On Datasets for Evaluating Architectures for Learning to Reason

Naveen Sundar Govindarajulu,[1] **Jean-Claude Paquin, Shreya Banerjee, Paul Mayol, Selmer Bringsjord***
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180, USA
[1]naveensundarg@gmail.com

Abstract

In our poster, we will introduce new datasets in propositional logic and first-order logic that can be used for learning to reason, and present some initial results on systems that use this data.

Introduction

There is a growing research interest in incorporating learning in reasoning systems. Such efforts fall largely into two different areas that we term **Area I** and **Area II**. We give a quick overview of what a reasoning system does before describing these two areas.

In general, a reasoning system can be modeled as a search through some space. This search usually relies on a number of hand-written heuristics. Theorem provers make this quite explicit, as one can specify these heuristics as an end-user. For instance, in first-order resolution theorem provers, the goal is to find a sequence of resolution operations using an initial set of clauses C that results in an empty clause. At any point in the search, the prover has to choose a set of clauses from an overall set of clauses it has derived. Theorem provers use heuristics such as the size of a clause, the complexity of a clause, age of a clause etc. to choose a clause.

Efforts in **Area I**, such as (Kaliszyk, Chollet, and Szegedy 2017), revolve around selecting or computing an appropriate set of heuristics using some form of learning while not tampering with the rest of the search process. Efforts in **Area II** aim to learn a function from scratch that does the entire search. While there has been quite significant progress in Area I, there has been very little progress in Area II. We feel that one of the main reasons for this state of affairs, is that there is no standard dataset that can be leveraged for Area II. Datasets for Area I are built up with learning heuristics as a goal and are not ideal for Area II systems.

In our poster, we will discuss new datasets in propositional logic and first-order logic that can be used for learning to reason and present some initial results based on this data.

*ONR

Why New Datasets?

Existing datasets that can be used for learning to reason are either too complex or do not show much variation in the samples. For example, the Mizar repository (Naumowicz and Kornilowicz 2009) has more than $50,000$ reasoning problems (theorems in a first-order logic) and have been used in **Area 1**, but these problems are too complex to be useful in bootstraping a learning system from scratch. On the other hand, simpler toy datasets such as the deduction task in bAbI (Weston et al. 2015) do not show that much variation. For example, figures 1a and 1b show answers and full proofs to two different questions in the bAbI dataset represented in the Slate proof assistant system (Bringsjord et al. 2008). Both the proofs can be obtained by applying the resolution inference rule twice. Moreover, the proofs are structurally similar and can be generated by switching constant symbols in a first-order proof. In fact, the entire dataset in the bAbI deduction task follows this one single proof pattern. Ideally, we want a dataset that has problems of varying levels of complexity.

The Problem

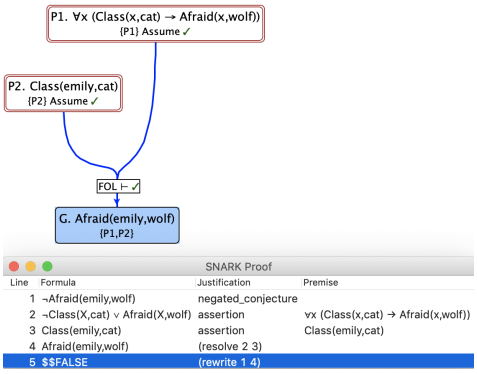
We present an abstract version of the problem we seek to solve. Assume that we have a formal logic $\mathcal{F} \equiv \langle L, I, \bot \rangle$, where L is the language of the formal logic, I is the inference system and $\bot \in L$. Any set of formulae Γ in the language L can be *consistent*, $\Gamma \nvdash_I \bot$, or *inconsistent* $\Gamma \vdash_I \bot$. Any reasoning problem $\Gamma \vdash \phi$ can be posed as a consistency problem $\Gamma + \neg\phi \vdash_I \bot$ if certain conditions C_1, C_2 are satisfied by I (Boolos, Burgess, and Jeffrey 2003).

C_1 The deduction theorem $\Gamma + \phi \vdash \psi \Rightarrow \Gamma \vdash \phi \rightarrow \psi$.

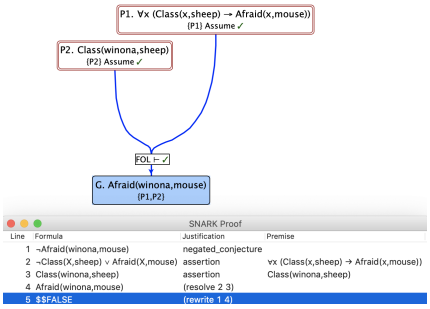
C_2 The law of excluded middle. $\{\} \vdash \phi \vee \neg\phi$.

In fact, resolution-based theorem provers such as Vampire (Kovács and Voronkov 2013) function in this manner by searching for a proof of $\bot$ from $\Gamma + \neg\phi$ in order to prove $\Gamma \vdash \phi$. Therefore, reasoning in logics with $\mathcal{C}$ can be reduced to consistency checking.

