Constraints, Graphs, Algebra, Logic, and Complexity

Moshe Y. Vardi

Rice University

Constraint Satisfaction Problem (CSP)

Input: (V, D, C):

- A finite set V of *variables*
- A finite 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 over D of arity $|t|$

Solution: $h : V \to D$

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

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

3-Colorability

$3\text{-}\mathrm{COLOR}$: 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 $\{(\langle u, v\rangle, \rho) : (u, v) \in E\}$, where $\rho = \{(R, G), (R, B), (G, R), (G, B), (B, R), (B, G)\}$.

Constraint Satisfaction

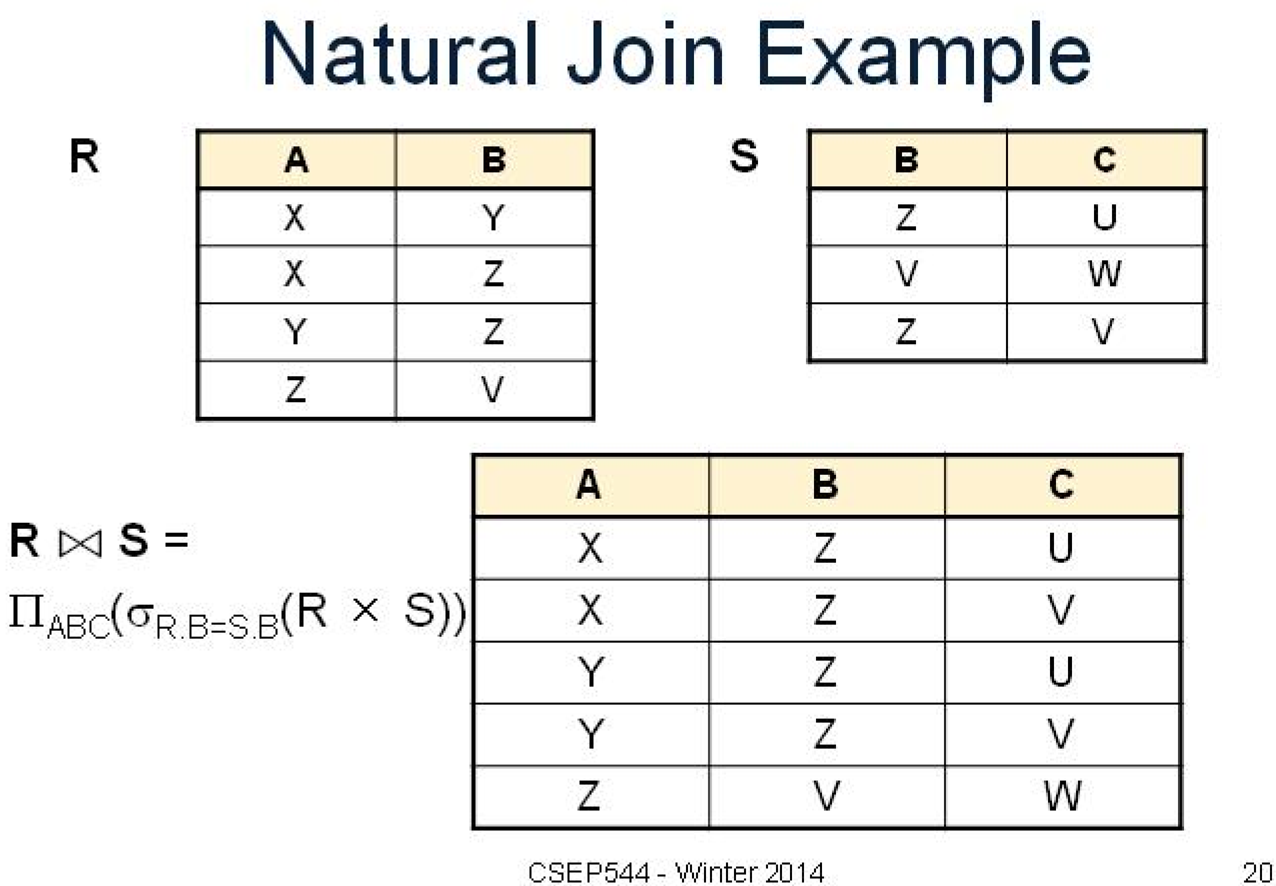
Applications:

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

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 $\bowtie_i R_i = \{t : t[S_i] \in R_i\}$.



