Constraint Satisfaction: an Introduction

Moshe Y. Vardi

Rice University
http://www.cs.rice.edu/∼vardi

Constraint Satisfaction

WWW index: 928,00 on Google (vs. 206,000 for "universal algebra")

Applications:

- belief maintenance
- machine vision
- natural language processing
- planning and scheduling
- temporal reasoning
- type reconstruction
- bioinformatics
- ...

Scientific presence: a conference and a journal.

Constraint-Satisfaction Problem (CSP)

Input: (V, D, C):

- A finite set V of *variables*
- A set D of *values*
- A finite set C of *constraints* restricting the values that tuples of variables can take.

Constraint: (t, R)

- t: a tuple of variables over V
- R: a relation of arity $|t|$

– Assume R unique given t and denote $R(t)$.

- Finite D: *discrete* constraints

Solution: assignment $h : V \rightarrow D$

- $h(t) \in R$: for all $(t, R) \in C$

Problem: Does (V, D, C) have a solution? I.e., is there an assignment of values to the variables such that all constraints are satisfied? (We typically want also to find a solution.)

Example: 3-Colorability

3-COLOR: Given an undirected graph $A = (V, E)$, is it 3-colorable?

- The variables are the nodes in V.
- The values are the elements in $\{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$.
- The constraints are $\{(\langle u, v\rangle, \rho) \quad : \quad (u, v) \in E\}$, where $\rho = \{(R, G), (R, B), (G, R), (G, B), (B, R), (B, G)\}$.

Example: Temporal Constraints

"It is evening; Paulo and Nikos decide to eat pizza together. Paulo needs 30 to 40 minutes to reach the pizzeria. Nikos gets there by bike; depending on his mood, it takes him either 10 to 20 minutes or 30 to 40 minutes. Nikos leaves the office between 7:20pm and 7:30pm. Paulo leaves home between 7:00pm and 7:10pm. Should they have Salami or Pepperoni?"

Formally:

- Variables range over $\mathcal{R}$
- A constraint over (x_i, x_j) is of the form

$$l_1 \leq x_j - x_i \leq r_1 \vee \ldots l_n \leq x_j - x_i \leq r_n$$

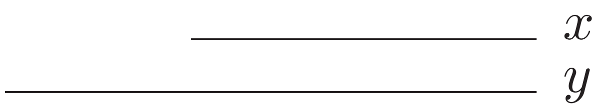
where the real intervals $[l_1, r_1], \ldots, [l_n, r_n]$ are pairwise disjoint.

Allen's Interval Algebra

Key: Reasoning with temporal intervals, rather than with temporal points

- Variables range over temporal intervals ($\mathcal{R}^2$)
- Constraints: *during*, *before*, *meets*, *overlaps*, *finishes*, etc. (13 relations)

Example: x *finishes* y



Example: Language Processing

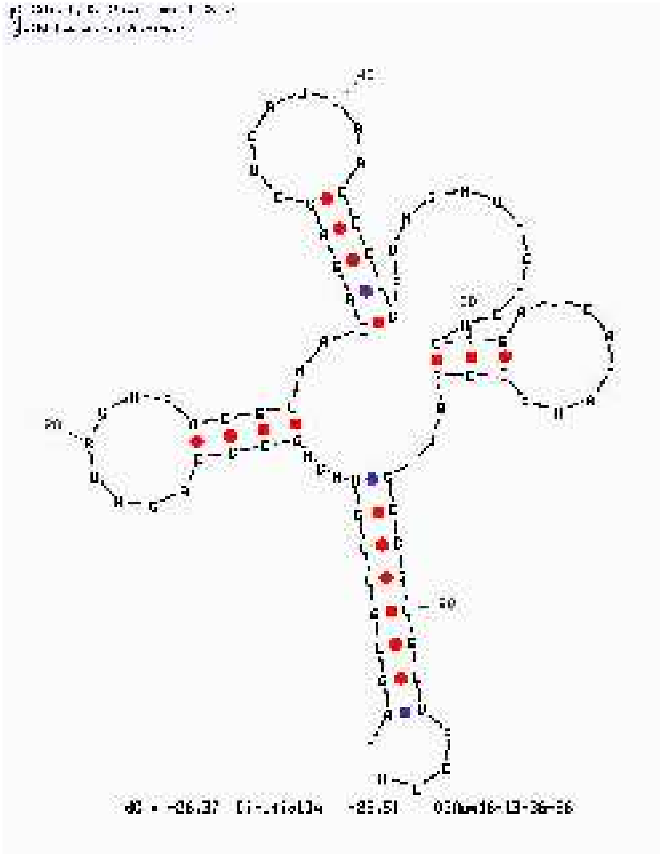
Syntactical Analysis of Input Sentences:

