Open Problems List

Arising from MathsCSP Workshop, Oxford, March 2006

Version 0.3, April 25, 2006

1 Complexity and Tractability of CSP

Question 1.0 (The Dichotomy Conjecture) Let $\mathcal{B}$ be a relational structure. The problem of deciding whether a given relational structure has a homomorphism to $\mathcal{B}$ is denoted CSP($\mathcal{B}$).

For which (finite) structures is CSP($\mathcal{B}$) decidable in polynomial time? Is it true that for any finite structure $\mathcal{B}$ the problem CSP($\mathcal{B}$) is either decidable in polynomial time or NP-complete?

Communicated by: Tomas Feder & Moshe Vardi (1993)

Question 1.1 A relational structure $\mathcal{B}$ is called *hereditarily tractable* if CSP($\mathcal{B}'$) is tractable for all substructures $\mathcal{B}'$ of $\mathcal{B}$. Which structures $\mathcal{B}$ are hereditarily tractable?

Communicated by: Pavol Hell

Question 1.2 A *weak near-unanimity term* is defined to be one that satisfies the following identities: $f(x, \ldots, x) = x$ and $f(x, y, ...y) = f(y, x, y, ...y) = ... = f(y, ..., y, x)$.

Is CSP($\mathcal{B}$) tractable for any (finite) structure $\mathcal{B}$ which is preserved by a weak near-unanimity term?

Communicated by: Benoit Larose, Matt Valeriote

Question 1.3 A constraint language[1] S is called *globally tractable* for a problem $\mathcal{P}$, if $\mathcal{P}(S)$ is tractable, and it is called *(locally) tractable* if for every finite $L \subseteq S$, $\mathcal{P}(L)$ is tractable.

These two notions of tractability do not coincide in the Abduction problem (see talk by Nadia Creignou).

- For which computational problems related to the CSP do these two notions of tractability coincide?
- In particular, do they coincide for the standard CSP decision problem?

Communicated by: Nadia Creignou

[1] That is, a (possibly infinite) set of relations over some fixed set.

Question 1.4 (see also Question 3.5) It has been shown that when a structure $\mathcal{B}$ has *bounded pathwidth duality* the corresponding problem CSP($\mathcal{B}$) is in the complexity class **NL** (see talk by Victor Dalmau). Is the converse also true (modulo some natural complexity-theoretic assumptions)?

Communicated by: Victor Dalmau

Question 1.5 Is there a good (numerical) parameterization for constraint satisfaction problems that makes them fixed-parameter tractable?

Question 1.6 Further develop techniques based on delta-matroids to complete the complexity classification of the Boolean CSP (with constants) with at most two occurrences per variable (see talk by Tomas Feder).

Communicated by: Tomas Feder

Question 1.7 Classify the complexity of uniform Boolean CSPs (where both structure and constraint relations are specified in the input).

Communicated by: Heribert Vollmer

Question 1.8 The *microstructure graph* of a binary CSP has vertices for each variable/value pair, and edges that join all pairs of vertices that are compatible with the constraints.

What properties of this graph are sufficient to ensure tractability? Are there properties that do not rely on the constraint language or the constraint graph individually?

2 Approximability and Soft Constraints

Question 2.1 Is it true that Max CSP(L) is **APX**-complete whenever Max CSP(L) is **NP**-hard?

Communicated by: Peter Jonsson

Question 2.2 Prove or disprove that Max CSP(L) is in **PO** if the core of L is supermodular on some lattice, and otherwise this problem is **APX**-complete.

The above has been proved for languages with domain size 3, and for languages containing all constants by a computer-assisted case analysis (see talk by Peter Jonsson). Develop techniques that allow one to prove such results without computer-assisted analysis.

Communicated by: Peter Jonsson

Question 2.3 For some constraint languages L, the problem Max CSP(L) is hard to approximate better than the random mindless algorithm on satisfiable or almost satisfiable instances. Such problems are called *approximation resistant* (see talk by Johan Hastad).

Is a single random predicate over Boolean variables with large arity approximation resistant?

What properties of predicates make a CSP approximation resistant?

What transformations of predicates preserve approximation resistance?

Communicated by: Johan Hastad

Question 2.4 Many optimisation problems involving constraints (such as MAX-SAT, MAX CSP, MIN-ONES SAT) can be represented using *soft constraints* where each constraint is specified by a *cost function* assigning some measure of cost to each tuple of values in its scope.

Are all tractable classes of soft constraints characterized by their multimorphisms? (see talk by Peter Jeavons)

Communicated by: Peter Jeavons

3 Algebra

Question 3.1 The Galois connection between sets of relations and sets of operations that preserve them has been used to analyse several different computational problems such as the satisfiability of the CSP, and counting the number of solutions.

How can we characterise the computational goals for which we can use this Galois connection?

Communicated by: Nadia Creignou

Question 3.2 For any relational structure $\mathcal{B} = (B, R_1, \ldots, R_k)$, let co-CSP($\mathcal{B}$) denote the class of structures which do *not* have a homomorphism to $\mathcal{B}$. It has been shown that the question of whether co-CSP($\mathcal{B}$) is definable in Datalog is determined by $Pol(\mathcal{B})$, the polymorphisms of the relations of $\mathcal{B}$ (see talk by Andrei Bulatov).

