Constraint Satisfaction vs. Dependence Logic

Phokion G. Kolaitis

UC Santa Cruz and IBM Research - Almaden

Joint work with

Lauri Hella

University of Tampere

Constraint Satisfaction and Dependence Logic

- Constraint Satisfaction is a ubiquitous problem in computer science.

  It was introduced by Ugo Montanari more than 40 years ago.

- Dependence Logic is a logical formalism for expressing and analyzing notions of dependence.

  It was developed by Jouko Väänänen about 10 years ago.

Question: What do constraint satisfaction and dependence logic have in common?

Constraint Satisfaction Problems

Input: (V, D, C)

- A finite set V of variables
- A finite domain D of values for the variables
- A set C of constraints (t, R) restricting the values that tuples of variables can take.
  - t: a tuple $t = (x_1, \ldots, x_m)$ of variables
  - R: a relation on D of arity $|t| = m$

Question: Does (V, D, C) have a solution?

Solution:

- An assignment of values to the variables such that all constraints are satisfied.
- Formally, a function $h : V \to D$ such that for every constraint $(t, R) \in C$, we have $h(t) = (h(x_1), \ldots, h(x_m)) \in R$.

Constraint Satisfaction

Fact: Numerous problems in computer science are constraint satisfaction problems.

- Boolean Satisfiability, Graph Colorability, ...
- Database Query Processing
- Planning and Scheduling
- Belief Maintenance
- Machine Vision

...

R. Dechter: "Constraint satisfaction has a unitary theoretical model with myriad practical applications."

Example: Boolean Satisfiability

3-SAT: Given a 3CNF-formula φ with variables $x_1, \ldots, x_n$ and clauses $c_1, \ldots, c_m$, is φ satisfiable?

3-SAT as a constraint satisfaction problem:

- Variables $x_1, \ldots, x_n$
- Domain $D = \{0, 1\}$
- Constraints $((x, y, z), R_i)$, $i = 0, 1, 2, 3$

Clause	Relation
$(x \vee y \vee z)$	$R_0 = \{0,1\}^3 - \{(0,0,0)\}$
$(\neg x \vee y \vee z)$	$R_1 = \{0,1\}^3 - \{(1,0,0)\}$
$(\neg x \vee \neg y \vee z)$	$R_2 = \{0,1\}^3 - \{(1,1,0)\}$
$(\neg x \vee \neg y \vee \neg z)$	$R_3 = \{0,1\}^3 - \{(1,1,1)\}$

Example: Graph Colorability

3-COLORABILITY: Given a graph $\mathbf{G} = (V, E)$, is it 3-colorable?

3-COLORABILITY as a constraint satisfaction problem:

- The variables are the nodes in V
- The domain is the set $D = \{R, G, B\}$ of three colors.
- For each edge $(u, v) \in E$, there is one constraint $((u, v), R)$, where R is the the $\neq$ relation on $\{R, G, B\}$, i.e.,

$$R = \{(R, G), (G, R), (R, B), (B, R), (B, G), (G, B)\}.$$

Algebraic Formulation of Constraint Satisfaction

Feder and Vardi - 1993:

Constraint Satisfaction $\equiv$ Homomorphism Problem.

- A homomorphism between two relational structures **A** and **B** is a function $h : A \to B$ such that for every relation symbol R in the vocabulary and every $(a_1, \ldots, a_n) \in A^n$,

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

- Every finite relational structure **B**, gives rise to a constraint satisfaction problem $\mathrm{CSP}(\mathbf{B})$: Given a finite relational structure **A**, is there a homomorphism $h : \mathbf{A} \to \mathbf{B}$?
- Conversely, every constraint satisfaction problem can be identified with a $\mathrm{CSP}(\mathbf{B})$, for some suitable **B**.

Constraint Satisfaction and the Homomorphism Problem

- 3-COLORABILITY $= \mathrm{CSP}(\mathbf{K}_3)$, there K_3 is the clique with 3 elements.
- k-COLORABILITY $= \mathrm{CSP}(\mathbf{K}_k)$, there K_k is the clique with k elements, $k \geq 2$.

Constraint Satisfaction and the Homomorphism Problem

- 3-COLORABILITY = CSP($\mathbf{K}_3$), there K_3 is the clique with 3 elements.
- k-COLORABILITY = CSP($\mathbf{K}_k$), there K_k is the clique with k elements, $k \geq 2$.
- POSITIVE NAE 3-SAT: Given a 3-CNF formula with only positive literals, is there a satisfying truth assignment such that in each clause not every variable is assigned value 1?