- Variables correspond to words and range over syntactical types.
- Types have to be in agreement, e.g., determiner-adjective-noun agreement.
- Agreement depends on gender, number, person, case, determiner type.

E.g., "*der schönen Mannes*".

Example: RNA Secondary Structure

Secondary Structure of RNA Sequences:



• RNA: sequence of base amino acids (Adenine, Guanine, Cytosine, Uracile)

• Variables correspond to bases and range over other bases for secondary interactions

• Constraints of different types:

– Pairing: A pairs with U, G pairs with C or U

– Stereochemical

Discrete Constraint Solving

Two basic algorithmic approaches:

- *Backtracking Search*: Repeatedly attempt to extend a partial solution by instantiating a new variable without violating constraints; backtrack when no extension possible.

- *Constraint Propagation*: Infer new constraints from existing constraints to reduce search-space size.

Complementarity: Both approaches are sound and complete, but solvers typically interleave propagation and search.

Example: DPLL

SAT(φ): (φ in CNF)

(1) If $\varphi \in \{0, 1\}$ return φ
(2) If φ contains p (resp., $\neg p$)
return SAT($\varphi[p \mapsto 1]$) (resp., SAT($\varphi[p \mapsto 0]$))
(3) choose $p \in Prop(\varphi)$ and $c \in \{0, 1\}$
return SAT($\varphi[p \mapsto c]$)$\vee$SAT($\varphi[p \mapsto 1 - c]$)

Commentary:

1. Termination
2. Propagation
3. Search

DPLL in Practice:

- Almost 50 years of algorithmic engineering
- Solves problems with *millions* of clauses
- SAT as a generic problem solver ("*NP-easiness*")

SAT in Practice: Bounded Model Checking

Abstract Hardware $H = (I, R, In, Tr)$

- I: Boolean input variables
- R: Boolean state variables
- $In(R)$: Boolean initial condition
- $Tr(I, R, R')$: Boolean transition relation

Invariance Verification Problem:

- *Model Checking*: Given hardware H and Boolean invariance condition $Inv(R)$, show that invariance holds in all reachable states of H – PSPACE-complete.
- *Bounded Model Checking*: Show that invariance holds all states reachable in k steps – NP-complete

Reduction of BMC to SAT: Create $k+1$ copies of variables: $I_0, \ldots, I_k, R_0, \ldots, R_k$.

$$In(R_0) \wedge Tr(I_0, R_0, R_1) \wedge \ldots \wedge Tr(I_{k-1}, R_{k-1}, R_k) \wedge \neg Inv(R_k)$$

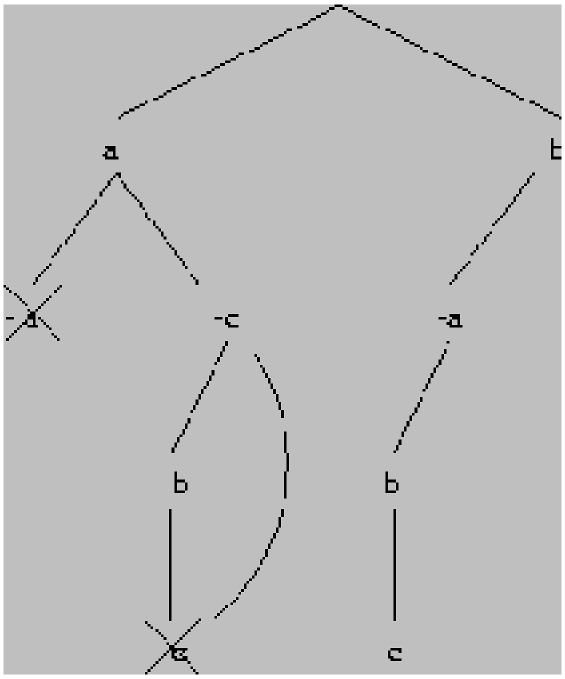
Note: For non-Boolean systems, CSP rather than SAT is needed.