Natural Join Example

R

A	B
X	Y
X	Z
Y	Z
Z	V

S

B	C
Z	U
V	W
Z	V

R $\bowtie$ S =

$\Pi_{ABC}(\sigma_{R.B=S.B}(R \times S))$

A	B	C
X	Z	U
X	Z	V
Y	Z	U
Y	Z	V
Z	V	W

CSEP544 - Winter 2014 20

Figure 1: Relations

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.

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 two relational structures.

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

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

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 $\mathbf{B}$.

CSP vs. Homomorphisms

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.

Uniform CSP vs. Non-Uniform CSP

Uniform CSP: $\{(\mathbf{A}, \mathbf{B}) : \exists$ homomorphism $h : \mathbf{A} \to \mathbf{B}\}$

Complexity of Uniform CSP: NP-complete

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

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

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 Non-Uniform CSP

Research Program: Identity the tractable cases of non-uniform CSP

- **Goal**: Understand complexity of a large class of problems in NP.

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)

Intuition: CSP is not expressive enough to diagonalize over PTIME.

“Evidence” for the 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

Intuition: Every undirected graph homomrphism problem is equivalent either to 2-COLOR or 3-COLOR.

More “Evidence”: Boolean CSP

$B = \{0, 1\}$

E.g.: 2-SAT

$\mathbf{B}$:

$x \vee y$:

0	1
1	0
1	1

$\neg x \vee y$:

0	0
0	1
1	1

$\neg x \vee \neg y$:

0	0
0	1
1	0

Dichotomy Theorem: (Schaefer, 1978)

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

- either $\mathbf{B}$ is trivial, Horn, anti-Horn, disjunctive, or affine, and $\mathrm{CSP}(\mathbf{B})$ is in PTIME,
- otherwise $\mathrm{CSP}(\mathbf{B})$ is NP-complete.

Dichotomy and Classification

Question: How far from CSP we need go to get a provable dichotomy?

Feder&V., 1993: It suffices to consider directed graphs to settle the Dichotomy Conjecture!

Classification Question:

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

- when is $\mathrm{CSP}(\mathbf{B})$ in PTIME?
- when is $\mathrm{CSP}(\mathbf{B})$ NP-complete?

Sources of Tractability

Empirical Observation: Feder&V., 1993

All known tractable CS problems can be explained as

- *combinatorial* (Datalog)
- *algebraic* (group-theoretic)

Classification Conjecture: (Feder&V., 1993)

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

- Datalog
- Group-Theoretic

Bulatov, 2002 showed that the group-theoretic explanation is too weak – more general algebraic techniques required.

Datalog and Non-Uniform CSP

Example: NON 2-COLORABILITY

$$
\begin{aligned}
O(X,Y) &\ :- \quad E(X,Y) \\
O(X,Y) &\ :- \quad O(X,Z), E(Z,W), E(W,Y) \\
Q &\ :- \quad O(X,X)
\end{aligned}
$$

Recall: Datalog $\subseteq$ PTIME

Define: $\overline{\mathrm{CSP}(\mathbf{B})} = \{\mathbf{A} : \mathbf{A} \notin \mathrm{CSP}(\mathbf{B})\}$.

Datalog vs. Non-Uniform CSP: Explanation for many tractability results – $\overline{\mathrm{CSP}(\mathbf{B})}$ is expressible in Datalog

Note: Datalog and $\overline{\mathrm{CSP}(\mathbf{B})}$ are monotone.

k-Datalog

‘ **Definition:**

- k-Datalog: Datalog with at most k variables per rule (NON 2-COLORABILITY is in 4-Datalog)
- $\exists\mathrm{IL}^k$: k-variable existential positive infinitary logic
  - variables: $x_1, \ldots, x_k$
  - no universal quantifiers
  - no negations
  - infinitary conjunctions and disjunctions

Facts: Fix $k \geq 1$

- k-Datalog $\subset$ $\exists\mathrm{IL}^k$
- $\exists\mathrm{IL}^k$ can be characterized in terms of *existential k-pebble games* between the *Spoiler* and the *Duplicator*.

