Artificial Intelligence 203 (2013) 19–34

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

# Complexity issues related to propagation completeness

Martin Babka, Tomáš Balyo, Ondřej Čepek, Štefan Gurský, Petr Kučera *, Václav Vlček

*Department of Theoretical Computer Science and Mathematical Logic, Faculty of Mathematics and Physics, Charles University in Prague, Malostranské nám. 25, 118 00 Praha 1, Czech Republic*

ARTICLE INFO

*Article history:*
Received 10 July 2012
Received in revised form 14 June 2013
Accepted 30 July 2013
Available online 7 August 2013

*Keywords:*
Boolean functions
Satisfiability
Knowledge compilation
Empowering implicates
Unit propagation
Propagation completeness

ABSTRACT

Knowledge compilation is a process of adding more information to a knowledge base in order to make it easier to deduce facts from the compiled base than from the original one. One type of knowledge compilation occurs when the knowledge in question is represented by a Boolean formula in conjunctive normal form (CNF). The goal of knowledge compilation in this case is to add clauses to the input CNF until a logically equivalent propagation complete CNF is obtained. A CNF is called propagation complete if after any partial substitution of truth values all logically entailed literals can be inferred from the resulting CNF formula by unit propagation. The key to this type of knowledge compilation is the ability to generate so-called empowering clauses. A clause is empowering for a CNF if it is an implicate and for some partial substitution of truth values it enlarges the set of entailed literals inferable by unit propagation.

In this paper we study several complexity issues related to empowering implicates, propagation completeness, and its relation to resolution proofs. We show several results: (a) given a CNF and a clause it is co-NP complete to decide whether the clause is an empowering implicate of the CNF, (b) given a CNF it is NP-complete to decide whether there exists an empowering implicate for it and thus it is co-NP complete to decide whether a CNF is propagation complete, and (c) there exist CNFs to which an exponential number of clauses must be added to make them propagation complete.

© 2013 Published by Elsevier B.V.

## 1. Introduction

One of the most studied problems in computer science, both theoretical and applied, is the satisfiability problem for CNF formulas (SAT). The difficulty of SAT depends on the class of CNF formulas to which the input formula belongs. There are various techniques and algorithms for SAT for different classes of CNF formulas ranging from linear algorithms for Horn, quadratic (2-CNF) and SLUR formulas [1,2] to the very complex variants of the exponential DPLL [3,4] and CDCL [5–8] procedures implemented in general purpose SAT solvers. Even the most complicated SAT solvers usually perform a task called unit propagation [3]. The goal of unit propagation is to infer as many logically entailed literals as possible from a partial truth assignment and the input formula. Although in general unit propagation is not a complete method (it does not infer all logically entailed literals), it is complete for the class of *propagation complete* (*PC*) CNF formulas [9].

PC formulas play an important role also in constraint programming, or more specifically, in CNF encodings of global constraints. There is a strong connection between propagation completeness of the CNF encoding and domain consistency

* Corresponding author. Tel.: +420 221 914 138; fax: +420 221 914 323.
*E-mail addresses:* babkys@gmail.com (M. Babka), biotomas@gmail.com (T. Balyo), ondrej.cepek@mff.cuni.cz (O. Čepek), stevko@mail.ru (Š. Gurský), kucerap@ktiml.mff.cuni.cz (P. Kučera), vlcek@ktiml.mff.cuni.cz (V. Vlček).

0004-3702/$ – see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.artint.2013.07.006

of the encoded constraint [10,11]. It has been studied for several concrete global constraints such as the AllDifferent constraint [12], the Sequence constraint [13], Regular, Among, and Generalized Sequence [10], or the Grammar constraints [14].