Important Search Technique: Backjumping

Backjumping vs. Backtracking:

- *Backtracking*: go up one level in the search tree.
- *Backjumping*: gp up many levels.

– If an assignment to x_1, x_2, x_3, x_4 cannot be extended to x_5, then changing the value of x_4 may not help!



Safe Jump: If an assignment to $x_1, \ldots, x_k$ cannot be extended to x_{k+1}, then x_j is a *safe jump* if the assignment to $x_1, \ldots, x_j$ also cannot be extended to x_{k+1}

But: Finding highest safe jumps is hard! Heuristics are needed.

Example: Leaf Deadend Backjumping

Suppose the partial assignment $x_1 \mapsto a_1, \ldots, x_k \mapsto a_k$ cannot be extended to the leaf variable x_{k+1}, that is, all assignments for x_{k+1} fail.

- For each possible value a of x_{k+1}, find largest j such that $x_1 \mapsto a_1, \ldots, x_j \mapsto a_j, x_{k+1} \mapsto a$ is consistent.
  - Check
    $x_1 \mapsto a_1, x_{k+1} \mapsto a$,
    $x_1 \mapsto a_1, x_2 \mapsto a_2, x_{k+1} \mapsto a$,
    etc.
- Maximal index (plus 1) is a safe jump.

Constraint Propagation

Basic Idea: Perform easy computations to eliminate local contradictions.

- *Resolution*: From $p \vee \alpha$ and $\neg p \vee \beta$, infer $\alpha \vee \beta$

Practical Constraint Propagation: polynomial global cost

Constraint Networks: Unary and binary constraints

- *Arc Consistency*: For each pair x, y of variables, for each $a \in R(x)$, there is $b \in R(y)$ such that $(a, b) \in R(x, y)$.

– If arc consistency is violated for $a \in R(x)$, delete a from $R(x)$ and delete pairs $(a, .)$ from $R(x, y)$ (efficient algorithms: *AC-1*, *AC-1*. etc.)

Path Consistency: For each triple x, y, z of variables, for each $(a, b) \in R(x, y)$, there is $c \in R(z)$ such that $(a, c) \in R(x, z)$ and $(b, c) \in R(y, z)$

– If path consistency is violated for $(a, b) \in R(x, y)$, delete (a, b) from $R(x, y)$ (efficient algorithms: *PC-1*, *PC-2*, etc.)

Homomorphisms

Homomorphism: Let $\mathbf{A} = (A, R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$ and $\mathbf{B} = (B, R_1^{\mathbf{B}}, \ldots, R_m^{\mathbf{B}})$ be relational structures.

$h : A \to B$ is a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if for every $i \leq m$ and every tuple $(a_1, \ldots, a_n) \in A^n$,

$$R_i^{\mathbf{A}}(a_1, \ldots, a_n) \implies R^{\mathbf{B}}(h(a_1), \ldots, h(a_n)).$$

The Homomorphism Problem: Given relational structures $\mathbf{A}$ and $\mathbf{B}$, is there a homomorphism $h : \mathbf{A} \to \mathbf{B}$?

Example: An undirected graph $\mathbf{A} = (V, E)$ is 3-colorable

$$\iff$$

there is a homomorphism $h : \mathbf{A} \to K_3$, where K_3 is the *3-clique*.

