Universal algebra and the constraint satisfaction problem

Libor Barto

Charles University in Prague

ASL North American Annual Meeting
University of Colorado, 20 May 2014

Constraint satisfaction problem (CSP)

- Common framework for many practical problems

Constraint satisfaction problem (CSP)

- Common framework for many practical problems
- This is the last time when the word “practical” appears

Constraint satisfaction problem (CSP)

- Common framework for some computational problems
  - Broad enough to include interesting examples
  - Narrow enough to make significant progress (on all problems within a class, rather than just a single computational problem)

Constraint satisfaction problem (CSP)

- Common framework for some computational problems
  - Broad enough to include interesting examples
  - Narrow enough to make significant progress (on all problems within a class, rather than just a single computational problem)
- Main achievement: better understanding why problems are easy or hard:

Constraint satisfaction problem (CSP)

- Common framework for some computational problems
  - Broad enough to include interesting examples
  - Narrow enough to make significant progress (on all problems within a class, rather than just a single computational problem)
- Main achievement: better understanding why problems are easy or hard:
  - Hardness comes from lack of symmetry
  - Symmetries of higher arity are important (not just automorphisms or endomorphisms) $\rightarrow$ **universal algebra** (not just group or semigroup theory)

Constraint satisfaction problem (CSP)

- Common framework for some computational problems
  - Broad enough to include interesting examples
  - Narrow enough to make significant progress (on all problems within a class, rather than just a single computational problem)
- Main achievement: better understanding why problems are easy or hard:
  - Hardness comes from lack of symmetry
  - Symmetries of higher arity are important (not just automorphisms or endomorphisms) $\rightarrow$ **universal algebra** (not just group or semigroup theory)
- Long term goal: go beyond CSP

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment $=$ mapping *variables* $\to$ *domain*

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment $=$ mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment = mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment $=$ mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution
- **Counting CSP:** How many solutions are there?

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment = mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution
- **Counting CSP:** How many solutions are there?
- **Max-CSP:** Find a map satisfying maximum number of constraints

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x,y,z),\ R_2(t,z),\ R_1(y,y,z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment $=$ mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution
- **Counting CSP:** How many solutions are there?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** Find a map satisfying $0.7 \times \mathit{Optimum}$ constraints

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment $=$ mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution
- **Counting CSP:** How many solutions are there?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** or: Find a map satisfying 0.3-fraction of constraints given 0.6-satisfiable instance

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment $=$ mapping *variables* $\to$ *domain*

- **Decision CSP:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution
- **Counting CSP:** How many solutions are there?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** or: Find a map satisfying 0.3-fraction of constraints given 0.6-satisfiable instance
- **Robust CSP:** Find an almost satifying assignment given an almost satisfiable instance

CSP over a structure (aka constraint language)

Interesting subproblems ... restrict the set of allowed relations

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite

Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

- What is the computational complexity for fixed $\mathbb{A}$?

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite

Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

- What is the computational complexity for fixed $\mathbb{A}$?
- **This talk:** Mainly decision $CSP(\mathbb{A})$

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

- What is the computational complexity for fixed $\mathbb{A}$?
- **This talk:** Mainly decision $CSP(\mathbb{A})$
- **Other interesting problems:**
  - restrict something else than the set of allowed relations
  - allow infinite A
  - allow weighted relations: mappigs $A^k \to Q \cup \{\infty\}$
  - (approximate) counting, Max-CSP, Approx Max-CSP

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

Examples:

- 3-*SAT*: $\mathbb{A} = (\{0, 1\}; x \vee y \vee z, x \vee y \vee \neg z, \dots)$

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

Examples:

- 3-SAT: $\mathbb{A} = (\{0, 1\}; x \vee y \vee z, x \vee y \vee \neg z, \dots)$
- 3-COL: $\mathbb{A} = (\{0, 1, 2\}; x \neq y)$

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

Examples:

- 3-SAT: $\mathbb{A} = (\{0, 1\}; x \vee y \vee z, x \vee y \vee \neg z, \dots)$
- 3-COL: $\mathbb{A} = (\{0, 1, 2\}; x \neq y)$
- $HORN$-3-SAT: $\mathbb{A} = (\{0, 1\}; x, \neg x, x \wedge y \rightarrow z)$

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

Examples:

- 3-*SAT*: $\mathbb{A} = (\{0, 1\}; x \vee y \vee z, x \vee y \vee \neg z, \dots)$
- 3-*COL*: $\mathbb{A} = (\{0, 1, 2\}; x \neq y)$
- *HORN*-3-*SAT*: $\mathbb{A} = (\{0, 1\}; x, \neg x, x \wedge y \to z)$
- *q*-*LIN*: $\mathbb{A} = (GF(q); \textit{affine subspaces})$.

