Concise Justifications Versus Detailed Proofs for Description Logic Entailments*

Stefan Borgwardt

Institute of Theoretical Computer Science, TU Dresden, Germany
stefan.borgwardt@tu-dresden.de

Logical reasoning is often seen as explainable by design. However, exploiting this explainability for working systems that are operated by humans still requires more research on what form such explanations should take. We focus here on Description Logics (DLs) [3], which is a popular family of knowledge representation formalisms and the basis of the Web Ontology Language OWL 2 [14]. Description logic ontologies contain *axioms* like

$$\begin{array}{rlrl}
\mathsf{Cow} \sqsubseteq \mathsf{Mammal}, & (1) \qquad & \mathsf{Cow} \equiv \forall \mathsf{eats.Grass}, & (4) \\
\mathsf{Cow} \sqsubseteq {=}4\, \mathsf{hasPart.Leg}, & (2) \qquad & \exists \mathsf{eats}.\top \sqsubseteq \mathsf{Animal}, & (5) \\
\mathsf{Mammal} \sqsubseteq \mathsf{Animal}, & (3) \qquad & \mathsf{Grass} \sqsubseteq \neg\mathsf{Animal}, & (6)
\end{array}$$

i.e. that cows are mammals with 4 legs, mammals are animals, cows are defined as anything that eats only grass, something that eats anything ($\top$) is an animal, and grass is different from animals.

This ontology has the unintended consequence that $\mathsf{Grass} \sqsubseteq \mathsf{Animal}$, i.e. every grass is an animal. The most popular way to explain such a consequence are so-called *justifications*, which are subset-minimal subsets of the ontology that still entail the consequence [4,7]. The idea is to isolate the possible cause(s) of the error. In our example, Axioms (1), (3), (4), and (5) constitute a justification for $\mathsf{Grass} \sqsubseteq \mathsf{Animal}$, i.e. we can ignore the other two axioms. This technique is very useful in real ontologies, which often have tens of thousands of axioms.

In this example, a careful reader may already be able to pinpoint the problematic axiom, but explaining precisely how the unintuitive consequence follows can be more challenging. For this reason, more recent research has rediscovered *proofs* as a means of providing more detailed explanations [1,2,5,6,9,12]. Proofs can contain intermediate steps between a justification and the conclusion, and can either be obtained from calculi of inference rules tailored for a specific DL [10,17] or constructed in a black-box fashion using heuristics [7,8], concept interpolation [16], or forgetting [1]. In our example, a proof may look as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{Cow} \equiv \forall \mathsf{eats.Grass}\ (4) \qquad \dfrac{\mathsf{Cow} \sqsubseteq \mathsf{Mammal}\ (1) \qquad \mathsf{Mammal} \sqsubseteq \mathsf{Animal}\ (3)}{\mathsf{Cow} \sqsubseteq \mathsf{Animal}}}{\forall \mathsf{eats.Grass} \sqsubseteq \mathsf{Animal}}}{\neg\exists \mathsf{eats}.\top \sqsubseteq \mathsf{Animal}} \qquad \exists \mathsf{eats}.\top \sqsubseteq \mathsf{Animal}\ (5)}{\top \sqsubseteq \mathsf{Animal}}}{\mathsf{Grass} \sqsubseteq \mathsf{Animal}}$$

* This work is supported by the DFG grant 389792660 (TRR 248) (see https://perspicuous-computing.science).

This proof describes that, because of Axioms (1), (3), and (4), everything that only eats grass is an animal. In particular, anything that does not eat ($\neg\exists\mathsf{eat}.\top$) vacuously eats only grass, and therefore is an animal. Since by Axiom (5), also everything that eats is an animal, everything that exists must be an animal ($\top \sqsubseteq \mathsf{Animal}$), which means in particular that grass is also an animal. Here, it is easier to point out the problems in the chain of reasoning (e.g. $\neg\exists\mathsf{eats}.\top \sqsubseteq \mathsf{Animal}$) and to identify Axiom 4 as the root cause that needs to be repaired.

Both justifications and proofs can be used as explanations, but different explanations can be easier or harder to understand than others. Therefore, different measures have been proposed to quantify their understandability. As the most basic, the *size* of a justification or proof, i.e. the number of involved axioms,[1] can indicate how easy it will be to understand [1,15], but does not reflect the complexity inherent in the individual axioms. In [8], the authors developed a *(cognitive) justification complexity* measure by assigning different weights to features of the axioms and computing how similar the goal consequence is to the axioms in the justification. The full definition of this measure is too long to repeat here, but it includes features such as the number of different constructors (e.g. $\neg$, $\exists$, $\forall$), the number of axiom types (e.g. subsumption $\sqsubseteq$ or equivalence $\equiv$), or whether a "hidden" consequence $\top \sqsubseteq \mathsf{A}$ is implied by the justification, but is not the goal consequence (e.g. $\top \sqsubseteq \mathsf{Animal}$ in the example). In [8], this approach was validated by several user studies relating justification complexity to the number of errors made by humans when working with the justifications.

