrm{Split1}$$

- The second split rules:

$$\frac{\Delta,\ C \sqsubseteq D}{\Delta,\ C \sqcap A \sqsubseteq D,\ C \sqcap B \sqsubseteq D}\ \mathrm{Split2}$$

  Given that $\top \sqsubseteq A \sqcup B$ is an axiom.

$$\frac{\Delta,\ C \sqsubseteq D}{\Delta,\ C \sqsubseteq D \sqcup A,\ C \sqsubseteq D \sqcup B}\ \text{Split2}$$

Given that $A \sqcap B \sqsubseteq \bot$ is an axiom.

– The rules of contraposition:

$$\frac{\Delta,\ C \sqsubseteq D}{\Delta,\ \neg D \sqsubseteq \neg C}\ \text{Contr}$$

$$\frac{\Delta,\ C \sqcap D \sqsubseteq E}{\Delta,\ C \sqsubseteq E \sqcup \neg D}\ \text{Contr}$$

$$\frac{\Delta,\ C \sqsubseteq D \sqcup E}{\Delta,\ C \sqcap \neg D \sqsubseteq E}\ \text{Contr}$$

– The rules of monotonicity:

$$\frac{\Delta,\ \forall R.C \sqsubseteq \forall R.D}{\Delta,\ C \sqsubseteq D}\ \text{Mon}$$

$$\frac{\Delta,\ \exists R.C \sqsubseteq \exists R.D}{\Delta,\ C \sqsubseteq D}\ \text{Mon}$$

Example 1. For a simple example of a derivation in the system let us go back to the example in the introduction. Figure 1 gives a derivation proving that the axioms

$$\texttt{RRated} \sqsubseteq \texttt{Movie} \tag{2}$$

$$\texttt{RRated} \equiv \texttt{Thriller} \sqcup \forall \texttt{hasViolenceLevel.High} \tag{3}$$

entail $\texttt{Thriller} \sqsubseteq \texttt{Movie}$.

$$\cfrac{\cfrac{\cfrac{\texttt{Thriller} \sqsubseteq \texttt{Movie}}{\texttt{Thriller} \sqsubseteq \texttt{RRated}}\ \text{Str (2)}}{\texttt{Thriller} \sqsubseteq \texttt{Thriller} \sqcup (\forall \texttt{hasViolenceLevel.High})}\ \text{Subst (3)}}{\cfrac{\texttt{Thriller} \sqsubseteq \texttt{Thriller}}{\emptyset}\ \text{Ax } (C \sqsubseteq C)}\ \text{Str } (C \sqsubseteq C \sqcup D)$$

Fig. 1. The derivation in Example 1.

Proposition 1. *The proof system is sound with respect to the standard semantics of ALC, i.e, if $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

The proof is an easy induction on the length of derivations.

Proposition 2. *The proof system is complete with respect to acyclic TBoxes, i.e., if Γ is an acyclic TBox and $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.*

Proof (sketch). The first thing to observe is that the set Γ, the TBox, can be eliminated. Thus we only need to show that if $\models \varphi$ then $\vdash \varphi$. The reason is that we may, by TBox-axioms and the substitution rule substitute in all the information of Γ in φ. The rest of the proof follows the usual Henkin style proof of completeness for the modal logic K. We will only give the outline of the proof here.

A set Δ of concepts is said to be consistent if for no $C_1, \ldots, C_k \in \Delta$,

$$\vdash C_1 \sqcap \ldots \sqcap C_k \sqsubseteq \bot.$$

Completeness follows from proving that any consistent set of concepts has a model, meaning a model in which each concept's interpretation is non-empty. To construct a model of a consistent set Δ of concepts, maximal consistent extensions of Δ are used as objects. The interpretation of the atomic concept A is the set of all maximal consistent sets extending Δ and including A. The role R is interpreted as holding between a and b if for all concepts C, if $\forall R.C \in a$ then $C \in b$. It is routine to check that this in fact is a model in which every concept in Δ is non-empty.

Observe that we in the proof used the fact that Γ is an acyclic TBox. The question of completeness in the general case is open.

