Enumerating All Solutions for Constraint Satisfaction Problems

Henning Schnoor

Ilka Schnoor

Institut für Theoretische Informatik

Leibniz Universität Hannover

Aachen, 24.2.2007

Generalization of many well-known problems

- 3SAT
- Horn-SAT
- Graph colorability
- . . . all kinds of combinatorial problems

A few satisfiability problems

- General SAT-problem: NP-complete.
- Restriction to 3SAT: NP-complete.
  $(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_4 \vee \overline{x_5})$
- Restriction to 2SAT: complete for NL.
  $(x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_1)$
- Restriction to Horn-formulas: complete for P.
  $((x_1 \wedge x_2 \wedge x_3) \rightarrow x_4) \wedge ((x_2 \wedge x_3, \wedge x_5) \rightarrow x_1)$

Constraint Satisfaction Problems: Definitions

- constraint language Γ: set of relations over domain D
- $\Gamma_{3SAT} := \{(x \vee y \vee z), (\overline{x} \vee y \vee z), (\overline{x} \vee \overline{y} \vee z), (\overline{x} \vee \overline{y} \vee \overline{z})\}$
- Γ-formula: $R_1(x_1, \ldots, x_k) \wedge \cdots \wedge R_n(y_1, \ldots, y_l)$ for $R_i \in \Gamma$

Constraint Satisfaction Problems: Definitions

- constraint language Γ: set of relations over domain D
- $\Gamma_{3SAT} := \{(x \vee y \vee z), (\overline{x} \vee y \vee z), (\overline{x} \vee \overline{y} \vee z), (\overline{x} \vee \overline{y} \vee \overline{z})\}$
- Γ-formula: $R_1(x_1, \ldots, x_k) \wedge \cdots \wedge R_n(y_1, \ldots, y_l)$ for $R_i \in \Gamma$
- solution: assignment I to the variables with $(I(x_1), \ldots, I(x_k)) \in R_1, \ldots, (I(y_1), \ldots, I(y_l)) \in R_n$
- constraint satisfaction problem $\mathrm{CSP}(\Gamma)$: Does a given Γ-formula have a solution?

Example: CSP can express colorability

- $D := \{0, 1, 2\}$, $R := \{(x, y) \mid x, y \in D, x \neq y\}$.

Example: CSP can express colorability

- $D := \{0, 1, 2\}$, $R := \{(x, y) \mid x, y \in D, x \neq y\}$.
- G a graph with edges $(u_1, v_1), \ldots, (u_n, v_n)$
- Define $\varphi := R(u_1, v_1) \wedge \cdots \wedge R(u_n, v_n)$.

Constraint Satisfaction Problems: Example

Example: CSP can express colorability

- $D := \{0, 1, 2\}\,,\ R := \{(x, y) \mid x, y \in D, x \neq y\}\,.$
- G a graph with edges $(u_1, v_1), \ldots, (u_n, v_n)$
- Define $\varphi := R(u_1, v_1) \wedge \cdots \wedge R(u_n, v_n)$.
- φ is satisfiable if and only if G is 3-colorable.

Corollary

In general, $\mathrm{CSP}(\Gamma)$ *is* NP*-complete.*

- Complexity classifications for 2- and 3-element domains

  Schaefer 1978, Bulatov 2003
- CSP (Γ) is either in P, or NP-complete for all known cases

We are interested in the complexity of enumerating all solutions for a given constraint formula.

Research Goal

Question

For Γ, decide if efficient enumeration for Γ-formulas is possible.

Efficient enumeration algorithm

On input φ, print all solutions of φ such that

- Time between two solutions is bounded by a polynomial
- Each solution $I \models \varphi$ is printed exactly once

An algorithm for the Boolean case

- Let φ be a formula with variables $x_1, \ldots, x_n$.