In [1], justification complexity was extended to measure proof complexity by viewing each inference step in a proof as a single "justification" of its inference. The *average step complexity* of a proof is the average of the justification complexities of each inference step in the proof, and similarly for the *maximum step complexity*. We will also consider ratios between these measures, e.g. the ratio of the proof size to the justification size, which reflects how many proof steps were needed per axiom in the original justification, or the ratio of the aggregated step complexity to the justification complexity. We consider both size and complexity, because in our dataset these two measures were not correlated,[2] so they reflect different aspects of the justifications and proofs. The size says more about the time required to read a proof than about the difficulty understanding the individual steps; e.g. a small proof can still contain very complex axioms. In the example, the size of the justification is 4 and the proof has size 9 (ratio 2.25). The justification's complexity is 400, the average step complexity of the proof is 236 (ratio 0.59), and the maximum is 380 (ratio 0.95). This indicates that the proof steps are (on average) easier to understand than the justification by itself, but there are still some harder steps, e.g. inferring $\neg\exists\mathsf{eats}.\top \sqsubseteq \mathsf{Animal}$ from $\forall\mathsf{eats}.\mathsf{Grass} \sqsubseteq \mathsf{Animal}$.

Since constructing proofs for consequences of large ontologies is challenging, in [1] proofs were obtained by first computing justifications for the consequences and then constructing proofs from these much smaller sets of axioms. Since the goal of that paper was to find small proofs, an implicit assumption was that

[1] Axioms that are used in several inference steps are only counted once.

[2] Except for a weak negative correlation of proof size and average step complexity.

small justifications would also yield small proofs. In this paper, we investigate this connection between justification complexity and proof complexity in more detail. We identify cases where small justifications do not lead to small proofs, and compare the justification complexity with the step complexity of a proof.

Dataset

The analysis is based on the dataset used in [1] to compare different methods of generating proofs. It consists of a series of 1.573 *tasks* that were extracted from the OWL 2 EL classification track of the 2015 OWL Reasoner Evaluation (ORE) [13]. Each task consists of a justification and a consequence. For example, for a single consequence $\mathsf{A} \sqsubseteq \mathsf{B}$ there could be 100 justifications in a given ontology, and each of those justifications was converted into 1 task for the dataset. Tasks that were the same up to renaming were merged and it was recorded how many of the original justifications are represented by each task in the final dataset. In total, this dataset covers 2.308.562 justifications from the ORE 2015 ontologies.

Three proof generators were used to construct proofs for each of the tasks. The first generator used the consequence-based reasoner ELK [10]. Since ELK returns all inference steps that can be used to derive a consequence, a post-processing step extracted a proof of minimal size. The other two proof generators used a formal technique called *forgetting* to generate proofs by eliminating predicates from the justification one-by-one, until only the predicates of the target consequence, e.g. $\mathsf{A} \sqsubseteq \mathsf{B}$, remained (for details, see [1]). Two forgetting tools were used for that purpose, LETHE [11] and FAME 1.0 [18]. The latter failed to generate proofs in some cases, and thus there are only 917 proofs in that part of the dataset. For some examples of actual proofs from the dataset, see the appendix or [1, 2].

Analysis

Figure 1 shows the relationship between justification size and proof size. The first diagram depicts the overall distribution of the 1573 tasks in the dataset. Individual data points are represented by semi-transparent shapes, i.e. stronger colors indicate more tasks with the same values. The second diagram is a histogram of all occurring ratios of $\frac{\text{proof size}}{\text{justification size}}$. The last is the same histogram adjusted by the number of times each task occurred in the original ORE 2015 dataset.

