Beyond LLM-Guided Common-Sense Reasoning for Natural Language Understanding

Moritz Bayerkuhnlein[1,*], Julian Britz[1] and Diedrich Wolter[1]

[1]Universität zu Lübeck, Institute for Software Engineering and Programming Languages, Ratzeburger Allee 160, 23562 Lübeck, Germany

Abstract

Mastering common-sense reasoning is often regarded as a major obstacle towards natural language understanding (NLU). The growing availability of large-scale knowledge bases and automated theorem provers has motivated studies to determine to which extent common sense knowledge as required in NLU is already provided by existing knowledge bases. However, large-scale knowledge bases may comprise too many axioms to be handled by an automated theorem prover. In 2024, Claudia Schon presented a paper at the FCR workshop demonstrating that similarity of language as captured by the embedding space of a large language model (LLM) can provide an effective heuristic for selecting relevant axioms. The work presented in this paper is motivated by her findings. We first show in a reproduction study that the results not only apply to theorem prover E, but also to prover Vampire. An analysis of the results reveals a complementary heuristic that identifies generally important axioms automatically, improving the performance of reasoning significantly.

Keywords

common-sense reasoning, natural language understanding (NLU), automated theorem proving, large language model (LLM)

1. Introduction

Natural language is inherently ambiguous, requiring listeners and readers to fill in missing pieces of information from context, their understanding of the world, and by logical inference. The importance of common-sense reasoning to interpret natural language has thus often been stressed in the literature, prominently in the works by Gary Marcus and Ernest Davis [1, for example]. We characterize reasoning in natural language understanding as the task of constructing a model in the sense of logic that is consistent with given natural language input as well as context and background knowledge. Setting aside the questions of which logic may be most suitable to represent semantics of language and how knowledge about language can be acquired, we are interested in computational principles that allow a logic model of a given natural language phrase to be constructed. As has already been shown by Schon [2, 3], this often cannot be accomplished by automated reasoning tasks alone, simply due to the high computational cost. Overcoming this computational bottleneck is a requirement to assess the extent of common-sense required in NLU tasks that is already covered by available knowledge bases.

Throughout the last decades of AI, several attempts have been made to compile common-sense knowledge bases on various levels of abstraction. Well-known examples on the side of fine-grained knowledge bases include the Cyc project [4] and its variants such as Next-KB [5]. On the side of general ontologies, SUMO [6] and DOLCE [7] provide formal knowledge that is also intended to provide the necessary glue to connect natural logic to formal reasoning. In case of the ontologies mentioned, a classic one (binary truth) is used as underlying logic and a connection to first-order theorem provers is possible. In this paper we consider Adimen-SUMO [8], a re-engineering of SUMO in first-order logic such that the axioms can be used with off-the-shelf automated theorem provers.

Perspectives on Humanities-Centred AI and Formal & Cognitive Reasoning Workshop 2025, (CHAI 2025 & FCR 2025), Joint Workshop at the 48th German Conference on Artificial Intelligence, September 16, 2025, Potsdam, Germany

*Corresponding author.

moritz.bayerkuhnlein@uni-luebeck.de (M. Bayerkuhnlein); diedrich.wolter@uni-luebeck.de (D. Wolter)

0000-0002-0919-9947 (M. Bayerkuhnlein); 0000-0001-9185-0147 (D. Wolter)

Applying automated theorem provers to large-scale knowledge bases quickly reveals a major challenge: The sheer size of a knowledge base – in conjunction with high computational costs of automated theorem proving – limits applicability. Clearly, not all elements of a knowledge base are relevant for proving consistency of a candidate model. Selecting the subset of relevant axioms offers means to improve applicability of theorem proving. But why should it be possible to design a clause selection method that is more effective than the advanced techniques already implemented in automated theorem provers? A potential answer to this question lies in the kind of proofs the theorem prover is requested to perform. Consider the task of proving that leaving your bike unlocked increases the likelihood of the bike being stolen. A potential reasoning chain would likely connect concepts such as "lock" as a means for fixing otherwise movable objects, "thieves" re-locating objects while overcoming security measures. In a large corpus of text we expect to find such reasoning chains to be spelled out or at least hinted at, hence the co-occurrence of "unlocked bike", "theft", and "stolen bike" is likely to occur.