Some SAT solvers try to avoid searching in the state subspaces with no solution by learning from conflicts, i.e. by performing *conflict driven clause learning* (*CDCL*) [5–8], the name CDCL is also used for the complete algorithm solving SAT problem. It is useful to learn clauses (called *empowering implicates* [9,15]) that allow unit propagation to infer more logically entailed literals after such a clause is added to the CNF formula than it was possible to infer before the addition. Therefore, to speed up the CDCL SAT solver search for a satisfying assignment, it is often very useful to learn (generate) empowering implicates and add them to the input CNF formula. Let us mention that today's most successful SAT solvers for real-world applications are the ones using CDCL procedure.

This process of adding empowering implicates to a CNF formula can be viewed as a special type of knowledge compilation where both the input and the output representation of the knowledge is a CNF formula. In general, knowledge compilation is a process of adding more information to a given knowledge representation in order to make it computationally easier to infer facts from the compiled representation [16,17], or a process of transforming a given knowledge representation into another knowledge representation which is more tractable with respect to fact deduction, such as transforming a CNF into a BDD [18]. Nevertheless, in this paper we are interested only in the very limited case of knowledge compilation that rests in adding empowering implicates to a CNF.

It has been shown in [9], along with other properties of PC formulas, that a formula $\varphi$ is PC if and only if there is no empowering implicate for $\varphi$. However, several complexity issues directly connected to propagation completeness and empowering implicates are left open in [9]. A short list of such questions is the following:

1. Given a CNF formula $\varphi$ and a clause $C$, what is the complexity of deciding whether $C$ is an empowering implicate for $\varphi$?
2. Given a CNF formula $\varphi$ that is not PC, how difficult is it to generate an empowering implicate for $\varphi$ by resolution, where the "level of difficulty" is measured by the length of the resolution proof?
3. Given a CNF formula $\varphi$, what is the complexity of deciding whether there exists an empowering implicate for $\varphi$?
4. Given a CNF formula $\varphi$ that is not PC, how many empowering implicates is it necessary to add to $\varphi$ in order to make it PC?

In this paper we tackle all of the above listed problems. After reviewing basic definitions and notation in Section 2, we derive several simple properties of empowering implicates in Section 3. We address the following four questions as follows:

1. In Section 3 we show that the first problem is co-NP complete. This is not a very difficult result, however, to the best of our knowledge, it was not stated in the related literature yet.
2. In Section 4 we tackle the second problem. We prove that for a non-PC CNF formula with $s$ occurrences of literals there always exists a resolution proof of length $O(s)$ of some empowering implicate. On the other hand, we construct examples of CNF formulas where a resolution proof of length $\Omega(s)$ is needed for any empowering implicate, which means that $\Theta(s)$ is an asymptotically tight bound for this problem. It is important to note that the upper bound result does not require the derived empowering implicate to be prime. We show (by a simple modification of results concerning refutation proofs [19,20]) that there exist CNF formulas such that in order to derive any prime empowering implicate of such CNF a resolution proof of an exponential length is needed.
3. Section 5 contains the main results of this paper which are connected to the third problem. It was proved in [9] that deciding about an existence of an empowering implicate is in $\Sigma_2^p$. Using the results from Section 4 we strengthen this result by showing that the problem belongs to $\Sigma_1^p = \mathrm{NP}$. Given the equivalence between propagation completeness and non-existence of empowering implicates proved in [9], this immediately implies that testing propagation completeness belongs to co-NP. Then we proceed with the hardness proof for this problem. We present a reduction from a well-known NP-complete 3-dimensional matching problem which proves that deciding for a CNF formula whether there exists an empowering implicate for it is NP-hard (and thus testing propagation completeness is coNP-hard).
4. The fourth question is answered in Section 5 as well by showing that there exist CNF formulas where an exponential number (both with respect to the number of variables and the number of clauses) of empowering implicates must be added in order to arrive at a PC formula. This strengthens the superpolynomial bound which follows from a combination of results in [9] and [21] using a superpolynomial lower bound for certain monotone circuits from [22]. The connection is discussed in detail in Section 2.5.

We close the paper by giving few concluding remarks in Section 6.

Download English Version:

https://daneshyari.com/en/article/376970

Download Persian Version:

https://daneshyari.com/article/376970

Daneshyari.com