We can see that proofs are around twice the size of the given justification, with LETHE sometimes producing slightly smaller proofs and a small number of proofs being 3–5 times larger. On the dataset of 1573 tasks, the average ratio is 2.53 (standard deviation (SD) 0.87) for ELK, and slightly smaller for LETHE and FAME. In the adjusted histogram, the average is 1.97 (SD 0.33) for ELK and very similar for the other two. This indicates that, on average, one inference step is necessary to incorporate each justification axiom into a proof and sometimes additional steps are needed. The correlation between proof size and justification size is very strong, i.e. $r > 0.8$ for all proof generators, and even $r > 0.95$ after adjustment. This means that small justifications do indeed generally induce small proofs, with only few exceptions. The most extreme outliers come from justifications similar to

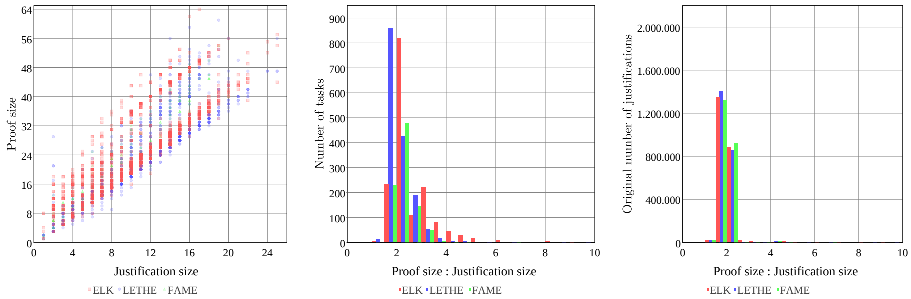

Fig. 1. Comparing justification size and proof size. 3 outliers have ratio > 10.

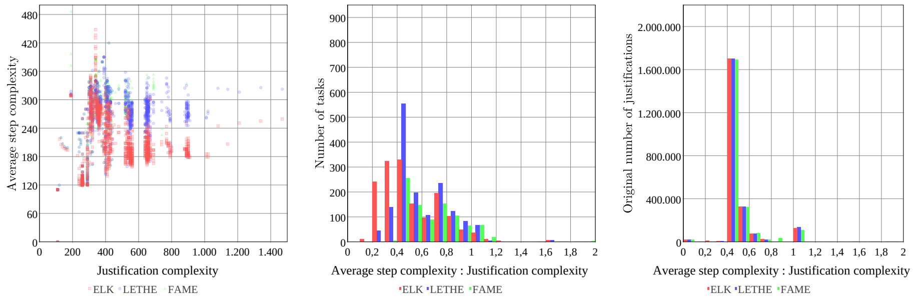

Fig. 2. Justification complexity vs. average step complexity. 2 outliers have ratio > 2.

$\{\mathsf{A} \equiv \mathsf{D} \sqcap \exists \mathsf{r.C} \sqcap \exists \mathsf{s.E} \sqcap \exists \mathsf{t.F},\ \mathsf{B} \equiv \mathsf{D} \sqcap \exists \mathsf{r.C} \sqcap \exists \mathsf{s.E}\}$. Elk needs multiple steps to decompose the conjunction ($\sqcap$) into statements like $\mathsf{A} \sqsubseteq \exists \mathsf{r.C}$, before composing it again to $\mathsf{A} \sqsubseteq \mathsf{D} \sqcap \exists \mathsf{r.C} \sqcap \exists \mathsf{s.E} \sqsubseteq \mathsf{B}$. The forgetting-based proof generators show a similar behavior, but this strongly depends on the order in which the predicates were eliminated. In the above example, forgetting D and E first yields the two axioms $\mathsf{A} \sqcap \exists \mathsf{r.C} \sqsubseteq \mathsf{B}$ and $\mathsf{A} \sqsubseteq \exists \mathsf{r.C}$, which gives $\mathsf{A} \sqsubseteq \mathsf{B}$ in one step by forgetting C.

There is only a moderate correlation between justification complexity and average step complexity (see Figure 2), with $r < 0.65$ ($r < 0.5$ after adjustment). Generally, the average step complexity is around half the justification complexity (average 0.52, SD 0.15 for Elk after adjustment), which illustrates that proofs break down the justifications into more comprehensible steps. But a sizable number of proofs has the same average step complexity as the justification, or an even higher one. Examples with a high ratio are similar to the ones mentioned above. In particular, if an equivalence axiom $A \equiv B$ is derived only from equivalence axioms in the justification, then the justification complexity is smaller than the step complexity of inferences involving both $\equiv$ and $\sqsubseteq$ (because they involve more than one axiom type), which increases the ratio. On the other end of the spectrum, there are proofs with an average step complexity of 20% of the justification complexity. This mainly happens when the justification contains many axioms, but they can be combined into a proof using very simple steps.