  POSITIVE NAE 3-SAT = CSP($\mathbf{B}$), where

  – $\mathbf{B} = (\{0,1\}, R^{\mathbf{B}})$ with $R^{\mathbf{B}} = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$;

  – each 3-CNF formula φ with only positive literals is encoded as $\mathbf{A}(\varphi)$, where $R^{\mathbf{A}(\varphi)} = \{(x,y,z) : x \vee y \vee z \text{ is a clause in } \varphi\}$.

Computational Complexity of Constraint Satisfaction

Fact:

- $\mathrm{CSP}(\mathbf{B})$ is in NP, for every $\mathbf{B}$.
- $\mathrm{CSP}(\mathbf{K}_2)$ (i.e., 2-COLORABILITY) is in PTIME.
- $\mathrm{CSP}(\mathbf{K}_k)$ (i.e., k-COLORABILITY) is NP-complete, for every $k \geq 3$.

Computational Complexity of Constraint Satisfaction

Fact:

- $\mathrm{CSP}(\mathbf{B})$ is in NP, for every $\mathbf{B}$.
- $\mathrm{CSP}(\mathbf{K}_2)$ (i.e., 2-$\textsc{Colorability}$) is in PTIME.
- $\mathrm{CSP}(\mathbf{K}_k)$ (i.e., k-$\textsc{Colorability}$) is NP-complete, for every $k \geq 3$.

Feder-Vardi Dichotomy Conjecture - 1993

For every $\mathbf{B}$, one of the following two holds:

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

The Fine Structure of NP

Theorem (Ladner - 1975)

If $\mathrm{PTIME} \neq \mathrm{NP}$, then there is a decision problem Q such that

- Q is in NP, but not in PTIME.
- Q is not NP-complete.

NP-complete
not NP-complete, not in PTIME
PTIME

The Fine Structure of NP

Theorem (Ladner - 1975)

If PTIME $\neq$ NP, then there is a decision problem Q such that

- Q is in NP, but not in PTIME.
- Q is not NP-complete.

NP-complete
not NP-complete, not in PTIME
PTIME

Feder-Vardi Dichotomy Conjecture

$\nearrow$	NP-complete
CSP(**B**)	not NP-complete, not in PTIME
$\searrow$	PTIME

Feder-Vardi Dichotomy Conjecture

Fact: Several special cases of this conjecture have been confirmed.

- **B** is an undirected graph (Hell-Nešetřil - 1990).
- **B** is a Boolean structure, i.e., $|B| = 2$ (Schaefer - 1978).
- **B** is a three-element structure, i.e., $|B| = 3$ (Bulatov - 2006).

Fact: The study of constraint satisfaction has been a meeting point of computational complexity, logic, and universal algebra.

Constraint Satisfaction and Logic

Fact:

- Each $\mathrm{CSP}(\mathbf{B})$ is expressible in Σ^1_1 (Existential SO Logic).
- Feder and Vardi identified a natural fragment of monadic Σ^1_1 that, in a precise sense "captures" constraint satisfaction.

Constraint Satisfaction and Logic

Fact:

- Each $\mathrm{CSP}(\mathbf{B})$ is expressible in Σ^1_1 (Existential SO Logic).
- Feder and Vardi identified a natural fragment of monadic Σ^1_1 that, in a precise sense "captures" constraint satisfaction.

Motivating Example:

- Recall that $\textsc{Positive NAE 3-Sat} = \mathrm{CSP}(\mathbf{B})$, where $\mathbf{B} = (\{0,1\}, R^{\mathbf{B}})$ with $R^{\mathbf{B}} = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$;
- $\textsc{Positive NAE 3-Sat}$ is definable by the Σ^1_1-sentence:

$$\exists S\ \forall x,y,z(R(x,y,z) \rightarrow (S(x) \vee S(y) \vee S(z)) \wedge (\neg S(x) \vee \neg S(y) \vee \neg S(z))).$$

Constraint Satisfaction and Logic

Fact:

- Each $\mathrm{CSP}(\mathbf{B})$ is expressible in Σ^1_1 (Existential SO Logic).
- Feder and Vardi identified a natural fragment of monadic Σ^1_1 that, in a precise sense "captures" constraint satisfaction.