Homomorphism Problems

Examples:

- k-Clique: $K_k \xrightarrow{h} (V, E)$?
- Hamiltonian Cycle: $(V, C_{|V|}, \neq) \xrightarrow{h} (V, E, \neq)$?
- Subgraph Isomorphism: $(V, E, \overline{E}) \xrightarrow{h} (V', E', \overline{E'})$?
- s-t Connectivity: $(V, E, \{\langle s, t\rangle\}) \overset{h}{\not\to} (\{0, 1\}, =, \neq)$?

Fact: (Levin, 1973)

The homomorphism problem is NP-complete.

CSP vs. Homomorphisms

From CSP to Homomorphism:

Given: $(V, D, \{C_1, \ldots, C_m\})$, where $C_i = (t_i, R_i)$.

Define $\mathbf{A}, \mathbf{B}$:

- $\mathbf{A} = (V, \{t_1\}, \ldots, \{t_m\})$
- $\mathbf{B} = (D, R_1, \ldots, R_m)$

Fact: (V, D, C) has a solution iff there is homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Homomorphisms vs. CSP

From Homomorphism to CSP:

Given: $\mathbf{A} = (A, R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$, $\mathbf{B} = (B, R_1^{\mathbf{B}}, \ldots, R_m^{\mathbf{B}})$.

Define (V, D, C):

- $V = A$: elements of $\mathbf{A}$ are variables.
- $D = B$: elements of $\mathbf{B}$ are values.
- $C = \{(t, R_i^{\mathbf{B}}) \ : \ t \in R_i^{\mathbf{A}}\}$: constraints derived from $\mathbf{A}, \mathbf{B}$.

Fact: There is homomorphism from $\mathbf{A}$ to $\mathbf{B}$ iff (V, D, C) has a solution.

Conclusion: CSP=Homomorphism Problem

- Feder&V., 1993
- Garey&Johnson, 1979: Homomorphism in, CSP not.

Introduction to Database Theory

Basic Concepts:

- *Relation Scheme*: a set of attributes
- *Tuple*: mapping from relation scheme to data values
- *Tuple Projection*: if t is a tuple on P, and $Q \subseteq P$, then $t[Q]$ is the restriction of t to Q.
- *Relation*: a set of tuples over a relation scheme
- *Relational Projection*: if R is a relation on P, and $Q \subseteq P$, then $R[Q]$ is the relation $\{t[Q] : t \in R\}$.
- *Join*: Let R_i be a relation over relation scheme S_i. Then $\bowtie_i R_i$ is a relation over the relation scheme $\cup_i S_i$ defined by $\bowtie_i R_i = \{t : t[S_i] \in R_i\}$.

Database Perspective of CSP

Given: $(V, D, \{C_1, \ldots, C_m\})$, where $C_i = (t_i, R_i)$.

Assume (wlog): Each t_i consists of distinct elements.

Database Perspective:

- V: attributes
- D: values
- (t_i, R_i): relation R_i over relation scheme t_i

Fact: (Bibel, Gyssens, Jeavons, Cohen)

$(V, D, \{C_1, \ldots, C_m\})$ has a solution iff $\bowtie_1^m R_i$ is nonempty.

Constraint Propagation

A database perspective on constraint propagation:

- *Projection Rule*: From (t, R) infer $(t, \pi_{t'}(R))$, where $t' \subseteq t$.

- *Join Rule*: From (t_1, R_1) and (t_1, R_2), infer $(t_1 \cup t_2, R_1 \bowtie R_2)$

Eaxmple: *Resolution* – From $\alpha = \theta \vee p$ and $\beta = \psi \vee \neg p$ infer $\theta \vee \psi$

- Sound and complete for refutation of clausal formulas

Soundness Proof:

- Consider $models(\varphi)$ as a relation (over a Boolean domain) with relation scheme $vars(\varphi)$.

- $(models(\alpha) \bowtie models(\beta))\,[vars(\theta) \cup vars(\psi)] = models(\theta \vee \psi)$

Arc and Path Consistency

A database perspective on constraint propagation:

Semijoin: Let R_i be a relation over relation scheme S_i. Then $R_1 \rhd\!< R_2$ is a relation over the relation scheme S_1 defined by $R_1 \rhd\!< R_2 = (R_1 \bowtie R_2)[S_1] = \{t[R_1] : t[S_1] \in R_1$ and $t[S_2] \in R_2\}$.