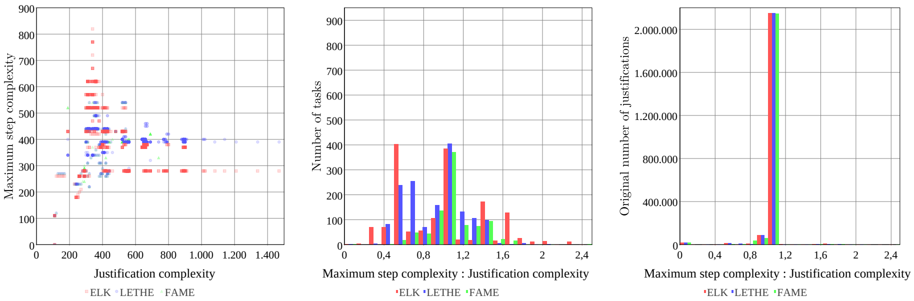

Fig. 3. Justification complexity vs. max. step complexity. 3 outliers have ratio > 2.5.

In Figure 3, the justification complexity and the maximum step complexity are strongly correlated, up to between $r = 0.66$ (Elk) and $r = 0.75$ (Fame) after adjustment. A large majority of the proofs for the ORE 2015 ontologies have a maximum step complexity that is roughly the same as the associated justification complexity (average 0.99, SD 0.12 for Elk after adjustment). The examples with extremely high or low ratios are again similar to the previous cases.

Discussion

In this existing dataset of proofs focused on the OWL 2 EL profile, we observed a relatively strong correlation between justification size/complexity and proof size/complexity. This means that the assumption that small justifications yield small proofs is generally justified. The main cause of outliers seems to be the handling of equivalence axioms, for which the proof generators are not optimized. Introducing tailored inference rules into Elk could improve the readability of proofs in these cases. For the forgetting-based proof generators, one can perhaps find a heuristic for choosing which predicates to forget first in order to reduce the complexity for such cases as well. In general, identifying outliers like this can help to fine-tune or debug proof generators.

What one cannot determine from this dataset is whether there is an ORE 2015 ontology that contains a single consequence with multiple justifications of different sizes or complexities, but where the easier justifications yield more difficult proofs. This is because the dataset aggregates isomorphic justifications into single tasks and does not track from which original consequences they were generated. To answer such more fine-grained questions, one would have to go back to the original ORE 2015 ontology corpus.

It is also still unclear which measures best describe the comprehensibility of proofs. In contrast to justifications, not just the complexity of single inference steps plays a role, but also the size and overall structure of the proof. Moreover, individual users may have different "inference rules" in their head than the ones produced by a proof generator (and different from other users), and this mismatch may increase the complexity of understanding a proof.

References

1. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding small proofs for description logic entailments: Theory and practice. In: Albert, E., Kovács, L. (eds.) Proc. of the 23rd Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'20). EasyChair Proceedings in Computing, vol. 73, pp. 32–67. EasyChair (2020). DOI: 10.29007/nhpp
2. Alrabbaa, C., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding proofs for description logic entailments in practice (extended abstract). In: Baader, F. (ed.) Workshop on Explainable Logic-Based Knowledge Representation (XLoKR'20) (2020), https://lat.inf.tu-dresden.de/XLoKR20/XLoKRpaper394.pdf
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017), http://dltextbook.org/
4. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}^+$. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) Proc. of the 30th German Conf. on Artificial Intelligence (KI'07). Lecture Notes in Artificial Intelligence, vol. 4667, pp. 52–67. Springer-Verlag (2007). DOI: 10.1007/978-3-540-74565-5_7
5. Borgida, A., Franconi, E., Horrocks, I.: Explaining $\mathcal{ALC}$ subsumption. In: Proc. of the 14th Eur. Conf. on Artificial Intelligence (ECAI'00). pp. 209–213 (2000), http://www.frontiersinai.com/ecai/ecai2000/pdf/p0209.pdf
6. Borgwardt, S., Hirsch, A., Kovtunova, A., Wiehr, F.: In the eye of the beholder: Which proofs are best? In: Borgwardt, S., Meyer, T. (eds.) Proc. of the 33rd Int. Workshop on Description Logics (DL'20). CEUR Workshop Proceedings, vol. 2663 (2020), http://ceur-ws.org/Vol-2663/paper-6.pdf
7. Horridge, M.: Justification Based Explanation in Ontologies. Ph.D. thesis, University of Manchester, UK (2011), https://www.research.manchester.ac.uk/portal/files/54511395/FULL_TEXT.PDF
8. Horridge, M., Bail, S., Parsia, B., Sattler, U.: Toward cognitive support for OWL justifications. Knowledge-Based Systems **53**, 66–79 (2013). DOI: 10.1016/j.knosys.2013.08.021
9. Kazakov, Y., Klinov, P., Stupnikov, A.: Towards reusable explanation services in protege. In: Artale, A., Glimm, B., Kontchakov, R. (eds.) Proc. of the 30th Int. Workshop on Description Logics (DL'17). CEUR Workshop Proceedings, vol. 1879 (2017), http://www.ceur-ws.org/Vol-1879/paper31.pdf
10. Kazakov, Y., Krötzsch, M., Simančík, F.: The incredible ELK. Journal of Automated Reasoning **53**(1), 1–61 (2013). DOI: 10.1007/s10817-013-9296-3
11. Koopmann, P., Schmidt, R.A.: LETHE: Saturation-based reasoning for non-standard reasoning tasks. In: Dumontier, M., Glimm, B., Gonçalves, R.S., Horridge, M., Jiménez-Ruiz, E., Matentzoglu, N., Parsia, B., Stamou, G.B., Stoilos, G. (eds.) Proc. of the 4th OWL Reasoner Evaluation Workshop (ORE'15). CEUR Workshop Proceedings, vol. 1387, pp. 23–30. CEUR-WS.org (2015), http://ceur-ws.org/Vol-1387/paper_9.pdf
12. McGuinness, D.L.: Explaining Reasoning in Description Logics. Ph.D. thesis, Rutgers University, NJ, USA (1996). DOI: 10.7282/t3-q0c6-5305
13. Parsia, B., Matentzoglu, N., Gonçalves, R.S., Glimm, B., Steigmiller, A.: The OWL Reasoner Evaluation (ORE) 2015 competition report. Journal of Automated Reasoning **59**(4), 455–482 (2017). DOI: 10.1007/s10817-017-9406-8
14. Patel-Schneider, P., Parsia, B., Motik, B.: OWL 2 web ontology language structural specification and functional-style syntax (second edition). W3C recommendation, W3C (2012), http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/