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite
Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

Examples:

- $3\text{-}SAT$: $\mathbb{A} = (\{0,1\}; x \vee y \vee z, x \vee y \vee \neg z, \dots)$
- $3\text{-}COL$: $\mathbb{A} = (\{0,1,2\}; x \neq y)$
- $HORN\text{-}3\text{-}SAT$: $\mathbb{A} = (\{0,1\}; x, \neg x, x \wedge y \to z)$
- $q\text{-}LIN$: $\mathbb{A} = (GF(q); \mathit{affine\ subspaces})$.
- Digraph reachability: $\mathbb{A} = (\{0,1\}; x, \neg x, x \leq y)$
- Graph reachability: $\mathbb{A} = (\{0,1\}, x, \neg x, x = y)$

CSP over a structure (aka constraint language)

Definition

$\mathbb{A} = (A; R_1, R_2, \dots)$: relational structure with A finite

Instance of $CSP(\mathbb{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where each R_i is in $\mathbb{A}$.

Examples:

- (NP-c) 3-SAT: $\mathbb{A} = (\{0, 1\}; x \vee y \vee z, x \vee y \vee \neg z, \dots)$
- (NP-c) 3-COL: $\mathbb{A} = (\{0, 1, 2\}; x \neq y)$
- (P-c) $HORN$-3-SAT: $\mathbb{A} = (\{0, 1\}; x, \neg x, x \wedge y \to z)$
- q-LIN: $\mathbb{A} = (GF(q); \textit{affine subspaces})$.
- (NL-c) Digraph reachability: $\mathbb{A} = (\{0, 1\}; x, \neg x, x \leq y)$
- (L-c) Graph reachability: $\mathbb{A} = (\{0, 1\}, x, \neg x, x = y)$

Decision CSP as model checking problem

$CSP(\mathbb{A})$:
Instance: Sentence ϕ in the language of $\mathbb{A}$ with $\exists$ and $\wedge$
Question: Is ϕ true in $\mathbb{A}$?

Decision CSP as model checking problem

$CSP(\mathbb{A})$:
Instance: Sentence ϕ in the language of $\mathbb{A}$ with $\exists$ and $\wedge$
Question: Is ϕ true in $\mathbb{A}$?

What about: Allow some other combination of $\{\exists, \forall, \wedge, \vee, \neg, =, \neq\}$.

Decision CSP as model checking problem

$CSP(\mathbb{A})$:
Instance: Sentence ϕ in the language of $\mathbb{A}$ with $\exists$ and $\wedge$
Question: Is ϕ true in $\mathbb{A}$?

What about: Allow some other combination of $\{\exists, \forall, \wedge, \vee, \neg, =, \neq\}$.

From 2^7 cases only 3 interesting (others reduce to these or are boring)

- $\{\exists, \wedge, (=)\}$ (CSP)
  open
- $\{\exists, \forall, \wedge, (=)\}$ (qCSP)
  open
- $\{\exists, \forall, \wedge, \vee\}$ (Positive equality free)
  solved - tetrachotomy P, NP-c, co-NP-c, PSPACE-c
  B.Martin, F.Madelaine 11

The dichotomy conjecture

A largest natural class of problems with a dichotomy?

The dichotomy conjecture

A largest natural class of problems with a dichotomy?

Conjecture (The dichotomy conjecture Feder and Vardi'93)

For every $\mathbb{A}$, decision $\mathrm{CSP}(\mathbb{A})$ is either in P or NP-complete.

The dichotomy conjecture

A largest natural class of problems with a dichotomy?

Conjecture (The dichotomy conjecture Feder and Vardi'93)

For every $\mathbb{A}$, decision $\mathrm{CSP}(\mathbb{A})$ is either in P or NP-complete.

- Evidence (in 93):
  - True for $|A| = 2$ Schaefer'78
  - True if $\mathbb{A} = (A; R)$, R is binary and symmetric Hell and Nešetřil'90

The dichotomy conjecture

A largest natural class of problems with a dichotomy?

Conjecture (The dichotomy conjecture Feder and Vardi'93)

For every $\mathbb{A}$, decision $\mathrm{CSP}(\mathbb{A})$ is either in P or NP-complete.

- Evidence (in 93):
  - True for $|A| = 2$ Schaefer'78
  - True if $\mathbb{A} = (A; R)$, R is binary and symmetric Hell and Nešetřil'90
- Feder and Vardi suggested that tractability is tied to "closure properties"

The dichotomy conjecture

A largest natural class of problems with a dichotomy?

