Reduction of CSP Dichotomy to Digraphs

CP2013 Uppsala.

Marcel Jackson LA TROBE UNIVERSITY

Jakub Bulín (Charles U), Dejan Delić (Ryerson U),
Todd Niven (La Trobe U and Monash U).

Outline

Fixed template CSPs

- Constraint language: $\Gamma = (D; R_1, \ldots, R_m)$
  - a fixed *domain* with a list of *relations*

Fixed template CSPs

- Constraint language: $\Gamma = (D; R_1, \ldots, R_m)$
  - a fixed *domain* with a list of *relations*
- *Usually D is finite. We will require the relation list to be finite.*

Fixed template CSPs

- Constraint language: $\Gamma = (D; R_1, \ldots, R_m)$
  - a fixed *domain* with a list of *relations*
- Instance of $CSP(\Gamma)$:

Fixed template CSPs

- Constraint language: $\Gamma = (D; R_1, \ldots, R_m)$
  - a fixed *domain* with a list of *relations*
- Instance of $CSP(\Gamma)$:
  - a finite set of variables V and
  - a finite set of constraints $\mathcal{C}$. Example constraint: $(v_1, \ldots, v_k) \in R_i$, for some i

Fixed template CSPs

- Constraint language: $\Gamma = (D; R_1, \ldots, R_m)$
  - a fixed *domain* with a list of *relations*
- Instance of $CSP(\Gamma)$:
  - a finite set of variables V and
  - a finite set of constraints $\mathcal{C}$. Example constraint: $(v_1, \ldots, v_k) \in R_i$, for some i
- Solution: an interpretation $\phi : V \to D$ such that $(\phi(v_1), \ldots, \phi(v_k)) \in R_i$ for each constraint

Fixed template CSPs

- Constraint language: $\Gamma = (D; R_1, \ldots, R_m)$
  - a fixed *domain* with a list of *relations*
- Instance of $CSP(\Gamma)$:
  - a finite set of variables V and
  - a finite set of constraints $\mathcal{C}$. Example constraint: $(v_1, \ldots, v_k) \in R_i$, for some i
- Solution: an interpretation $\phi : V \to D$ such that $(\phi(v_1), \ldots, \phi(v_k)) \in R_i$ for each constraint

Example

A directed graph $\Gamma = (D; \to)$, with $\to$ binary

Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:

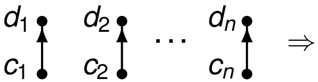
Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:

$$\begin{pmatrix} d_{1,1} \\ \vdots \\ d_{k,1} \end{pmatrix}, \begin{pmatrix} d_{1,2} \\ \vdots \\ d_{k,2} \end{pmatrix}, \ldots, \begin{pmatrix} d_{1,n} \\ \vdots \\ d_{k,n} \end{pmatrix} \in R \; \Rightarrow \begin{pmatrix} f(d_{1,1}, \ldots, d_{1,n}) \\ \vdots \\ f(d_{k,1}, \ldots, d_{k,n}) \end{pmatrix} \in R$$

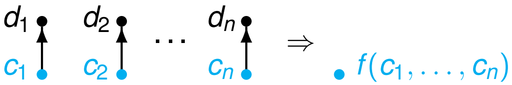
Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:



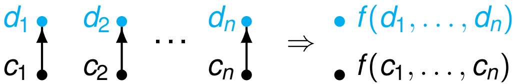
Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:



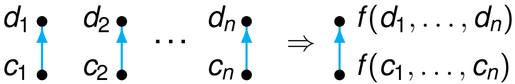
Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:



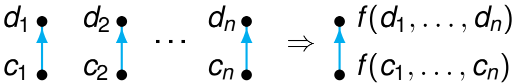
Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:



Polymorphism of $\Gamma = (D; R_1, \ldots, R_m)$

A function $f : D^n \to D$ preserving each R_i pointwise:



Example

- “automorphism” = “bijective unary polymorphism”
- “endomorphism” = “unary polymorphism”

