Maltsev constraints revisited

Ross Willard

University of Waterloo, CAN

Dagstuhl Seminar 15301
July 21, 2015

In the beginning ...

$\mathbb{D} = (D, \Gamma)$ – the template

In this lecture, D and Γ are always finite.

CSP Dichotomy Conjecture (Feder-Vardi, 1990s)

For every $\mathbb{D}$, CSP($\mathbb{D}$) is in P or is NP-complete.

How far are we from solving the conjecture?

Assume $\mathbb{D}$ is core.

What we know (in terms of polymorphisms)

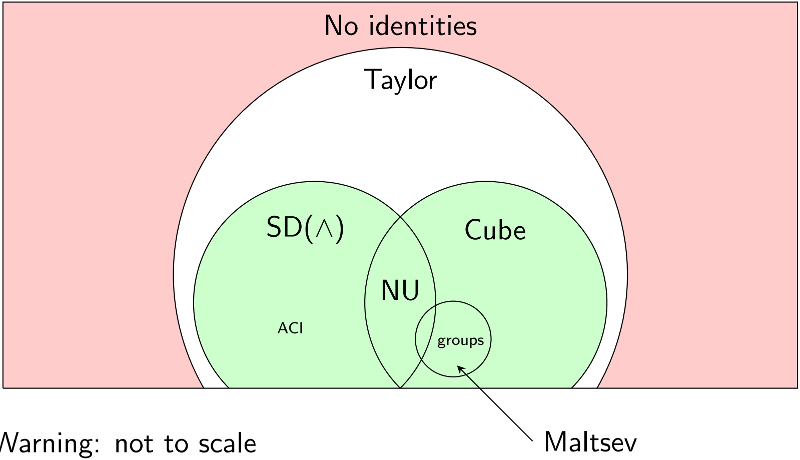
1. If the polymorphisms of $\mathbb{D}$ satisfy no interesting identities, then CSP($\mathbb{D}$) is NP-complete (BJK, 2004).

2. If $\mathbb{D}$ has polymorphism(s) satisfying "SD($\wedge$)" identities **or** "cube" identities, then CSP($\mathbb{D}$) is in P.

   - "SD($\wedge$)" $\Leftrightarrow$ WNUs of all arities ≥ 3.
     - Solvable by local consistency (Barto-Kozik, 2009).
   - "cube" $\Leftrightarrow$ $\boxed{\text{CENSORED}}$.
     - Solvable by the "few subpowers algorithm" (IMMVW, 2007).

In pictures,. . .

Core templates (Red = NP-hard, Green = in P)



Warning: not to scale

Cube vs. Maltsev constraints – A primer

1. Feder-Vardi algorithm for **subgroup constraints**.
   - $\exists$ a group such that $m(x, y, z) := xy^{-1}z$ is a polymorphism.
   - Algorithm adapted from computational group theory.

2. Bulatov's algorithm for **Maltsev constraints** (2002).
   - Polymorphism satisfying $m(x, x, z) = z$ and $m(x, z, z) = x$.
   - Algorithm requires significant universal algebra.

3. Bulatov-Dalmau "simple algorithm" for Maltsev constraints (2006).
   - It's simple.

4. Few subpowers algorithm (IMMVW): the extension of the B-D algorithm to its natural boundary of applicability (cube).

The Bulatov-Dalmau algorithm – Summary

Fix a "cube polymorphism" $c(x_1, \ldots, x_n)$.

Given a CSP($\mathbb{D}$) instance $(V, \mathcal{C})$:

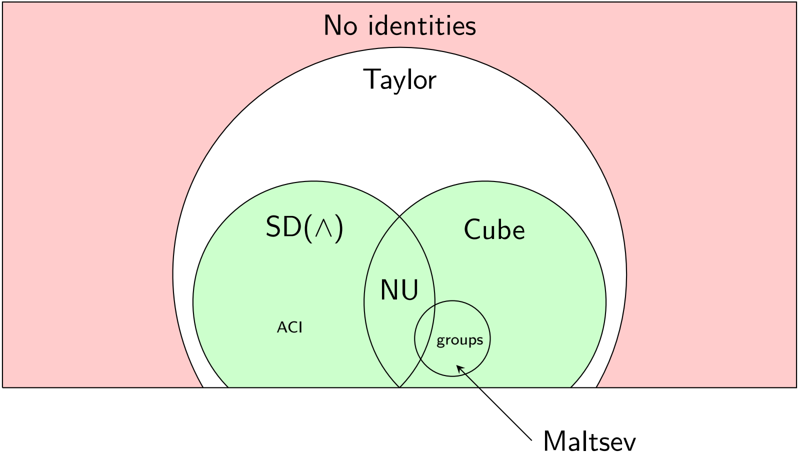
1. Enumerate the constraints.