Such co-occurrence is exactly what semantic similarity as determined by LLMs represents. By inspecting the embedding of a given sentence, i.e., its mapping to a neural activation of the LLM, it might thus be possible to retrieve related words or phrases. Schon [3] proposed selecting axioms that pin down the semantics of such words, which was shown to be effective in an experimental study.

In this paper we re-visit the approach by Schon [3] and reproduce the study, also considering an additional theorem prover. We are motivated to identify further indicators for relevant axioms that an automated theorem prover cannot discover on its own. In particular, we are motivated to study whether intrinsic properties of a knowledge base also provide means to identify relevant axioms. Our approach is motivated by the hypothesis that a manually constructed knowledge base would be orthogonal in the sense of stating general axioms exactly once. If such *core axioms* exist which are used in many proofs over a given knowledge base, it would be important to feed them to the theorem prover. To sum up, our work addresses the following hypotheses:

Hypothesis 1 A language model that captures semantic similarity in terms of alternative wordings can also estimate relevance of axioms in a knowledge base that serves as a lexicon in natural language understanding.

Hypothesis 2 A manually designed knowledge base contains relevant axioms that cannot be discovered from semantic similarity alone, but which are essential in the construction of a valid model.

The contribution of this paper is to describe an experimental analysis using the Adimen-SUMO ontology and theorem provers E [9] and Vampire [10]. We demonstrate the existence of core axioms in Adimen-SUMO, i.e., a set of axioms that must be considered for achieving a proof.

2. Preliminaries and Problem Statement

Traditionally, the domains for automated reasoning, planning, and understanding have been narrow and application specific. There have been monumental efforts in the common-sense reasoning community to estimate and capture the knowledge that is relevant to make reasoning about the everyday world possible. The largest and longest-standing project (in all of artificial intelligence) is Cyc, an ontology that encodes everyday concepts in a logic-like language. Unfortunately, Cyc is closed source and not integrated with standard theorem provers. In this study we focus on the Adimen-SUMO ontology [8], a re-engineering of SUMO's upper and mid-level ontology in first-order logic which is suitable for automated theorem proving.

Atomic truths in Adimen-SUMO are represented as *axioms*, which can be used to prove theorems. Adimen-SUMO contains around 8000 axioms. Similar to theorem proving in other domains, the amount of axioms provided increases the potential search space for a theorem prover, making it difficult to find proofs in a reasonable time frame. For common-sense reasoning in the context of natural language understanding, this is especially problematic, as often implications are omitted from the natural language sentences, which makes it difficult to identify the relevant axioms for a given proof task. Theorem

proving has thus to rely on axiom selection, a technique to select which axioms will be considered in a proof attempt.

2.1. Axiom Selection

Every theorem will require a different set of axioms to be proven, which encourages the use of selection strategies to favor dynamic selection over selecting axioms based on fixed rules. These techniques are also referred to as *clause selection* since the conjunction of all axioms constitutes the background knowledge for a proof. While clause selection appears to be a more natural term when considering the task from a theorem prover's point of view, axiom selection better captures the outside view of filtering which axioms from a common sense ontology are passed to the theorem prover.

A prominent example of such a strategy is the SUMO Inference Engine (SInE) selection strategy [11]. The SInE strategy is a trigger-based method for axiom selection in automated theorem proving. It begins with symbols from the conjecture and recursively selects axioms containing them, using a trigger condition that avoids overly common symbols. A benevolence parameter b allows slightly more frequent symbols to trigger axioms, improving flexibility. Clearly, the SInE strategy is based on syntactic similarity of symbols, which is why we refer to it as **syntactic axiom selection**.

The widespread use of LLMs enables axiom selection based on so-called semantic similarity. Trained on vast text corpora, LLMs capture co-occurrence relationships between words and phrases which are referred to as semantic relationships. Assuming common-sense knowledge bases to reflect concepts from natural language as part of the LLM training data, LLMs can be applied to select axioms based on a semantic similarity of the concepts involved. Computationally, concepts are mapped by the LLM to an (internal) embedding space. Embedding spaces are real-valued vector spaces that are commonly equipped with a cosine similarity $u^T v \cdot ||u||^{-1} \cdot ||v||^{-1}$ for $u, v \in \mathbb{R}^n$. In vector-based selection, axioms are selected from an averaged embedding of individual symbols. The SeVEn (Sentence-based Vector Encoding) extends this approach by embedding entire axioms as natural language sentences. Each axiom A from a knowledge base KB is first translated into a natural language sentence $S = t(A)$, which is then encoded into a vector $v_S(A) = f(S)$ using a sentence embedding model. The resulting sentence-based representation of the knowledge base, $v_S(\text{KB})$, maintains the structure of vector-based selection while enabling deeper semantics, which is why we refer to it as **semantic axiom selection**.

