Identifying Bottlenecks in Practical SAT-Based Model Finding for First-Order Logic Ontologies with Datasets

Shirly Stephen, Torsten Hahmann
School of Computing and Information Science,
University of Maine, Orono, Maine 04473
{shirly.stephen, torsten.hahmann}@maine.edu

Abstract

Satisfiability of first-order logic (FOL) ontologies is typically verified by translation to propositional satisfiability (SAT) problems, which is then tackled by a SAT solver. Unfortunately, SAT solvers often experience scalability issues when reasoning with FOL ontologies and even moderately sized datasets. While SAT solvers have been found to capably handle complex axiomatizations, finding models of datasets gets considerably more complex and time-intensive as the number of clause exponentially increases with increase in individuals and axiomatic complexity. We identify FOL *definitions* as a specific bottleneck and demonstrate via experiments that the presence of many defined terms of the highest arity significantly slows down model finding. We also show that removing *optional definitions* and substituting these terms by their definiens leads to a reduction in the number of clauses, which makes SAT-based model finding practical for over 100 individuals in a FOL theory.

Introduction

First-order logic is one of the standards for describing formal ontologies in many domains. However, an ontology can only serve its purpose if we can verify its logical consistency in two forms: (1) check its *internal consistency* to rule out contradictions by generating some model, and (2) check its *external consistency* with datasets that are representative of the ontology's intended purpose. Existing work in FOL ontology consistency checking is limited to satisfiability verification of terminological axioms (TBox), and they rarely contain facts/assertions about individuals (i.e., they typically lack an ABox). Model finding for FOL ontologies is not only theoretically incomplete but has not been very successful in practice either except for tiny, often trivial models with less than 20 individuals. In this work we identify a specific bottleneck that can be remedied for model finding to scale better in practice despite its theoretical undecidability and intractability. Our experiments are specifically designed to test the hypothesis that: "*Additional defined terms negatively impact model finding time, and rewriting ABox facts that use optional definitions with their definiens can speed and scale up model finding in practice*".

Preliminaries

A FOL ontology $\mathcal{T}$ is a set of FOL sentences. All nonlogical symbols in $\mathcal{T}$, i.e., constants, functions, and predicates, form its *signature*, denoted by $\lambda(T)$. Each $\mathcal{T}$ admits a set of *interpretations* over a nonempty domain $\mathcal{D}$ of individuals. Any interpretation assigns each constant in $\lambda(\mathcal{T})$ an individual in $\mathcal{D}$ and each n-ary predicate Ω in $\lambda(\mathcal{T})$ a relation $\Psi(\Omega) : \mathcal{D}^n \rightarrow \{\text{True}, \text{False}\}$. An interpretation $\mathcal{I}$ for which all sentences in $\mathcal{T}$ are true is called a *model*. To facilitate model finding using SAT solvers, a FOL ontology is typically converted to an equisatisfiable clausal normal form (FOL-CNF), where each FOL sentence is represented as a universally quantified conjunction of disjunctions of atoms. A SAT-based model finder then propositionalizes all FOL-CNF clauses by instantiating them, for a fixed domain size, d. Then each FOL-CNF clause leads to an exponentially growing number d^v of propositional clauses, where v is the number of variables in the FOL-CNF clause. The number of propositional variables in the SAT representation is highly influenced by the number and arity of predicates: each predicate of arity n results in d^n propositional variables. The number of propositional variables determines the search space, which consists of 2^{V_P} possible interpretations. However, the search space can, be greatly reduced if the number of predicates of highest arity is kept low. This is the idea tested here experimentally, leveraging the fact that many ontologies have lots of defined predicates, which we can easily dispense off during model finding while still keeping the semantics of the ontology intact.

Formalization

Knowledge in a FOL ontology can be divided into an ABox, and TBox. The ABox contains class and relational assertions about individuals, and is a set of ground clauses. The TBox contains terminological axioms that describe the concepts and their roles within a domain of interest. We adapt a distinction of three types of sentences in a FOL ontology to formalize the ideas central to our experiments.

- An *explicit definition* is a special type of TBox sentence of the form $\forall x_1, \ldots, x_n[\Omega(x_1, \ldots, x_n) \leftrightarrow \alpha(x_1, \ldots, x_n)]$, where Ω is the defined n-ary predicate, and α is a formula with x_1 to x_n as free variables and $\lambda(\mathcal{T}) \setminus \Omega$ as the only nonlogical symbols.

- *Optional definitions* are explicit definitions of predicates not used in other sentences in the TBox.
- A *defined assertion* is a sentence $\sigma \in$ ABox, such that all occurrences of predicates Ω_i in the sentence that are optional definitions in the TBox are substituted by their definiens α_i.

Experimental Setup

To investigate the impact of *optional definitions* vs *defined assertions* on model finding we compare reasoning performance of the model finder Paradox (Claessen and Sörensson 2003) with 13 syntactic variants of an ontology (TBox + ABox) constructed as described below.

