On the computational complexity of non-dictatorial aggregation

L. Kirousis[1] P. Kolaitis[2] J. Livieratos[1]

[1]Department of Mathematics
National and Kapodistrian University of Athens

[2]Computer Science Department
UC Santa Cruz and IBM Research - Almaden

RAMiCS, 2018

Outline

1. Introduction: Preference Aggregation
2. Preliminaries
   - Framework
   - Statement of problems
   - Characterizations (previous results)
3. Algorithms (new results)
   - Universal Algebra
   - Possibility Domains
   - Uniform Possibility Domains

Preference Aggregation

- $\mathcal{O} = \{o_1, \ldots, o_k\}$, $k \geq 3$.
- $n \geq 2$ individuals,
- $<_i$: ranking of i-th agent [strict total ordering of $\mathcal{O}$].
- X set of strict total orderings of $\mathcal{O}$.

▶ Social Welfare Function $SWF : X^n \mapsto X$.

Preference Aggregation

- $\mathcal{O} = \{o_1, \ldots, o_k\}$, $k \geq 3$.
- $n \geq 2$ individuals,
- $<_i$: ranking of i-th agent [strict total ordering of $\mathcal{O}$].
- X set of strict total orderings of $\mathcal{O}$.

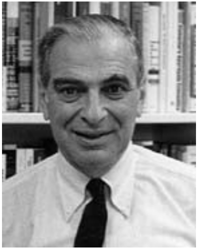
▶ Social Welfare Function $SWF : X^n \mapsto X$.

▶ J. K. Arrow (1951): Showed that non-trivial preference aggregation is impossible given only some mild axioms that a SWF should satisfy.

John Kenneth Arrow, 1921-2017

The conditions are:

1. Universality (unrestricted domain):
   $SWF : X^n \mapsto X$,
2. Independency (independency of irrelevant alternatives): ranking o_i, o_j does not depend on the ranking of any other o_l.
3. Pareto efficiency (unanimity): if everyone ranks o_i higher than o_j, o_i should be ranked higher by the SWF.

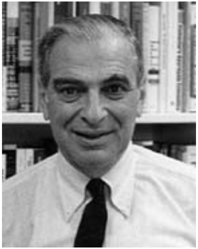


J. K. Arrow

John Kenneth Arrow, 1921-2017

The conditions are:

1. Universality (unrestricted domain): $SWF : X^n \mapsto X$,
2. Independency (independency of irrelevant alternatives): ranking o_i, o_j does not depend on the ranking of any other o_l.
3. Pareto efficiency (unanimity): if everyone ranks o_i higher than o_j, o_i should be ranked higher by the SWF.



J. K. Arrow

And then we have:

Theorem (General Possibility, 1951)

Any SWF satisfying 1, 2, 3 is a dictatorship: the outcome is the ranking of a single individual.

Outline

Abstract Domains

- n voters, m issues.
- A_j set of possible positions for j-th issue, $|A_j| \geq 2$.
- $X \subseteq \prod_{j=1}^{m} A_j$ set of feasible voting patterns.
- $X_j = A_j$.

Abstract Domains

- n voters, m issues.
- A_j set of possible positions for j-th issue, $|A_j| \geq 2$.
  - Boolean framework: $|A_j| = 2$ for all $j = 1, \ldots, m$.
- $X \subseteq \prod_{j=1}^{m} A_j$ set of feasible voting patterns.
- $X_j = A_j$.

Abstract Domains

- n voters, m issues.
- A_j set of possible positions for j-th issue, $|A_j| \geq 2$.
  - Boolean framework: $|A_j| = 2$ for all $j = 1, \ldots, m$.
- $X \subseteq \prod_{j=1}^{m} A_j$ set of feasible voting patterns.
- $X_j = A_j$.

Non-degeneracy conditions

Aggregators

$\bar{f} = (f_1, \ldots, f_m) : X^n \mapsto X$ m-tuple of n-ary functions $f_j : X_j^n \mapsto X_j$:

Aggregators

$\bar{f} = (f_1, \ldots, f_m) : X^n \mapsto X$ m-tuple of n-ary functions $f_j : X_j^n \mapsto X_j$:

1. Universality and IIA are "built in".
2. Conservativeness (supportiveness):

$$\text{if } x_j = (x_j^1, ..., x_j^n) \in A_j^n \text{ then } f_j(x_j) \in \{x_j^1, \ldots, x_j^n\},$$

for all $j = 1, \ldots, m$.

Aggregators

$\bar{f} = (f_1, \dots, f_m) : X^n \mapsto X$ m-tuple of n-ary functions $f_j : X_j^n \mapsto X_j$:

1. Universality and IIA are "built in".
2. Conservativeness (supportiveness):

$$\text{if } x_j = (x_j^1, ..., x_j^n) \in A_j^n \text{ then } f_j(x_j) \in \{x_j^1, \dots, x_j^n\},$$

   for all $j = 1, \dots, m$.