4 Bounded proof system

In this section we define our proof system with bounded cognitive resources. Taking the restriction on working memory capacity into account, we use a bound on the length of formulas that may occur in the proof-steps of our system. To make the model of working memory load more realistic, we also want to take into account that only certain parts of the formulas may be relevant to a proof-step, whereas other parts need not be stored in the working memory. For this purpose we use "black boxes" for concepts, called *abstraction concepts*, and denote them by $[\![C]\!]$, in which case the structure of the concept C is "hidden".

Also, by starting the derivation witnessing an entailment $\Gamma \vdash C \sqsubseteq D$ with the goal $[\![C]\!] \sqsubseteq [\![D]\!]$ instead of $C \sqsubseteq D$, the process of parsing formulas is modeled by the rule of Inspection (see below).

4.1 Abstraction concepts

Let L be the base language (atomic concepts, roles and individuals). For every concept C in L we introduce a new atomic concept $[\![C]\!]$. This new language is denoted by L_{abs}. The concept $[\![C]\!]$ is extensionally the same concept as C; we will see that we can derive $\vdash [\![C]\!] \equiv C$. However, since we are interested in the proof system where we only allow formulas of a certain length or complexity it may help to regard a complex concept as atomic, and this can be done with the help of abstraction concepts $[\![C]\!]$.

It should be noted that different intuitions on "black boxes" lead to slightly different implementations of the abstraction concepts. We have used a version where two "black boxes" which are syntactically the same are identified.

We need a rule for unfolding abstraction concepts in the following way:

– The inspection rule:

$$\frac{\Delta,\ \varphi}{\Delta,\ \varphi[\llbracket C\rrbracket^*/\llbracket C\rrbracket]}\ \text{Inspect}$$

Here $\varphi[\llbracket C\rrbracket^*/\llbracket C\rrbracket]$ is the result of substituting *one* of the occurrences of $\llbracket C\rrbracket$ in φ with $\llbracket C\rrbracket^*$, where $\llbracket C\rrbracket^*$ is defined by

– $\llbracket A\rrbracket^*$ is A if A is atomic.
– $\llbracket\neg C\rrbracket^*$ is $\neg\llbracket C\rrbracket$.
– $\llbracket C \sqcap D\rrbracket^*$ is $\llbracket C\rrbracket \sqcap \llbracket D\rrbracket$.
– $\llbracket C \sqcup D\rrbracket^*$ is $\llbracket C\rrbracket \sqcup \llbracket D\rrbracket$.
– $\llbracket\exists R.C\rrbracket^*$ is $\exists R.\llbracket C\rrbracket$.
– $\llbracket\forall R.C\rrbracket^*$ is $\forall R.\llbracket C\rrbracket$.

We also need to modify the axioms (A1) and (A2) slightly in this bounded version of the proof system:

– A1′. $A \equiv \llbracket C\rrbracket$ if $A \equiv C$ is in Γ; and $A \sqsubseteq \llbracket C\rrbracket$ if $A \sqsubseteq C$ is in Γ.
– A2′. $A \equiv \bar{A} \sqcap \llbracket C\rrbracket$ if $A \sqsubseteq C$ is in Γ. Here $\bar{A}$ is a new atomic concept.

Let us denote provability in this system by $\vdash_{\text{abs}}$: $\Gamma \vdash_{\text{abs}} C \sqsubseteq D$ if there is a derivation in the system starting with $\llbracket C\rrbracket \sqsubseteq \llbracket D\rrbracket$ and ending with $\emptyset$.

It is easy to show that:

Proposition 3. *$\Gamma \vdash \varphi$ iff $\Gamma \vdash_{abs} \varphi$.*

4.2 Length of formulas

The length of a concept is defined by

– $|\llbracket C\rrbracket| = |\top| = |\bot| = |A| = 1$
– $|C \sqcap D| = |C \sqcup D| = 1 + |C| + |D|$
– $|\neg C| = 1 + |C|$
– $|\exists R.C| = |\forall R.C| = 2 + |C|$.