We pose the learning problem as a standard classification task. Let $con(\Gamma) \in \{0, 1\}$ denote whether Γ is consistent $con(\Gamma) = 1$ or not $con(\Gamma) = 0$. Given a training data D_{train} of sentences and their consistency information,



(a) **bAbI 1** Emily is a cat. Cats are afraid of wolves. What is emily afraid of?



(b) **bAbI 2** Winona is a sheep. Sheep are afraid of mice. What is winona afraidof?

Figure 1: bAbI problems

$\{\langle \Gamma_1, con(\Gamma_i)\rangle \mid 1 \leq i \leq n\}$, the goal is to learn a function that approximates con and is evaluated on a test set D_{test}.

Preview: Data and Data Generation

We look at two different formal logics: propositional calculus and first-order logic. We randomly generate sentences and their consistency labels. The generation process is slightly different for the two different logics.

Propositional Logic

For propositional logic, we generate formula in conjunctive normal form. Each formula ϕ is the form of a disjunction $l_1 \ldots l_i \ldots l_n$. Each literal l is an atom P or its negation P. We generate a radom set of formulae Γ by randomly generating u clauses $c_1, c_2, \ldots, c_u$ where each clause has v random literals drawn from a set of w atomic propositions. For each such random set of sentences, we run a theorem prover to check whether the sentence is consistent or not. Using this method, we have three distinct datasets for propositional logic with $u, v, w = 3$, $u, v, w = 4$ and $u, v, w = 5$.

First-order Logic

First-order Logic Due to the expressivity of first-order logic, naïve random generation of formulae can quickly lead to very difficult to solve problems or degenerate problems that do not have real-world analogs but are also difficult to solve. For instance, biconditionals such as $\phi \leftrightarrow \phi$, where ϕ has two or more nested quantifiers, can cause some state-of-the-art theorem provers into running for an unbounded amount of time. To address this, we use sorted first-order logic with a given set of relation, function, and constant symbols, along with certain complexity constraints. The sorts prevent us from generating nonsensical formulae. Given a sorted signature σ, we generate a certain number of unique formulae Γ and apply first-order natural deduction inference rules till we have a proof ρ of a certain complexity. One such problem and a corresponding proof in an imaginary mechanical domain is given below. See Figure 2.

Sample FOL Problem

1. If gear x is connected to gear y and gear y is connected to gear z, then gear x is connected to gear z.
2. If gear x and gear y are connected, and gear x is broken, gear y is broken too.
3. Gear 1 is connected to gear 2.
4. Gear 2 is connected to gear 3.
5. Gear 3 is enclosed by box 1.
6. All boxes are connected.
7. Lever 1 either fixes or breaks all gears.
8. Lever 1 breaks gear 1.
9. If x encloses y and y is broken, x is broken.

Question: Is box 20 broken? **Answer**: Yes

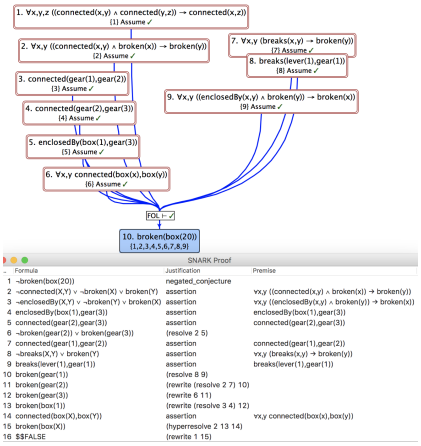


Figure 2: **Proof for the Sample FOL Problem** A resolution proof for the sample FOL problem given above.

References

Boolos, G. S.; Burgess, J. P.; and Jeffrey, R. C. 2003. *Computability and Logic (Fifth Edition)*. Cambridge, UK: Cambridge University Press.

Bringsjord, S.; Taylor, J.; Shilliday, A.; Clark, M.; and Arkoudas, K. 2008. Slate: An Argument-Centered Intelligent Assistant to Human Reasoners. In Grasso, F.; Green, N.; Kibble, R.; and Reed, C., eds., *Proceedings of the 8th International Workshop on Computational Models of Natural Argument (CMNA 8)*, 1–10. Patras, Greece: University of Patras.

Kaliszyk, C.; Chollet, F.; and Szegedy, C. 2017. Holstep: A Machine Learning Dataset for Higher-order Logic Theorem Proving. *arXiv preprint arXiv:1703.00426*.

Kovács, L., and Voronkov, A. 2013. First-order theorem proving and vampire. In *International Conference on Computer Aided Verification*, 1–35. Springer.

Naumowicz, A., and Kornilowicz, A. 2009. A Brief Overview of Mizar. In Berghofer, S.; Nipkow, T.; Urban, C.; and Wenzel, M., eds., *Theorem Proving in Higher Order Logics*, volume 5674 of *Lecture Notes in Computer Science (LNCS)*. Berlin: Springer. 67–72.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.