Compact Propositional Encodings of First-Order Theories

Deepak Ramachandran and Eyal Amir
Computer Science Department
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{dramacha,eyal}@cs.uiuc.edu

1 Introduction

A *propositionalization* of a theory in First-Order Logic (FOL) is a set of propositional sentences that is satisfiable iff the original theory is satisfiable. We cannot translate arbitrary FOL theories to propositional logic because FOL is only semi-decidable. However, when possible, it is often advantageous to do so because we can use optimized, efficient SAT solvers (e.g. [Moskewicz *et al.*, 2001]) to solve the resulting SAT problem. Propositionalization is used frequently in Planning [Kautz and Selman, 1996], Relational Data Mining [Krogel *et al.*, 2003], and Formal Verification [Kropf, 1999].

Current propositional encodings (*naive prop.*), based on [Gilmore, 1960], result in prohibitively large propositional encodings even for moderate applications and assume a known finite domain for the theory. However despite their drawbacks, they are the most efficient solutions known so far. Examples of naive prop. are given in Table 1.

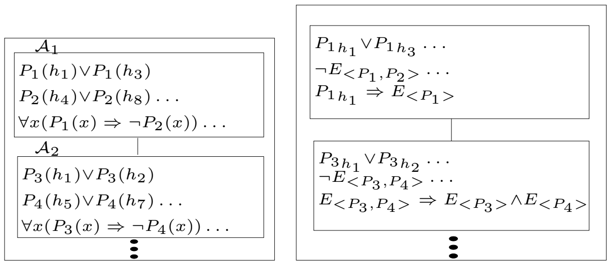
We briefly describe a novel, systematic approach to translating two important subsets of FOL into propositional logic. Our approach leverages structure in the FOL formulation to provide significantly more compact propositional encodings without requiring a finite fixed domain.

Specifically, we present algorithms for translating two important decidable subsets of FOL: (1) function-free *monadic*, and (2) the *Bernays-Schöenfinkel-Ramsey* class (see [Börger *et al.*, 1996]) in which all existential quantifiers must occur before all universal ones (all arity is allowed for predicates, with equality, but no functions). These subsets cover important problems in AI and computer science, such as expressive planning, data mining, constraint satisfaction, propositional modal logic, and quantified boolean formulae (QBF).

Our algorithms generate propositional encodings of these subsets of FOL as follows. They start by grouping axiom sets into a tree of partitions following the approach of [Amir and McIlraith, 2004]. Then, they translate each partition separately using only a restricted set of constants that depend on the structure of the partitioning. Finally, they combine the translated parts into a single propositional theory.

FOL Theory	Naive Prop.
$\neg(P(a)\wedge Q(b))$	$\neg(P_a\wedge Q_b)$
$\forall x P(x) \Rightarrow \forall y \neg Q(y)$	$(P_a\wedge P_b\wedge P_c \Rightarrow (\neg Q_a\wedge\neg Q_b\wedge\neg Q_c))$
$\exists z(R(z,c)\wedge\neg Q(a))$	$(R(a,c)\wedge\neg Q(a))\vee(R(c,c)\wedge\neg Q(a))$

Table 1: Naive propositionalization


$\mathcal{A}_1$
$P_1(h_1)\vee P_1(h_3)$
$P_2(h_4)\vee P_2(h_8)\ldots$
$\forall x(P_1(x)\Rightarrow\neg P_2(x))\ldots$
$\mathcal{A}_2$
$P_3(h_1)\vee P_3(h_2)$
$P_4(h_5)\vee P_4(h_7)\ldots$
$\forall x(P_3(x)\Rightarrow\neg P_4(x))\ldots$
$P_{1\,h_1}\vee P_{1\,h_3}\cdots$
$\neg E_{<P_1,P_2>}\cdots$
$P_{1\,h_1}\Rightarrow E_{<P_1>}$
$P_{3\,h_1}\vee P_{3\,h_2}\cdots$
$\neg E_{<P_3,P_4>}\cdots$
$E_{<P_3,P_4>}\Rightarrow E_{<P_3>}\wedge E_{<P_4>}$

(a) Partitioning $\{\mathcal{A}_i\}_{i\leq n}$ (b) Part-Prop-Monadic ($\{\mathcal{A}_i\}_{i\leq n}$)

Figure 1: The partitioning of the theory $\mathcal{A}$.