TBoxes: The experiments are conducted using a spatial ontology from the $CODI$ family (Hahmann 2018). The ontology relates spatial entities of various dimensions via qualitative spatial relations. Its medium size (26 axioms plus definitions) and complexity (see Table 1 for the number of variables and length of FOL-CNF clauses) make the ontology ideal for studying the effects of variations in model size, signature size, or number of axioms. For our experiments, the base case TBox, i.e., case 1 uses four primitives (i.e. undefined terms *Cont*, *Leq*, *S*, *ZEX*) and 12 defined terms (*MinDim*, *MaxDim*, *PointRegion*, *Point*, *Curve*, *ArealRegion*, *Lt*, *Gt*, *Geq*, *EqDim*, *Covers*, *P*) while the remaining TBoxes (cases 2-13) add combinations of five optionally defined terms (*PP*, *C*, *PO*, *Inc*, *SC*)[1] as shown in Table 1.

ABoxes: A map of lynx habitat in Maine forms the master ABox from which samples for all experiments are drawn. It consists of 173 spatial objects, each either a $Point$ (27), $Curve$ (90), or $ArealRegion$ (56), and 3,665 spatial relation assertions (245 positive ones and 3,420 negated ones). Samples ABoxes are constructed with increasing numbers of individuals (denoted by n), including all relational assertions that mention only the selected individuals, and distinctness assertions between the n individuals. Each experiment is averaged over 10 or more different samples for each sample size (i.e., domain size of sought models) with sizes currently ranging from 40 to 80. To ensure comparability of the results, the same ABox is relied upon for all 13 cases. The ABoxes only differ in that depending on the case's definition set, some facts are replaced by the corresponding defined assertions. That is, any of the 5 optional definitions not used in a specific case (i.e., the definition is not included in the TBox for that case) are substituted by their definiens, thus creating 13 variants of the same ABox. Hence the 13 cases (TBox + ABox) are logically equivalent, and they have the same number of models, but are syntactically different.

All experiments are run on an Intel Xeon CPU E5-2620 v3 at 2.40 GHz (with 12 cores) with 64GB RAM and 64bit Windows 10 Pro. Measured times are CPU time (see Table 1) for the process that runs Paradox.

Table 1: Summary of the TBoxes of the 13 cases experimented with. Each row represents one case, with the included optional definitions, statistics of the resulting FOL-CNF, and statistics of the propositionalized versions for samples sizes 40, 60, and 80. The abbreviations denote: C: FOL-CNF clauses; P_v: propositional variables; I: sample size (i.e., distinct individuals in the ABox samples); $\sim$: Paradox could not find a model.

TBox Case	**Optional Definitons**					**FOL - CNF** *#C*	Propositional CNF *#P_v*			**Model Finding Time** (TBox + ABox)		
	PP	C	PO	Inc	SC							
							$I = 40$	$I = 60$	$I = 80$	$I = 40$	$I = 60$	$I = 80$
1						65	13120	29280	51840	7	55	576
2	√	-	-	-	-	68	14720	32880	58240	10	62	560
3	-	√	-	-	-	68	14720	32880	58240	53	4500	8220
4	√	√	-	-	-	71	16320	36480	64640	52	1590	8710
5	-	-	√	-	-	68	14720	32880	58240	21	594	3780
6	√	-	√	-	-	71	16320	36480	64640	22	563	3910
7	√	√	√	-	-	74	17920	40080	71040	80	3020	17200
8	-	-	-	√	-	76	14720	32880	58240	125	3380	$\sim$
9	√	-	-	√	-	79	16320	36480	64640	159	4370	$\sim$
10	√	√	√	√	-	85	19520	43680	77440	1090	18500	$\sim$
11	-	-	-	-	√	72	14720	32880	58240	889	23000	$\sim$
12	√	-	-	-	√	75	16400	36600	64800	885	28800	$\sim$
13	√		√	√	√	92	21120	47280	83840	5740	$\sim$	$\sim$

Preliminary Results

Preliminary results show an exponential increase in runtime with increasing sample size, but an exponential increase in runtime is also observed as more optional definitions are included without any change in the number of possible interpretations. With all five optional definitions, the runtime increases by three order of magnitudes, e.g., from 7s to 5,740s for samples size 40, even though the number of propositional variables does not double. Thus, the elimination of optional definitions and the use of defined assertions can drastically speed and scale up model finding.

Conclusion and Future Work

This work is a first, if small, systematic study of the feasibility of model construction for a mid-sized but fairly complex FOL ontology with an ABox. The idea underlying our work is a novel approach to write FOL ABox assertions that helps scale model construction to models with 120 or more individuals. This can easily be implemented as a preprocessing step for existing model finders, with a corresponding postprocessing step that completes a model with interpretations for the (temporarily removed) defined predicates.

The results we obtained are good enough to leverage the use of *defined assertions* to further test (1) whether the approach extends to other model finders, such as iProver, that do not fully propositionalize the FOL ontology; and (2) to what extent the complexity and semantic restrictiveness of definitions impacts model finding performance.

Acknowledgement: This material is based in part upon work supported by the National Science Foundation under Grant Number III-1565811.

References

Claessen, K., and Sörensson, N. 2003. New techniques that improve MACE-style finite model building. In *Workshop on Model Computation at CADE 2003.*

Hahmann, T. 2018. Decomposition operations in a multidimensional theory of qualitative space. In *Proc. of Intern. Conf. on Formal Ontologies in Information Systems (FOIS 2018).*

[1] https://github.com/gruninger/colore/tree/master/ontologies/multidim_space_codi