Building on both approaches, [3] proposed a hybrid strategy that unifies semantic and syntactic selection. The idea is to first apply SeVEn to select axioms semantically related to the conjecture. Then, SInE is applied to this enriched conjecture set, adding further axioms based on syntactic triggers.

Formally, let SInE(KB, $\{A_1, \ldots, A_n\}$) denote the set of axioms selected by SInE given a set of conjectures, and SeVEn(KB, C) denote the axioms selected by SeVEn for a conjecture C. The combined selection strategy is defined as a **union** of both methods.

2.2. Problem Statement

Like [3], we consider white box proofs in the Adimen-SUMO ontology [8]. In Adimen-SUMO, truth tests are provable conjectures used to check logical consistency and reasoning accuracy. A set of 8010 truth-tests has been automatically constructed [12], from which we randomly selected a subset of 1000 for the experiments. We count how many proofs the theorem prover will be able to achieve using different axiom selection strategies. Our research objective is to identify an axiom selection method that allows most proofs to be achieved. As a baseline we consider the theorem's prover built-in selection techniques.

3. Experiments

We compare the following axiom selection strategies:

standard full ontology baseline without any pre-selection, i.e., only using the prover's built-in method;

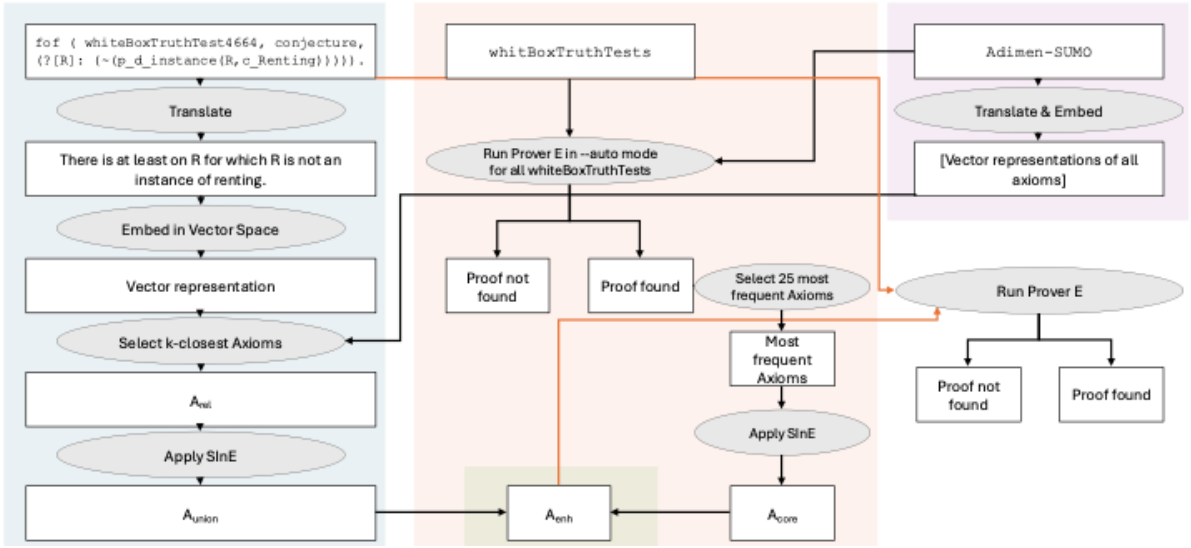

Figure 1: Overview of the selection process. Steps of the approach proposed by [3], combining syntactic and semantic selection (left and right). Our contribution is the identification and integration of *core axioms* into the hybrid selection pipeline (middle).

syntactic axiom selection using the SInE algorithm;

semantic axiom selection using SeVEn based on semantic similarity, computed via sentence embeddings;

union combined semantic and syntactic selection approach following [3] and

enhanced-union extension of the union method, enriched with frequently used axioms (core axioms)

3.1. Validation of Prior Work

Our experiments validate the findings of [3], with nearly identical results (Figure 3). The slight difference is due to the sampling of 1,000 White-box Truth tests. The syntactic selection strategy, via SInE was run with a benevolence of 3 and recursion depth of 2. Semantic selection using SeVEn with a threshold of 1,500 axioms (increased from prior work to match the average output of SInE) powered by a pre-trained all-MiniLM-L6-v2[1] sentence embedding model, and a hybrid union approach that combines both methods. The hybrid method employs the pre-trained all-MiniLM-L6-v2 sentence embedding model, selecting the 160 closest axioms per conjecture[2].