Motivating Example:

- Recall that $\textsc{Positive NAE 3-Sat} = \mathrm{CSP}(\mathbf{B})$, where $\mathbf{B} = (\{0,1\}, R^{\mathbf{B}})$ with $R^{\mathbf{B}} = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$;
- $\textsc{Positive NAE 3-Sat}$ is definable by the Σ^1_1-sentence:

$$\exists S\ \forall x,y,z(R(x,y,z) \rightarrow (S(x) \vee S(y) \vee S(z)) \wedge (\neg S(x) \vee \neg S(y) \vee \neg S(z))).$$

Definition: MMSNP is the fragment of monadic Σ^1_1 such that

- all first-order quantifiers are universal;
- no inequalities $\neq$ occur;
- relation symbols from the vocabulary occur only negatively.

MMSNP vs. Constraint Satisfaction

Definition: Let ψ be an MMSNP-sentence. The model checking problem $\mathrm{MC}(\psi)$ of ψ asks: Given a structure **A**, does **A** $\models \psi$?

Theorem (Feder-Vardi 1993, Kun-Nešetřil 2008)

- For every **B**, there is an MMSNP-sentence ψ such that $\mathrm{CSP}(\mathbf{B}) = \mathrm{MC}(\psi)$.
- For every MMSNP-sentence ψ, there is a structure **B** such that $\mathrm{MC}(\psi)$ is PTIME-equivalent to $\mathrm{CSP}(\mathbf{B})$.

Corollary: There is a dichotomy in the complexity of constraint satisfaction if and only if there is a dichotomy in the complexity of the model checking problem for MMSNP.

Dependence logic

Fact: Various notions of dependence and independence are encountered in computer science and mathematics:

- Functional dependencies in relational databases;
- Independence in linear algebra;
- Independence in probability theory.

Fact: Dependence logic is a formalism for expressing and analyzing notions of dependence and independence.

- It was introduced by Jouko Väänänen in 2007.
- The origins of dependence logic can be traced to partially ordered quantifiers (Henkin - 1961) and independence-friendly logic (Hintikka-Sandu - 1989).

Relational Databases and Database Dependencies

In 1970, E.F. Codd introduced the relational database model.

- A relational database is a finite collection $R_1, \ldots, R_m$ of finite relations.
- Every relation R_i can be thought of as a table; the columns of each table have names, called attributes.

  TEACHES(instructor, course, term)

- In general, data are not arbitrary; instead, data obey certain semantic restrictions that are called database dependencies.
- Functional Dependencies (FDs) are the most widely used and extensively studied database dependencies.

Functional Dependencies

Definition: R a relation, X and Y lists of attributes of R.

- R satisfies the functional dependency $X \to Y$ if for all tuples s and s' in R such that $s[X] = s'[X]$, we have $s[Y] = s'[Y]$.
- Informally, the values of the attributes in Y are a function of the values of the attributes in X.

Examples: TEACHES(instructor, course, term)

- instructor, term $\to$ course holds if no instructor teaches more than one courses each term.
- course, term $\to$ instructor holds if no course in a given term is taught by more than one instructors.

The Implication Problem for Functional Dependencies

Definition: Σ a set of FDs, $X \rightarrow Y$ a FD.
$\Sigma \models X \rightarrow Y$ if for every relation R that satisfies every FD in Σ, we have that R satisfies $X \rightarrow Y$.

Examples: Armstrong's Axioms - 1974

- Reflexivity: If $Y \subseteq X$, then $\models X \rightarrow Y$.
- Augmentation: $X \rightarrow Y \models XZ \rightarrow YZ$, for every Z.
- Transitivity: $\{X \rightarrow Y,\ Y \rightarrow Z\} \models X \rightarrow Z$.

Theorem (Beeri-Bernstein - 1979)
The implication problem for functional dependencies is solvable in linear time.

Functional Dependencies and Dependence Logic

Characteristics of Dependence Logic:

- Functional dependencies form the basic building blocks of Dependence Logic: they are atoms with the attributes as their free variables.
- Dependence Logic augments functional dependencies with the standard constructs of first-order logic, i.e., with Boolean connectives and first-order quantifiers.

Differences between Dependence Logic and First-Order Logic

- Team semantics, instead of Tarskian semantics
- Second-order interpretation of disjunction.