15. Peñaloza, R., Sertkaya, B.: Understanding the complexity of axiom pinpointing in lightweight description logics. Artificial Intelligence **250**, 80–104 (2017). DOI: 10.1016/j.artint.2017.06.002
16. Schlobach, S.: Explaining subsumption by optimal interpolation. In: Alferes, J.J., Leite, J.A. (eds.) Proc. of the 9th Eur. Conf. on Logics in Artificial Intelligence (JELIA'04). Lecture Notes in Computer Science, vol. 3229, pp. 413–425. Springer-Verlag (2004). DOI: 10.1007/978-3-540-30227-8_35
17. Simancik, F., Kazakov, Y., Horrocks, I.: Consequence-based reasoning beyond horn ontologies. In: Walsh, T. (ed.) Proc. of the 22nd Int. Joint Conf. on Artificial (IJCAI'11). pp. 1093–1098. IJCAI/AAAI (2011). DOI: 10.5591/978-1-57735-516-8/IJCAI11-187
18. Zhao, Y., Schmidt, R.A.: FAME: an automated tool for semantic forgetting in expressive description logics. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Proc. of the 9th Int. Conf. on Automated Reasoning (IJCAR'18). Lecture Notes in Computer Science, vol. 10900, pp. 19–27. Springer-Verlag (2018). DOI: 10.1007/978-3-319-94205-6_2

A Examples

Figures 4–6 show some of the examples discussed above. Proofs are shown as directed acyclic hypergraphs, with hyperedges indicated by blue boxes. The justification is given by the nodes with an incoming hyperedge that has no premises and that is labeled by “asserted” or “Asserted Conclusion”.