3.2. Core Axioms

The combined use of syntactic and semantic selection methods fails to cover a wide enough range of proofs. We observe that conjectures without successful proofs often show higher cosine similarity to their 160 nearest axioms, suggesting that overly specific selections can exclude essential axioms. Since theorem provers rely on axioms to guide simplification steps, missing such axioms hinders proof discovery. To address this, Prover E was run in auto mode on all $8,010$ White-box Truth tests, allowing it to optimize SInE parameters per conjecture. The 25 axioms most frequently used in successful proofs were then identified as *core axioms* (see Figure 2 for an example).
As such we obtain an axiom selection strategy featuring the core axioms discovered in the previous step, which we refer to as `enhanced`. To improve core axiom selection without greatly increasing their number, a SInE strategy was applied to the 25 core axioms (benevolence 1, depth 1), an overview is

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[2] Parameters and models are based on the results of [3]

```
fof(mergeA594, axiom,
  ![REL]: (p__d__instance(REL, c__BinaryRelation) =>
    (p__d__instance(REL, c__IrreflexiveRelation) <=>
      (![INST]: ~p__d__holds3(REL, INST, INST))))
)
```

Figure 2: Example of a core axiom discoved using the White-box Truth tests in TPTP-FOF representation, defining the irreflexive property of binary relations. Stating that if a relation REL is a binary relation, then it is irreflexive if and only if there is no instance INST such that REL holds between INST and itself. In other words, an irreflexive relation never relates an entity to itself.

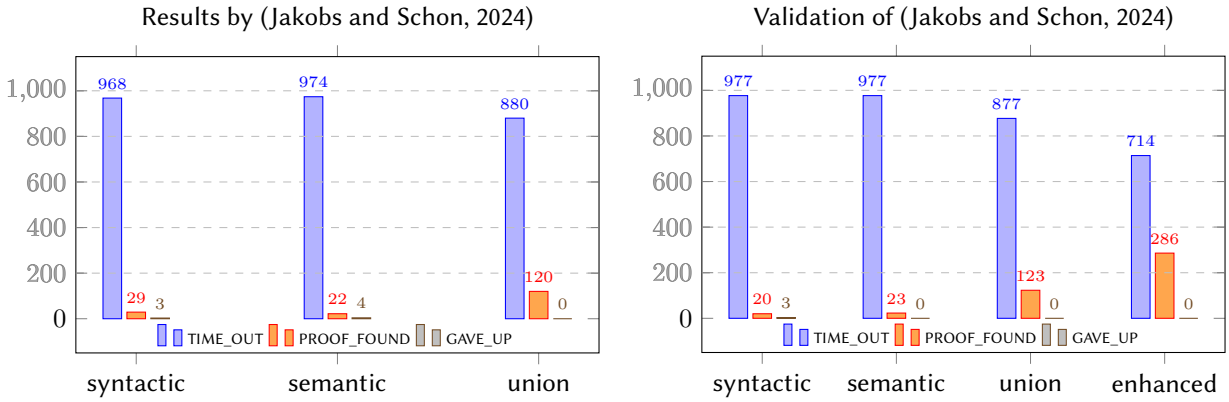

Figure 3: Side-by-side comparison of validation results from Jakobs and Schon [3]. TIME_OUT indicates the prover exceeded the 15s CPU limit, PROOF_FOUND denotes success, and GAVE_UP means the prover stopped searching due to internal heuristics.

