

Arc Consistency and Friends

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

Hubie Chen

Universitat Pompeu Fabra
Barcelona, Spain

CanaDAM - May 2009

Constraint Satisfaction

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

Structures and Homomorphisms

- ▶ A *signature* is a set of relation symbols $\sigma = \{R, \dots, \}$; each symbol R has an associated *arity* k_R
- ▶ Example: to talk about graphs, signature $\sigma = \{E\}$ with $k_E = 2$
- ▶ A *structure* \mathbf{B} over σ consists of:
 - ▶ A *universe* B , a set (finite here)
 - ▶ A relation $R^{\mathbf{B}} \subseteq B^{k_R}$ for each $R \in \sigma$
- ▶ Given two structures \mathbf{A}, \mathbf{B} over the same signature σ , a homomorphism is a mapping $h : A \rightarrow B$ such that

$$\begin{aligned} & \text{for each tuple } (a_1, \dots, a_k) \in R^{\mathbf{A}}, \\ & \text{it holds that } (h(a_1), \dots, h(a_k)) \in R^{\mathbf{B}} \end{aligned}$$

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

- ▶ Warning: I will mix terminology a bit (e.g. refer to elements of A as *variables*)

A Framework of Problems

- ▶ Official definition: CSP is to decide, given two structures \mathbf{A} , \mathbf{B} , if there exists hom. $\mathbf{A} \rightarrow \mathbf{B}$
- ▶ A heavily studied restriction of the CSP:

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

- ▶ Now, for each structure \mathbf{B} we have a different computational problem $\text{CSP}(\mathbf{B})$

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 $\text{CSP}(\mathbf{B})$
- ▶ First results by Schaefer '78 (two-elt. structures), Hell & Nešetřil '90 (undirected graphs)
- ▶ Another type of problem: given an algorithm (or algorithmic technique), describe the structures

$$\{\mathbf{B} \mid \text{CSP}(\mathbf{B}) \text{ solvable by algorithm} \}$$

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(**B**) world) arc consistency and extensions thereof can be studied and understood using algebraic tools

Arc Consistency: the algorithm

- ▶ Given an instance (\mathbf{A}, \mathbf{B}) of the hom. problem...
- ▶ To each variable $a \in A$, associate a set S_a
- ▶ We maintain invariant: for any hom. $h : \mathbf{A} \rightarrow \mathbf{B}$ 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 $(a_1, \dots, a_k) \in R^{\mathbf{A}}$:
let $T = R^{\mathbf{B}} \cap (S_{a_1} \times \dots \times S_{a_k})$ 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” \Rightarrow instance really unsat.
 - ▶ If AC outputs “?” \Rightarrow instance may be sat. or unsat.
- ▶ Incompleteness follows from P not equal NP (as AC runs in P)
- ▶ A quick unconditional proof:
 - ▶ Let \mathbf{B} be K_3
 - ▶ We have $E^{\mathbf{B}} = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\}$
 - ▶ The projection of $E^{\mathbf{B}}$ 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 \mathbf{A} be any graph with no hom. to \mathbf{B} , e.g. $\mathbf{A} = K_4$
 - ▶ On (\mathbf{A}, \mathbf{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 \mathbf{B} , a *power structure* $\mathcal{P}(\mathbf{B})$ such that

AC returns “?” on $(\mathbf{A}, \mathbf{B}) \Leftrightarrow$ there is a hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B})$

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

A characterization of when AC works

- ▶ Although AC is not complete in general, there are some structures \mathbf{B} where AC solves (“is complete for”) $\text{CSP}(\mathbf{B})$: for all structures \mathbf{A}

there is a hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B}) \Rightarrow$ there is a hom. $\mathbf{A} \rightarrow \mathbf{B}$

- ▶ Theorem (Feder & Vardi '93, Dalmau & Pearson '99): AC solves $\text{CSP}(\mathbf{B})$ if and only if there is a hom. from $\mathcal{P}(\mathbf{B}) \rightarrow \mathbf{B}$
- ▶ Proof (\Rightarrow):
 - ▶ Suppose AC solves $\text{CSP}(\mathbf{B})$.
 - ▶ There is a hom. $\mathcal{P}(\mathbf{B}) \rightarrow \mathcal{P}(\mathbf{B})$, hence there is a hom. $\mathcal{P}(\mathbf{B}) \rightarrow \mathbf{B}$
- ▶ Proof (\Leftarrow):
 - ▶ Suppose there is a hom. $\mathcal{P}(\mathbf{B}) \rightarrow \mathbf{B}$
 - ▶ Let \mathbf{A} be such that there is a hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B})$
 - ▶ Compose the two homs. to get a hom. $\mathbf{A} \rightarrow \mathbf{B}$

What can we do with this characterization?