The length of a formula is the sum of the lengths of the constituting concepts:

$$|C \equiv D| = |C \sqsubseteq D| = |C| + |D|.$$

The length of a set of formulas is the sum of the lengths of the formulas in the set. In the bounded system, only sets limited to a certain length are allowed to appear:

Definition 1. *$\Gamma \vdash^k_{abs} C \sqsubseteq D$ if there is a derivation (in the system described above) starting from $\llbracket C\rrbracket \sqsubseteq \llbracket D\rrbracket$ and ending in $\emptyset$ in which only sets of length less than or equal to k appear.*

Example 2. In Example 1 formulas of length six appear, however the subformula $\forall$`hasViolenceLevel.High`) is never "used". In fact we have that $\Gamma \vdash^4_{\text{abs}}$ `Thriller` $\sqsubseteq$ `Movie` as the derivation in Figure 2 shows.

$$
\begin{array}{cl}
[\![\mathtt{Thriller}]\!] \sqsubseteq [\![\mathtt{Movie}]\!] & \\ \hline
 & \text{Inspect} \\
[\![\mathtt{Thriller}]\!] \sqsubseteq \mathtt{Movie} & \\ \hline
 & \text{Str (2)} \\
[\![\mathtt{Thriller}]\!] \sqsubseteq [\![\mathtt{RRated}]\!] & \\ \hline
 & \text{Inspect} \\
[\![\mathtt{Thriller}]\!] \sqsubseteq \mathtt{RRated} & \\ \hline
 & \text{Sub (3)} \\
[\![\mathtt{Thriller}]\!] \sqsubseteq [\![\mathtt{Thriller} \sqcup (\forall \mathtt{hasViolenceLevel.High})]\!] & \\ \hline
 & \text{Inspect} \\
[\![\mathtt{Thriller}]\!] \sqsubseteq [\![\mathtt{Thriller}]\!] \sqcup [\![(\forall \mathtt{hasViolenceLevel.High})]\!] & \\ \hline
 & \text{Str } (C \sqsubseteq C \sqcup D) \\
[\![\mathtt{Thriller}]\!] \sqsubseteq [\![\mathtt{Thriller}]\!] & \\ \hline
 & \text{Ax } (C \sqsubseteq C) \\
\emptyset &
\end{array}
$$

Fig. 2. The derivation of Example 2.

4.3 A measure of reasoning complexity

Knowing that $\Gamma \models \varphi$ we may use the proof formalism to define different complexity measures. We have chosen to focus on the length of the shortest derivation witnessing

$$\Gamma \vdash^{6}_{\text{abs}} [\![C]\!] \sqsubseteq [\![D]\!].$$

The reasons for focusing on the length bound six are both pragmatic, it gives a measure that is feasibly computable, but also theoretic, the average number of chunks that fit in the working memory of a human is in the range $5-9$ [20].

5 Explanation generator

A proof searching computer program was implemented in Haskell that uses the standard breadth-first search algorithm with parallel computing threads. For computability reasons, we decided on adding some additional restrictions to the system including:

- When a proof step consists of more than one formulas, the program searches only for proofs that completely remove one of the formulas before modifying the other ones.
- The program only searches for proofs with a certain subformula property: a rule application may not introduce a new concept. For example, the program never searches for proofs that apply the substitution rule to the axiom $C \sqcup \top \equiv \top$ and replace a $\top$ with $D \sqcup \top$, as it introduces a new concept D.
- Proofs are acyclic: a proof step cannot appear more than once in a proof.

These modifications are real restrictions, and some justifications become unprovable. However, for the purposes of this paper, we have managed to find proofs for all the justifications needed using this restricted system.

In [13] a simple complexity measure on justifications was presented together with empirical data from an experiment conducted with 14 students. They were presented with six correct (even though the participants were not aware of this fact) justifications (EM1–3, and HM1–3) and asked to decide the correctness of the justifications. We would like to compare our complexity measure with the one presented in [13] using the empirical data form that experiment.

We decided to measure the complexity, i.e., the length of the shortest proof in the system described above for five of the justifications, the results can be found in Table 1. One of the justifications, HM1, uses constructors not in ALC, and we decided to exclude that justification altogether. Also, the justification EM1 uses an RBox, which we have not implemented in our system; however we decided to slightly modify EM1 to get EM1' and compute the complexity of EM1' instead.

The empirical data from the experiment can be found in Table 1, together with the computed complexities.

