Arc Consistency and Friends
(joint work with Berit Grussien, Manuel Bodirsky, and Victor Dalmau)

Hubie Chen

Universitat Pompeu Fabra
Barcelona, Spain

CanaDAM - May 2009
The constraint satisfaction problem (CSP) is a general search problem.

In an instance of the CSP:

- We are given: set of variables $A$, set $B$ called domain, constraints
- Want to decide: does there exist assignment $A \rightarrow B$ satisfying all constraints?

Many combinatorial problems can be naturally cast as cases of the CSP:

- boolean satisfiability ($A =$ variables, $B = \{0, 1\}$)
- graph coloring ($A =$ vertices, $B =$ colors)
- scheduling ($A =$ events, $B =$ time slots)

Complexity: NP-complete in general (in usual formulations)

We will cast the CSP as the problem of deciding if there is a homomorphism between two given relational structures.
A signature is a set of relation symbols $\sigma = \{R, \ldots, \}$; each symbol $R$ has an associated arity $k_R$.

Example: to talk about graphs, signature $\sigma = \{E\}$ with $k_E = 2$.

A structure $\mathcal{B}$ over $\sigma$ consists of:

- A universe $B$, a set (finite here)
- A relation $R^\mathcal{B} \subseteq B^{k_R}$ for each $R \in \sigma$

Given two structures $\mathcal{A}, \mathcal{B}$ over the same signature $\sigma$, a homomorphism is a mapping $h : \mathcal{A} \rightarrow \mathcal{B}$ such that

$$\text{for each tuple } (a_1, \ldots, a_k) \in R^\mathcal{A},$$
$$\text{it holds that } (h(a_1), \ldots, h(a_k)) \in R^\mathcal{B}$$

(over all symbols $R \in \sigma$)

Warning: I will mix terminology a bit (e.g. refer to elements of $A$ as variables)
Official definition: CSP is to decide, given two structures $A, B$, if there exists hom. $A \rightarrow B$

A heavily studied restriction of the CSP:

$$CSP(B) = \{ A \mid \text{exists hom. } A \rightarrow B \}$$

Now, for each structure $B$ we have a different computational problem $CSP(B)$
A Framework of Problems

- This restriction of CSP allows one to capture & place into a unified framework many different cases of CSP that have been studied:
  - 2-SAT
  - Horn-SAT
  - 3-Colorability
  - $H$-Colorability
  - Solving systems of equations

- Research problem: classify the complexity of all problems CSP($\mathcal{B}$)

- First results by Schaefer ’78 (two-elt. structures), Hell & Nesetril ’90 (undirected graphs)

- Another type of problem: given an algorithm (or algorithmic technique), describe the structures

\[ \{ \mathcal{B} \mid \text{CSP}(\mathcal{B}) \text{ solvable by algorithm} \} \]
Arc Consistency

- Efficient (polytime) algorithm for detecting unsat. instances
- Idea: make local inferences using one variable at a time
- Generalizes *unit propagation* for CNF-SAT
- Basic reasoning technique in constraint solving, used in practice
- Theme of this talk:
  show how (in the CSP(\mathbf{B}) world) arc consistency and extensions thereof can be studied and understood using algebraic tools
Arc Consistency: the algorithm

- Given an instance \((A, B)\) of the hom. problem...
- To each variable \(a \in A\), associate a set \(S_a\)
- We maintain invariant: for any hom. \(h : A \rightarrow B\) and any \(a \in A\), it must hold that \(h(a) \in S_a\)
- Initialize \(S_a = B\) for each \(a \in A\)
- Iterate until convergence:
  for each tuple \((a_1, \ldots, a_k) \in R^A:\)
  let \(T = R^B \cap (S_{a_1} \times \ldots \times S_{a_k})\) then, for each \(i\), set \(S_{a_i} = \pi_i(T)\)
- If for some \(a \in A\) it holds that \(S_a = \emptyset\), output “unsatisfiable”
  Else output “?”
A sound but incomplete procedure

- AC is “sound but incomplete”
  - If AC outputs “unsatisfiable” ⇒ instance really unsat.
  - If AC outputs “?” ⇒ instance may be sat. or unsat.