Conjecture (The dichotomy conjecture Feder and Vardi'93)

For every $\mathbb{A}$, decision $\mathrm{CSP}(\mathbb{A})$ is either in P or NP-complete.

- Evidence (in 93):
  - True for $|A| = 2$ Schaefer'78
  - True if $\mathbb{A} = (A; R)$, R is binary and symmetric Hell and Nešetřil'90
- Feder and Vardi suggested that tractability is tied to "closure properties"
- $\rightarrow$ algebraic approach Bulatov, Jeavons, Krokhin'00

Disclaimer

Most of the definitions will be imprecise

Almost no theorem is true as stated

PP and UA

Simulation

- If $\mathbb{A}$ "can simulate" $\mathbb{B}$ then $CSP(\mathbb{A})$ is at least as hard as $CSP(\mathbb{B})$.

Simulation

- If $\mathbb{A}$ "can simulate" $\mathbb{B}$ then $CSP(\mathbb{A})$ is at least as hard as $CSP(\mathbb{B})$.
- **What does simulate mean?**

Simulation

- If $\mathbb{A}$ "can simulate" $\mathbb{B}$ then
  $CSP(\mathbb{A})$ is at least as hard as $CSP(\mathbb{B})$.
- **What does simulate mean?**
- (Slightly imprecise) answer:
  "can simulate" means "positively primitively (pp) interprets"

Simulation

- If $\mathbb{A}$ "can simulate" $\mathbb{B}$ then
  $CSP(\mathbb{A})$ is at least as hard as $CSP(\mathbb{B})$.
- **What does simulate mean?**
- (Slightly imprecise) answer:
  "can simulate" means "positively primitively (pp) interprets"
- Special case of pp-interpretability is pp-definability

Simulation

- If $\mathbb{A}$ "can simulate" $\mathbb{B}$ then
  $CSP(\mathbb{A})$ is at least as hard as $CSP(\mathbb{B})$.
- **What does simulate mean?**
- (Slightly imprecise) answer:
  "can simulate" means "positively primitively (pp) interprets"
- Special case of pp-interpretability is pp-definability
- Assume $\mathbb{A}, \mathbb{B}$ have the same domain.
  $\mathbb{A}$ pp-defines $\mathbb{B}$ $=$ relations in $\mathbb{B}$ definable using
  relations in $\mathbb{A}$, and $\exists, =, \wedge$.

Example of pp-definability

- $\mathbb{A} = (A; R)$, where R is ternary

Example of pp-definability

- $\mathbb{A} = (A; R)$, where R is ternary
- $\mathbb{B} = (B; S, T)$, where S is binary and T is unary
  - $S(x, y) \quad \text{iff} \quad (\exists z)\ R(x, y, z) \wedge R(y, y, x)$
  - $T(x) \quad \text{iff} \quad R(x, x, x)$

Example of pp-definability

- $\mathbb{A} = (A; R)$, where R is ternary
- $\mathbb{B} = (B; S, T)$, where S is binary and T is unary
  - $S(x, y) \quad \text{iff} \quad (\exists z)\ R(x, y, z) \wedge R(y, y, x)$
  - $T(x) \quad \text{iff} \quad R(x, x, x)$
- Each instance of $\mathrm{CSP}(\mathbb{B})$, eg.

$$T(z),\ S(x, y)$$

Example of pp-definability

- $\mathbb{A} = (A; R)$, where R is ternary
- $\mathbb{B} = (B; S, T)$, where S is binary and T is unary
  - $S(x, y) \quad \text{iff} \quad (\exists z)\ R(x, y, z) \wedge R(y, y, x)$
  - $T(x) \quad \text{iff} \quad R(x, x, x)$
- Each instance of $\mathrm{CSP}(\mathbb{B})$, eg.

$$T(z),\ S(x, y)$$

- can be rewritten to an equivalent instance of $\mathrm{CSP}(\mathbb{A})$

$$R(z, z, z),\ R(x, y, w),\ R(y, y, x)$$

Example of pp-definability

- $\mathbb{A} = (A; R)$, where R is ternary
- $\mathbb{B} = (B; S, T)$, where S is binary and T is unary
  - $S(x, y)$ iff $(\exists z)\ R(x, y, z) \wedge R(y, y, x)$
  - $T(x)$ iff $R(x, x, x)$
- Each instance of $\mathrm{CSP}(\mathbb{B})$, eg.

$$T(z),\ S(x, y)$$

- can be rewritten to an equivalent instance of $\mathrm{CSP}(\mathbb{A})$

$$R(z, z, z),\ R(x, y, w),\ R(y, y, x)$$

- Thus $\mathrm{CSP}(\mathbb{A})$ is at least as hard as $\mathrm{CSP}(\mathbb{B})$

