A Definability Dichotomy for Finite Valued CSPs*

Anuj Dawar and Pengming Wang

University of Cambridge Computer Laboratory
{anuj.dawar, pengming.wang}@cl.cam.ac.uk

May 28, 2015

Abstract

Finite valued constraint satisfaction problems are a formalism for describing many natural optimization problems, where constraints on the values that variables can take come with rational weights and the aim is to find an assignment of minimal cost. Thapper and Živný have recently established a complexity dichotomy for finite valued constraint languages. They show that each such language either gives rise to a polynomial-time solvable optimization problem, or to an NP-hard one, and establish a criterion to distinguish the two cases. We refine the dichotomy by showing that all optimization problems in the first class are definable in fixed-point language with counting, while all languages in the second class are not definable, even in infinitary logic with counting. Our definability dichotomy is not conditional on any complexity-theoretic assumption.

1 Introduction

Constraint Satisfaction Problems (CSPs) are a widely-used formalism for describing many problems in optimization, artificial intelligence and many other areas. The classification of CSPs according to their tractability has been a major area of theoretical research ever since Feder and Vardi [6] formulated their dichotomy conjecture. The main aim is to classify various constraint satisfaction problems as either tractable (i.e. decidable in polynomial time) or NP-hard and a number of dichotomies have been established for special cases of the CSP as well as generalizations of it. In particular, Cohen et al. [4] extend the algebraic methods that have been very successful in the classification of CSPs to what they call *soft constraints*, that is constraint problems involving optimization rather than decision problems. In this context, a recent result by Thapper and Živný [7] established a complexity dichotomy for *finite valued CSPs* (VCSPs). This is a formalism for defining optimization problems that can be expressed as sums of explicitly given rational-valued functions. As Thapper and Živný argue, the formalism is general enough to include a wide variety of natural optimization problems. They show that every finite valued CSP is either in P or

*This is an extended abstract to an article currently in submission for CSL 2015. A full version can be accssed under http://arxiv.org/abs/1504.03543

NP-hard and provide a criterion, in terms of the existence of a definable XOR function, that determines which of the two cases holds.

In this paper we are interested in the *definability* of constraint satisfaction problems in a suitable logic. Definability in logic has been a significant tool for the study of CSPs for many years. A particular logic that has received attention in this context is Datalog, the language of inductive definitions by function-free Horn clauses. A dichotomy of definability has been established in the literature, which shows that every constraint satisfaction problem on a fixed template is either definable in Datalog or it is not definable even in the much stronger C^{ω}—an infinitary logic with counting. This result has not been published as such but is an immediate consequence of results in [2] where it is shown that every CSP satisfying a certain algebraic condition is not definable in C^{ω}, and in [3] where it is shown that those that fail to satisfy this condition have bounded width and are therefore definable in Datalog. The definability dichotomy so established does not line up with the (conjectured) complexity dichotomy as it is known that there are tractable CSPs that are not definable in Datalog.

Our main result is a definability dichotomy for finite valued CSPs. In the context of optimization problems involving numerical values, Datalog is unsuitable so we adopt as our yardstick definability in fixed-point logic with counting (FPC). This is an important logic that defines a natural and powerful proper fragment of the polynomial-time decidable properties (see [5]). It should be noted that C^{ω} properly extends the expressive power of FPC and therefore undefinability results for the former yield undefinability results for the latter. We establish that every finite valued CSP is either definable in FPC or undefinable in C^{ω}. Moreover, this dichotomy lines up exactly with the complexity dichotomy of Thapper and Živný. All the valued CSPs they determine are tractable are in fact definable in FPC, and all the ones that are NP-hard are provably not in C^{ω}.

The positive direction of our result builds on the recent work of Anderson et al. [1] showing that solutions to explicitly given instances of linear programming are definable in FPC. Thapper and Živný show that for the tractable VCSPs the optimal solution can be found by solving their basic linear programming (BLP) relaxation. Thus, to establish the definability of these problems in FPC it suffices to show that the reduction to the BLP is itself definable in FPC.

For the negative direction, we use the reductions used in [7] to establish NP-hardness of VCSPs and show that these reductions can be carried out within FPC. We start the chain of reductions from the standard CSP form of 3-SAT, which is not definable in C^{ω} as a consequence of results from [2].

2 Background

Definition 1. *Let D be a finite domain. A* valued constraint language Γ *over D is a set of functions, where each $f \in \Gamma$ has an associated arity $m = \mathrm{ar}(f)$ and $f : D^m \to \mathbb{Q}^+$.*

Definition 2. *An instance of the* valued constraint satisfaction problem (VCSP) *over a valued constraint language Γ is a pair $I = (V, C)$, where V is a finite set of* variables *and C is a finite set of* constraints*. Each constraint in C is a triple (σ, f, q), where $f \in \Gamma$, $\sigma \in V^{\mathrm{ar}(f)}$ and $q \in \mathbb{Q}$.*

A solution *to an instance I of* VCSP(Γ) *is an assignment $h : V \to D$ of values in D to the variables in V. The* cost *of the solution h is given by $cost_I(h) := \sum_{(\sigma,f,q)\in C} q \cdot f(h(\sigma))$. The valued constraint satisfaction problem is then to find a solution with minimal cost.*

Given a valued constraint language Γ, there are certain natural closures Γ' of this set of functions for which the computational complexity of VCSP(Γ) and VCSP(Γ') coincide. We consider the so-called *expressive power* of Γ, which consists of functions that can be defined by minimising a cost function over a fixed VCSP(Γ)-instance I over some projection of the variables in I.