Justifacion	Success rate (%)	Mean time (s)	Horridge, et.al.	$\vdash^{6}_{\mathrm{abs}}$
EM1'	57^{*}	103^{*}	654^{*}	11
EM2	57	163	703	14
EM3	29	134	675	18
HM2	7	100	1395	13
HM3	57	157	1406	≤ 34

Table 1. The results of the experiments in [13] together with the two theoretical measures of complexity: the one from [13] and the proposed one. Observe that there is a small mismatch for the justification EM1', the experiment used the slightly more complex justification EM1 and the Horridge, et.al. complexity was also calculated with EM1.

Two comments should be made regarding the results in the table. By including rules for RBoxes in the system the complexity of EM1 should be expected to be $3 - 4$ steps higher than that of EM1', putting the complexity of EM1 on par with that of EM2.

Horridge, et. al. comments on the justification HM3 as follows.

> For item HM3 the explanation is a little different. [...] The high success rate was therefore due to an error in reasoning, that is, a failure in understanding rather than a failure in the model — the MSc students got the right answer, but for the wrong reasons.

Thus we think the correct success rate of HM3 should be considerably lower, close to zero.

As expected, our complexity measure assigns a too low measure of complexity to the hard problems, especially HM2. The explanation for this is probably that the search complexity for the smallest proof is not part of the complexity measure. This might indicate that to define a better measure of complexity, that correlates better with data from empirical studies, we need to engineer a more deterministic proof system in which the search itself is present.

Having said that, it should be noted that if the goal is to present a model for *explaining* justification (or any entailments), the smallest proof is well-suited. In particular, a shortened version of the proofs in which only the steps including axioms from the TBox are presented can, and should, be used as explanations.

For example, let us focus on the running example again. The shortened version of the proof of that justification is presented in Figure 3. In this example, only the steps including axioms from the TBox are shown. In an interactive framework, the proof could be presented in such a way making it possible to show the hidden parts of the deduction. We believe that such interactive presentations would constitute good explanations for entailments in description logics.

$$
\begin{array}{c}
C_1 \sqsubseteq \bot \\
\vdots \\
\dfrac{\top \wedge C_1 \sqsubseteq \bot}{\top \wedge (\neg C_2) \sqsubseteq \bot} \quad C_1 \sqsubseteq \neg C_2 \\
\vdots \\
\dfrac{\exists R.\top \sqcup \neg \exists R.\top \sqsubseteq C_2}{C_2 \sqcup \neg \exists R.\top \sqsubseteq C_2} \quad \exists R.\top \sqsubseteq C_2 \\
\vdots \\
\dfrac{\dfrac{C_2 \sqcup \forall R.C_5 \sqsubseteq C_2 \sqcup C_2}{C_2 \sqcup \forall R.C_5 \sqsubseteq C_3 \sqcup C_2} \quad C_3 \sqsubseteq C_2}{C_2 \sqcup \forall R.C_5 \sqsubseteq (C_4 \sqcup \forall R.C_5) \sqcup C_2} \quad C_4 \sqcup \forall R.C_5 \equiv C_3 \\
\vdots \\
\emptyset
\end{array}
$$

Fig. 3. A shortened proof of the justification (1).

6 Conclusion

We have presented a method for automatically generating explanations in description logic that target human understandability. In fact, the explanations are based on proofs in proof systems with bounded cognitive resources. An example of a specific proof system was given and the derived complexity measure was evaluated against a small set of empirical data. The results point in a positive direction, but deeper empirical studies are needed to evaluate the understandability of the generated explanations.

We believe that the approach presented in this paper is promising for generating understandable explanations intended for non-expert users, *e.g.*, engineers, doctors, and users of the semantic web. The explanations can be presented either in the language of description logic or, using straight-forward translations, in natural language.