A Simple Algorithm

An algorithm for the Boolean case

- Let φ be a formula with variables $x_1, \ldots, x_n$.
- If $\varphi \wedge \overline{x_1} \in \mathrm{SAT}$: enumerate all solutions of $\varphi \wedge \overline{x_1}$
- If $\varphi \wedge x_1 \in \mathrm{SAT}$: enumerate all solutions of $\varphi \wedge x_1$

A Simple Algorithm

An algorithm for the Boolean case

- Let φ be a formula with variables $x_1, \ldots, x_n$.
- If $\varphi \wedge \overline{x_1} \in \mathrm{SAT}$: enumerate all solutions of $\varphi \wedge \overline{x_1}$
- If $\varphi \wedge x_1 \in \mathrm{SAT}$: enumerate all solutions of $\varphi \wedge x_1$

Theorem (Creignou, Hébrard, 1997)

- *In the Boolean case, there is no other algorithm.*

A Simple Algorithm

- Algorithm needs decision for "Γ-formulas with literals."
- Works if CSP for Γ plus $\underbrace{(x = 0)}_{\overline{x}}$ and $\underbrace{(x = 1)}_{x}$ is tractable.

- Algorithm needs decision for "Γ-formulas with literals."
- Works if CSP for Γ plus $\underbrace{(x = 0)}_{\overline{x}}$ and $\underbrace{(x = 1)}_{x}$ is tractable.
- Generalization to non-Boolean case:
  - Γ^+ is Γ plus clauses $(x = \alpha)$ for $\alpha \in D$.
  - Γ^+ is "Γ with literals."

A Simple Algorithm

- Algorithm needs decision for "Γ-formulas with literals."
- Works if CSP for Γ plus $\underbrace{(x = 0)}_{\overline{x}}$ and $\underbrace{(x = 1)}_{x}$ is tractable.
- Generalization to non-Boolean case:
  - Γ^+ is Γ plus clauses $(x = \alpha)$ for $\alpha \in D$.
  - Γ^+ is "Γ with literals."

CSP (Γ^+) is SAT for formulas $\varphi \wedge (x_1 = 4) \wedge (x_2 = 1) \ldots$

Satisfiability problem with literals.

Non-Boolean Generalization

Theorem (Cohen, 2004)

If CSP $(\Gamma^+) \in \mathrm{P}$*, then* Γ *has an efficient enumeration algorithm.*

Proof

Generalization of previous algorithm.

Non-Boolean Generalization

Theorem (Cohen, 2004)

If CSP $(\Gamma^+) \in \mathrm{P}$, *then* Γ *has an efficient enumeration algorithm.*

Proof

Generalization of previous algorithm.

Converse?

- For the Boolean case, the converse holds as well.

  Creignou, Hébrard, 1997

- What about arbitrary domains?

- A weaker version of the converse does hold.

Theorem

Equivalent:

1. $\mathsf{CSP}\,(\Gamma^+) \in \mathrm{P}$.
2. *Solutions for* Γ*-formuls can be enumerated in lexicographical ordering, with different order for each variable.*

- For x_1, we demand $0 < 2 < 4 < \dots$
- For x_2, we demand $4 < 3 < 1 < \dots$

Proposition

- *If* CSP $(\Gamma^{+}) \in \mathrm{P}$*, there is an efficient enumeration algorithm.*
- *If* CSP $(\Gamma) \notin \mathrm{P},$ *there is no efficient enumeration algorithm.*

Proposition

- *If* CSP$(\Gamma^+) \in \mathrm{P}$*, there is an efficient enumeration algorithm.*
- *If* CSP$(\Gamma) \notin \mathrm{P}$*, there is no efficient enumeration algorithm.*

We are interested in the remaining cases:

CSP$(\Gamma) \in \mathrm{P}$, and CSP$(\Gamma^+) \notin \mathrm{P}$.

- In this case, we can add *some*, but not *all* literals to Γ and remain tractable.

Typical Enumeration Strategy

Situation

- $D = \{0, 1, 2\}$
- CSP for Γ plus $(x = 0), (x = 1)$ tractable,
- CSP for Γ plus $(x = 0), (x = 1), (x = 2)$ not tractable.