In the following, we briefly describe our methods with examples, and then present experimental results from applying these methods to an encoding of the Pigeonhole problem.

2 Partitioned Prop. for the Monadic Class

Figure 1 shows an example of the partitioned prop. of a theory in the monadic class (all predicates have arity 1). The theory $\mathcal{A}$ on the left is partitioned into $\mathcal{A}_1,\mathcal{A}_2\ldots\mathcal{A}_n$. There is an edge between every two partitions that share symbols. The partitioning is done in such a way that *message-passing* [Amir and McIlraith, 2004], a reasoning procedure that involves sending messages between partitions when a formula is in the intersection language, is sound and complete.

Figure 1 b) shows the propositional encoding of the partitioned theory. There are two kinds of propositonal symbols, those of the form P_a, which stand for $P(a)$, and $E_{<P,Q\ldots>}$ which stand for $\exists x[P(x)\wedge Q(x)\ldots]$. It can be shown that every monadic FOL formula can be converted into an equivalent propositional sentence using only these symbols. To ensure that the propositional symbols have the right semantics we need to add some consistency axioms to the propositionalization. For eg. $P_{1_{h_1}}\Rightarrow E_{<P_1>}$ ensures that if there is some constant h_1 for which $P_1(h_1)$ holds, then $\exists x P_1(x)$ holds as well and $E_{<P_3,P_4>}\Rightarrow E_{<P_3>}\wedge E_{<P_4>}$ ensures that if $\equiv\exists x[P_3(x)\wedge P_4(x)]$ is true, then $\exists x P_3(x)$ and $\exists x P_4(x)$ is true as well.

The union of these partitioned propositionalizations can be

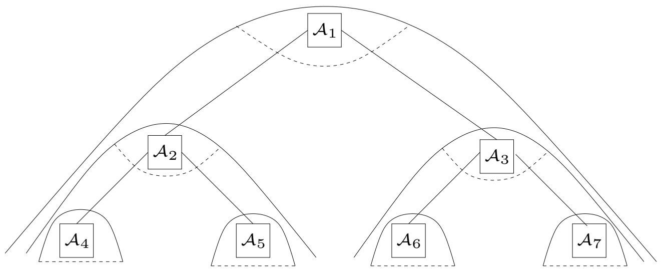


Figure 2: The solid boundaries represents the constants and the broken ones represent the predicates in each separate propositionalization.

shown to be equisatisfiable with $\mathcal{A}$. The proof relies on the completeness of the message passing procedure. Briefly, it can be shown that if the intersection graph of the partitioning is a tree, for every formula α s.t. $\mathcal{A} \vdash \alpha$, there is a proof s.t. each step uses either axioms that are completely within the same partition $\mathcal{A}_i$ or messages transmitted to $\mathcal{A}_i$ from other partitions. The proof of this relies on the following theorem:

Theorem 1 (Craig's Interpolation Theorem). *If α and β are first order formula s.t. $\alpha \vdash \beta$, then there is a formula $\gamma \in \mathcal{L}(L(\alpha) \cap L(\beta))$ such that $\alpha \vdash \gamma$ and $\gamma \vdash \beta$.*

3 Partitioned Prop. for the Ramsey Class

It is known that the naive partitioning is correct for the Ramsey class without any additional semantic assumptions [Börger *et al.*, 1996]. The number of propositional symbols used is $|P||C|^k$ where k is the maximum arity of any predicate. This can be significantly reduced by partitioning. But it is not enough to propositionalize each partition individually. For example, consider the case where $P(c)$ and $\forall x[P(x) \Rightarrow Q(a)]$ are in different partitions. Because we propositionalize these separately, $P(c) \Rightarrow Q(a)$ will not be in the propositional theory and thus $Q(a)$ cannot be deduced.