Polymorphisms and complexity

- Polymorphisms determine computational complexity
  - Jeavons, Cohen, Gyssens (1997)

Polymorphisms and complexity

- Polymorphisms determine computational complexity
  - Jeavons, Cohen, Gyssens (1997)
- Complexity theoretic issues are determined by the "equational theory" of the polymorphisms.
  - Bulatov, Jeavons, Krokhin (2001)

Polymorphisms and complexity

- Polymorphisms determine computational complexity
  - Jeavons, Cohen, Gyssens (1997)
- Complexity theoretic issues are determined by the "equational theory" of the polymorphisms.
  - Bulatov, Jeavons, Krokhin (2001)

Problem	Polymorphism Algebra
HORNSAT	$\begin{array}{c\|cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$

The Dichotomy Conjecture

Feder & Vardi 1994

Conjecture: For all Γ, $CSP(\Gamma)$ is either in $\mathtt{P}$ or is $\mathtt{NP}$-complete

The Dichotomy Conjecture

Feder & Vardi 1994

Conjecture: For all Γ, $CSP(\Gamma)$ is either in $\mathtt{P}$ or is $\mathtt{NP}$-complete

- Polymorphisms used to facilitate algorithmic solution

The Dichotomy Conjecture

Feder & Vardi 1994

Conjecture: For all Γ, *CSP*(Γ) is either in P or is NP-complete

- Polymorphisms used to facilitate algorithmic solution
- Furthermoreover, *CSP*(Γ) is P-time equivalent to one over a digraph

The Dichotomy Conjecture

Feder & Vardi 1994

Conjecture: For all Γ, $CSP(\Gamma)$ is either in $\mathtt{P}$ or is $\mathtt{NP}$-complete

- Polymorphisms used to facilitate algorithmic solution
- Furthermoreover, $CSP(\Gamma)$ is $\mathtt{P}$-time equivalent to one over a digraph

Algebraic conjectures

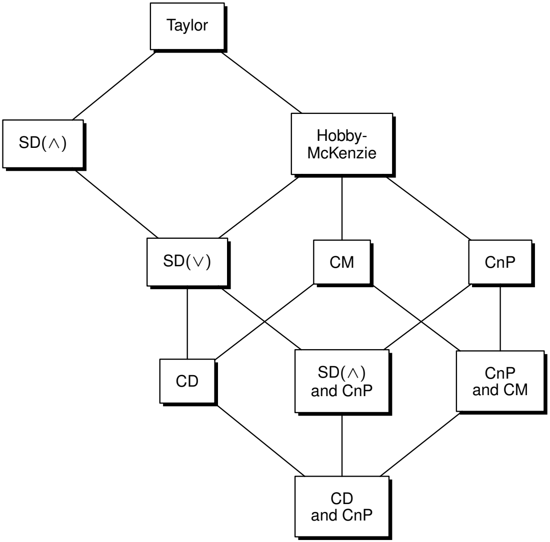
- Algebraic dichotomy conjecture: a characterisation of tractable CSPs
  - Bulatov, Jeavons, Krokhin
- conjectured/known polymorphism classifications of $\mathtt{NL}$, $\mathtt{L}$, and amenability to specific kinds of algorithms

Outline

Main Result

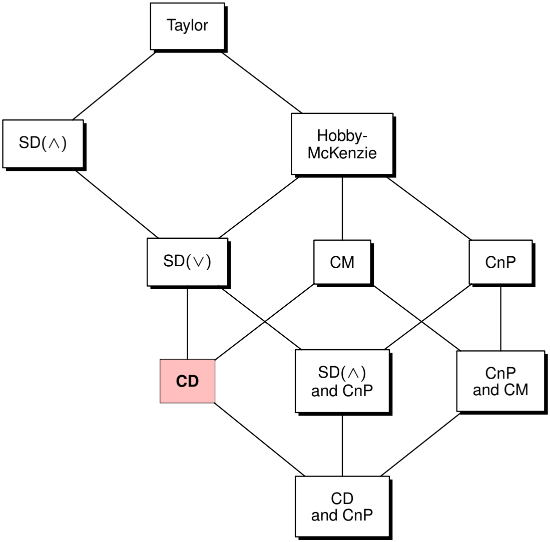
Every *CSP*(Γ) is *logspace* equivalent to a CSP over a balanced directed graph with (almost) the same polymorphism properties