The Main Ingredients of Dependence Logic

Team Semantics

- Tarskian semantics: structure **A**, formula φ, assignment s of values from B to the free variables of φ.
- Single asssigments cannot give meaning to an FD $X \to Y$. A set of assignments, i.e., a relation R is needed to give meaning to $X \to Y$. Sets of assignments are called teams.

The Main Ingredients of Dependence Logic

Team Semantics

- Tarskian semantics: structure **A**, formula φ, assignment s of values from B to the free variables of φ.
- Single asssigments cannot give meaning to an FD $X \to Y$. A set of assignments, i.e., a relation R is needed to give meaning to $X \to Y$. Sets of assignments are called teams.

Semantics of Disjunction: What does it mean to say that
$R \models$ (instructor, term $\to$ course) $\vee$ (course, term $\to$ instructor)?

The Main Ingredients of Dependence Logic

Team Semantics

- Tarskian semantics: structure **A**, formula φ, assignment s of values from B to the free variables of φ.
- Single asssigments cannot give meaning to an FD $X \to Y$. A set of assignments, i.e., a relation R is needed to give meaning to $X \to Y$. Sets of assignments are called teams.

Semantics of Disjunction: What does it mean to say that
$R \models$ (instructor, term $\to$ course) $\vee$ (course, term $\to$ instructor)?

- Pedantic Answer:
  $R \models$ instructor, term $\to$ course or
  $R \models$ course, term $\to$ instructor.
- Imaginative Answer: There are R_1, R_2 s.t. $R = R_1 \cup R_2$,
  $R_1 \models$ instructor, term $\to$ course and
  $R_2 \models$ course, term $\to$ instructor.

Dependence logic D: Syntax

Definition: Let τ be a relational vocabulary.
D(τ)-formulas are defined by the following grammar:

$$\begin{aligned}\varphi \quad ::= \quad & x_1 = x_2 \mid \neg\,(x_1 = x_2) \mid R(x_1, \ldots, x_n) \mid \neg R(x_1, \ldots, x_n) \mid \\ & \mathrm{dep}(x_1, \ldots, x_n; y) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid \forall x \varphi \mid \exists x \varphi, \\ & \text{where } R \in \tau.\end{aligned}$$

Note:

- D(τ)-formulas are assumed to be in negation normal form: negations may occur only in front of equality atoms or relational atoms.
- Dependence atoms $\mathrm{dep}(x_1, \ldots, x_n; y)$ occur only positively.

Dependence logic D: Team Semantics

Definition: A team on $\mathbf{A}$ is a set T of assignments $s : V \to A$, for some fixed set $V = \mathrm{dom}(T)$ of variables.

Team Semantics: $\mathbf{A}, T \models \varphi$

- Atomic or negated atomic formula θ
  $\mathbf{A}, T \models \theta$ if $\mathbf{A}, s \models \theta$, for every $s \in T$.
- Dependence atom $\mathrm{dep}(\boldsymbol{x}; y)$
  $\mathbf{A}, T \models \mathrm{dep}(x_1, \ldots, x_n; y)$ if there is $f : A^n \to A$ such that for all $s \in T$, we have that $s(y) = f(s(x_1), \ldots, s(x_n))$.
- Conjunction
  $\mathbf{A}, T \models \varphi \wedge \psi$ if $\mathbf{A}, T \models \varphi$ and $\mathbf{A}, T \models \psi$.
- Disjunction
  $\mathbf{A}, T \models \varphi \vee \psi$ if there are $T', T'' \subseteq T$ such that
  $T' \cup T'' = T, \quad \mathbf{A}, T' \models \varphi, \quad \mathbf{A}, T'' \models \psi$.

Dependence logic D: Team Semantics (continued)

Team Semantics: $\mathbf{A}, T \models \varphi$

- Universal quantifier
  $\mathbf{A}, T \models \forall x \psi$ if $\mathbf{A}, T[A/x] \models \psi$,
  where
  $T[A/x] = \{s[a/x] : s \in T, a \in A\}$.
- Existential quantifier
  $\mathbf{A}, T \models \exists x \psi$ if there is $F \colon T \to A$ such that $\mathbf{A}, T[F/x] \models \psi$,
  where
  $T[F/x] = \{s[F(s)/x] : s \in T\}$.
- If ψ is a D-sentence, then $\mathbf{A} \models \psi$ if $\mathbf{A}, \{\emptyset\} \models \psi$.