Existential k-Pebble Games

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

- **Spoiler**: *places* on or *removes* a pebble from an element of $\mathbf{A}$.
- **Duplicator**: tries to duplicate move on $\mathbf{B}$.

$\mathbf{A}$: $a_1, a_2, \ldots, a_l$ $\mathbf{B}$: $b_1, b_2, \ldots, b_l$

- *Spoiler wins*: $h(a_i) = b_i$, $1 \leq i \leq l$ is not a homomorphism.
- *Duplicator wins*: otherwise.

Facts: (Kolaitis&V., 1995):

- $\mathbf{B}$ satisfies the same $\exists \mathrm{IL}^k$ sentences as $\mathbf{A}$ iff the Duplicator wins the existential k-pebble game on $\mathbf{A}, \mathbf{B}$.
- There is a PTIME algorithm to decide whether the *Spoiler* or the *Duplicator* wins the existential k-pebble game.

k-Datalog and CSP

Theorem: (Kolaitis&V., 1998): TFAE for $k \geq 1$ and a structure $\mathbf{B}$:

- $\overline{\mathrm{CSP}(\mathbf{B})}$ is expressible in k-Datalog
- $\overline{\mathrm{CSP}(\mathbf{B})}$ is expressible in $\exists \mathrm{IL}^k$
- $\mathrm{CSP}(\mathbf{B}) = \{\mathbf{A}$: *Duplicator wins the exist.*$k-$pebble game on$\mathbf{A}$ and $\mathbf{B}\}$.

Intuition: $\overline{\mathrm{CSP}(\mathbf{B})} \in k$-Datalog implies that existence of homomorphism is equivalent to the Duplicator winning the existential k-pebble game.

Consequence (Kolaitis&V., 2000): Solving CSP using Datalog coresponds to standard constraint propagation.

Classification Questions

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

- Is $\overline{\mathrm{CSP}(\mathbf{B})}$ in k-Datalog, for a fixed $k > 0$?
- Is $\overline{\mathrm{CSP}(\mathbf{B})}$ in k-Datalog, for some $k > 0$?

Group Theory

Example: Affine satisfiability - linear equations mod 2

$x_1 - x_2 + x_3 = 1$, $x_1 + x_2 - x_3 = 1$

Definition: $CSP(\mathbf{B}) \in Subgroup$ if there is a finite group G such that each k-ary relation in $\mathbf{B}$ is a *coset* of G^k.

Theorem: Feder&V., 1993

$CSP(\mathbf{B}) \in Subgroup$ implies $CSP(\mathbf{B}) \in PTIME$.

Jeavons et al.: a more general algebraic framework

The Product Operation

Definition: Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. The *product* of these graphs is the graph $G_1 \times G_2 = (V_1 \times V_2, E_1 \times E_2)$, where $(\langle u, u' \rangle, \langle v, v' \rangle) \in E_1 \times E_2$ iff $(u, v) \in E_1$ and $(u', v') \in E_2$.

Note: This definition can be extended to pairs of relational structures.

Polymorphisms

Definition: Let $\mathbf{B} = (B, R_1^{\mathbf{B}}, \ldots, R_m^{\mathbf{B}})$ be a relational structure. A k-ary *polymorphism* is a homomorphism $f : \mathbf{B}^k \to \mathbf{B}$ (*closure condition*).

Feder&V., 93: Study $Poly(\mathbf{B})$ – set of polymorphisms of $\mathbf{B}$

Theorem: [Jeavons, Cohen&Gyssens, 1997]
$Poly(\mathbf{B_1}) = Poly(\mathbf{B_2}) \Rightarrow CSP(\mathbf{B_1}) \equiv_p CSP(\mathbf{B_2})$.

Conclusion: $Poly(\mathbf{B})$ characterizes the complexity of $CSP(\mathbf{B})$.