- Incompleteness follows from P not equal NP (as AC runs in P)

- A quick unconditional proof:
  - Let B be $K_3$
  - We have $E^B = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\}$
  - The projection of $E^B$ onto either coordinate always gives \{1, 2, 3\}, so no inference will take place: algorithm terminates with $S_a = \{1, 2, 3\}$ for all $a \in A$
  - Let A be any graph with no hom. to B, e.g. $A = K_4$
  - On (A, B), AC algorithm returns “?” on an unsat. instance
The power structure

- We can give an algebraic condition that holds when AC algorithm returns “?”
- This will allow us to study AC from a number of angles using algebraic tools
- We can define, for each structure $B$, a *power structure* $\mathcal{P}(B)$ such that

  \[
  \text{AC returns “?” on } (A, B) \iff \text{there is a hom. } A \rightarrow \mathcal{P}(B)
  \]

- **Definition of $\mathcal{P}(B)$:**
  - Universe is $\mathcal{P}(B) \setminus \{\emptyset\}$
  - For each $R$, we have
    \[
    R^{\mathcal{P}(B)} = \{(\pi_1(S), \ldots, \pi_k(S)) \mid S \subseteq R^B, S \neq \emptyset\}
    \]
- **Key point:** from $B$ there is a derived structure $\mathcal{P}(B)$ that characterizes AC in the above way
A characterization of when AC works

- Although AC is not complete in general, there are some structures $B$ where AC solves ("is complete for") CSP($B$): for all structures $A$
  
  there is a hom. $A \rightarrow \mathcal{P}(B) \Rightarrow$ there is a hom. $A \rightarrow B$

- Theorem (Feder & Vardi '93, Dalmau & Pearson '99): AC solves CSP($B$) if and only if there is a hom. from $\mathcal{P}(B) \rightarrow B$

- Proof ($\Rightarrow$):
  
  Suppose AC solves CSP($B$).
  
  There is a hom. $\mathcal{P}(B) \rightarrow \mathcal{P}(B)$, hence there is a hom. $\mathcal{P}(B) \rightarrow B$

- Proof ($\Leftarrow$):
  
  Suppose there is a hom. $\mathcal{P}(B) \rightarrow B$
  
  Let $A$ be such that there is a hom. $A \rightarrow \mathcal{P}(B)$
  
  Compose the two homs. to get a hom. $A \rightarrow B$
What can we do with this characterization?

Theorem (Feder & Vardi ’93, Dalmau & Pearson ’99): AC solves CSP($\mathcal{B}$) if and only if there is a hom. from $\mathcal{P}(\mathcal{B}) \rightarrow \mathcal{B}$

What can we do with this characterization (and others like it)?

1. Characterization implies that given a structure $\mathcal{B}$, it is *decidable* if AC solves CSP($\mathcal{B}$): simply check for a hom. from $\mathcal{P}(\mathcal{B}) \rightarrow \mathcal{B}$
   ⇒ helps us understand which structures are solvable by AC

2. Also helps us understand the class of structures
   \[ \{ \mathcal{B} \mid \text{CSP(}\mathcal{B}\text{) solvable by AC} \} \] in aggregate: can prove
   - $\mathcal{B}$ solvable by AC, $\mathcal{B}'$ is an expansion of $\mathcal{B}$ by pp-def. relations
     ⇒ $\mathcal{B}'$ solvable by AC
   - $\mathcal{B}$ solvable by AC, $\mathcal{B}'$ is hom. equivalent to $\mathcal{B}$
     ⇒ $\mathcal{B}'$ solvable by AC
Look-Ahead Arc Consistency (LAAC)

- **LAAC algorithm:**
  - Arbitrarily pick a variable $a$
  - Try to find a value $b$ such that after $a$ set to $b$, AC returns “?”
  - If no such value found, terminate and output “?”
  - If such a value found, fix $a$ to $b$, eliminate $a$, and repeat

- Algorithm either outputs “?” or a satisfying assignment
  Note: “one-sided error”, but on the other side

- Only conceptual primitive used other than picking/setting variables is AC!

- **Theorem (Chen & Dalmau ’04):** LAAC solves CSP($\mathcal{B}$) if and only if there is a hom. $h : \mathcal{P}(\mathcal{B}) \times \mathcal{B} \rightarrow \mathcal{B}$ such that $h(\{b\}, c) = b$ for all $b, c \in \mathcal{B}$

- **Notes:** can decide if LAAC solves CSP($\mathcal{B}$); LAAC solves 2-SAT
 Peek Arc Consistency (PAC)