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if
  - The domain of $\mathbb{B}$ is a pp-definable subset of A^k modulo a pp-definable equivalence

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if
  - The domain of $\mathbb{B}$ is a pp-definable subset of A^k modulo a pp-definable equivalence
  - The relations of $\mathbb{B}$ are "pp-definable" from $\mathbb{A}$
    (m-ary relation on B is defined as a km-ary relation on A)

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if
  - The domain of $\mathbb{B}$ is a pp-definable subset of A^k modulo a pp-definable equivalence
  - The relations of $\mathbb{B}$ are "pp-definable" from $\mathbb{A}$ (m-ary relation on B is defined as a km-ary relation on A)
- If $\mathbb{A}$ pp-interprets the structure corresponding to 3-*SAT* then $\mathrm{CSP}(\mathbb{A})$ is NP-complete BJK

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if
  - The domain of $\mathbb{B}$ is a pp-definable subset of A^k modulo a pp-definable equivalence
  - The relations of $\mathbb{B}$ are “pp-definable” from $\mathbb{A}$ (m-ary relation on B is defined as a km-ary relation on A)
- If $\mathbb{A}$ pp-interprets the structure corresponding to 3-*SAT* then $\mathrm{CSP}(\mathbb{A})$ is NP-complete BJK
- This explains NP-completness for all known NP-complete CSPs...

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if
  - The domain of $\mathbb{B}$ is a pp-definable subset of A^k modulo a pp-definable equivalence
  - The relations of $\mathbb{B}$ are "pp-definable" from $\mathbb{A}$ (m-ary relation on B is defined as a km-ary relation on A)
- If $\mathbb{A}$ pp-interprets the structure corresponding to 3-*SAT* then $\mathrm{CSP}(\mathbb{A})$ is NP-complete BJK
- This explains NP-completness for all known NP-complete CSPs...

Conjecture (The algebraic dichotomy conjecture Bulatov, Jeavons, Krokhin)

If $\mathbb{A}$ does not interpret 3-*SAT* then $\mathrm{CSP}(\mathbb{A})$ is in P.

pp-interpretation, the borderline?

- $\mathbb{A}$ pp-interprets $\mathbb{B}$ if
  - The domain of $\mathbb{B}$ is a pp-definable subset of A^k modulo a pp-definable equivalence
  - The relations of $\mathbb{B}$ are "pp-definable" from $\mathbb{A}$ (m-ary relation on B is defined as a km-ary relation on A)
- If $\mathbb{A}$ pp-interprets the structure corresponding to 3-SAT then $\mathrm{CSP}(\mathbb{A})$ is NP-complete BJK
- This explains NP-completness for all known NP-complete CSPs...

Conjecture (The algebraic dichotomy conjecture Bulatov, Jeavons, Krokhin)

If $\mathbb{A}$ does not interpret 3-SAT then $\mathrm{CSP}(\mathbb{A})$ is in P.

Similar conjectures and hardness results about L, NL
Larose, Tesson

PP and UA

- Operation $t : A^k \to A$ is compatible with relation $R \subseteq A^n$, if R is closed under coordinate-wise application of t.

PP and UA

- Operation $t : A^k \to A$ is compatible with relation $R \subseteq A^n$, if R is closed under coordinate-wise application of t.
- Operation $t : A^k \to A$ is a polymorphism of $\mathbb{A}$ if it is compatible with every relation in $\mathbb{A}$

  polymorphism with $k = 1 =$ endomorphism
  polymorphism with $k > 1 =$ higher arity symmetry

PP and UA

- Operation $t : A^k \to A$ is compatible with relation $R \subseteq A^n$, if R is closed under coordinate-wise application of t.
- Operation $t : A^k \to A$ is a polymorphism of $\mathbb{A}$ if it is compatible with every relation in $\mathbb{A}$

  polymorphism with $k = 1$ = endomorphism
  polymorphism with $k > 1$ = higher arity symmetry
- $\mathrm{Pol}(\mathbb{A}) = (A; \textit{all polymorphisms of}\, \mathbb{A})$ ... the algebra of polymorphisms

PP and UA

- Operation $t : A^k \to A$ is compatible with relation $R \subseteq A^n$, if R is closed under coordinate-wise application of t.
- Operation $t : A^k \to A$ is a polymorphism of $\mathbb{A}$ if it is compatible with every relation in $\mathbb{A}$

  polymorphism with $k = 1$ = endomorphism
  polymorphism with $k > 1$ = higher arity symmetry
- $\mathrm{Pol}(\mathbb{A}) = (A; \textit{all polymorphisms of } \mathbb{A})$ ... the algebra of polymorphisms
- **Old theorem**: $\mathbb{A}$ pp-defines $\mathbb{B}$ iff $\mathrm{Pol}(\mathbb{A}) \subseteq \mathrm{Pol}(\mathbb{B})$
  Geiger'68, Bondarchuk, Kaluznin, Kotov, Romov'69