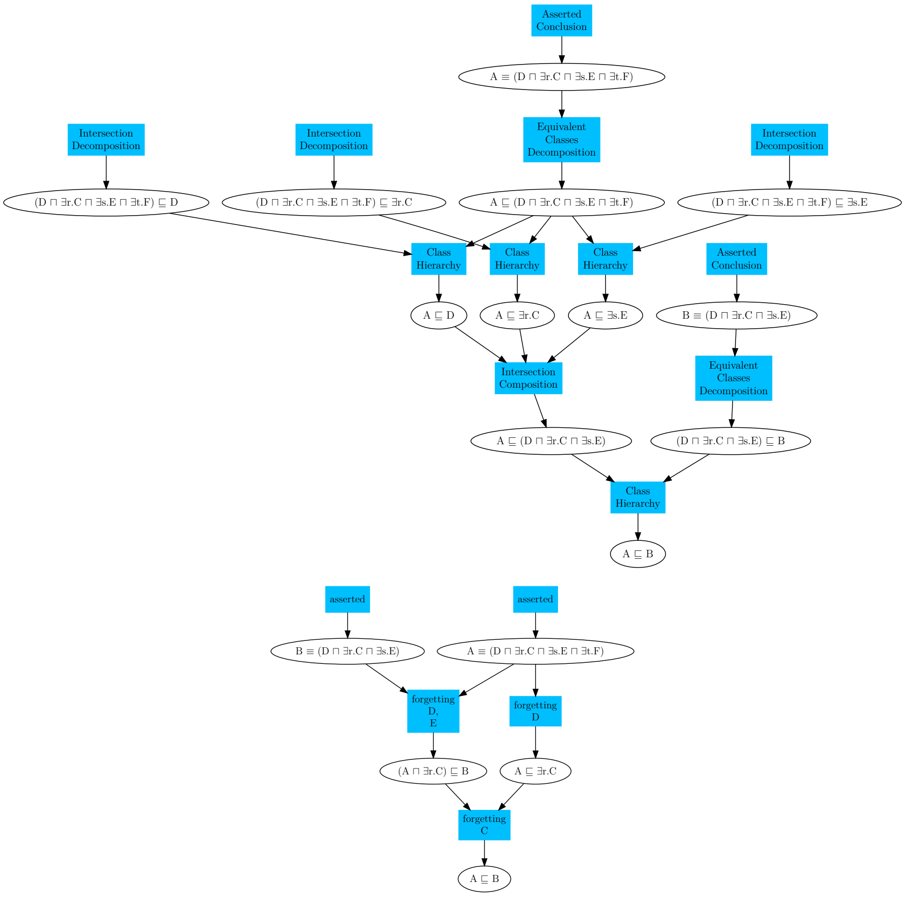
Asserted Conclusion
$A \equiv (D \sqcap \exists r.C \sqcap \exists s.E \sqcap \exists t.F)$
Intersection Decomposition
Intersection Decomposition
Equivalent Classes Decomposition
Intersection Decomposition
$(D \sqcap \exists r.C \sqcap \exists s.E \sqcap \exists t.F) \sqsubseteq D$
$(D \sqcap \exists r.C \sqcap \exists s.E \sqcap \exists t.F) \sqsubseteq \exists r.C$
$A \sqsubseteq (D \sqcap \exists r.C \sqcap \exists s.E \sqcap \exists t.F)$
$(D \sqcap \exists r.C \sqcap \exists s.E \sqcap \exists t.F) \sqsubseteq \exists s.E$
Class Hierarchy
Class Hierarchy
Class Hierarchy
Asserted Conclusion
$A \sqsubseteq D$
$A \sqsubseteq \exists r.C$
$A \sqsubseteq \exists s.E$
$B \equiv (D \sqcap \exists r.C \sqcap \exists s.E)$
Intersection Composition
Equivalent Classes Decomposition
$A \sqsubseteq (D \sqcap \exists r.C \sqcap \exists s.E)$
$(D \sqcap \exists r.C \sqcap \exists s.E) \sqsubseteq B$
Class Hierarchy
$A \sqsubseteq B$
asserted
asserted
$B \equiv (D \sqcap \exists r.C \sqcap \exists s.E)$
$A \equiv (D \sqcap \exists r.C \sqcap \exists s.E \sqcap \exists t.F)$
forgetting D, E
forgetting D
$(A \sqcap \exists r.C) \sqsubseteq B$
$A \sqsubseteq \exists r.C$
forgetting C
$A \sqsubseteq B$

Fig. 4. A task with small justification size of 2, but a large proof generated by Elk of size 12 (shown at the top), i.e. a ratio of 6. The justification complexity is 340, the average step complexity is 363 (ratio 1.07), and the maximum step complexity is 620 (ratio 1.82). The corresponding proof generated by Lethe (bottom) has size 5 (ratio 2.5), average step complexity 320 (ratio 0.94), and maximum step complexity 340 (ratio 1). Many larger examples with more conjuncts exist in the dataset.