- ▶ Theorem (Feder & Vardi '93, Dalmau & Pearson '99): AC solves $\text{CSP}(\mathbf{B})$ if and only if there is a hom. from $\mathcal{P}(\mathbf{B}) \rightarrow \mathbf{B}$
- ▶ What can we do with this characterization (and others like it)?
 1. Characterization implies that given a structure \mathbf{B} , it is *decidable* if AC solves $\text{CSP}(\mathbf{B})$: simply check for a hom. from $\mathcal{P}(\mathbf{B}) \rightarrow \mathbf{B}$
 \Rightarrow helps us understand which structures are solvable by AC
 2. Also helps us understand the class of structures $\{ \mathbf{B} \mid \text{CSP}(\mathbf{B}) \text{ solvable by AC} \}$ in aggregate: can prove
 - ▶ \mathbf{B} solvable by AC, \mathbf{B}' is an expansion of \mathbf{B} by pp-def. relations
 $\Rightarrow \mathbf{B}'$ solvable by AC
 - ▶ \mathbf{B} solvable by AC, \mathbf{B}' is hom. equivalent to \mathbf{B}
 $\Rightarrow \mathbf{B}'$ 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 $\text{CSP}(\mathbf{B})$ if and only if there is a hom. $h : \mathcal{P}(\mathbf{B}) \times \mathbf{B} \rightarrow \mathbf{B}$ such that $h(\{b\}, c) = b$ for all $b, c \in B$
- ▶ Notes: can decide if LAAC solves $\text{CSP}(\mathbf{B})$;
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 $I(\mathcal{P}(\mathbf{B})^n)$ denote the induced substructure of $\mathcal{P}(\mathbf{B})^n$ containing all elements of $\mathcal{P}(\mathbf{B})^n$ with at least one coordinate a singleton.
PAC solves $\text{CSP}(\mathbf{B}) \Leftrightarrow \forall n \geq 1$, there's a hom. $I(\mathcal{P}(\mathbf{B})^n) \rightarrow \mathbf{B}$
- ▶ Note: induced substructure of a structure defined similarly to induced subgraph of a graph

Peek Arc Consistency (PAC)

- ▶ Theorem:
PAC solves $\text{CSP}(\mathbf{B}) \Leftrightarrow \forall n \geq 1$, 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 $a \in A$, have a hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B})$ sending a to singleton
 - ▶ Combining these homs., get hom. $\mathbf{A} \rightarrow \mathcal{P}(\mathbf{B})^{|A|}$ setting each $a \in A$ to a tuple having some singleton
 - ▶ This is a hom. $\mathbf{A} \rightarrow I(\mathcal{P}(\mathbf{B})^{|A|})$
 - ▶ Compose it with the hom. in theorem statement to get hom. $\mathbf{A} \rightarrow \mathbf{B}$

Peek Arc Consistency (PAC)

- ▶ Theorem:
PAC solves $\text{CSP}(\mathbf{B}) \Leftrightarrow \forall n \geq 1$, there's a hom. $I(\mathcal{P}(\mathbf{B})^n) \rightarrow \mathbf{B}$
- ▶ Proof (\Rightarrow):
 - ▶ Suppose PAC solves $\text{CSP}(\mathbf{B})$
 - ▶ Want to show that there is a hom. $I(\mathcal{P}(\mathbf{B})^n) \rightarrow \mathbf{B}$
 - ▶ Consider input $\mathbf{A} = I(\mathcal{P}(\mathbf{B})^n)$; we show that PAC returns “?” on this
 - ▶ Let $a = (S_1, \dots, S_n)$ 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 \mathbf{A} to $\mathcal{P}(\mathbf{B})$ sending a to a singleton $\{b\}$
 - ▶ Hence, PAC returns “?” on (\mathbf{A}, \mathbf{B})
 - ▶ By assumption that PAC solves $\text{CSP}(\mathbf{B})$, get hom. $\mathbf{A} \rightarrow \mathbf{B}$

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 $\text{CSP}(\mathbf{B}) \Leftrightarrow \forall n \geq 1$, there's a hom. from $\dots \rightarrow \mathbf{B}$
- ▶ Tractability results (in progress): SAC solves structures preserved by
 - ▶ any majority operation
 - ▶ any 2-semilattice operation

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

- ▶ 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 $\mathcal{P}(\mathbf{B})$ that characterize higher forms of consistency?
- ▶ Class of structures \mathbf{B} for which bounded width solves $\text{CSP}(\mathbf{B})$ has been classified (Barto & Kozik)
- ▶ Open question 2: can singleton arc consistency solve all such problems?
- ▶ Thanks!