PP and UA

- Operation $t : A^k \to A$ is compatible with relation $R \subseteq A^n$, if R is closed under coordinate-wise application of t.
- Operation $t : A^k \to A$ is a polymorphism of $\mathbb{A}$ if it is compatible with every relation in $\mathbb{A}$

  polymorphism with $k = 1$ = endomorphism
  polymorphism with $k > 1$ = higher arity symmetry
- $\mathrm{Pol}(\mathbb{A}) = (A; \textit{all polymorphisms of}\, \mathbb{A})$ ... the algebra of polymorphisms
- **Old theorem**: $\mathbb{A}$ pp-defines $\mathbb{B}$ iff $\mathrm{Pol}(\mathbb{A}) \subseteq \mathrm{Pol}(\mathbb{B})$
  Geiger'68, Bondarchuk, Kaluznin, Kotov, Romov'69
- More generally: $\mathbb{A}$ pp-interprets $\mathbb{B}$ iff $\mathrm{Pol}(\mathbb{B})$ interprets $\mathrm{Pol}(\mathbb{A})$
  Birkhoff'35, Bodirsky'08

PP and UA

- Operation $t : A^k \to A$ is compatible with relation $R \subseteq A^n$, if R is closed under coordinate-wise application of t.
- Operation $t : A^k \to A$ is a polymorphism of $\mathbb{A}$ if it is compatible with every relation in $\mathbb{A}$

  polymorphism with $k = 1$ = endomorphism
  polymorphism with $k > 1$ = higher arity symmetry
- $\mathrm{Pol}(\mathbb{A}) = (A; \textit{all polymorphisms of } \mathbb{A})$ ... the algebra of polymorphisms
- **Old theorem**: $\mathbb{A}$ pp-defines $\mathbb{B}$ iff $\mathrm{Pol}(\mathbb{A}) \subseteq \mathrm{Pol}(\mathbb{B})$
  Geiger'68, Bondarchuk, Kaluznin, Kotov, Romov'69
- More generally: $\mathbb{A}$ pp-interprets $\mathbb{B}$ iff $\mathrm{Pol}(\mathbb{B})$ interprets $\mathrm{Pol}(\mathbb{A})$
  Birkhoff'35, Bodirsky'08
- Interpretations closely connected to central objects of study in UA: varieties and Mal'tsev conditions

On the algebraic approach

Interpretations closely connected to central objects of study in UA: varieties and Mal'tsev conditions

On the algebraic approach

Interpretations closely connected to central objects of study in UA: varieties and Mal'tsev conditions

- Important conditions on $\mathbb{A}$ correspond to previously studied conditions for $\mathrm{Pol}(\mathbb{A})$

On the algebraic approach

Interpretations closely connected to central objects of study in UA: varieties and Mal'tsev conditions

- Important conditions on $\mathbb{A}$ correspond to previously studied conditions for $\mathrm{Pol}(\mathbb{A})$
- We can use UA to identify interesting special cases

On the algebraic approach

Interpretations closely connected to central objects of study in UA: varieties and Mal'tsev conditions

- Important conditions on $\mathbb{A}$ correspond to previously studied conditions for $\mathsf{Pol}(\mathbb{A})$
- We can use UA to identify interesting special cases
- Sometimes operations are directly used in algorithms

On the algebraic approach

Interpretations closely connected to central objects of study in UA: varieties and Mal'tsev conditions

- Important conditions on $\mathbb{A}$ correspond to previously studied conditions for $\mathrm{Pol}(\mathbb{A})$
- We can use UA to identify interesting special cases
- Sometimes operations are directly used in algorithms

Theorem

The following are equivalent.

1. $\mathbb{A}$ *does not interpret (=cannot simulate) 3-SAT*
2. ...

... *Taylor, Hobby, McKenzie, Bulatov, Maróti, Siggers, ...*

33. ...
34. $\mathrm{Pol}(\mathbb{A})$ *contains an operation* t *such that* $t(a, a, \ldots, a) = a$ *and* $t(a_1, a_2, \ldots, a_k) = t(a_2, \ldots, a_k, a_1)$ *for all* $a, a_i \in A$
    B, Kozik'10

Results

Results

- Better understanding of pre-algebraic results

Results

- Better understanding of pre-algebraic results
- Far broader special cases solved. The dichotomy conjecture is true:
  - if $|A| = 3$ Bulatov'06
  - if $|A| = 4$ Marković et al.
  - if $\mathbb{A}$ contains all unary relations Bulatov'03, Barto'11
  - if $\mathbb{A} = (A; R)$ where R is binary, without sources or sinks Barto, Kozik, Niven'09