Dependence logic: Expressive Power

Theorem (Väänänen - 2007)
For sentences, $D = \Sigma^1_1$ (Existential Second-Order Logic)

Dependence logic: Expressive Power

Theorem (Väänänen - 2007)
For sentences, $\mathrm{D} = \Sigma_1^1$ (Existential Second-Order Logic)

Fagin's Theorem - 1974
On the class of all finite structures, $\Sigma_1^1 = \mathrm{NP}$.

Corollary:
On the class of all finite structures, $\mathrm{D} = \mathrm{NP}$. Hence, every constraint satisfaction problem $\mathrm{CSP}(\mathbf{B})$ is D-definable.

Dependence logic: Expressive Power

Theorem (Väänänen - 2007)
For sentences, $\mathsf{D} = \Sigma^1_1$ (Existential Second-Order Logic)

Fagin's Theorem - 1974
On the class of all finite structures, $\Sigma^1_1 = \mathrm{NP}$.

Corollary:
On the class of all finite structures, $\mathsf{D} = \mathrm{NP}$. Hence, every constraint satisfaction problem $\mathrm{CSP}(\mathbf{B})$ is D-definable.

Theorem (Jarmo Kontinen - 2013)
3-SAT is polynomial-time reducible to the model-checking problem of the quantifier-free D-formula

$$\mathrm{dep}(x; y) \vee \mathrm{dep}(u; v) \vee \mathrm{dep}(u; v).$$

Constraint Satisfaction vs. Dependence Logic

Question:
What is the exact connection between dependence logic and constraint satisfaction?

Constraint Satisfaction vs. Dependence Logic

Question:
What is the exact connection between dependence logic and constraint satisfaction?

Main Result:
There is natural fragment of dependence logic that, in a precise sense, captures exactly the class of all constraint satifaction problems $\mathrm{CSP}(\mathbf{B})$.

Uniform Dependence Atoms

Uniform dependence atom: $\mathrm{udep}(x_1, \ldots, x_n; y_1, \ldots, y_n)$

Semantics: $\mathbf{A}, T \models \mathrm{udep}(x_1, \ldots, x_n; y_1, \ldots, y_n)$ if there is a unary function $g : A \to A$ such that for every $s \in T$, we have that

$$s(y_1) = g(s(x_1)), \ldots, s(y_n) = g(s(x_n)).$$

Uniform Dependence Atoms

Uniform dependence atom: $\mathrm{udep}(x_1, \ldots, x_n; y_1, \ldots, y_n)$

Semantics: $\mathbf{A}, T \models \mathrm{udep}(x_1, \ldots, x_n; y_1, \ldots, y_n)$ if there is a unary function $g : A \to A$ such that for every $s \in T$, we have that

$$s(y_1) = g(s(x_1)), \ldots, s(y_n) = g(s(x_n)).$$

Uniform k-valued dependence atom:
$\mathrm{udep}[k](x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n)$, where $\alpha_1, \ldots, \alpha_n$ are k-valued variables ranging over the set $[k] = \{1, \ldots, k\}$.

Semantics: $\mathbf{A}, T \models \mathrm{udep}[k](x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n)$ if there is a unary function $h : A \to [k]$ such that for every $s \in T$, we have that

$$s(\alpha_1) = h(s(x_1)), \ldots, s(\alpha_n) = h(s(x_n)).$$

Universal Monotone Uniform Dependence Logic

- QF-MUD[k]: Quantifier-free monotone dependence logic with uniform k-valued dependence atoms

$$\varphi \quad ::= \quad \alpha = \underline{i} \mid \neg R(\boldsymbol{x}) \mid \mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha}) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2),$$

  where $i \in [k]$.

- QF-MUD $= \bigcup_{k \geq 1}$ QF-MUD[k].
- $\forall$-MUD[k]: Universal monotone dependence logic with uniform k-valued dependence atoms

$$\varphi \quad ::= \quad \psi \mid \forall x \varphi \mid \forall \alpha \varphi,$$

  where $\psi \in$ QF-MUD[k].

- $\forall$-MUD $= \bigcup_{k \geq 1} \forall$-MUD[$k$].