- future investigations can be restricted to the special case of digraphs with no loss of generality



Polymorphism landscape

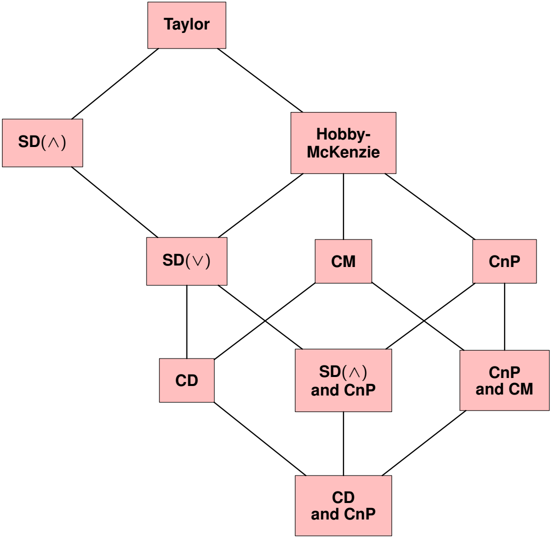
Each node depicts an algebraic property relevant to CSPs



Polymorphism landscape

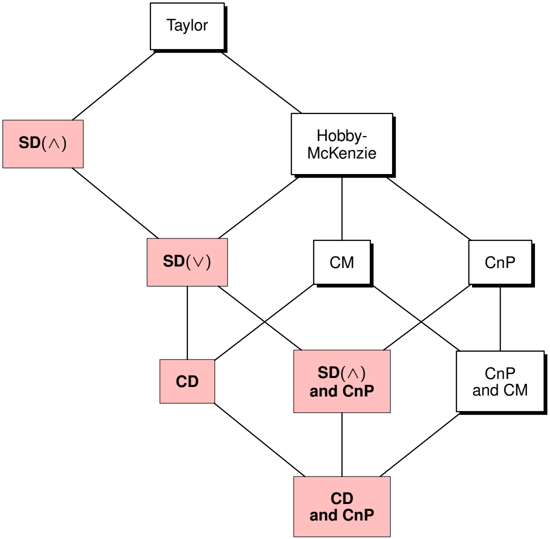
Each node depicts an algebraic property relevant to CSPs

$$x = \nu(y, x, x, \dots, x) = \nu(x, y, x, \dots, x) = \dots = \nu(x, x, \dots, x, y)$$



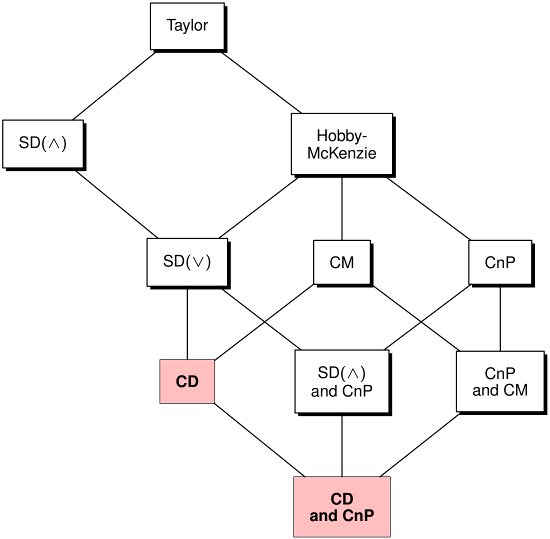
Polymorphism landscape

Algebraic dichotomy conjecture: $CSP(\Gamma)$ is tractable if it satisfies "Taylor" and is NP-complete otherwise