Results

- Better understanding of pre-algebraic results
- Far broader special cases solved. The dichotomy conjecture is true:
  - if $|A| = 3$ Bulatov'06
  - if $|A| = 4$ Marković et al.
  - if $\mathbb{A}$ contains all unary relations Bulatov'03, Barto'11
  - if $\mathbb{A} = (A; R)$ where R is binary, without sources or sinks Barto, Kozik, Niven'09
- Applicability of known algorithmic principles understood

Results

- Better understanding of pre-algebraic results
- Far broader special cases solved. The dichotomy conjecture is true:
  - if $|A| = 3$ Bulatov'06
  - if $|A| = 4$ Marković et al.
  - if $\mathbb{A}$ contains all unary relations Bulatov'03, Barto'11
  - if $\mathbb{A} = (A; R)$ where R is binary, without sources or sinks Barto, Kozik, Niven'09
- Applicability of known algorithmic principles understood
  - Describing all solutions
    Idziak, Markovic, McKenzie, Valeriote, Willard'07

Results

- Better understanding of pre-algebraic results
- Far broader special cases solved. The dichotomy conjecture is true:
  - if $|A| = 3$ Bulatov'06
  - if $|A| = 4$ Marković et al.
  - if $\mathbb{A}$ contains all unary relations Bulatov'03, Barto'11
  - if $\mathbb{A} = (A; R)$ where R is binary, without sources or sinks Barto, Kozik, Niven'09
- Applicability of known algorithmic principles understood
  - Describing all solutions Idziak, Markovic, McKenzie, Valeriote, Willard'07
  - Local consistency (constraint propagation) Barto, Kozik'09, Bulatov

Results

- Better understanding of pre-algebraic results
- Far broader special cases solved. The dichotomy conjecture is true:
  - if $|A| = 3$ Bulatov'06
  - if $|A| = 4$ Marković et al.
  - if $\mathbb{A}$ contains all unary relations Bulatov'03, Barto'11
  - if $\mathbb{A} = (A; R)$ where R is binary, without sources or sinks Barto, Kozik, Niven'09
- Applicability of known algorithmic principles understood
  - Describing all solutions Idziak, Markovic, McKenzie, Valeriote, Willard'07
  - Local consistency (constraint propagation) Barto, Kozik'09, Bulatov
  - All known tractable cases solvable by a combination of these two

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)
- (k, l)-algorithm: Derive the strongest constraints on k variables which can be deduced by "considering" l variables at a time.

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)
- (k, l)-algorithm: Derive the strongest constraints on k variables which can be deduced by "considering" l variables at a time.
- If a contradiction is found, answer "no"
  otherwise answer "yes"

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)
- (k, l)-algorithm: Derive the strongest constraints on k variables which can be deduced by “considering” l variables at a time.
- If a contradiction is found, answer “no”
  otherwise answer “yes”
- “no” answers are always correct

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)
- (k, l)-algorithm: Derive the strongest constraints on k variables which can be deduced by "considering" l variables at a time.
- If a contradiction is found, answer "no"
  otherwise answer "yes"
- "no" answers are always correct
- if "yes" answers are correct for every instance of $\mathrm{CSP}(\mathbb{A})$ we say that $\mathbb{A}$ has width (k, l).

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)
- (k, l)-algorithm: Derive the strongest constraints on k variables which can be deduced by "considering" l variables at a time.
- If a contradiction is found, answer "no"
  otherwise answer "yes"
- "no" answers are always correct
- if "yes" answers are correct for every instance of $\mathrm{CSP}(\mathbb{A})$ we say that $\mathbb{A}$ has width (k, l).
- if $\mathbb{A}$ has width (k, l) for some k, l then $\mathbb{A}$ has bounded width

Local consistency

Roughly: $\mathbb{A}$ has bounded width iff $\mathrm{CSP}(\mathbb{A})$ can be solved by checking local consistency

More precisely:

- Fix $k \leq l$ (integers)
- (k, l)-algorithm: Derive the strongest constraints on k variables which can be deduced by "considering" l variables at a time.
- If a contradiction is found, answer "no"
  otherwise answer "yes"
- "no" answers are always correct
- if "yes" answers are correct for every instance of $\mathrm{CSP}(\mathbb{A})$ we say that $\mathbb{A}$ has width (k, l).
- if $\mathbb{A}$ has width (k, l) for some k, l then $\mathbb{A}$ has bounded width

Various equivalent formulations (bounded tree width duality, definability in Datalog)

Example of $(2,3)$-consistency

Let $\mathbb{A} = (\{0,1\}; x = y, x \neq y)$

Consider the instance

$$x = y,\ y = z,\ z = w,\ x \neq w$$

Example of $(2,3)$-consistency

Let $\mathbb{A} = (\{0,1\}; x = y, x \neq y)$

Consider the instance

$$x = y,\ y = z,\ z = w,\ x \neq w$$

- By looking at $\{x, y, z\}$ we see (using $x = y$ and $y = z$) that $x = z$.