The Algebraic Approach to CSP: Study $Poly(\mathbf{B})$.

Algebraic Approach

Definition: A *Maltsev operation* is a ternary function f such that $f(a,a,b) = f(b,a,a) = b$ for all a, b in its domain.

- Example: $x \cdot y^{-1} \cdot z$

Theorem [Bulatov, 2002]
If $Poly(\mathbf{B})$ contains a Maltsev operation, then $CSP(\mathbf{B})$ is in PTIME.

Back to Datalog

Definition: A k-ary *near-unanimity operation* is a k-ary function f such that $f(x_1, x_2, \ldots, x_k) = a$ whenever at least $k-1$ of the x_i's equal a.

Example: Majority is a near-unanimity operation.

Theorem: [Feder&V., 1993]
If $Poly(\mathbf{B})$ contains a near-unanimity function, then $\overline{CSP}(\mathbf{B})$ is definable in Datalog.

More on Datalog

Definition: A k-ary *weak near-unanimity operation* is a k-ary function f such that $(a, a, \cdots, a) = a$, and $f(b, a, \cdots, a) = f(a, b, a, \cdots, a) = \cdots = f(a, a, \cdots, b)$, for all a, b in the domain.

Definition: A structure B is a *core* if every homomorphism $h : \mathbf{B} \to \mathbf{B}$ is an isomorphism.

WLOG: *Restrict* attention to cores

Theorem: [Barto&Kozik, 2009]
$\overline{CSP}(\mathbf{B})$ is definable in Datalog iff $Poly(B)$ contains weak near-unanimity operations for all sufficiently large arities. This condition can be checked in exponential time.

The Algebraic Conjecture

Definition: A *cyclic operation* is a k-ary function f such that $f(a_1, a_2 \ldots, a_k) = f(a_2, \ldots, a_k, a_1)$ for all $a_1, \ldots, a_k$ in its domain.

Algebraic Conjecture (early 2010s): [Barto, Bulatov, Jeavons, Kozik, Krokhin]
$CSP(\mathbf{B})$ is in PTIME iff $Poly(\mathbf{B})$ contains a cyclic operation operation of arity at least 2.

Theorem: [Barto, Bulatov, Jeavons, Kozik, Krokhin]
If $Poly(\mathbf{B})$ does not contain a cyclic operation, then $CSP(\mathbf{B})$ is NP-complete.

FOCS 2017: Dichotomy Conjecture Resolved!

- *A Dichotomy Theorem for Nonuniform CSPs*, Andrei A. Bulatov
- *The Proof of CSP Dichotomy Conjecture*, Dmitriy Zhuk

Uniform Tractability

General Problem: CSP$(\mathcal{C}, \mathcal{D})$, where $\mathcal{C}, \mathcal{D}$ are classes of structures

- Is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, where $\mathbf{A} \in \mathcal{C}$ and $\mathbf{B} \in \mathcal{D}$.

Question: When is CSP$(\mathcal{C}, \mathcal{D})$ tractable?

- Non-uniform case: CSP$(All, \mathbf{B})$ for a fixed structure $\mathbf{B}$.

Another imortant case: When is CSP$(\mathcal{C}, All)$ tractable?

Bounded Treewidth

Definition: A *tree decomposition* of a structure $\mathbf{A} = (A, R_1, \ldots, R_m)$ is a labeled tree T such that

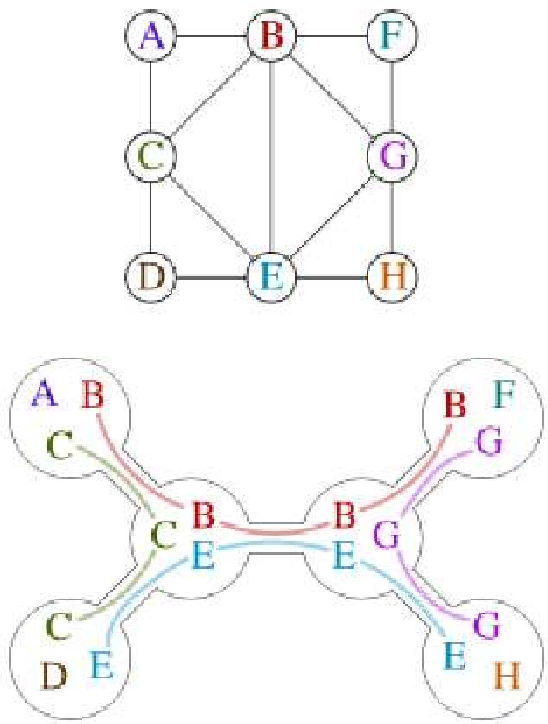
- Each label is a non-empty subset of A;
- For every R_i and every $(a_1, \ldots, a_n) \in R_i$, there is a node whose label contains $\{a_1, \ldots, a_n\}$.
- For every $a \in A$, the nodes whose label contain a form a subtree.

