

Finite Domain and Symbolic Inference Methods for Extensions of First-Order Logic

Johan Wittocx

*Dept. of Computer Science, K.U. Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium
E-mail: johan.wittocx@cs.kuleuven.be*

In this dissertation, we investigate various sorts of reasoning on finite structures and theories in the logic $\text{FO}(\cdot)$, a rich extension of classical logic with, amongst others, inductive definitions and aggregates. In particular, we study the tasks of constraint propagation, grounding, model revision, and debugging for $\text{FO}(\cdot)$.

Keywords: first-order logic, constraint propagation, grounding, finite model generation

1. Introduction

One of the long term goals of the research field Knowledge Representation and Reasoning is to build a *knowledge base system* (KBS). In such a system, knowledge about a domain of discourse is stored and different tasks are solved by applying various inference methods on that knowledge. An example is a KBS storing knowledge about course scheduling at a university. By applying suitable forms of inference, schedules can be generated automatically, hand-made schedules can be checked, existing schedules can be revised, etc., all using the same background knowledge. In our work, we contribute to the goal of building a KBS by investigating various forms of inference that are useful to solve many practical tasks.

Specifically, we study the following forms of inference in the context of finite structures: constraint propagation, grounding, model revision, and debugging. Also, we investigate symbolic, i.e., structure-independent, constraint propagation. As underlying logic, we use $\text{FO}(\cdot)$ [1]. This logic extends order-sorted first-order logic with inductive definitions, aggregates, integer arithmetic, and partial functions. While deductive query answering is well-known to be undecidable for FO and,

a fortiori, for the extension $\text{FO}(\cdot)$, the tasks we study are in **NP**.

In the rest of this abstract, we summarize our results for the various forms of inference. We will use the following terminology. An n -ary three-valued relation over a domain D is a function mapping each tuple in D^n to one of the truth values *true*, *false*, or *unknown*. Such a three-valued relation \tilde{R} is said to *approximate* a relation S if $\{\bar{d} \mid \tilde{R}(\bar{d}) = \text{true}\} \subseteq S \subseteq \{\bar{d} \mid \tilde{R}(\bar{d}) \in \{\text{true}, \text{unknown}\}\}$. A *partial* or *three-valued* finite structure over a vocabulary Σ is a structure with a finite domain D that assigns three-valued relations over D to each symbol in Σ . Such a partial structure \tilde{I} approximates a structure J if for every symbol $P \in \Sigma$, the three-valued relation assigned to P by \tilde{I} approximates the relation assigned to P by J .

2. Constraint propagation [4]

Constraint propagation for $\text{FO}(\cdot)$ is the task of deriving, from a given partial structure \tilde{I} and theory T , atoms that are true, respectively false, in all models of T approximated by \tilde{I} . Constraint propagation is a base form of inference and can be used, e.g., for building finite model generators and configuration systems, and for query answering in incomplete databases. Finite model generators in turn are applied to solve constraint satisfaction problems such as scheduling and planning.

Complete propagation, i.e., deriving all atoms that are certainly true, respectively false, is **NP-hard**. This is an unacceptably high complexity for most practical applications of propagation. Therefore, we develop a — necessarily incomplete — propagation algorithm with polynomial data-complexity, based on rewriting theories to a suitable normal form, called INF.

In the dissertation, we explain that the propagation on INF theories can be represented by a logic program. This has two benefits. First, existing al-

gorithms to evaluate logic programs can be applied to implement our propagation method. Secondly, techniques from logic programming analysis can be used to analyze propagation on a theory.

Finally, we show that our propagation method can also be executed when a theory, but no partial structure is given. This ‘symbolic’ algorithm computes two queries for each symbol P . Answers to these queries correspond to atoms over P that are certainly true, respectively false. The symbolic algorithm has applications in contexts where data is subject to frequent changes.

3. Grounding [5]

In the context of our work, grounding is defined as the task of reducing a first-order theory T and finite partial structure \tilde{I} to a propositional theory T_g such that the models of T_g are precisely the models of T approximated by \tilde{I} . Grounding is used as a preprocessing phase in many logic-based reasoning systems. It serves to provide the user with a rich input language, while enabling the system to rely on efficient propositional solvers, e.g., SAT solvers, to perform the actual reasoning.

A common practice to efficiently ground a theory T consists of carefully adding redundant information, called *bounds* to T . The bounds help a grounder to avoid computing unnecessary parts of the grounding T_g . We show that symbolic constraint propagation can be used to compute such bounds. We also present results about correctly adding the bounds to a theory. Experiments show that our method may drastically reduce grounding size and time.

4. Model revision [3]

Model revision is the task of changing a given finite model M of a theory T to a new model of M' of T that satisfies some additional requirements. Model revision is the form of inference applied to solve tasks like rescheduling and reconfiguration. Often, a requirement is that the new model M' is sufficiently close to the old one M .

We present an algorithm to solve model revision problems. The algorithm works by iteratively increasing the search space to find a new model. In each step, a grounding-based model generator is applied to verify whether a new model exists in the current search space.

5. Debugging [6]

One of the benefits of using a KBS to solve a computational problem is that typically, the logic theory describing the problem is more compact and readable than a program to solve the same problem in a standard programming language. Nevertheless, bugs are introduced when writing theories and debugging even small theories is often difficult in practice. This is especially the case when the bug makes the theory unsatisfiable.

We present a method for debugging unsatisfiable theories. Our method consists of interactively guiding a user through relevant parts of a proof of the unsatisfiability of the theory in the context of a finite partial structure. We show that such proofs can be generated by a finite model generator.

6. Conclusion

We investigated several useful forms of inference for the rich logic $\text{FO}(\cdot)$. Our results contribute to the goal of building a practical knowledge base system. Moreover, they show that first-order logic, usually associated with deduction, can be seen as a constraint propagation language. Many of the results are applied in the IDP system [2], a model generator for $\text{FO}(\cdot)$ that is competitive with the best ASP solvers. The dissertation is available at the URL <http://lirias.kuleuven.be/handle/123456789/265361>.

References

- [1] M. Denecker and E. Ternovska, A logic of non-monotone inductive definitions, *ACM Transactions on Computational Logic*, **9**(2)(2008), Article 14.
- [2] The IDP system. <http://dtai.cs.kuleuven.be/krr/software.html>.
- [3] J. Wittocx, B. De Cat, and M. Denecker, Towards computing revised models for FO theories, in *INAP*, S. Abreu and D. Seipel, eds, 2009, pp. 199–212.
- [4] J. Wittocx, M. Mariën, and M. Denecker, Approximate reasoning in first-order logic theories, in *KR*, G. Brewka and J. Lang, eds, AAAI Press, 2008, pp. 103–112.
- [5] J. Wittocx, M. Mariën, and M. Denecker, Grounding FO and FO(ID) with bounds, *Journal of Artificial Intelligence Research*, **38**(2010), 223–269.
- [6] J. Wittocx, H. Vlaeminck, and M. Denecker, Debugging for model expansion, in *ICLP*, volume 5649 of *LNCS*, P. M. Hill and D. S. Warren, eds, Springer, 2009, pp. 296–311.