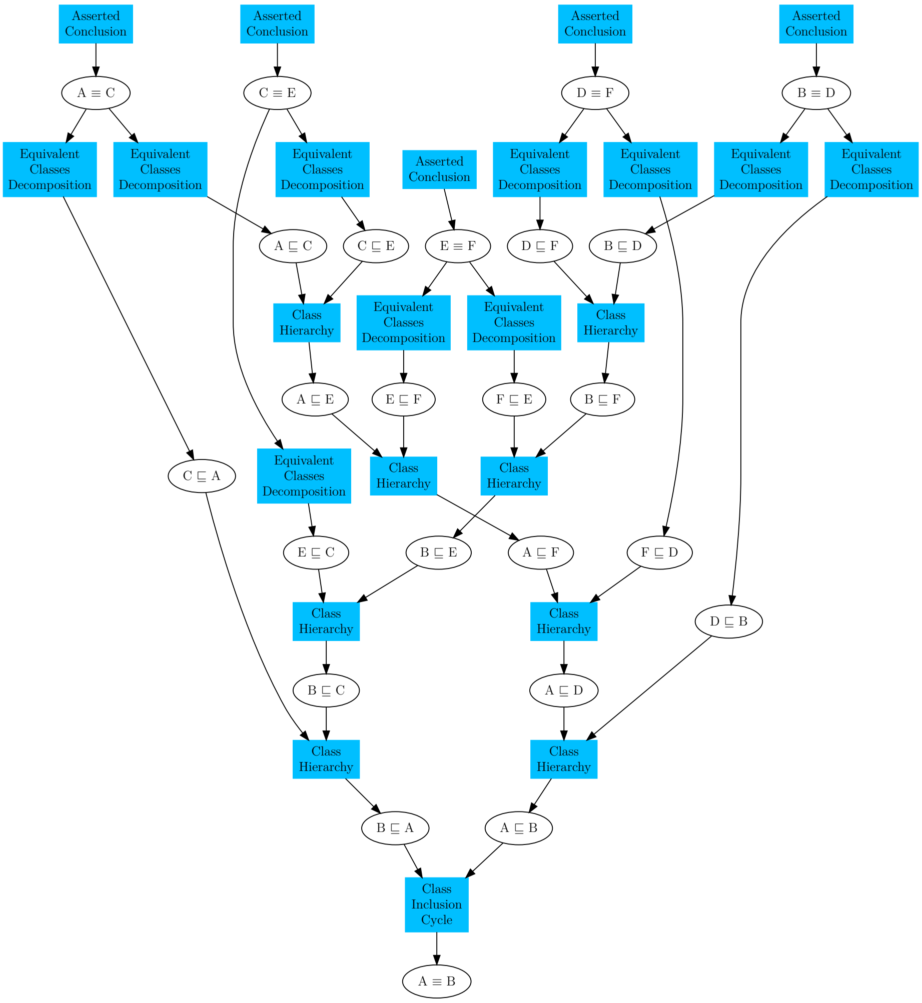

Fig. 5. Another example of an unnecessarily large proof generated by Elk. The justification contains 5 axioms and the proof is of size 24 (ratio 4.8). The justification complexity is 150 and the proof has an average step complexity of 197 (ratio 1.31) and a maximum step complexity of 260 (ratio 1.73).

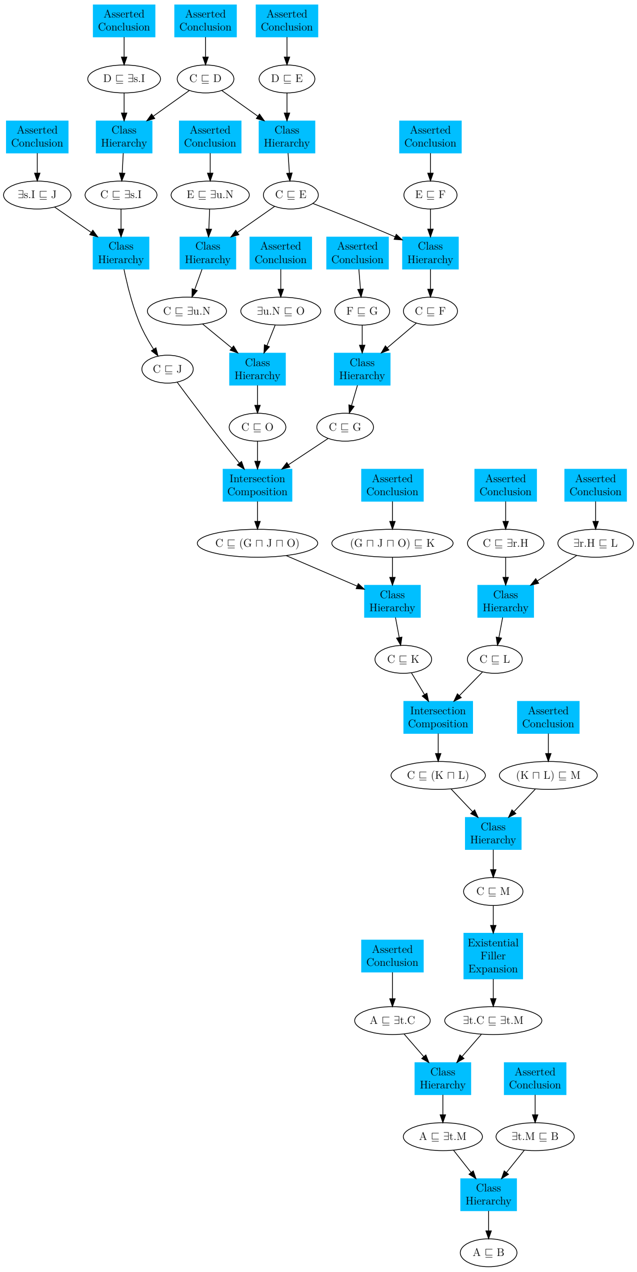

Fig. 6. From the other end of the spectrum, here is a task with a large justification size (14) and justification complexity (900). The ELK proof has size 29 (ratio 2.07), average step complexity 196 (ratio 0.22), and maximum step complexity 280 (ratio 0.32). The proof is a little too detailed, but the justification alone does not provide enough information for easy comprehension.

