A Rendezvous of Logic, Complexity, and Algebra

Hubie Chen
presented by Martin Kupec

Definition 1. *A* constraint *over a constraint language* Γ *is an expression of the form* $R(v_1, \ldots, v_k)$ *where* R *is a relation of arity* k *contained in* Γ*, and the* v_i *are variables. A constraint is satisfied by a mapping* f *defined on the* v_i *if* $(f(v_1), \ldots, f(v_k)) \in R$*.*

Example 2. We demonstrate that 3-SAT can be viewed as a problem of the form $\mathsf{CSP}(\Gamma)$ for a boolean constraint language Γ. Define the relations $R_{0,3}$, $R_{1,3}$, $R_{2,3}$, and $R_{3,3}$ by

$$\begin{array}{rcl} R_{0,3} & = & \{0,1\}^3 \setminus \{(0,0,0)\} & \equiv & (x \vee y \vee z) \\ R_{1,3} & = & \{0,1\}^3 \setminus \{(1,0,0)\} & \equiv & (\neg x \vee y \vee z) \\ R_{2,3} & = & \{0,1\}^3 \setminus \{(1,1,0)\} & \equiv & (\neg x \vee \neg y \vee z) \\ R_{3,3} & = & \{0,1\}^3 \setminus \{(1,1,1)\} & \equiv & (\neg x \vee \neg y \vee \neg z) \end{array}$$

Definition 3. *We say that a relation* $R \subseteq D^k$ *is* pp-definable *(short for* primitive positive definable*) from a constraint language* Γ *if for some* $m \geq 0$ *there exists a finite conjunction* $\mathcal{C}$ *consisting of constraints and equalities* $(u = v)$ *over variables* $\{v_1, \ldots, v_k, x_1, \ldots, x_m\}$ *such that*

$$R(v_1, \ldots, v_k) \equiv \exists x_1 \ldots \exists x_m \mathcal{C}.$$

That is, R *contains exactly those tuples of the form* $(g(v_1), \ldots, g(v_k))$ *where* g *is an assignment that can be extended to a satisfying assignment of* $\mathcal{C}$*. We use* $\langle \Gamma \rangle$ *to denote the set of all relations that are pp-definable from* Γ*.*

Example 4. Let $S = \{(0,1), (1,0)\}$ be the disequality relation. The following is a pp-definition of S from the constraint language Γ_3 (3-SAT).

$$S(y,z) = \exists x (R_{0,3}(x,y,z) \wedge R_{1,3}(x,y,z) \wedge R_{2,3}(z,y,x) \wedge R_{3,3}(z,y,x)).$$

Proposition 5. *Let* Γ *and* Γ' *be finite constraint languages. If* $\Gamma' \subseteq \langle \Gamma \rangle$*, then* $\mathsf{CSP}(\Gamma')$ *reduces to* $\mathsf{CSP}(\Gamma)$*.*

Definition 6. *An operation* $f : D^m \to D$ *is a* polymorphism *of a relation* $R \subseteq D^k$ *if for any choice of* m *tuples* $(t_{11}, \ldots, t_{1k}), \ldots, (t_{m1}, \ldots, t_{mk})$ *from* R*, it holds that the tuple obtained from these* m *tuples by applying* f *coordinate-wise,* $(f(t_{11}, \ldots, t_{m1}), \ldots, f(t_{1k}, \ldots, t_{mk}))$*, is in* R*.*

Definition 7. *The set of polymorphisms of* Γ *is defined as follows.*

$$\mathsf{Pol}(\Gamma) = \{f : \forall R \in \Gamma, f \text{ is a polymorphism of } R\}.$$

Definition 8. *The set of relations having all operations in* O *as a polymorphism is denoted by* $\mathsf{Inv}(O)$*.*

$$\mathsf{Inv}(O) = \{R : \forall f \in O, f \text{ is a polymorphism of } R\}.$$

Theorem 9. *Let* Γ *be a finite constraint language over a finite domain* D*. It holds that* $\langle \Gamma \rangle = \mathsf{Inv}(\mathsf{Pol}(\Gamma))$*.*