Polymorphism landscape

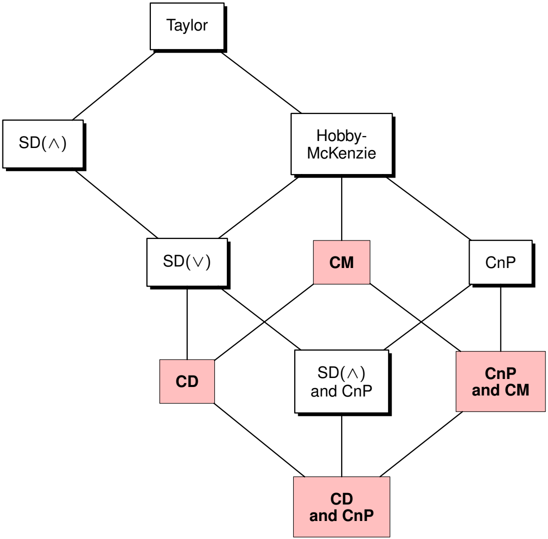
CSPs solvable by local consistency check (Barto and Kozik).



Polymorphism landscape

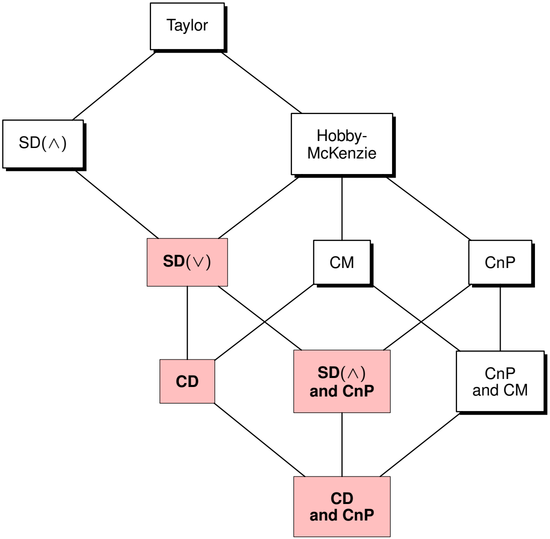
Strict width.

(Many authors)



Polymorphism landscape

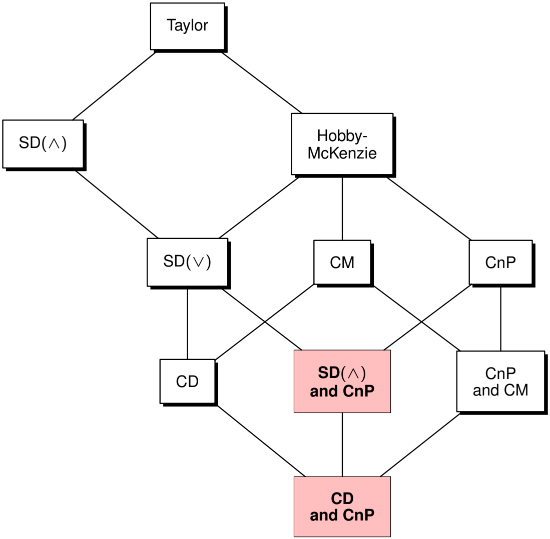
CSPs solvable by generalised Gaussian elimination algorithm (Barto + Berman, Idziak, Marković, McKenzie, Valeriote, Willard)



Polymorphism landscape

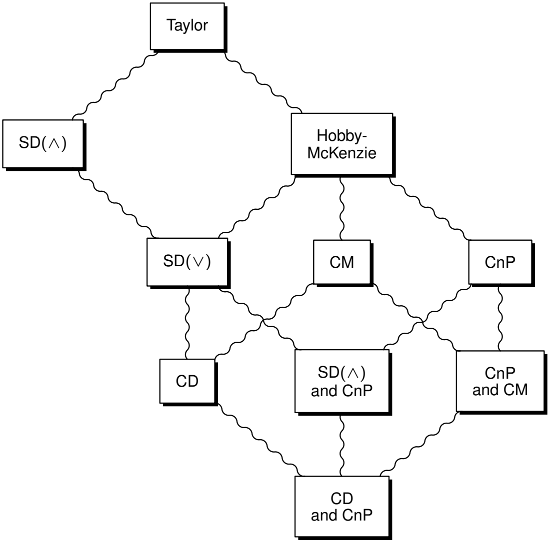
Algebraic $\mathtt{NL}$ conjecture:

Solvability in nondeterministic logspace??