References

1. Adler, J.E., Rips, L.J.: Reasoning: Studies of Human Inference and its Foundations. Cambridge University Press (2008)
2. Anderson, J.R., Lebiere, C.: The atomic components of thought. Lawrence Erlbaum, Mahwah, N.J. (1998)
3. Borgida, A., Franconi, E., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F.: Explaining ALC subsumption. In: ECAI. pp. 209–213 (2000)
4. Braine, M.D.S., O'Brien, D.P.: Mental logic. L. Erlbaum Associates (1998)
5. Engström, F., Nizamani, A.R., Strannegård, C.: Generating comprehensible explanations in description logic, `http://gup.ub.gu.se/publication/199115`, full version with appendix
6. Fitch, F.B.: Symbolic Logic: an Introduction. Ronald Press, New York (1952)
7. Gentzen, G.: Investigation into logical deduction, 1935. In: Szabo, M.E. (ed.) The collected papers of Gerhard Gentzen. North-Holland Amsterdam (1969)
8. Geuvers, H., Nederpelt, R.: Rewriting for Fitch style natural deductions. In: Rewriting Techniques and Applications. pp. 134–154. Springer (2004)
9. Gilhooly, K., Logie, R., Wetherick, N., Wynn, V.: Working memory and strategies in syllogistic-reasoning tasks. Memory & Cognition 21(1), 115–124 (1993)
10. Gobet, F., Lane, P.: The CHREST architecture of cognition: the role of perception in general intelligence. In: Artificial General Intelligence 2010, Lugano, Switzerland. Atlantis Press (2010)
11. Hitch, G., Baddeley, A.: Verbal reasoning and working memory. The Quarterly Journal of Experimental Psychology 28(4), 603–621 (1976)
12. Holyoak, K.J., Morrison, R.G.: The Cambridge handbook of thinking and reasoning. Cambridge University Press (2005)
13. Horridge, M., Bail, S., Parsia, B., Sattler, U.: Toward cognitive support for OWL justifications. Knowledge-Based Systems 53(0), 66 – 79 (2013)
14. Horridge, M., Parsia, B., Sattler, U.: Justification oriented proofs in owl. In: The Semantic Web–ISWC 2010, pp. 354–369. Springer (2010)
15. Jaskowski, S.: The theory of deduction based on the method of suppositions. Studia Logica 1, 5–32 (1934)
16. Johnson-Laird, P.N.: Mental models. Harvard University Press (1983)
17. Kosslyn, S., Smith, E.: Cognitive Psychology: Mind and Brain. Upper Saddle River, NJ: Prentice-Hall Inc (2006)
18. Laird, J., Newell, A., Rosenbloom, P.: Soar: An architecture for general intelligence. Artificial Intelligence 33(3), 1–64 (1987)
19. McGuinness, D.L., Borgida, A.: Explaining subsumption in description logics. In: IJCAI (1). pp. 816–821 (1995)
20. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
21. Negri, S., von Plato, J.: Structural Proof Theory. Cambridge University Press (2001)
22. Prawitz, D.: Natural Deduction. A Proof-Theoretical Study, Stockholm Studies in Philosophy, vol. 3. Almqvist & Wiksell, Stockholm (1965)
23. Rips, L.: The Psychology of Proof. Bradford (1996)
24. Robinson, A., Voronkov, A.: Handbook of Automated Reasoning. Elsevier Science (2001)
25. Sheeran, M., Stålmarck, G.: A tutorial on Stålmarck's proof procedure for propositional logic. Formal Methods in Systems Design 16(1), 23–58 (Jan 2000)

26. Smullyan, R.M.: First-Order Logic. Dover Publications, New York, second corrected edn. (1995), first published in 1968 by Springer Verlag, Berlin, Heidelberg, New York
27. Stenning, K., van Lambalgen, M.: Human reasoning and cognitive science. Bradford Books, MIT Press (2008)
28. Strannegård, C., Engström, F., Nizamani, A.R., Rips, L.: Reasoning about truth in first-order logic. Journal of Logic, Language and Information 22(1), 115–137 (2013), doi: 10.1007/s10849-012-9168-y
29. Strannegård, C., Nizamani, A.R., Sjöberg, A., Engström, F.: Bounded kolmogorov complexity based on cognitive models. In: Kühnberger, K.U., Rudolph, S., Wang, P. (eds.) Artificial General Intelligence, Lecture Notes in Computer Science, vol. 7999, pp. 130–139. Springer Berlin Heidelberg (2013)
30. Strannegård, C., Ulfsbäcker, S., Hedqvist, D., Gärling, T.: Reasoning processes in propositional logic. Journal of Logic, Language and Information 19(3), 283–314 (2010)
31. Toms, M., Morris, N., Ward, D.: Working memory and conditional reasoning. The Quarterly Journal of Experimental Psychology 46(4), 679–699 (1993)
32. Wang, P.: From NARS to a Thinking Machine. In: Proceedings of the 2007 Conference on Artificial General Intelligence. pp. 75–93. IOS Press, Amsterdam (2007)