APPLICATIONS OF UNIVERSAL ALGEBRA AND LOGIC TO THE CONSTRAINT SATISFACTION PROBLE

organized by
Anuj Dawar, Phokion Kolaitis, Benoit Larose, and Matt Valeriote

Workshop Summary

The purpose of the workshop was to bring together researchers from the computational complexity, logic, and universal algebra communities in order to advance the understanding of the constraint satisfaction problem (CSP). The primary motivating problem in the area is the Dichotomy Conjecture formulated by Feder and Vardi in 1993, which asserts that, for every constraint language Γ over an arbitrary finite domain, either the class of constraint satisfaction problems with constraint relations from Γ (denoted CSP(Γ)) is in **P** or it is **NP**-complete. Several other conjectures and open problems (some of which may be more manageable) related to the Dichotomy Conjecture have emerged in recent years and prior to the workshop an extensive list of problems was compiled and circulated to the workshop participants. In addition, a number of survey papers were posted on the workshop website and made available to the participants.

Each morning of the workshop, two hour-long talks were given, with those on the first three days providing introductions to the main threads of the workshop. The talks on the last two days were arranged during the week, with their subjects being determined by developments during the workshop. The following table contains the schedule of workshop talks:

Day	Title	Speaker
Monday (March 31)	Benoit Larose	A universal algebra primer for CSP
Monday	Phokion Kolaitis	Logic and constraint satisfaction, Part I
Tuesday	Matt Valeriote	Universal algebra and the CSP
Tuesday	Anuj Dawar	Logic and constraint satisfaction, Part II
Wednesday	Eric Allender	Complexity classes below P
Wednesday	Martin Grohe	Treewidth
Thursday	Victor Dalmau	Dualities and the CSP
Thursday	Andrei Bulatov	Coloured graphs and the CSP
Friday	Ross Willard	Few subpowers and the CSP
Friday	Ralph McKenzie	Dichotomy for smooth graphs

On Monday afternoon a problem session was held, moderated by Moshe Vardi, with the goal of identifying a number of problems that participants would like to work on during the afternoon group sessions. On subsequent afternoons, small working groups were established to discuss and work on some of these problems. At the end of each day a wrap-up session was held during which each group provided a progress report.

The following is a list of some of the themes and problems that arose from this session and that were explored during the week:

(1) Width hierarchy

Most known tractability results for CSP's employ some combination of two distinct algorithmic paradigms; the first is a distant relative of Gaussian elimination, while the second is keyed on establishing definability in Datalog (which, in turn, yields a polynomial-time constraint propagation algorithm).

A natural parameter associated with each Datalog program (and closely related to the program's running time) is the maximum number of distinct variables appearing in the rules of that program. Let k-Datalog be the collection of all Datalog programs with k distinct variables per rule. It is well known (and not hard to show) that the expressive power of Datalog programs increases with the number of distinct variables.

If a class of CSP's is definable in Datalog then one may define the "width" of this class to be the smallest k for which the class is definable in k-Datalog. At this point, all CSP's currently known to be definable in Datalog are definable in 3-Datalog and so a central question is whether or not there is a proper hierarchy for CSP widths.

For those participants not familiar with Datalog, Albert Atserias ran a tutorial on this subject on Tuesday afternoon. It turns out that there are several competing notions of width for CSP's and some effort was made in one of the working groups to unify these various notions. It is known that there are certain algebraic invariants of a constraint language that may be used to detect its width and these invariants were studied during the workshop.

(2) Algorithmic classification

There are a number of refinements of the Dichotomy Conjecture that correlate tractability (or related conditions) with properties of a constraint language and one issue that arises from this is the complexity of determining if a given constraint language satisfies the property in question. For example, it is conjectured that the tractability of CSP(Γ) is related to the relations in the constraint language Γ having a so-called weak near unanimity polymorphism and so it is desirable to know the complexity of determining if Γ has such a polymorphism. The precise complexity of this particular problem is currently open at the moment, but the complexity of related questions have been worked out.

In this vein, a number of sessions were held to consider the existential inverse satisfiability problem (or $\exists -$ INV SAT). An instance of this problem consists of a finite set $\mathcal{S}$ of relations over some finite set A along with a relation R and the problem is to determine if R can be defined from the relations in $\mathcal{S}$ via a primite positive formula. It is known that this class of problems lies in co-NEXPTIME and, during the workshop, was conjectured to be complete for this class. A special case of this problem is obtained by restricting the relation R in an instance to be of the form $\{a\}$ for some $a \in A$. Using that graph 3-colorability is NP-complete, Ralph McKenzie was able to show that this restricted version of the problem is co-NP-complete. Ross Willard employed the clone membership problem (shown to be EXPTIME-complete by Harvey Friedman) to prove that $\exists -$ INV SAT is EXPTIME-hard.