Example of $(2,3)$-consistency

Let $\mathbb{A} = (\{0,1\}; x = y, x \neq y)$

Consider the instance

$$x = y,\ y = z,\ z = w,\ x \neq w$$

- By looking at $\{x, y, z\}$ we see (using $x = y$ and $y = z$) that $x = z$.
- By looking at $\{x, z, w\}$ we see (using $x = z$ and $z = w$) that $x = w$.

Example of $(2,3)$-consistency

Let $\mathbb{A} = (\{0,1\}; x = y, x \neq y)$

Consider the instance

$$x = y,\ y = z,\ z = w,\ x \neq w$$

- By looking at $\{x, y, z\}$ we see (using $x = y$ and $y = z$) that $x = z$.
- By looking at $\{x, z, w\}$ we see (using $x = z$ and $z = w$) that $x = w$.
- By looking at $\{x, w\}$ we now see a contradiction

Example of $(2,3)$-consistency

Let $\mathbb{A} = (\{0,1\}; x = y, x \neq y)$

Consider the instance

$$x = y,\ y = z,\ z = w,\ x \neq w$$

- By looking at $\{x, y, z\}$ we see (using $x = y$ and $y = z$) that $x = z$.
- By looking at $\{x, z, w\}$ we see (using $x = z$ and $z = w$) that $x = w$.
- By looking at $\{x, w\}$ we now see a contradiction

In fact, $\mathbb{A}$ has width $(2,3)$, that is, such reasoning is always sufficient for an instance of $\mathrm{CSP}(\mathbb{A})$.

Bounded width

- The problems *q-LIN* do not have bounded width
  Feder, Vardi‘93

Bounded width

- The problems *q-LIN* do not have bounded width
  Feder, Vardi'93
- If $\mathbb{A}$ can simulate *q-LIN* then $\mathbb{A}$ does not have bounded width
  Larose, Zádori'07

Bounded width

- The problems *q-LIN* do not have bounded width
  Feder, Vardi'93
- If $\mathbb{A}$ can simulate *q-LIN* then $\mathbb{A}$ does not have bounded width
  Larose, Zádori'07
- Thus the "obvious" necessary condition for bounded width is that $\mathbb{A}$ cannot simulate *q-LIN*.

Bounded width

- The problems *q-LIN* do not have bounded width
  Feder, Vardi'93
- If $\mathbb{A}$ can simulate *q-LIN* then $\mathbb{A}$ does not have bounded width
  Larose, Zádori'07
- Thus the "obvious" necessary condition for bounded width is that $\mathbb{A}$ cannot simulate *q-LIN*.
- It is sufficient:

Bounded width

- The problems *q-LIN* do not have bounded width
  Feder, Vardi'93
- If $\mathbb{A}$ can simulate *q-LIN* then $\mathbb{A}$ does not have bounded width
  Larose, Zádori'07
- Thus the "obvious" necessary condition for bounded width is that $\mathbb{A}$ cannot simulate *q-LIN*.
- It is sufficient:

Theorem

The following are equivalent.

1. $\mathbb{A}$ *cannot simulate q-LIN*
2. $\mathbb{A}$ *has bounded width* *B, Kozik'09*
3. $\mathbb{A}$ *has width* $(2, 3)$ *B; Bulatov*

Bonus: Robust approximation

- **Task:** Find an almost satisfying assignment given an almost satisfiable instance

Bonus: Robust approximation

- **Task:** Find an almost satisfying assignment given an almost satisfiable instance
- More precisely: Find an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of the constraints given an instance which is $(1 - \varepsilon)$ satisfiable, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$ (g should only depend on $\mathbb{A}$).