Let $\mathcal{B}$ be a core, F the set of all idempotent polymorphisms of $\mathcal{B}$ and $\mathbf{V}$ the variety generated by the algebra (B, F). Is it true that co-CSP($\mathcal{B}$) is definable in Datalog if and only if $\mathbf{V}$ omits types **1** and **2** (that is, the local structure of any finite algebra in $\mathbf{V}$ does not contain a G-set or an affine algebra)?

Communicated by: Andrei Bulatov

Question 3.3 Does every tractable clone of polynomials over a group contain a Mal'tsev operation?

Communicated by: Pascal Tesson

Question 3.4 Classify (w.r.t. tractability of corresponding CSPs) clones of polynomials of semigroups.

Communicated by: Pascal Tesson

Question 3.5 Is it true that for any structure $\mathcal{B}$ which is invariant under a near-unanimity operation the problem CSP($\mathcal{B}$) is in the complexity class **NL**? Does every such structure have bounded pathwidth duality? (see also Question 1.4)

Both results are known to hold for a 2-element domain (Dalmau) and for majority operations (Dalmau,Krokhin).

Communicated by: Victor Dalmau, Benoit Larose

Question 3.6 Is it decidable whether a given structure is invariant under a near-unanimity function (of some arity)?

Communicated by: Benoit Larose

Question 3.7 Let L be a fixed finite lattice. Given an integer-valued supermodular function f on L^n, is there an algorithm that maximizes f in polynomial time in n if the function f is given by an oracle?

The answer is yes if L is a distributive lattice (see "Supermodular Functions and the Complexity of Max-CSP", Cohen, Cooper, Jeavons, Krokhin, Discrete Applied Mathematics, 2005). More generally, the answer is yes if L is obtained from finite distributive lattices via Mal'tsev products (Krokhin, Larose – see talk by Peter Jonsson). The smallest lattice for which the answer is not known is the 3-diamond.

Communicated by: Andrei Krokhin

Question 3.8 Find the exact relationship between width and relational width. (It is known that one is bounded if and and only if the other is bounded.)

Also, what types of width are preserved under natural algebraic constructions?

Communicated by: Victor Dalmau

4 Logic

Question 4.1 The (basic) Propositional Circumscription problem is defined as follows:

> **Input:** a propositional formula ϕ with atomic relations from a set S, and a clause c.
>
> **Question:** is c satisfied in every minimal model of ϕ?

It is conjectured (Kirousis, Kolaitis) that there is a trichotomy for this problem, that it is either in P, coNP-complete or in Π_2^P, depending on the choice of S. Does this conjecture hold?

Communicated by: Nadia Creignou

Question 4.2 The Inverse Satisfiability problem is defined as follows:

> **Input:** a finite set of relations S and a relation R.
>
> **Question:** is R expressible by a CNF(S)-formula without existential variables?

A dichotomy theorem was obtained by Kavvadias and Sideri for the complexity of this problem with constants. Does a dichotomy hold without the constants? Are the Schaefer cases still tractable?

Communicated by: Nadia Creignou

Question 4.3 Let LFP denote classes of structures definable in first-order logic with a least-fixed-point operator, let HOM denote classes of structures which are closed under homomorphisms, and let co-CSP denote classes of structures defined by *not* having a homomorphism to some fixed target structure.

- Is LFP $\cap$ HOM $\subseteq$ Datalog?
- Is LFP $\cap$ co-CSP $\subseteq$ Datalog? (for finite target structures)
- Is LFP $\cap$ co-CSP $\subseteq$ Datalog? (for ω-categorical target structures)

Communicated by: Albert Atserias, Manuel Bodirsky

Question 4.4 (see also Question 3.2) Definability of co-CSP($\mathcal{B}$) in k-Datalog is a sufficient condition for tractability of CSP($\mathcal{B}$), which is sometimes referred to as having *width* k. There is a game-theoretic characterisation of definability in k-Datalog in terms of $(\exists, k)$-pebble games (see talk by Phokion Kolaitis).

- Is there an algorithm to decide for a given structure $\mathcal{B}$ whether co-CSP($\mathcal{B}$) is definable in k-Datalog (for a fixed k)?
- Is the width hierarchy strict? The same question when $\mathcal{B}$ is ω-categorical, but not necessarily finite?

Communicated by: Phokion Kolaitis, Manuel Bodirsky

Question 4.5 Find a good logic to capture CSP with "nice" (e.g., ω-categorical) infinite templates.

Communicated by: Iain Stewart

5 Graph Theory

Question 5.1 The LIST HOMOMORPHISM problem for a (directed) graph H is equivalent to the problem CSP(H^*) where H^* equals H together with all unary relations.

- It is conjectured that the list homomorphism problem for a reflexive digraph is tractable if H has the X-underbar property (which is the same as having the binary polymorphism *min* w.r.t. some total ordering on the set of vertices), and NP-complete otherwise.
- It is conjectured that the list homomorphism problem for an irreflexive digraph is tractable if H is preserved by a majority operation, and NP-complete otherwise.