- *Arc Consistency*:

$$R(x) \leftarrow R(x) \rhd\!< R(x,y)$$

$$R(x,y) \leftarrow R(x,y) \rhd\!< R(x)$$

- *Path Consistency*:

$$R(x,y) \leftarrow R(x,y) \rhd\!< (R(x,z) \bowtie R(y,z))$$

Conjunctive Queries

Conjunctive Query: First-order logic without $\forall, \vee, \neg$; written as a rule

$$Q(X_1, \ldots, X_n) \ :- \ R(X_3, Y_2, X_4), \ldots, S(X_2, Y_3)$$

Significance: most common SQL queries (*Select-Project-Join*)

Example:

$$GrandParent(X, Y) \ :- \ Parent(X, Z), Parent(Z, Y)$$

Conjunctive Query Containment (CQC)

Definition: $Q_1 \sqsubseteq Q_2$ iff $Q_1(B) \subseteq Q_2(B)$ for every database B.

Fundamental problem in database theory:

- Optimization of SPJ Queries
- Information Integration
- Data Warehousing
- ...

Fact: $Q_1 \sqsubseteq Q_2$ holds iff Q_1 logically implies Q_2.

- *Decidability*: Bernays-Schönfinkel Class ($\exists^*\forall^*$)

CQC vs. Homomorphism Problem

Canonical Database B^Q:

- Each variable in Q is a distinct element
- Each subgoal $R(X_3, Y_2, X_4)$ of Q gives rise to a tuple $R(X_3, Y_2, X_4)$ in B^Q
- Each *distinguished variable* X_i gives rise to a tuple $P_i(X_i)$ in B^Q

Fact: (Chandra&Merlin, 1977)

For conjunctive queries Q_1 and Q_2, TFAE:

- $Q_1 \sqsubseteq Q_2$
- There is a homomorphism $h : B^{Q_2} \to B^{Q_1}$.
- $Q_2(B^{Q_1})$ is nonempty

Homomorphism Problem vs. CQC

Given: $\mathbf{A} = (A, R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$, $\mathbf{B} = (B, R_1^{\mathbf{B}}, \ldots, R_m^{\mathbf{B}})$.

Define conjunctive query $Q_{\mathbf{A}}$ over variables in A:

$$Q_{\mathbf{A}} \ :- \ \bigwedge_{i=1}^{m} \bigwedge_{t \in R_i^{\mathbf{A}}} R_i(t)$$

Define conjunctive query $Q_{\mathbf{B}}$ over variables in B:

$$Q_{\mathbf{B}} \ :- \ \bigwedge_{i=1^m} \bigwedge_{t \in R_i^{\mathbf{B}}} R_i(t)$$

Fact: For relational structures $\mathbf{A}$ and $\mathbf{B}$, TFAE:

- There is a homomorphism $h : \mathbf{A} \to \mathbf{B}$.
- $Q_{\mathbf{B}} \sqsubseteq Q_{\mathbf{A}}$
- $Q_{\mathbf{A}}(\mathbf{B})$ is nonempty

“Same Difference”

Conclusion: (Kolaitis&V., 1998)

CSP, Homomorphism, CQ containment, and CQ evaluation all amount to the same problem!

Potential Promise: Synergy between AI and DB.

Uniform CSP vs. Non-Uniform CSP

Uniform CSP:

$$\{(\mathbf{A}, \mathbf{B}) : \exists \text{ homomorphism } h : \mathbf{A} \rightarrow \mathbf{B}\}$$

Complexity of Uniform CSP: NP-complete

Non-uniform CSP: Fix a structure $\mathbf{B}$

$$\mathrm{CSP}(\mathbf{B}) = \{\mathbf{A} : \exists \text{ homomorphism } h : \mathbf{A} \rightarrow \mathbf{B}\}$$

Complexity of Non-Uniform CSP: Depends on $\mathbf{B}$

- $\mathrm{CSP}(K_2)$ is in PTIME (2-Colorability)
- $\mathrm{CSP}(K_3)$ is NP-complete (3-Colorability)