$\boxed{\text{CENSORED}}$

4. Bulatov-Dalmau give a clever way to $\boxed{\text{CENSORED}}$

Reminiscent of (and generalizes) Gaussian elimination – without having to consider linear equations!

Moving forward



Early optimism: the "white space" should all be in P, solved by combining local consistency and the B-D algorithm.

- But attempts to "glue" the two together have (so far) failed.

The problem, as I see it

1. The Bulatov-Dalmau algorithm is **too simple**.
2. It has encouraged us to not "look under the hood" and see what is "really going on" in cube (or Maltsev) CSP instances.
   - In particular: how linear systems arise in such instances.

Thesis

It should be possible to solve Maltsev (and cube) CSP instances via a mixture of local consistency and "local" Gaussian elimination – not requiring "global" small generating sets.

3. If true, then such a new algorithm could potentially extend beyond the natural boundary of the few subpowers algorithm.

Problem

Understand linear systems in Maltsev CSP instances.

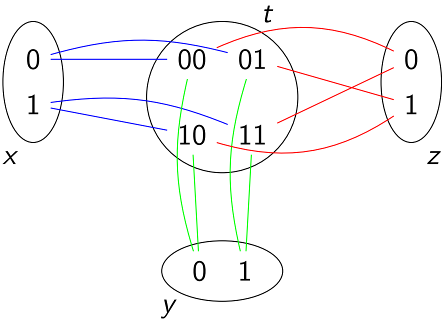
Outline of rest of talk

1. Overly simplistic example suggesting how linear equations arise in binary, subgroup-constraint CSP instances.
2. Generalization by dismissive hand-waving.
3. Some serious problems that arise, vaguely explained.
4. Whimpering, inconclusive finish.

How linear systems arise

Basic gadget

Example: consider three variables x, y, z with domain $\{0, 1\}$:



Introduce a fourth variable t with domain $\{0, 1\}^2$.

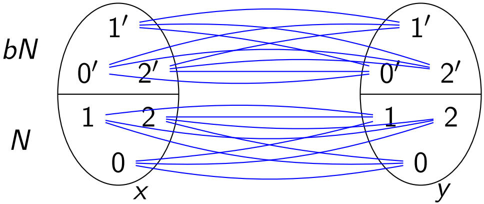
Add constraints between t and x, y, z encoding the two projections and $\oplus$.

This gadget defines $x \oplus y = z$ via binary subgroup constraints.

Variant: subgroups of $(\mathbf{S}_3)^2$

Start with the group $\mathbf{S}_3 = \{1, a, a^2\} \cup \{b, ba, ba^2\} = N \cup bN$.

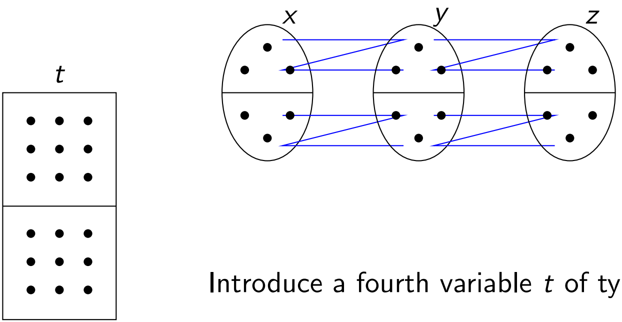
Identify (i.e., coordinatize) each coset of N with a copy of $\mathbb{Z}_3$.



Also define $E = N^2 \cup (bN)^2$; it is a subgroup of $(\mathbf{S}_3)^2$.

$\therefore$ Given two variables x, y of type $\mathbf{S}_3$, we can constrain them by E.

Next, consider three variables x, y, z of type $\mathbf{S}_3$, constrained by E.

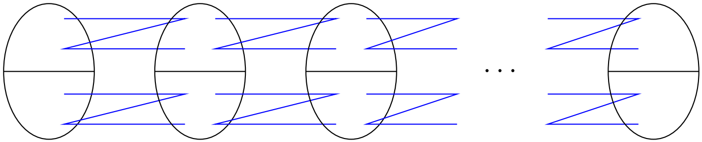


Introduce a fourth variable t of type $\mathbf{E}$.