Universal Monotone Uniform Dependence Logic

$$\varphi \quad ::= \quad \alpha = \underline{i} \mid \neg R(\boldsymbol{x}) \mid \mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha}) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \\ \forall x \varphi \mid \forall \alpha \varphi.$$

Remarks:

- Analogously to MMSNP, the logics QF-MUD and ∀-MUD allow no inequalities and only negative occurrences of $R \in \tau$.
- $\mathrm{udep}[k](x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n)$ is expressed by the D-formula
  $$\forall y \exists \beta (\mathrm{dep}(y;\beta) \wedge \textstyle\bigwedge_{i\in[n]}(y = x_i \rightarrow \beta = \alpha_i)).$$
  This formula violates the syntactic restrictions of ∀-MUD[k]:
  – It contains existential quantification;
  – It contains inequalities between first-order variables.

Constraint Satisfaction vs. Dependence Logic

Theorem: Constraint Satisfaction is PTIME-equivalent to the Model Checking Problem for $\forall$-MUD.

- For every structure **B**, there is a $\forall$-MUD-sentence $\varphi_{\mathbf{B}}$ such that $\mathrm{CSP}(\mathbf{B})$ is PTIME-equivalent to $\mathrm{MC}(\varphi_{\mathbf{B}})$.
- For every $\forall$-MUD-sentence φ, there is a structure $\mathbf{B}_\varphi$ such that $\mathrm{MC}(\varphi)$ is PTIME-equivalent to $\mathrm{CSP}(\mathbf{B}_\varphi)$.

Corollary: The Feder-Vardi Dichotomy Conjecture for $\mathrm{CSP}(\mathbf{B})$ holds if and only if a dichotomy in the complexity of the Model Checking Problem for $\forall$-MUD holds.

From Constraint Satisfaction to Dependence Logic

Theorem A: ($\forall$-MUD captures CSP)
Assume that $\tau = \{R\}$. For every τ-structure $\mathbf{C}$ with $|C| = k$, there is a $\forall$-MUD[k]-sentence $\varphi_{\mathbf{C}}$ such that for every τ-structure $\mathbf{A}$,

$$\mathbf{A} \in \mathrm{CSP}(\mathbf{C}) \text{ if and only if } \mathbf{A} \models \varphi_{\mathbf{C}}.$$

From Constraint Satisfaction to Dependence Logic

Theorem A: ($\forall$-MUD captures CSP)
Assume that $\tau = \{R\}$. For every τ-structure $\mathbf{C}$ with $|C| = k$, there is a $\forall$-MUD[k]-sentence $\varphi_{\mathbf{C}}$ such that for every τ-structure $\mathbf{A}$,

$$\mathbf{A} \in \mathrm{CSP}(\mathbf{C}) \text{ if and only if } \mathbf{A} \models \varphi_{\mathbf{C}}.$$

Theorem: (Feder-Vardi - 1993)
For every structure $\mathbf{B}$, there is a structure $\mathbf{C}$ over a vocabulary with a single binary relation symbol such that $\mathrm{CSP}(\mathbf{B})$ is PTIME-equivalent to $\mathrm{CSP}(\mathbf{C})$.

Corollary: For every structure $\mathbf{B}$, there is a $\forall$-MUD-sentence $\varphi_{\mathbf{B}}$ such that $\mathrm{CSP}(\mathbf{B})$ is PTIME-equivalent to $\mathrm{MC}(\varphi_{\mathbf{B}})$.

From Constraint Satisfaction to Dependence Logic

- To prove Theorem A, it suffices to find a QF-MUD[k]-formula $\theta_{\mathbf{C}}$ such that $\mathbf{A} \in \mathrm{CSP}(\mathbf{C})$ if and only if $\mathbf{A}, F \models \theta_{\mathbf{C}}$, where F is the full team consisting of all assignments
$$s : \{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n\} \to A \cup [k].$$
This is so because $\mathbf{A}, F \models \theta_{\mathbf{C}}$ if and only if $\mathbf{A} \models \varphi_{\mathbf{C}}$, where $\varphi_{\mathbf{C}}$ is the sentence $\forall \boldsymbol{x} \forall \boldsymbol{\alpha}\, \theta_{\mathbf{C}}$.

- Observe next that if $\mathbf{A}, T \models \mathrm{udep}[k](\boldsymbol{x}, \boldsymbol{\alpha})$, then there is a homomorphism $h : (A, R_{T,\boldsymbol{x}}) \to ([k], R_{T,\boldsymbol{\alpha}})$, where $R_{T,\boldsymbol{x}}$ is the relation $\{s(\boldsymbol{x}) : s \in T\}$, and similarly for $R_{T,\boldsymbol{\alpha}}$. Thus, if $R^{\mathbf{A}} \subseteq R_{T,\boldsymbol{x}}$ and $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathbf{C}}$, then $\mathbf{A} \in \mathrm{CSP}(\mathbf{C})$.