Typical Enumeration Strategy

Situation

- $D = \{0, 1, 2\}$
- CSP for Γ plus $(x = 0), (x = 1)$ tractable,
- CSP for Γ plus $(x = 0), (x = 1), (x = 2)$ not tractable.
- For every solution, exchanging 2 with a 1 again gives solution.
- If $I = (2, 1, 1, 0, 0, 1, 2, 1)$ is solution, so is
  $I' = (1, 1, 1, 0, 0, 1, 1, 1)$.

This is a typical situation.

Approach

- CSP for Γ plus "literal clauses" $(x = 0)$ and $(x = 1)$ is tractable

- CSP for Γ plus “literal clauses” $(x = 0)$ and $(x = 1)$ is tractable

Enumeration Approach

- Previous algorithm: get solutions in which only 0 and 1 appear

- CSP for Γ plus "literal clauses" $(x = 0)$ and $(x = 1)$ is tractable

Enumeration Approach

- Previous algorithm: get solutions in which only 0 and 1 appear

$$(1, 0, 2, 2, 0)$$

$$(2, 0, 1, 1, 0)$$

$$(1, 0, 1, 2, 0)$$

- CSP for Γ plus "literal clauses" $(x = 0)$ and $(x = 1)$ is tractable

Enumeration Approach

- Previous algorithm: get solutions in which only 0 and 1 appear

$$(1, 0, 2, 2, 0) \rightarrow (1, 0, 1, 1, 0)$$
$$(2, 0, 1, 1, 0) \rightarrow (1, 0, 1, 1, 0)$$
$$(1, 0, 1, 2, 0) \rightarrow (1, 0, 1, 1, 0)$$

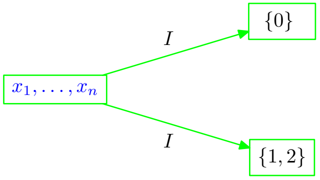
- Others obtained by changing *some* of the 1s into 2s.

Typical Enumeration Strategy

- The 1 is a "placeholder" for either 1 or 2.

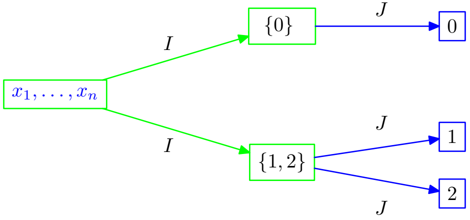
Typical Enumeration Strategy

- The 1 is a “placeholder” for either 1 or 2.

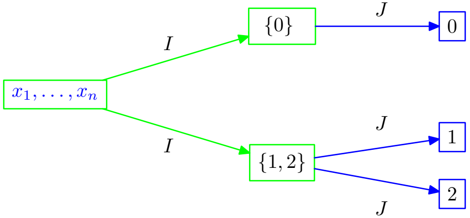


Typical Enumeration Strategy

- The 1 is a "placeholder" for either 1 or 2.

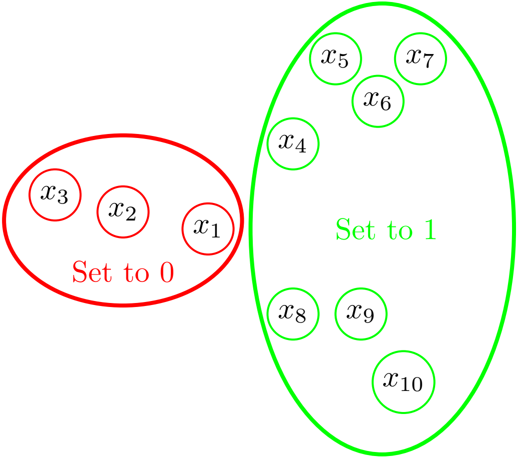


- The 1 is a "placeholder" for either 1 or 2.



