Artificial Intelligence 257 (2018) 1–23

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Constants and finite unary relations in qualitative constraint reasoning

Peter Jonsson

Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden

ARTICLE INFO

Article history:
Received 30 May 2017
Received in revised form 1 December 2017
Accepted 13 December 2017
Available online 20 December 2017

Keywords:
Constraint satisfaction
Qualitative reasoning
Computational complexity

ABSTRACT

Extending qualitative CSPs with the ability of restricting selected variables to finite sets of possible values has been proposed as an interesting research direction with important applications, cf. “Qualitative constraint satisfaction problems: an extended framework with landmarks” by Li, Liu, and Wang (2013) [48]. Previously presented complexity results for this kind of extended formalisms have typically focused on concrete examples and not on general principles. We propose three general methods. The first two methods are based on analysing the given CSP from a model-theoretical perspective, while the third method is based on directly analysing the growth of the representation of solutions. We exemplify the methods on temporal and spatial formalisms including Allen's algebra and RCC-5.

1. Introduction

This introductory section is divided into two parts where we first discuss the background of this article and thereafter describe our results.

1.1. Background

Qualitative reasoning has a long history in artificial intelligence and the combination of qualitative reasoning and constraint reasoning has been a very productive field. A large number of constraint-based formalisms for qualitative reasoning have been invented, most notably within temporal and spatial reasoning, and they have been investigated from many different angles. It has been noted that a particular extension to qualitative CSPs is highly relevant: Cohn and Renz [25, p. 578] observe the following

> One problem with this [constraint-based] approach is that spatial entities are treated as variables which have to be instantiated using values of an infinite domain. How to integrate this with settings where some spatial entities are known or can only be from a small domain is still unknown and is one of the main future challenges of constraint-based spatial reasoning.

and Li, Liu, and Wang [48, p. 33] write

E-mail address: peter.jonsson@liu.se.

https://doi.org/10.1016/j.artint.2017.12.003
0004-3702/

> There is a growing consensus, however, that breakthroughs are necessary to bring spatial/temporal reasoning theory closer to practical applications. One reason might be that the current qualitative reasoning scheme uses a rather restricted constraint language: constraints in a qualitative CSP are always taken from the *same* calculus and only relate variables from the same *infinite* domain. This is highly undesirable, as constraints involving restricted variables and/or multiple aspects of information frequently appear in practical tasks such as urban planning and spatial query processing.

That is, they regard the question of how to extend constraint formalisms with constants and other unary relations[1] as being very important; the same observation has been made in a wider context by Kreutzmann and Wolter [44]. An interesting recent example where such extensions of qualitative formalisms are necessary is the article on spatial query processing by Nikolaou and Koubarakis [56].

Given a (finite or infinite) set of values D, we let $D_c = \{\{d\} \mid d \in D\}$ (i.e. the set of constant relations over D) and $D_f = \{D' \subseteq D \mid D' \text{ is finite}\}$ (i.e. the set of finite unary relations over D). Let us consider finite-domain CSPs for a moment. For every finite constraint language Γ over D, the computational complexity of CSP($\Gamma \cup D_f$) is known due to results by Bulatov [17]. This is an important complexity result in finite-domain constraint satisfaction and it has been reproven several times using different methods [2,18]. Very recently, the complexity of CSP($\Gamma \cup D_c$) and CSP(Γ) has also been determined [19,63].

The situation is radically different when considering infinite-domain CSPs where similar powerful results are not known. This can, at least partly, be attributed to the fact that infinite-domain CSPs constitute a much richer class of problems than finite-domain CSPs: for every computational problem X, there is an infinite-domain constraint language Γ_X such that X and CSP(Γ_X) are polynomial-time Turing equivalent [9]. Finite domain CSPs are, on the other hand, always members of NP. Hence, the majority of computational problems cannot be captured by finite-domain CSPs.

Nevertheless, there exist concrete examples where interesting qualitative and/or infinite-domain CSPs have been extended with finite unary relations and/or constant relations. A very early example is the article by Jonsson and Bäckström [38] (see also Koubarakis [43]) where several temporal formalisms (including the point algebra and Allen's algebra) are extended by unary relations (and also other relations). A more recent example is the article by Li et al. [48] where the point algebra and Allen's algebra are once again considered. Li et al. also study several other formalisms including the cardinal relation algebra and RCC-5 and RCC-8 over two-dimensional regions and where constants are assumed to be polygonal regions. The results for the temporal formalisms by Jonsson and Bäckström are not completely comparable with the results by Li et al.: Jonsson and Bäckström's approach is based on linear programming while Li et al. use methods based on enforcing consistency and computational geometry. Consistency-enforcing methods have certain advantages such as lower time complexity and easier integration with existing constraint solving methods. At the same time, the linear programming method allows for more expressive extensions with retained tractability. Both consistency-based and LP-based methods have attracted attention lately, cf. Giannakopoulou et al. [30] and Kreutzmann and Wolter [44], respectively, and generalisations of the basic concepts have been proposed and analysed by de Leng and Heintz [26].

We see that this line of research has to a large extent been based on analysing concrete examples. The approach in this article will be different: instead of studying concrete examples, we study basic principles and aim at providing methods that are applicable to many different constraint formalisms.

1.2. Our results

We present three different methods. The first two methods are based on analysing the given CSP from a model-theoretical perspective. The third method is more of a toolbox for proving that the size of solutions (i.e., the number of bits needed for representing a solution) grows in a controlled way, and that problems consequently are in NP. We will now describe these methods in slightly more detail.

Method I. The first method is based on exploiting *ω-categoricity*. This is a model-theoretical property of constraint languages and other mathematical structures that have gained a lot of attention in the literature. Briefly speaking, a constraint language Γ is ω-categorical if Γ is the unique countable model (up to isomorphism) of the set of all first-order sentences that are true in Γ. One of the interesting aspects of ω-categorical constraint languages is that they in some respects resemble constraint languages over finite domains: this is expressed by a famous result proved by Engeler, Ryll-Nardzewski, and Svenonius (see Theorem 11). From a model-theoretical point of view, ω-categoricity is a very strong assumption. Nevertheless, many interesting CSP problems can be formulated using ω-categorical constraint languages: examples include the point algebra, RCC-5, RCC-8, and Allen's algebra. Among the ω-categorical constraint languages, the *model-complete cores* are particularly interesting. Such constraint languages allow us to define gadgets that can be used for simulating constants. This method is applicable to a wide selection of CSP(Γ) problems when Γ is ω-categorical. The drawback with the method is that it may be difficult to compute the gadgets used for simulating constants. Given that Γ is an ω-categorical model-complete core and that the gadgets can be computed efficiently, we verify (based on results by Bodirsky [5]) that CSP(Γ) is polynomial-time equivalent to CSP($\Gamma \cup D_c$). To demonstrate the strength of this method, we apply it to an extended version of Allen's algebra.

[1] Finite unary relations are sometimes referred to as *landmarks* in the AI literature. We will use the standard mathematical term throughout the article.