- The idea of the proof is to build $\theta_{\mathbf{C}}$ using disjunctions in such a way that if $\mathbf{A}, F \models \theta_{\mathbf{C}}$, then there is a subteam T of F satisfying the conditions above.

From Constraint Satisfaction to Dependence Logic

Example: $\textsc{Positive 1-in-3 3-Sat} = \mathrm{CSP}(\mathbf{B})$, where

$$\mathbf{B} = (\{0,1\}, R^{\mathbf{B}}) \text{ with } R^{\mathbf{B}} = \{(1,0,0),(0,1,0),(0,0,1)\}.$$

Here, we have that

$$\varphi_{\mathbf{B}} := \forall x_1 \forall x_2 \forall x_3 \forall \alpha_1 \forall \alpha_2 \forall \alpha_3 ((\eta_{\mathbf{B}} \wedge \psi_{\mathbf{B}}) \vee \neg R(x_1, x_2, x_3) \vee \nu_{\mathbf{B}}),$$

where

- $\eta_{\mathbf{B}} := \mathrm{udep}[2](x_3; \alpha_3) \vee (\alpha_3 = \underline{0} \wedge \mathrm{udep}[2](x_2; \alpha_2))$
  $\vee \mathrm{udep}[2](x_1, x_2, x_3; \alpha_1, \alpha_2, \alpha_3)$
- $\psi_{\mathbf{B}} := (\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{0}) \vee$
  $(\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{0}) \vee (\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{1})$
- $\nu_{\mathbf{B}} := (\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{1}) \vee (\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{0}) \vee$
  $(\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{0}) \vee (\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{1}) \vee$
  $(\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{1})$.

From Dependence Logic to Constraint Satisfaction

Theorem B: (MMSNP is at least as expressive as $\forall$-MUD)
For every $\forall$-MUD-sentence φ be a sentence, there is a MMSNP-sentence φ^* such that for every structure $\mathbf{A}$

$$\mathbf{A} \models \varphi \text{ if and only if } \mathbf{A} \models \varphi^*.$$

Recall the following result:

Theorem (Feder-Vardi - 1993)
For every MMSNP-sentence ψ, there is a structure $\mathbf{B}$ such that $\mathrm{MC}(\psi)$ is equivalent to $\mathrm{CSP}(\mathbf{B})$ via PTIME-reductions.

Corollary: For every $\forall$-MUD-sentence φ, there is a structure $\mathbf{B}_\varphi$ such that $\mathrm{MC}(\varphi)$ is PTIME-equivalent to $\mathrm{CSP}(\mathbf{B}_\varphi)$.

From Dependence Logic to MMSNP

- To prove Theorem B, translate inductively QF-MUD[k] to the extension of MMSNP with k-valued variables.

- The translation of each QF-MUD-formula ψ is of the form
$$\exists \mathbf{P} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha} (R(\boldsymbol{x\alpha}) \to \psi^+),$$
where ψ^+ is quantifier free. Here, R is an extra relation symbol interpreted by the team on which ψ is evaluated.

- One can prove that
$$\mathbf{A}, T \models \psi \quad \text{iff} \quad (\mathbf{A}, R_{T,\boldsymbol{x\alpha}}) \models \exists \mathbf{P} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha} (R(\boldsymbol{x\alpha}) \to \psi^+).$$

- This extends easily to $\forall$-MUD[k]-sentences
$$\mathbf{A} \models \forall \boldsymbol{x} \forall \boldsymbol{\alpha} \psi \quad \text{iff} \quad \mathbf{A} \models \exists \mathbf{P} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha} \psi^+.$$

- Theorem B follows from this, as the k-valued variables can be easily eliminated from sentences of MMSNP.

Concluding Remarks

- We identified a fragment of dependence logic that captures constraint satisfaction, up to polynomial-time equivalence.
- This result implies that a complexity classification of the model checking problem for universal dependence logic is at least as hard as settling the Feder-Vardi dichotomy conjecture for constraint satisfaction.
- What is the exact expressive power of $\forall$-MUD?
  - Is $\mathrm{CSP}(\mathbf{B})$ definable in $\forall$-MUD for every $\mathbf{B}$?
  - Is $\forall$-MUD a proper fragment of MMSNP?