Maltsev CSPs are definable in 2-sorted Datalog

Dejan Delic
Department of Mathematics
Ryerson University
Toronto, Canada

June 14, 2017

Datalog, Linear Arc Consistency, and SLAC

- A *Datalog Program* is a finite set of rules of the form

$$T_0 \leftarrow T_1, T_2, \ldots, T_n$$

  where T_i's are atomic formulas. T_0 is the **head** of the rule, and the part to the right of the arrow is the **body**.
- Two kinds of relational predicates:
  1. intentional - IDBs: those occurring at least once in the head of some rule and which are not part of the vocabulary of the template (they are derived by the computation)
  2. extensional - EDBs: they are relations from the template and do not change during computation; i.e. they cannot appear in the head of any rule.
  3. Special IDB: nullary (Boolean) predicate called the *goal*

- The rule

$$T_0 \leftarrow T_1, T_2, \ldots, T_n$$

  is linear if at most one atomic formula in its body is an IDB.
- A Datalog program is linear if so are all its rules.
- The semantics of Datalog programs are generally defined in terms of fixed-point operators.

- If a linear Datalog program terminates with success on a finite relational instance, we say that the instance of the CSP is linear arc consistent, or LAC, for short.
- We can define Singleton Linear Arc Consistency (SLAC) by restricting the instance to a singleton in any coordinate and running the program on it. We eliminate the vertices which fail the LAC test, until the instance “stabilizes" and the resulting instance is SLAC, if nonempty.
- SLAC enforces the following property: if $(y, x_{j_1}, \ldots, x_{j_m}, y)$ is a path pattern based at y (a cycle), then there is a realization of the pattern with $y = a$, for all $a \in \mathbb{A}_y$. (we can assume that all x_j's are distinct.)

Local Consistency

Theorem

(Barto, Kozik, 2009) CSPs solvable by local consistency checks (equivalently, in Datalog) are precisely the CSPs with templates of bounded relational (tree)width.

- Over time, the same two authors, weakened the necessary and sufficient conditions for local consistency: from (2,3)-minimality and Prague strategies, to weak Prague strategies, culminating with SLAC (Kozik, July 2016, ArXiv)

Question: Are local consistency methods sufficient to solve *all* CSPs?
Answer: NO! Feder and Vardi (1998), using the result of Razborov about unsolvability of the Bipartite Matching by certain classes of Boolean circuits showed that the CSPs with group templates (and, therefore, Maltsev templates) cannot be solved by local consistency checks.

Reduction to a Binary Case

- We would like to reduce the problem to the one where all the constraints are binary.
- Is there any advantage to doing this first?
- Yes - SLAC solves *cyclic systems of linear equations* over a finite Abelian group. Such a system consists of equations of the form $x_i + a = x_j$.
- Such a system defines an equivalence relation on the set of its variables whose blocks can be computed in logspace and has a number of nice properties, among others the “rectangulation" between distinct equivalence blocks.

- Let d be the arity of the longest relation in the constraint language.
- With a little bit of work, we can convert the instance into one in which all relations are d-ary.
- Then, we can consider a 2-sorted instance (essentially, a bipartite graph with coloured edges), with the set of variables having two sorts: one for the original variables $x_1, \ldots, x_n$ and the other one for all distinct d-tuples of the original variables $(x_{i_1}, \ldots, x_{i_d})$, with a number of binary constraints between the variables of different sorts.
- It can be shown (as in Bulin-D.-Jackson-Niven) that the generated 2-sorted instance is still Maltsev.
- The reduction to the binary constraint case can also be done as in Barto-Kozik papers, via $\mathbb{A}^{\lceil \frac{d}{2} \rceil}$.

Definition

Let $\mathbb{A}$ be an algebra and $B \leq A$ its subuniverse (subalgebra). We say that $\mathbb{B}$ is an absorbing subuniverse if there is a polymorphism t such that

$$t(B, B, \ldots, B, A), t(B, B, \ldots, A, B), \ldots, t(A, B, \ldots, B, B) \subseteq B$$

Theorem

(Kearnes 1991, Valeriote 1990) A finite simple idempotent absorption-free algebra $\mathbb{A}$ is one of the following two types:

1. *$\mathbb{A}$ has skew-free congruences, i.e.*

$$Con(\mathbb{A}^n) \cong \mathbf{2}^n$$

2. *$\mathbb{A}$ is Abelian, and, therefore, a simple affine module*

$$\mathbb{A} = (V_K; \{x - y + z, \lambda x + \mu y \quad (\lambda + \mu = 1)\}),$$

   where V_K is a finite dimensional vector space over a finite field K and $\lambda, \mu \in K$

Finite simple idempotent algebras can also be of the third type but those fail to be absorption-free rather miserably: they have an element which is absorptive with respect to *all* operations.