The *treewidth* $\mathrm{tw}(\mathbf{A})$ of $\mathbf{A}$ is defined by

$$\mathrm{tw}(\mathbf{A}) \;=\; \min_T\{\max\{\text{label size in } T\}\} - 1$$

Note: Generalizes the *treewidth* of a *graph*.

Figure 2: Treewidth Decomposition of Width 2



Bounded Treewidth and CSP

$$\mathcal{T}_k = \{\mathbf{A} : \ \mathrm{tw}(\mathbf{A}) \leq k\}$$

Theorem: (Freuder, 1990)

$\mathrm{CSP}(\mathcal{T}_k, \mathrm{All})$ is in PTIME.

Note:

- Complexity is exponential in k.
- Determining treewidth of $\mathbf{B}$ is NP-hard.
- Checking if treewidth is k is in linear time.

Complexity of Query Evaluation

Expression Complexity: Fix $\mathbf{B}$

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

Data Complexity: Fix Q

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

Exponential Gap: (V., 1982)

- Data complexity of FO: *LOGSPACE*
- Expression complexity of FO: *PSPACE-complete*

Mystery: practical query evaluation

Variable-Confined Queries

Definition: FO^k is first-order logic with at most k variables.

In Practice: (V., 1995)

- Queries often can be rewritten to use a small number of variables.
- Variable-confined queries have lower expression complexity.
- E.g.: expression complexity of FO^k is *PTIME-complete*

CSP and Database Queries

Theorem: Chandra&Merlin, 1977

Given $\mathbf{A}$, we can construct in polynomial time an existential, positive, conjunctive first-order query $Q_{\mathbf{A}}$ such that $h : \mathbf{A} \to \mathbf{B}$ iff $Q_{\mathbf{A}}(\mathbf{B})$ is nonempty.

Definition: The *core* of a structure is its (unique) minimal homomorphic substructure. Let $\mathcal{C}_k$ consists of structures with cores of treewidth at most k.

Lemma: Chandra&Merlin, 1977 $Q_{\mathbf{A}}$ is logically equivalent to $Q_{\mathbf{core(A)}}$

Theorem: [Kolaitis&V., 1998]
$\mathbf{core}(\mathbf{A})$ has treewidth k iff $Q_{\mathbf{A}}$ is expressible in existential, positive FO^{k+1}.

Corollary [Dalmau&Kolaitis&V., 2002]
CSP$(\mathcal{C}_k, All)$ is tractable; can be solved using k-Datalog.

Lower Bounds

Theorem: [Grohe, 2005]
Assume $FPT \neq W[1]$. Then CSP$(\mathcal{A}, All)$ is tractable only if $\mathcal{A} \subseteq \mathcal{C}_k$.

Theorem: [Atserias&Bulatov&Dalmau, 2007]
CSP$(\mathcal{A}, All)$ is solvable by k-Datalog only if $\mathcal{A} \subseteq \mathcal{C}_k$.

Research Direction: When is CSP$(\mathcal{C}, \mathcal{D})$ tractable?

In Conclusion

CSP: a paradigmatic problem with connection to

- Graph theory,
- Algebra, and
- Logic,

with several outstanding open questions of theoretical and practical importance.

- Counting CSP, e.g, #SAT
- Fine-grained complexity of CSP, e.g., NLOGSPACE
- $\cdots$