Definition 3. *Let Γ be a valued constraint language over D. We say a function $f : D^m \to \mathbb{Q}$, is* expressible *in Γ, if there is some instance $I_f = (V_f, C_f) \in$ VCSP(Γ) and a tuple $\bar{v} = (v_1, \dots, v_m) \in V_f^m$ such that*

$$f(\bar{x}) = \min_{h \in H_{\bar{x}}} cost_{I_f}(h),$$

where $H_{\bar{x}} := \{h : V_f \to D \mid h(v_i) = x_i \ , \ 1 \leq i \leq m\}$. We then say the function f is expressed *by the instance I_f and the tuple $\bar{v}$, and call the set of all functions that can be expressed by an instance of* VCSP(Γ) *the* expressive power *of Γ, denoted by $\langle\Gamma\rangle$.*

Another common notion in the study of constraint satisfaction problems, and of structure homomorphisms more generally, is the *core* of a structure. In the context of valued constraint satisfaction, this is defined as follows.

Definition 4. *We call a valued constraint language Γ over domain D a* core *if for for all $a \in D$, there is some instance $I_a \in$ VCSP(Γ) such that in every minimal cost solution over I_a, some variable is assigned a. A valued constraint language Γ' over a domain $D' \subseteq D$ is a* sub-language *of Γ if it contains exactly the functions of Γ restricted to D'. We say Γ' is a* core of Γ*, if Γ' is a sub-language of Γ and also a core.*

Furthermore, we also consider the closure of Γ under parameterized definitions. That is, we define Γ_c, the language obtained from Γ by allowing functions that are obtained from those in Γ by fixing some parameters.

Our main result relies on the dichotomy result from [7]. It states that any constraint language Γ falls into two cases: Either VCSP(Γ) is solvable in polynomial time, using its formulation as a linear program, the so-called *basic linear program*, or it is NP-hard. For the precise statement, we need to introduce an additional notion. We say the property (XOR) holds for a valued constraint language Γ over domain D if there are $a, b \in D, a \neq b$, such that $\langle\Gamma\rangle$ contains a binary function f with argmin $f = \{(a, b), (b, a)\}$.

Theorem 5 (Theorem 3.3, [7]). *Let Γ be a core over some finite domain D.*

- *Either for each instance I of* VCSP(Γ)*, the optimal solutions of I are the same as the basic linear program of I;*
- *or property* (XOR) *holds for Γ_c and* VCSP(Γ) *is* NP*-hard.*

Our definability dichotomy is in terms of two logics that have proven to be useful in the study of classical (non-valued) CSPs already. The first one, fixed-point logic with counting (FPC), is an extension of inflationary fixed-point logic with the ability to express the cardinality of definable sets. We also consider C^{ω}—the infinitary logic with counting, and finitely many variables. We use the following notion of definability. For now, assume that instances of VCSP(Γ) and rational numbers are represented as structures over the vocabularies σ_{Γ} and $\sigma_{\mathbb{Q}}$ respectively (which we later define formally).

Definition 6. *Let Γ be a finite valued constraint language. We say* VCSP(Γ) *is* definable *in a logic L, if there is a L-interpretation from σ_{Γ} to $\sigma_{\mathbb{Q}}$ that takes a structure encoding an instance I of* VCSP(Γ) *to a structure encoding its optimal value $q \in \mathbb{Q}$.*

3 Main Result

Our main result puts the earlier dichotomy of finite valued CSPs into terms of definability. We show that the tractable cases of VCSP(Γ) are exactly those that can be defined within FPC, while the intractable ones can not be defined even in the more powerful logic C^{ω}.

Theorem 7. *Let Γ be a finite valued constraint language over a domain D.*

- *Then, either* VCSP(Γ) *is definable in* FPC*;*
- *or there is a core Γ' of Γ, such that* (XOR) *holds for Γ'_c, and* VCSP(Γ) *is not definable in C^{ω}.*

References

[1] M. Anderson, A. Dawar, and B. Holm. Maximum matching and linear programming in fixed-point logic with counting. In *Proceedings of the 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 173–182, 2013.

[2] A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666 – 1683, 2009.

[3] L. Barto and M. Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61, 2014.

[4] D. Cohen, M.C. Cooper, P. Jeavons, and A. Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983 – 1016, 2006.

[5] A. Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2:8–21, 2015.

[6] T. Feder and M.Y. Vardi. Computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1998.

[7] J. Thapper and S. Živný. The complexity of finite-valued CSPs. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pages 695–704. ACM, 2013.