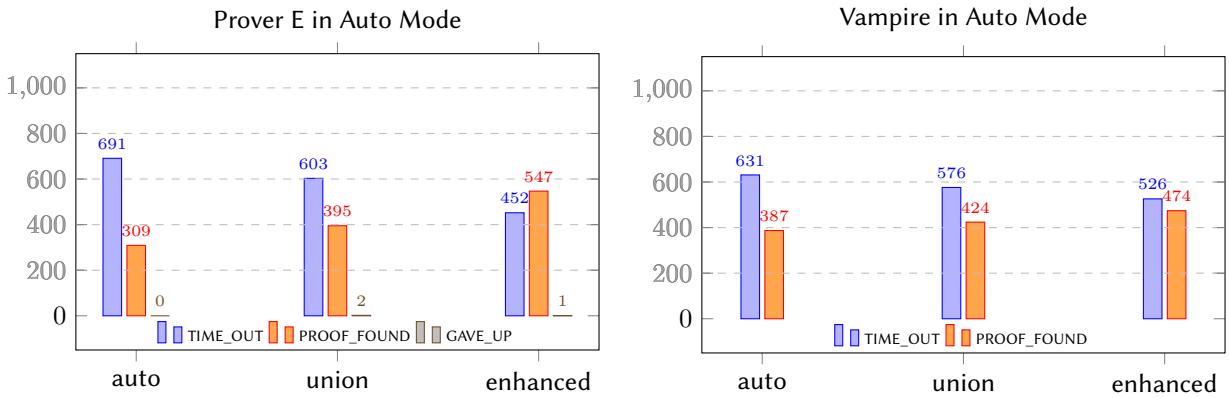

Figure 4: Axiom selection in combination with an automatic theorem prover (ATP) in Auto Mode, as used during CASC.

shown in Figure 1. The resulting axioms were combined with those from the union-based method, balancing frequency and structural relevance. On the same 1,000 White-box Truth tests, this enhanced approach yielded 286 proofs (see Figure 3).

3.3. Running ATP in Competition Mode

To explore how internal heuristics interact with external axiom selection we evaluate different automatic theorem provers using their automatic mode also used during CASC competitions. In addition to Prover E which was used in the original study we selected Vampire 4.9 [10] due to being a current and long-standing winner of the CASC competitions [13] FOF (First-order-form) bracket.

From the results evaluating Prover E in its competition mode, we obtained three subsets of the

Table 1
Cardinality of White-Box truth test subsets and average proof times (in seconds) for each method. The sets $\mathcal{W}_{\texttt{auto}}$, $\mathcal{W}_{\texttt{union}}$, and $\mathcal{W}_{\texttt{enhanced}}$ correspond to the subsets of truth tests for which a proof was found by the respective selection strategy.

Subset	Cardinality	`auto`	`union`	`enhanced`
$\mathcal{W}_{\texttt{auto}}$	309	5.48	0.55	0.40
$\mathcal{W}_{\texttt{union}}$	395	–	1.30	1.36
$\mathcal{W}_{\texttt{enhanced}}$	547	–	–	1.66

White-box Truth tests $\mathcal{W}$, identified by the axiom selection methods $\mathcal{W}_{\texttt{auto}}, \mathcal{W}_{\texttt{union}}$ and $\mathcal{W}_{\texttt{enhanced}}$. We can report that the obtained subsets are in fact subsets of their cardinally superior set, such that $\mathcal{W}_{\texttt{auto}} \subset \mathcal{W}_{\texttt{union}} \subset \mathcal{W}_{\texttt{enhanced}} \subset \mathcal{W}$.

Furthermore, the versions featuring the additional selection strategies of `union` and `enhanced` result in comparably faster proof times on average. Overall, restricting the search space through a combined syntactic and semantic selection process, supplemented by core axioms, consistently improves the prover's performance. In automatic mode, the average time to find a proof was 5.48 seconds; with the selection strategies in place, it was reduced to less than a second (see Table 1). A slight increase was measured from extending the `union` strategy with the `core axioms`, which can be explained by simply having slightly more axioms to choose from for the set $\mathcal{W}_{\texttt{union}}$. Finally, presumably the largest of the subset $\mathcal{W}_{\texttt{enhanced}}$ showed a reduced average time to proof with *core axioms* in place compared to the automatic mode.

3.4. Discussion

While we have demonstrated the existence of these core axioms for proving conjectures in common-sense ontologies, there are some clear limitations.

First of all, the White-box Truth tests used to evaluate the selection strategies are designed to validate theorem provers on formal ontologies like Adimen-SUMO. As of now, similar truth tests that reflect natural language expressions directly are not available in the same quantity or quality. This makes it unclear to which extent the effects we observe, especially the impact of core axioms, will carry over to an actual NLU setting, or if they are specific to this kind of synthetic evaluation.

Second, we can only extract core axioms from successful proof attempts. This introduces a bias toward axioms that appear in *easy* or already solvable proofs. Axioms that are essential but only show up in harder proofs, or in cases that currently fail, are missed entirely by this method.