We can add constraints between t and x, y, z encoding the two projections and "$(t_1, t_2) \mapsto t_1 + t_2 \pmod 3$ on strands."

In this fashion this gadget encodes "$x + y = z \pmod 3$" on each of the two "strands" of blocks.

Now consider having many variables $x_1, \ldots, x_n$ all of type $\mathbf{S}_3$, mutually constrained by E.



Call this a **component**, having two strands.

By introducing variables of type $\mathbf{E}$, we can encode <u>pairs</u> of 3-variable linear equations (one on each strand).

- They need not be the <u>same</u> equation!

In this fashion we encode two systems Σ, Σ' of linear equations, one on each strand.

Consistency can be checked by running Gaussian elimination on each of the two systems.

Let's boogie

Just for fun: encode several system-pairs $(\Sigma_1, \Sigma_1'), \ldots, (\Sigma_k, \Sigma_k')$ on disjoint sets $X_1, \ldots, X_k$ of variables of type $\mathbf{S}_3$.

For each component X_i:

- Introduce a variable v_i of type $\{0, 1\}$.
- Pick $x_i \in X_i$ and constrain x_i, v_i by the parity relation.

Finally, encode your favourite system Δ of 3-variable $\mathbb{Z}_2$-linear equations on $\{v_1, \ldots, v_k\}$, using the gadget $\{0, 1\}^2$.

Algorithm to test consistency:

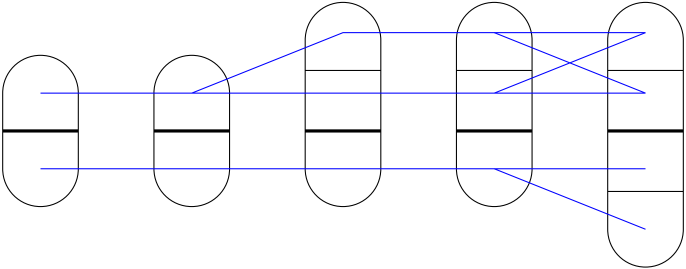
1. For each $i = 1, \ldots, k$, run G.E. on Σ_i and (separately) on Σ_i'.
   - If Σ_i or Σ_i' is inconsistent, delete the strand and update the value of v_i.
   - If for some i, both strands are inconsistent, answer NO.
2. Run G.E. on Δ (with updated values for the v_i's).
3. If consistent, answer YES.

Dismissive hand-waving

General picture

Assume Maltsev (or cube) template, binary constraints.

1. Universal algebra $\Rightarrow$ a theory of "linear equations on strands."
   - Vector spaces arise from "minimal abelian congruences."
   - Each congruence block is "coordinatizable" over a finite field.
   - Gadgets $\Leftrightarrow$ algebras whose minimal congruences form an M_n.
2. Strands obtained by propagation of gadget constraints.



Each strand encodes a linear system.

Serious problems

First problem

Problem 1

A component may have exponentially many different strands.

However, there is a fixed bound (depending on the template) on the number of parts in the “is-connected-to” partition of strands.

Conjecture 1

Connected strands encode the “same” linear system.

Second problem

Recall the example where a $\mathbb{Z}_2$-component "acted on" the strands of several $\mathbf{S}_3$-components.

Problem 2

In general, it can be much worse: a component can act on its own strands!

Conjecture 2

(Assume cube): That's OK! For each connected part of the partition of strands, there is a single "virtual" system determining all the strands, and which doesn't get twisted in knots.

Whimpering finish

Wasn't this lecture supposed to mention algorithms?

Original thesis: it should be possible to solve Maltsev (and cube) CSP instances by a mixture of local consistency and G.E. applied to components.

Unclear if my work will lead to this. An important step to solve:

Computational subproblem (assume cube)

Given a binary $(2, \infty)$-minimal CSP instance and variables $x_1, \ldots, x_k, y$, decide whether "$x_1, \ldots, x_k$ determine y" (in the sense that any solutions agreeing at $x_1, \ldots, x_k$ also agree at y).

- The Bulatov-Dalmau algorithm easily solves this (sigh ...).

Conjecture 3

(Assume cube) If $x_1, \ldots, x_k$ determine y, the potatoes at each x_i and y are subdirectly irreducible, every x_i is essential, $k \geq 2$, and $\boxed{\text{CENSORED}}$, then this must be "explained" by the linear system(s) of a component containing $\{x_1, \ldots, x_k, y\}$.

Thank you!