- Idea: take a peek at each variable
- PAC algorithm:
  - For each variable-value pair \((a, b)\), set \(a\) to \(b\) and check to see what AC returns
  - If for some variable \(a\) it holds that all values \(b\) result in AC returning “unsat”, return “unsat”
  - Else return “?”
- Algorithm returns “unsat” or “?” as AC does
- Theorem (Bodirsky and Chen): Let \(I(\mathcal{P}(\mathcal{B})^n)\) denote the induced substructure of \(\mathcal{P}(\mathcal{B})^n\) containing all elements of \(\mathcal{P}(\mathcal{B})^n\) with at least one coordinate a singleton. PAC solves \(\text{CSP}(\mathcal{B}) \iff \forall n \geq 1, \text{there’s a hom. } I(\mathcal{P}(\mathcal{B})^n) \rightarrow \mathcal{B}\)
- Note: induced substructure of a structure defined similarly to induced subgraph of a graph
Theorem:
PAC solves CSP($\mathbf{B}$) ⇔ ∀n ≥ 1, there’s a hom. $I(\mathcal{P}(\mathbf{B})^n) \rightarrow \mathbf{B}$

Proof ($\Leftarrow$):

- Easier direction: have algebraic char., want to show alg. works
- Suppose PAC doesn’t return unsat. on ($\mathbf{A}, \mathbf{B}$)
- Then for each $a \in \mathbf{A}$, have a hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B})$ sending $a$ to singleton
- Combining these homs., get hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B})^{|\mathbf{A}|}$ setting each $a \in \mathbf{A}$ to a tuple having some singleton
- This is a hom. $\mathbf{A} \rightarrow I(\mathcal{P}(\mathbf{B})^{|\mathbf{A}|})$
- Compose it with the hom. in theorem statement to get hom. $\mathbf{A} \rightarrow \mathbf{B}$
Theorem:
PAC solves CSP(B) ⇔ ∀n ≥ 1, there’s a hom. I(P(B)^n) → B

Proof (⇒):
- Suppose PAC solves CSP(B)
- Want to show that there is a hom. I(P(B)^n) → B
- Consider input A = I(P(B)^n); we show that PAC returns “?” on this
- Let a = (S_1, . . . , S_n) be any variable from A
- Know that by def., some S_i is a singleton {b}
- Consider the mapping π_i that projects onto the i-th coordinate
- This is a hom. from A to P(B) sending a to a singleton {b}
- Hence, PAC returns “?” on (A, B)
- By assumption that PAC solves CSP(B), get hom. A → B
Singleton Arc Consistency (SAC)

- Idea: like PAC, but perform propagation
- SAC algorithm:
  - Loop until no more changes:
    - For each variable-value pair \((a, b)\), set \(a\) to \(b\) and run AC; if AC returns “unsat” then remove all instances of \((a, b)\) from instance
    - If for some variable \(a\) there are no values \(b\), return “unsat”; else return “?”
  - Algorithm returns “unsat” or “?” as AC and PAC do
- Theorem: SAC solves \(\text{CSP}(B) \iff \forall n \geq 1, \) there’s a hom. from \(\ldots \rightarrow B\)
- Tractability results (in progress): SAC solves structures preserved by
  - any majority operation
  - any 2-semilattice operation
A hierarchy

- Theorem: $\text{AC} \subset \text{LAAC} \subseteq \text{PAC} \subset \text{SAC}$
- Here, an algorithm denotes the set of structures $B$ that it solves
- Note: proof of inclusion $\text{LAAC} \subseteq \text{PAC}$ is algebraic, uses algebraic characterizations of algorithms
  - I know of no proof based just on the algorithm descriptions
Perspective

- Algebraic tools have been used to understand the family of problems CSP(\(B\))
- Work on understanding when problems fall in/out of complexity classes:
  - Sufficient condition for NP-hardness - Bulatov et al.
  - Sufficient conditions for hardness for other comp. classes - Larose & Tesson
  - Classification results for large families of structures - Bulatov
- Work on understanding when certain algorithmic techniques work:
  - Maintaining a succinct representation / “few subalgebras” - Dalmau, Berman et al.
  - Bounded width - Larose & Zadori, Atserias et al., Bulatov, Kiss & Valeriote, Barto & Kozik, ...
- In contrast, here, we use algebraic tools to understand particular, concrete algorithms
Open questions

- Bounded width algorithms \( \approx \) generalizations of arc consistency where inference performed on \( k \) variables at a time
- Open question 1: are there analogs of the structure \( \mathcal{P}(\mathcal{B}) \) that characterize higher forms of consistency?
- Class of structures \( \mathcal{B} \) for which bounded width solves \( \text{CSP}(\mathcal{B}) \) has been classified (Barto & Kozik)
- Open question 2: can singleton arc consistency solve all such problems?
- Thanks!