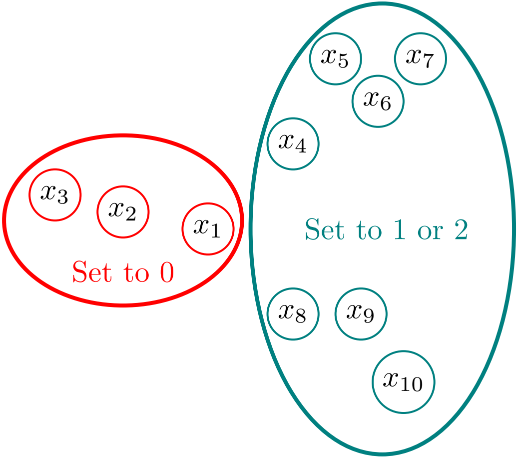
Real solutions are refinements of "partial solutions."

Looking for the right combinations



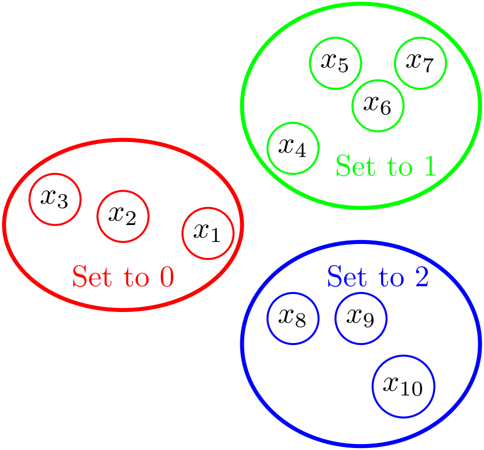
- 0/1-solution

Looking for the right combinations



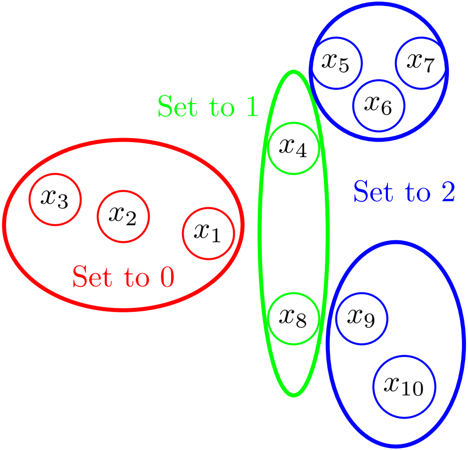
- 0/1-solution
- 1 can be 1 or 2

Looking for the right combinations



- 0/1-solution
- 1 can be 1 or 2
- Which combinations are solutions?

Looking for the right combinations



- 0/1-solution
- 1 can be 1 or 2
- Which combinations are solutions?

From Partial to Real Solutions I

Problem

Given a $0, 1$-solution, generate all “fitting” refinements.

Approach

- Looking for possibilities to change 1 into 2 and get a solution.
- This is a problem involving only the values 1 and 2

From Partial to Real Solutions I

Problem

Given a $0, 1$-solution, generate all "fitting" refinements.

Approach

- Looking for possibilities to change 1 into 2 and get a solution.
- This is a problem involving only the values 1 and 2
- A Boolean Constraint Satisfaction Problem.

From Partial to Real Solutions II

Approach

- Split up the problem into two Boolean problems:
  - Enumerate the 0-1-solutions
  - For each of these, enumerate the “fitting” solutions

Using knowledge about the Boolean case, we can prove exactly when this works.

Generalizations

Generalizations of approach

- Larger domains than $\{0, 1, 2\}$
- Equivalence classes of solutions

There are restrictions on the possible orders.

Theorem

For constraint languages with certain algebraic properties on the three-element domain, our algorithms cover all cases.

Proof Idea

- Usual implementation techniques for CSP do not work
- Main idea: Reduction from Boolean cases with stricter implementation

Conclusion & Future Research

Summary

- New enumeration algorithms
- Cases when CSP with literals NP-complete, but efficient enumeration possible
- Restrictions on possible enumeration orders
- Known algebraic proof techniques are only of limited use

Future Research

- Achieve full classification
- Apply new algebraic methods

Thank You!