However, it can be shown that there exists an ordering $\prec$ of the vertices in the intersection graph of the partitioning such that a correct propositionalization results if we propositionalize the predicates of each partition $\mathcal{A}_i$ w.r.t. every constant that occurs in some partition $\mathcal{A}_j$, $A_j \prec A_i$. For example, in a prop. of the theory in figure 1 a), $P_{3_{h_3}}$ will be in the language, but $P_{1_{h_2}}$ will not. There are n such orderings, one for each choice of a partition as the root of the tree. In the case of a binary tree with n vertices (figure 2) each prop. will have $\frac{\log n}{n}|P||C|^k$ variables. Thus the problem of satisfiability of $\mathcal{A}$ reduces to n independent SAT instance of size $\frac{\log n}{n}|P||C|^k$. The running time will be $O(n \cdot 2^{\frac{\log n}{n}|P||C|^k})$ compared to the naive prop. which takes $O(2^{|P||C|^k})$.

4 Experimental Results

In figure 3 we present the results of some experiments with our algorithms on two different problems. The first is a first-order encoding of the pigeonhole problem that belongs to the Ramsey class. The upper two graphs compare our Ramsey prop. with the naive prop. for this problem, on the basis of the number of variables created and the running time for the SAT solver. Our algorithm does substantially better, enabling the solution of problems that are outside the scope of the naive encoding. The size of the encoding for the partitioned Ramsey prop. grows as $O(p \log p)$ versus $O(p^2)$ for the naive prop, where p is the number of pigeons. The running time for our SAT solver shows orders of magnitude speedup.

The bottom two graphs present a similar comparison for a *partitioned pigeonhole* problem (i.e. where each pigeon can be assigned only to a fixed subset of the holes) which is in both the Ramsey and Monadic classes. We compare both the Monadic and Ramsey partitioned prop. with naive prop. For the naive prop., the size of the encoding grows as $O(p^2)$ versus $O(p)$ for the partitioned monadic prop.

The experiments were done on a 3.2 GHz Linux machine with Zchaff [Moskewicz *et al.*, 2001] as the SAT solver.

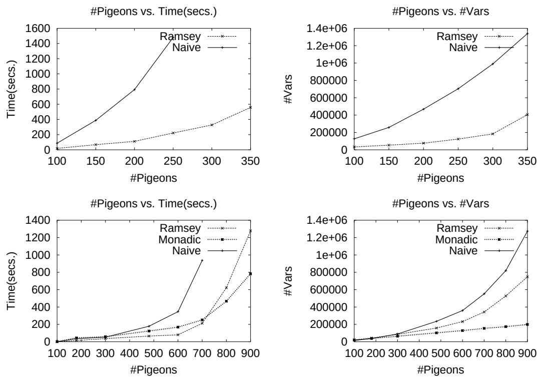


Figure 3: Experimental Results

References

[Amir and McIlraith, 2004] E. Amir and S. McIlraith. Partition-based logical reasoning for first-order and propositional theories. *Artificial Intelligence*, 2004. Accepted for publication.

[Amir, 2001] E. Amir. Efficient approximation for triangulation of minimum treewidth. In *UAI'01*, pages 7–15. Morgan Kaufmann, 2001.

[Börger *et al.*, 1996] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer-Verlag, 1996.

[Gilmore, 1960] P.C. Gilmore. A proof method for quantification theory: It's justification and realization. *IBM Journal of Research and Development*, 4(1):28–35, January 1960.

[Kautz and Selman, 1996] H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *AAAI'96*, 1996.

[Krogel *et al.*, 2003] M. Krogel, N. Lavrac, and S. Wrobel. Comparative evaluation of approaches to propositionalization. In *ILP*, pages 197–214, 2003.

[Kropf, 1999] T. Kropf, editor. *Introduction to Formal Hardware Verification*. Springer, 1999.

[Moskewicz *et al.*, 2001] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, 2001.