Bonus: Robust approximation

- **Task:** Find an almost satisfying assignment given an almost satisfiable instance
- More precisely: Find an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of the constraints given an instance which is $(1 - \varepsilon)$ satisfiable, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$ (g should only depend on $\mathbb{A}$).
- Algorithms for 2-*SAT* and *HORN-SAT* based on linear programming and semidefinite programming Zwick'98

Bonus: Robust approximation

- **Task:** Find an almost satisfying assignment given an almost satisfiable instance
- More precisely: Find an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of the constraints given an instance which is $(1 - \varepsilon)$ satisfiable, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$ (g should only depend on $\mathbb{A}$).
- Algorithms for 2-*SAT* and *HORN-SAT* based on linear programming and semidefinite programming Zwick'98
- *q-LIN* has no robust polynomial algorithm (assuming P $\neq$ NP) Hastad'01

Bonus: Robust approximation

- **Task:** Find an almost satisfying assignment given an almost satisfiable instance
- More precisely: Find an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of the constraints given an instance which is $(1 - \varepsilon)$ satisfiable, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$ (g should only depend on $\mathbb{A}$).
- Algorithms for 2-*SAT* and *HORN-SAT* based on linear programming and semidefinite programming Zwick'98
- *q-LIN* has no robust polynomial algorithm (assuming P $\neq$ NP) Hastad'01
- If $\mathbb{A}$ can simulate *q-LIN* then $\mathrm{CSP}(\mathbb{A})$ has no robust algorithm Dalmau, Krokhin'11

Bonus: Robust approximation 2

- If $\mathbb{A}$ can simulate *q-LIN* then $\mathrm{CSP}(\mathbb{A})$ has no robust algorithm Dalmau, Krokhin'11

Bonus: Robust approximation 2

- If $\mathbb{A}$ can simulate *q-LIN* then $\mathrm{CSP}(\mathbb{A})$ has no robust algorithm Dalmau, Krokhin'11
- Conjecture of Guruswami and Zhou: this is the only obstacle

Bonus: Robust approximation 2

- If $\mathbb{A}$ can simulate *q-LIN* then $\mathrm{CSP}(\mathbb{A})$ has no robust algorithm Dalmau, Krokhin'11
- Conjecture of Guruswami and Zhou: this is the only obstacle

Theorem (B, Kozik'12)

The following are equivalent (assuming $P \neq NP$)

- $\mathbb{A}$ *cannot simulate q-LIN*
- $\mathrm{CSP}(\mathbb{A})$ *has a robust polynomial algorithm*
- *canonical semidefinite programming relaxation correctly decides* $\mathrm{CSP}(\mathbb{A})$

Bonus 2: Counting CSP

- The complexity is also controlled by $\mathrm{Pol}(\mathbb{A})$

Bonus 2: Counting CSP

- The complexity is also controlled by $\mathrm{Pol}(\mathbb{A})$
- A necessary condition for tractability found
  Bulatov, Dalmau'03
  (inspiration: the other algorithm for decison CSPs)

Bonus 2: Counting CSP

- The complexity is also controlled by Pol($\mathbb{A}$)
- A necessary condition for tractability found
  Bulatov, Dalmau'03
  (inspiration: the other algorithm for decison CSPs)
- A stronger necessary condition for tractability found
  Bulatov, Grohe'05

Bonus 2: Counting CSP

- The complexity is also controlled by Pol($\mathbb{A}$)
- A necessary condition for tractability found
  Bulatov, Dalmau'03
  (inspiration: the other algorithm for decison CSPs)
- A stronger necessary condition for tractability found
  Bulatov, Grohe'05
- The stronger condition is sufficient
  Bulatov'08, Dyer and Richerby'10

Final remarks

For decision CSPs

- Easy criterion for hardness
- Theory gives generic reduction between any two NP-complete CSPs (instead of ad hoc reductions)
- Applicability of known algorithms understood
- The dichotomy conjecture still open in general

Final remarks

For decision CSPs

- Easy criterion for hardness
- Theory gives generic reduction between any two NP-complete CSPs (instead of ad hoc reductions)
- Applicability of known algorithms understood
- The dichotomy conjecture still open in general

For other variants (Approx-CSP, Valued CSP, infinite)

- Universal algebra also relevant Cohen, Cooper, Creed, Jeavons, Živný; Raghavendra; Bodirsky, Pinsker
- More or less the same criterion for easiness/hardness
- Easiness comes from "symmetry"
- One needs symmetry of higher arity (e.g. polymorphisms) rather than just automorphisms or endomorphisms

Final remarks

For decision CSPs

- Easy criterion for hardness
- Theory gives generic reduction between any two NP-complete CSPs (instead of ad hoc reductions)
- Applicability of known algorithms understood
- The dichotomy conjecture still open in general

For other variants (Approx-CSP, Valued CSP, infinite)

- Universal algebra also relevant Cohen, Cooper, Creed, Jeavons, Živný; Raghavendra; Bodirsky, Pinsker
- More or less the same criterion for easiness/hardness
- Easiness comes from "symmetry"
- One needs symmetry of higher arity (e.g. polymorphisms) rather than just automorphisms or endomorphisms

Beyond CSPs

- ???
- There is ≥ 1 examples Raghavendra

We need coffee!

Thank you!