Do these conjectures hold?

Communicated by: Tomas Feder & Pavol Hell

Question 5.2 "An island of tractability?"

Let $\mathfrak{A}_m$ be the class of all relational structures of the form $(A, E_1, \ldots, E_m)$ where each E_i is an irreflexive symmetric binary relation and the relations E_i together satisfy the following 'fullness' condition: *any two distinct elements x, y are related in exactly one of the relations E_i.*

Let $\mathcal{B}_m$ be the single relational structure $(\{1, \ldots, m\}, E_1, \ldots, E_m)$ where each E_i is the symmetric binary relation containing all pairs xy except the pair ii. (Note that the relations E_i are not irreflexive.)

The problem CSP$(\mathfrak{A}_m, \mathcal{B}_m)$ is defined as: Given $\mathcal{A} \in \mathfrak{A}_m$, is there a homomorphism from $\mathcal{A}$ to $\mathcal{B}_m$?

When $m = 2$, this problem is solvable in polynomial time - it is the recognition problem for split graphs (see "Algorithmic Graph Theory and Perfect Graphs", M. C. Golumbic, Academic Press, New York, 1980) When $m > 3$, this problem is NP-complete (see "Full constraint satisfaction problems", T. Feder and P. Hell, to appear in SIAM Journal on Computing).

What happens when $m = 3$? Is this an "island of tractability"? Quasi-polynomial algorithms are known for this problem (see "Full constraint satisfaction problems", T. Feder and P. Hell, ,to appear in SIAM Journal on Computing, and "Two algorithms for list matrix partitions", T. Feder, P. Hell, D. Kral, and J. Sgall, SODA 2005). Note that a similar problem for $m = 3$ was investigated in "The list partition problem for graphs", K. Cameron, E. E. Eschen, C. T. Hoang and R. Sritharan, SODA 2004.

Communicated by: Tomas Feder & Pavol Hell

Question 5.3 Finding the *generalized hypertree-width*, $w(H)$ of a hypergraph H is known to be NP-complete. However it is possible to compute a hypertree-decomposition of H in polynomial time, and the *hypertree-width* of H is at most $3w(H) + 1$ (see talk by Georg Gottlob).

Are there other decompositions giving better approximations of the generalized hypertree-width that can be found in polynomial time?

Communicated by: Georg Gottlob

Question 5.4 It is known that a CSP whose constraint hypergraph has bounded fractional hypertree width is tractable (see talk by Daniel Marx).

Is there a hypergraph property more general than bounded fractional hypertree width that makes the associated CSP polynomial-time solvable?

Are there classes of CSP that are tractable due to structural restrictions and have unbounded fractional hypertree width?

Communicated by: Georg Gottlob, Daniel Marx

Question 5.5 Prove that there exist two functions $f_1(w), f_2(w)$ such that, for every w, there is an algorithm that constructs in time $n^{f_1(w)}$ a fractional hypertree decomposition of width at most $f_2(w)$ for any hypergraph of fractional hypertree width at most w (See talk by Daniel Marx).

Communicated by: Daniel Marx

Question 5.6 Turn the connection between the Robber and Army game and fractional hypertree width into an algorithm for approximating fractional hypertree width.

Communicated by: Daniel Marx

Question 5.7 Close the complexity gap between (H, C, K)-colouring and $\sharp(H, C, K)$-colouring (see talk by Dimitrios Thilikos)

Find a tight characterization for the fixed-parameter tractable (H, C, K)-colouring problems.

- For the (H, C, K)-colouring problems, find nice properties for the non-parameterised part $(H - C)$ that guarantee fixed-parameter tractability.
- Clarify the role of loops in the parameterised part C for fixed-parameter hardness results.

Communicated by: Dimitrios Thilikos

6 Constraint Programming and Modelling

Question 6.1 In a constraint programming system there is usually a *search* procedure that assigns values to particular variables in some order, interspersed with a *constraint propagation* process which modifies the constraints in the light of these assignments.

Is it possible to choose an ordering for the variables and values assigned which changes each problem instance as soon as possible into a new instance which is in a tractable class? Can this be done efficiently? Are there useful heuristics?

Question 6.2 The time taken by a constraint programming system to find a solution to a given instance can be dramatically altered by modelling the problem differently.

Can the efficiency of different constraint models be objectively compared, or does it depend entirely on the solution algorithm?

Question 6.3 For practical constraint solving it is important to eliminate *symmetry*, in order to avoid wasted search effort.

Under what conditions is it tractable to detect the symmetry in a given problem instance?

7 Notes

- Representations of constraints - implicit representation - effect on complexity
- Unique games conjecture - structural restrictions that make it false - connections between definability and approximation
- MMSNP - characterise tractable problems apart from CSP

- Migrate theoretical results to tools
- What restrictions do practical problems actually satisfy?
- Practical parallel algorithms - does this align with tractable classes?
- Practically relevant constraint languages ("global constraints")
- For what kinds of problems do constraint algorithms/heuristics give good results?