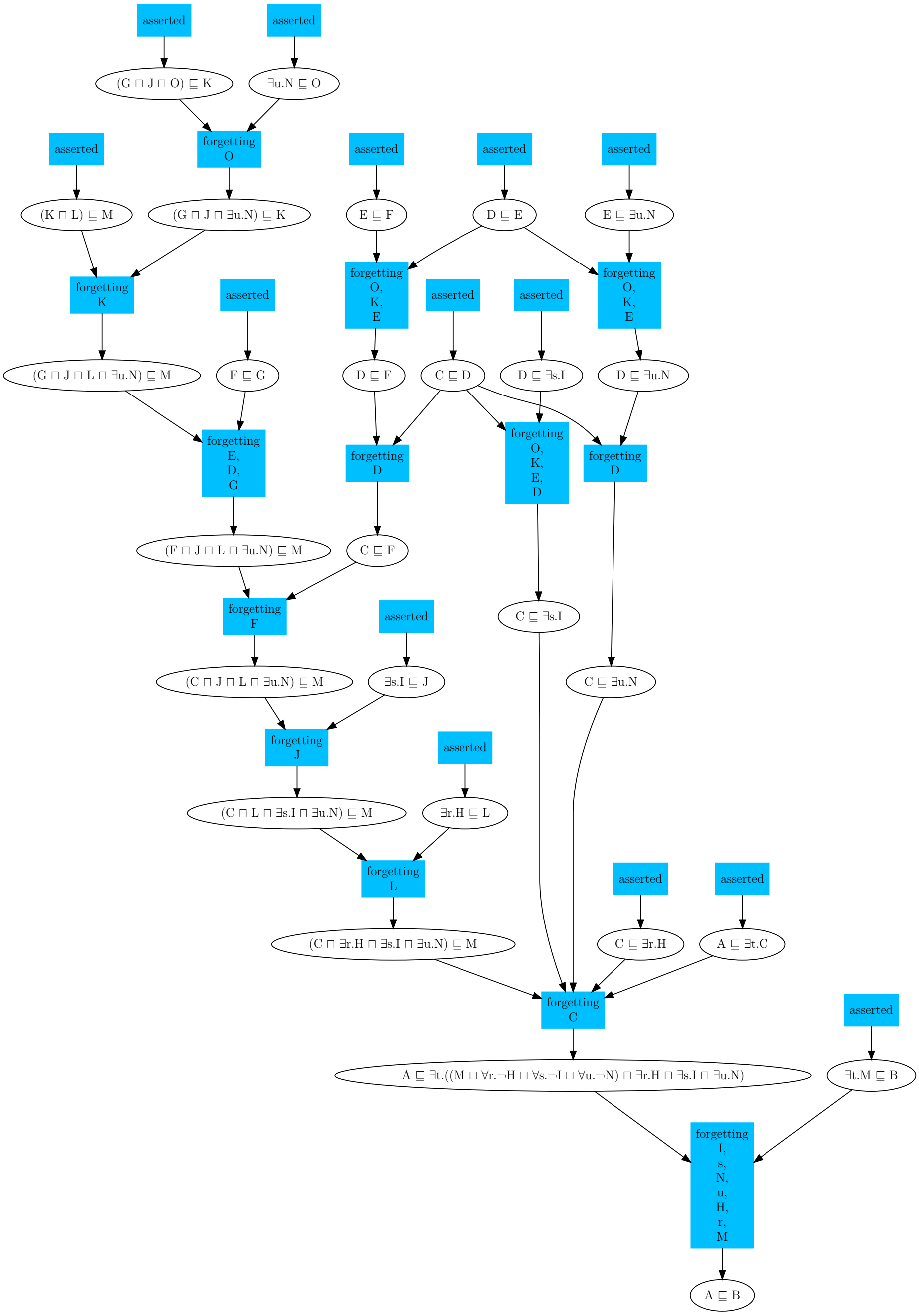

Fig. 7. The proof generated by LETHE for the same task as in Figure 6. It uses more complex intermediate expressions, but a different forgetting order could have produced a similar proof as the one generated by ELK. The proof has size 27 (ratio 1.93), average step complexity 275 (ratio 0.31), and maximum step complexity 400 (ratio 0.44).