Congruence skew-free property in conjunction with the lack of absorption yields the key ingredient in a number of Barto-Kozik proofs.

Proposition

Every simple, idempotent, congruence skew-free which lies in a Maltsev variety is functionally complete.

which yields:

Proposition

Let $\mathbb{A}_1, \ldots, \mathbb{A}_k$ be isomorphic simple, absorption-free, congruence skew-free algebras lying in a Maltsev variety. If $\mathbb{R} \leq_{sp} \prod_i \mathbb{A}_i$ is such that $\pi_i \vee \pi_j = 1_{\mathbb{R}}$, then $\mathbb{R} = \prod_i \mathbb{A}_i$. In addition, $\mathbb{R}$ is absorption-free.

In the case when all $\mathbb{A}_i$'s are isomorphic simple affine modules, the full rectangulation fails but one can work around it using the following basic fact:

Proposition

Every subdirect product of simple affine modules is isomorphic to a direct product of some of the factors in the subdirect product.

This can be viewed as a theorem about basic properties of the solution set of a system of linear equations over a finite field. We are interested in it in the context of a binary constraint language.

Given two copies of the same simple affine module $\mathbb{A}_1$ and $\mathbb{A}_2$, the binary constraint between them is either

1. the graph of an affine isomorphism between them; or,
2. full direct product $\mathbb{A}_1 \times \mathbb{A}_2$.

So, we can view that constraint as either:

1. $x_i + a = x_j$, for some $a \in V$; or
2. $0x_i + 0x_j = 0$ (no restriction on the values of x_i and x_j.)

Analysis of CES

- Given a system of linear equations of the form $x_i + a = x_j$, we introduce an equivalence relation $x_i \sim x_j$ if and only if x_i and x_j are constrained by such an equation.
- The equivalence classes can be computed in logspace (Reingold)
- In order to solve such a syste, we need to be able to fix values for all variables in a $\sim$-class consistently (this is what it means to solve a CES)
- How can such a system fail to have a solution? We derive $0 = a$, which amounts to finding a cyclic pattern based at, say x_i, such that 0 is connected to a in the domain for x_i.
- This can be done in SLAC.
- This idea was used in Egri, Hell, Larose, Rafiey (2013) to design a logspace algorithm that solves the list homomorphism problem for n-permutable digraphs.

Algorithm

1. Transform the instance into a binary constraint language.
2. Run multisorted version of SLAC.

Finally, we emulate Kozik's proof that SLAC solves CSPs over templates of bounded width.

- The usual two ways of shrinking:
  1. Absorption (preserves 1-consistency and SLAC - Kozik's prrof does not require anything beyond being Taylor)
  2. Creating restrictions over isomorphic simple absorption-free domains and rectangulation with regard to the remaining absorption-free domains.
- Since the instance is binary, 1-consistency is easy to establish under the second way of "shrinking".

The key to proving SLAC in the reduced instance is the following claim, in the binary constraint context:

Corollary

Let $\mathbb{B}_1, \mathbb{B}_2, \mathbb{A}_1, \ldots, \mathbb{A}_k$ *be algebras which lie in a Maltsev variety, such that all* $\mathbb{A}_i$*'s are simple, absorption-free, and pairwise isomorphic. Let* $\mathbb{R} \leq_{sp} \mathbb{B}_1 \times \mathbb{B}_2 \times \prod_i \mathbb{A}_i$ *be the solution set of a path pattern, such that*

- B_1 *and* B_2 *are absorption-free.*
- *for each pair* i, j *such that* $1 \leq j \leq 2 < i \leq k+2$ *we have* $\pi_j \vee \pi_i = 1_{\mathbb{R}}$.

Then, $\mathbb{R} = proj_{1,2}(\mathbb{R}) \times proj_{3,\ldots,k+2}(\mathbb{R})$.

and, in the affine case, it turns out that the following property of a CES over a simple affine module is all that is needed: for any two $a, b \in \mathbb{A}$,

$$(a_1, a_2, \ldots, a_n) \text{ is a solution } \Leftrightarrow (a_1 + k, a_2 + k, \ldots, a_n + k) \text{ is a solution}$$

where we assume the canonical labelling of domains by elements of the underlying vector space V and $k \in V$.

Anything Else?

Yes - I can prove the Dichotomy Conjecture for all templates with a binary constraint language, by using a slightly more general argument. If we remove the Maltsev condition in the congruence skew-free case, we potentially lose functional completeness. However, I can prove the rectangulation property without it. (the paper is on ArXiv.)