That said, our results suggest that there is an effect, i.e., some axioms appear to act as bridges, filling in gaps that neither syntactic nor semantic selection strategies can easily cover. In other words, demonstrating the existence of such core axioms shows that several axioms cannot be identified using LLMs. These axioms are not just frequent, they appear to carry structural or semantic importance that helps reasoning succeed. Being able to identify likely core axioms in advance could help improve reasoning performance in settings where no prior proof data is available, and could enable logic-based NLU.

4. Conclusions and Future Work

In this study we set out to reproduce a study by Schon [3] in order to get a deeper understanding of how word similarity applied to symbol names helps to guide an automated theorem prover. We were able to reproduce the results, also using Vampire as alternative theorem prover. These findings suggest that Hypothesis 1 holds, i.e., language similarity measures derived from co-occurrence offers a helpful heuristics for activating axioms or sub-theories that are relevant to prove a given fact.

We could identify a set of axioms from Adimen-SUMO which often is used in successful proofs. Explicitly activating this set of axioms for every proof significantly increased the proof success rate.

This suggests that Hypothesis 2 holds too, i.e., there exists a set of axioms specific to the underlying knowledge-base that is not identified by word-level similarity as it represents general concepts.

Taking both observations together and setting them into context with previous findings by Schon [3], we may conclude that there exists significant opportunities for improving automated theorem proving when applied to commonsense reasoning. In future work we wish to shift attention more towards NLU by using reasoning to infer models (in a logic sense) from given sentences. Due to language leaving out pieces of information that are obvious to humans, this step requires means to select not only axioms but also facts that are related to a given sentence, considering background knowledge as well as context.

Declaration on Generative AI

Section “Preliminaries” was translated and rephrased using GenAI Tools.

References

[1] E. Davis, G. Marcus, Commonsense reasoning and commonsense knowledge in artificial intelligence, Communications of the ACM 58 (2015) 92–103. doi:10.1145/2701413.

[2] C. Schon, Using the meaning of symbol names to guide first-order logic reasoning, in: FCR@KI 2024, 2024, pp. 19–27.

[3] O. Jakobs, C. Schon, Context-specific selection of commonsense knowledge using large language models, in: KI 2024: Advances in Artificial Intelligence, 2024, pp. 218–231.

[4] D. Lenat, R. Guha, Cyc: A midterm report, Communications of the ACM 33 (1990) 32–49.

[5] K. D. Forbus, T. Hinrichs, Analogy and qualitative representations in the companion cognitive architecture, AI Magazine 38 (2017) 34–42.

[6] I. Niles, A. Pease, Toward a standard upper ontology, in: C. Welty, B. Smith (Eds.), Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), 2001, pp. 2–9.

[7] S. Borgo, R. Ferrario, A. Gangemi, N. Guarino, C. Masolo, D. Porello, E. M. Sanfilippo, L. Vieu, A. Galton, O. Kutz, DOLCE: A descriptive ontology for linguistic and cognitive engineering, Applied Ontology 17 (2022) 45–69. doi:10.3233/AO-210259.

[8] J. Álvez, P. Lucio, G. Rigau, Adimen-SUMO: Reengineering an ontology for first-order reasoning, Int. J. Semant. Web Inf. Syst. 8 (2012) 80–116. doi:10.4018/jswis.2012100105.

[9] S. Schulz, S. Cruanes, P. Vukmirovic, Faster, higher, stronger: E 2.3, in: CADE 27 – 27th International Conference on Automated Deduction, 2019, pp. 495–507.

[10] L. Kovács, A. Voronkov, First-order theorem proving and Vampire, in: N. Sharygina, H. Veith (Eds.), Computer Aided Verification, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–35.

[11] M. Suda, Aiming for the goal with SInE, in: Vampire 2018 and Vampire 2019: The 5th and 6th Vampire Workshops, volume 71 of *EPiC Series in Computing*, 2020, pp. 38–44.

[12] J. Álvez, M. Hermo, P. Lucio, G. Rigau, Automatic white-box testing of first-order logic ontologies, J. Log. Comput. 29 (2019) 723–751. doi:10.1093/LOGCOM/EXZ001.

[13] G. Sutcliffe, The CADE ATP system competition – CASC, AI Magazine 37 (2016) 99–101. doi:10.1609/AIMAG.V37I2.2620.