Polymorphism landscape

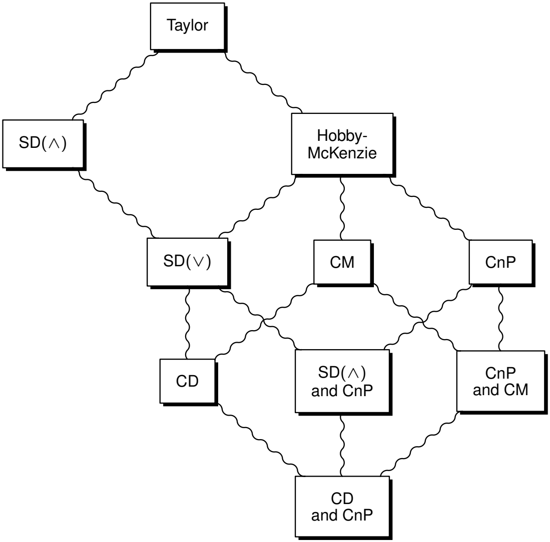
Algebraic L conjecture: Solvability in logspace??



Polymorphism landscape

Corollary of main result:

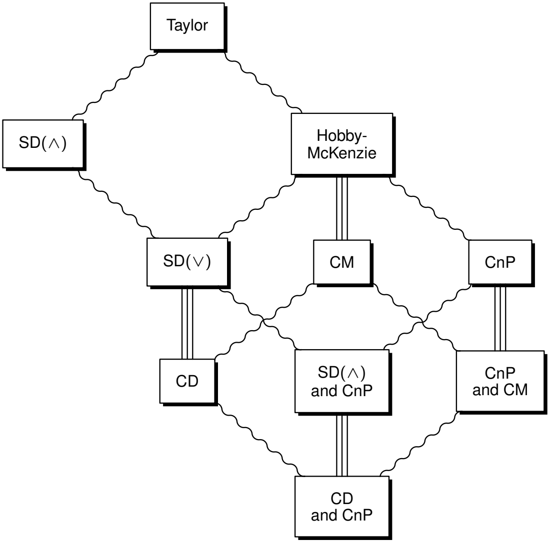
- *All regions can be distinguished using digraph CSPs*



Polymorphism landscape

Corollary of main result:

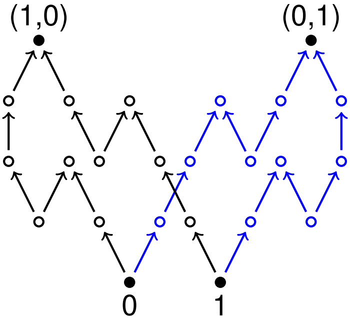
- *All regions can be distinguished using digraph CSPs*
- *The conjectures can be resolved by considering digraph CSPs only*



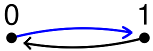
Polymorphism landscape

Note:
It is possible to find other logspace translations that obliterate some of these distinctions

Outline



(*Above.*) *The digraph obtained by applying the construction to the following two element cyclic digraph*



What *can't* be done

- The landscape of list homomorphism problems over digraphs is substantially restricted (Hell, Rafiey).
- "Logspace equivalent" cannot naturally be replaced by "first order equivalent", at least for *balanced* digraphs.

What *can't* be done

- The landscape of list homomorphism problems over digraphs is substantially restricted (Hell, Rafiey).
- "Logspace equivalent" cannot naturally be replaced by "first order equivalent", at least for *balanced* digraphs.

Extensions

- The main result *does* neverthe-more-or-less apply to infinite domain CSPs (with finitely many relations).
- Approximation CSPs? Counting CSPs? Surjective CSPs? . . . CSPs?