Complexity of Query Evaluation

Measuring Complexity:

- *Combined Complexity:*

$$\{\langle Q, \mathbf{B}\rangle : Q(\mathbf{B}) \text{ is nonempty}\}$$

- *Expression Complexity:* Fix $\mathbf{B}$

$$\{Q : Q(\mathbf{B}) \text{ is nonempty}\}$$

- *Data Complexity:* Fix Q

$$\{\mathbf{B} : Q(\mathbf{B}) \text{ is nonempty}\}$$

Known: Typically, exponential gap between data complexity and expression/combined complexity (V., 1982).

Database Perspective of CSP

Uniform CSP:

$$\{(\mathbf{A}, \mathbf{B}) : Q_{\mathbf{A}}(\mathbf{B}) \text{ is nonempty}\}$$

Complexity of Uniform CSP: combined complexity

Non-uniform CSP: Fix a structure $\mathbf{B}$

$$\mathrm{CSP}(\mathbf{B}) = \{\mathbf{A} : Q_{\mathbf{A}}(\mathbf{B}) \text{ is nonempty}\}$$

Complexity of Non-Uniform CSP: expression complexity

Complexity of CQC

In Conjunctive Query Containment ($Q_1 \sqsubseteq Q_2$?),
we are typically interested in *uniform* results.

Research Program:

Identity the tractable cases of uniform CQC.

Results: Tractable cases of $Q_1 \sqsubseteq Q_2$

- Saraiya, 1991:

  Every predicate occurs at most *twice* in Q_1

- Chekuri–Rajaraman, 1997:

  Q_2 has *querywidth* at most k, where k is a fixed positive integer.

- Gottlob–Leone–Scarcello, 1999:

  Q_2 has *hypertreewidth* at most k, where k is a fixed positive integer.

Complexity of Non-Uniform CSP

Research Program:

Identity the tractable cases of non-uniform CSP (Schaefer, 1978, ...,Jeavons et al, 1990s)

Dichotomy Conjecture: (Feder–V., 1993)

For every structure $\mathbf{B}$,

- either $\mathrm{CSP}(\mathbf{B})$ is in PTIME
- or $\mathrm{CSP}(\mathbf{B})$ is NP-complete.

Recall: $P \neq NP \Rightarrow NP - NPC - P \neq \emptyset$ (Ladner, 1975)

Dichotomy Conjecture: $CSP - NPC - P = \emptyset$

Evidence for Dichotomy Conjecture

"Evidence 1": (Hell–Nešetril, 1990)

Let $\mathbf{B}$ be an *undirected* graph.

- $\mathbf{B}$ bipartite $\Longrightarrow$ $\mathrm{CSP}(\mathbf{B})$ is in PTIME
- $\mathbf{B}$ non-bipartite $\Longrightarrow$ $\mathrm{CSP}(\mathbf{B})$ is NP-complete

"Evidence 2": (Schaefer, 1978)

Let $\mathbf{B}$ have a *Boolean* domain, then

- either $\mathrm{CSP}(\mathbf{B})$ is in PTIME
- or $\mathrm{CSP}(\mathbf{B})$ is NP-complete.

"Evidence 3": (Bulatov, 2002)

Let $\mathbf{B}$ have a *3-element* domain, then

- either $\mathrm{CSP}(\mathbf{B})$ is in PTIME
- or $\mathrm{CSP}(\mathbf{B})$ is NP-complete.

The Dichotomy Questions

Question 1: Does the Dichotomy Conjecture hold?

Question 2: The *Classification Broblem*

For a given structure $\mathbf{B}$,

- Decide if $\mathrm{CSP}(\mathbf{B})$ is in PTIME.
- Decide if $\mathrm{CSP}(\mathbf{B})$ is NP-complete.

Question 3: Explaining the dichotomy (if true)

Classification Conjecture: (Feder–V., 1993)

Two explanations for tractability of $\mathrm{CSP}(\mathbf{B})$

- *combinatorial* (Datalog) (V.)
- *algebraic* (group-theoretic) (Krokhin)