(3) Special classes of instances

While the notion of a constraint language provides a useful parametrization of the CSP into subclasses, another natural way to define subclasses is to restrict the instances of CSP(Γ) to fall within a specified class of structures. For example, it is known that for any finite constraint language Γ, and any k, if one restricts instances of CSP(Γ) to those of tree-width k then the problem class is tractable.

During the workshop a systematic analysis of the computational complexity of the subclass of CSP(G), where G is a finite graph, whose instances are finite planar graphs was initiated. A number of tractable and NP-complete cases were uncovered; for example tractability occurs when G is the complete graph on $n \neq 3$ vertices, while NP-completeness arises when G is a cycle of odd length. A number of substantial technical difficulties arose in this investigation and so much more work needs to be done on this problem.

(4) Maltsev Conditions

A key observation made early on in the study of the CSP is that the computational complexity of CSP(Γ) is determined by the set of polymorphisms of the constraint language Γ. There is good evidence to suggest that in fact the tractability of CSP(Γ) can be expressed as a special type of term condition, called a Maltsev condition, on the algebra of polymorphisms of Γ. Indeed, several of the most general tractability results to date can be expressed in this manner.

The experience garnered by universal algebraists suggests that a natural setting in which to extend some of the known tractability results is to that of constraint languages that support Gumm terms as polymorphisms, or equivalently whose algebras of polymorphisms generate congruence modular varieties. This class of constraint languages will prove to be an important testing ground for the main conjectures and will require the development of hybrid polynomial-time algorithms. A discussion of how to approach this was initiated during the workshop.

While not directly related to the CSP, the resolution of a conjecture of Zadori's on the nature of certain Maltsev conditions over finite relational structures could help to clarify matters. A number of the participants spent time considering special cases of Zadori's conjecture and proposed some related problems.

Perhaps one of the most striking claims made during the workshop occurred during the presentation of Andrei Bulatov on his technology for analyzing the local structure of finite algebras. He announced that under some rather general conditions, expressed in the language of coloured finite graphs, a constraint language will be tractable. While his condition is not a Maltsev condition, it does provide some support for the Definability-in-Datalog Conjecture and more generally, the Tractability Conjecture.

(5) Fragments

The subclasses of the CSP that can be defined via a finite constraint language can be expressed within the logic MMSNP (Monadic Monotone Strict NP). Even though the expressive power of MMSNP is quite large, it turns out that, in some sense, CSP's and MMSNP are computationally equivalent. Given the difficulties encountered in trying to settle the Dichotomy Conjecture for CSP's, it seems sensible to test the conjecture over fragments of MMSNP. Restricting MMSNP formulas by allowing a single set-variable one obtains the fragment $\mathrm{MM_1SNP}$. A study of this fragment was initiated during the workshop and some partial results, dealing with instances arising from templates with no loops, were obtained. In these cases, tractability was shown to occur when all obstructions have size at most 2 and that NP-hardness arises in all other cases. The following special case was completely classified: given a set $S \subseteq \{1, \ldots, k\}$, determine for which graphs $G = (V, E)$ it is the case that G can be 2-colored such that on every (directed) k-path the number of 1's belongs to S.

It is known that, in general, the complement of CSP(B) is not definable in Datalog and it is also known that definability within certain fragments of MMSNP entails definability in Datalog. For example, if the complement of CSP(B) is definable within the logic $\exists L^{\omega;+}_{\infty\omega}$ then it is also definable in Datalog. In order to understand the limits of this sort of result, a working group was formed to consider least fixed point (LFP) logic in connection with the CSP. In order to show that for CSP's, definability in LFP implies definability in Datalog it was noticed that it is sufficient to show that for structures $\mathbf{A}$ and $\mathbf{B}$, whenever $\mathbf{A} \leq_k \mathbf{B}$ (meaning that Duplicator has a winning strategy in the one-sided k-pebble game) then there are structures $\mathbf{A}', \mathbf{B}'$ such that $\mathbf{A} \to \mathbf{A}'$, $\mathbf{B}' \to \mathbf{B}$, and $\mathbf{A}' \equiv_k \mathbf{B}'$ (meaning that Duplicator has a winning strategy in the two-sided k-pebble game). An approach to verifying this condition was identified and some progress made.

A refinement of Schaefer's Theorem (that establishes a dichotomy for CSP's over 2-element [or Boolean] domains) provides details of the subclasses of the CSP that belong to complexity classes below P. By examining Hubie Chen's proof of the dichotomy for quantified CSP's over a Boolean domain, a working group was able to show that the same picture emerges for quantified CSP's below P, i.e., tractable quantified Boolean CSP's are no harder than their unquantified counterparts.