Theorem 10. *Let* Γ *and* Γ' *be finite constraint languages. If* $\mathsf{Pol}(\Gamma) \subseteq \mathsf{Pol}(\Gamma')$*, then* $\Gamma' \subseteq \langle \Gamma \rangle$ *and* $\mathsf{CSP}(\Gamma')$ *reduces to* $\mathsf{CSP}(\Gamma)$*.*

Definition 11. *A* clone *is a set of operations that*

- *contains all* projections*, that is, the operations* $\pi_i^m : D^m \to D$ *with* $1 \le i \le m$ *such that* $\pi_i^m(d_1, \ldots, d_m) = d_i$ *for all* $d_1, \ldots, d_m \in D$*, and*
- *is closed under composition, where the composition of an arity* n *operation* $f : D^n \to D$ *and* n *arity* m *operations* $f_1, \ldots, f_n : D^m \to D$ *is defined to be the arity* m *operation* $g : D^m \to D$ *such that* $g(d_1, \ldots, d_m) = f(f_1(d_1, \ldots, d_m), \ldots, f_n(d_1, \ldots, d_m))$ *for all* $d_1, \ldots, d_m \in D$*.*

Proposition 12. *For all constraint languages* Γ*, the set of operations* $\mathsf{Pol}(\Gamma)$ *is a clone.*

Theorem 13. *(Schaefer's theorem – algebraic formulation) Let* Γ *be a finite boolean constraint language. The problem* $\mathsf{CSP}(\Gamma)$ *is polynomial-time tractable if* Γ *has one of the following six operations as a polymorphism:*

- *the constant operation* 0*,*
- *the constant operation* 1*,*
- *the boolean AND operation* $\wedge$*,*
- *the boolean OR operation* $\vee$*,*
- *the operation* majority*,*
- *the operation* minority*.*

Otherwise, the problem $\mathsf{CSP}(\Gamma)$ *is NP-complete.*

Algorithms used for tractability proof.

Arc consistency algorithm
Input: an instance of the CSP.
1 For each variable v, define D_v to be $\cap_C \pi_v(C)$ where the intersection is over all constraints C.
2 For each constraint $R(v_1, \ldots, v_k)$, replace R with $R \cap (D_{v_1} \times \cdots \times D_{v_k})$.
If R becomes empty, then terminate and report "unsatisfiable".
3 If any relations were changed in step 2, goto step 1. Otherwise, halt.

Algorithm for majority polymorphism
Input: an instance ϕ of the CSP with variable set V.
1 For each non-empty subset $W = \{w_1, \ldots, w_l\}$ of V of size $l \le 3$,
add the constraint $D^l(w_1, \ldots, w_l)$ to ϕ.
2 For each constraint $R(w_1, \ldots, w_l)$ of ϕ with $l \le 3$, compute the set
$R' = \{(f(w_1), \ldots, f(w_l)) \mid f : \{w_1, \ldots, w_l\} \to D$ is a partial solution of the instance $\phi\}$.
Then, replace R with R'.
If R becomes empty, terminate and report "unsatisfiable".
3 If any relations were changed in step 2, goto step 2 and repeat it. Otherwise, halt.

Theorem 14. *A clone over* $\{0, 1\}$ *either contains only essentially unary operations, or contains one of the following four operations:*

- *the boolean AND operation* $\wedge$*,*
- *the booelan OR operation* $\vee$*,*
- *the operation* majority*,*
- *the operation* minority*.*

Lemma 15. *If* Γ *is a finite boolean constraint language such that* $\mathsf{Pol}(\Gamma)$ *contains only essentially unary operations that act as permutations, then for any finite boolean constraint language* Γ'*, it holds that* $\mathsf{CSP}(\Gamma')$ *reduces to* $\mathsf{CSP}(\Gamma)$*.*

Proposition 16. *Let* Φ *be an instance of* Quantified Horn-SAT *having prefix class* Π_2*. The formula* Φ *is true if and only if for every assignment* $f \in [\le 1, \mathsf{false}]_\Phi$*, there exists an extension* $f' : Y_\Phi \cup X_\Phi \to \{\mathsf{true}, \mathsf{false}\}$ *of* f *satisfying all clauses of* Φ*.*