   - Stronger notion than unanimity (equivalent in the Boolean framework).
   - The social outcome for each issue must be equal to at least one individual's position.

Aggregators cont.

$$\bar{f} \begin{pmatrix} x_1^1 & \cdots & x_j^1 & \cdots & x_m^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \cdots & x_j^i & \cdots & x_m^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & \cdots & x_j^n & \cdots & x_m^n \end{pmatrix}$$

Aggregators cont.

$$\begin{pmatrix} \color{red}{f_1} & & & & \\ \color{red}{\frown} & & & & \\ x_1^1 & \cdots & x_j^1 & \cdots & x_m^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \cdots & x_j^i & \cdots & x_m^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & \cdots & x_j^n & \cdots & x_m^n \\ \color{red}{\smile} & & & & \\ \shortparallel & & & & \\ f_1(x_1) & & & & \end{pmatrix}$$

Aggregators cont.

$$\begin{pmatrix} \color{red}{f_1} & \cdots & \color{red}{f_j} & \cdots & \color{red}{f_m} \\ \color{red}{\frown} & & \color{red}{\frown} & & \color{red}{\frown} \\ x_1^1 & \cdots & x_j^1 & \cdots & x_m^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \cdots & x_j^i & \cdots & x_m^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & \cdots & x_j^n & \cdots & x_m^n \\ \color{red}{\smile} & & \color{red}{\smile} & & \color{red}{\smile} \\ \shortparallel & & \shortparallel & & \shortparallel \\ f_1(x_1) & \cdots & f_j(x_j) & \cdots & f_m(x_m) \end{pmatrix}$$

Aggregators cont.

$$\begin{pmatrix} f_1 & \cdots & f_j & \cdots & f_m \\ \frown & & \frown & & \frown \\ x_1^1 & \cdots & x_j^1 & \cdots & x_m^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \cdots & x_j^i & \cdots & x_m^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & \cdots & x_j^n & \cdots & x_m^n \\ \smile & & \smile & & \smile \\ \shortparallel & & \shortparallel & & \shortparallel \\ (f_1(x_1), & \ldots, & f_j(x_j), & \ldots, & f_m(x_m)) \end{pmatrix}$$

Aggregators cont.

$$\begin{pmatrix} \color{red}{f_1} & \cdots & \color{red}{f_j} & \cdots & \color{red}{f_m} \\ \color{red}{\frown} & & \color{red}{\frown} & & \color{red}{\frown} \\ x_1^1 & \cdots & x_j^1 & \cdots & x_m^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \cdots & x_j^i & \cdots & x_m^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & \cdots & x_j^n & \cdots & x_m^n \\ \color{red}{\smile} & & \color{red}{\smile} & & \color{red}{\smile} \\ \shortparallel & & \shortparallel & & \shortparallel \\ \color{red}{(f_1(x_1),} & \color{red}{\ldots,} & \color{red}{f_j(x_j),} & \color{red}{\ldots,} & \color{red}{f_m(x_m))} \end{pmatrix}$$

$$= \bar{f}(x^1, \ldots, x^n) \in X.$$

Majority/minority aggregators

- $\bar{f} = (f_1, \ldots, f_m)$ is a majority/minority aggregator if for all j, $f_j(x, x, y) = f_j(x, y, x) = f_j(y, x, x) = x \mid y$.
- X admits a majority aggregator if and only if it admits a ternary $\bar{f} = (f_1, \ldots, f_m)$ such that, for all j, B_j (binary) $f_j \restriction_{B_j} = maj : \{0,1\}^3 \mapsto \{0,1\}$ s.t.

$$\mathrm{maj}(x, y, z) = \begin{cases} x & \text{if } x = y \text{ or } x = z, \\ y & \text{if } y = z. \end{cases}$$

- Accordingly, it admits a minority aggregetor if and only if it admits a ternary $\bar{f} = (f_1, \ldots, f_m)$ such that, for all j, B_j (binary) $f_j \restriction_{B_j} = \oplus : \{0,1\}^3 \mapsto \{0,1\}$, where $\oplus(x, y, z) = x \oplus y \oplus z$ is the binary addition modulo 2.

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, \ldots, n\}$ such that: $(f_1, \ldots, f_m) = (pr_d^n, \ldots, pr_d^n)$.

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, ..., n\}$ such that: $(f_1, ..., f_m) = (pr_d^n, ..., pr_d^n)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \ldots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, \ldots, n\}$ such that: $(f_1, \ldots, f_m) = (pr_d^n, \ldots, pr_d^n)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \ldots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Definition

X is a possibility domain if, for some $n \in \mathbb{N}$, there exists a n-ary non-dictatorial aggregator for X.

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, \ldots, n\}$ such that: $(f_1, \ldots, f_m) = (pr_d^n, \ldots, pr_d^n)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \ldots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Definition

X is a possibility domain if, for some $n \in \mathbb{N}$, there exists a n-ary non-dictatorial aggregator for X.

Example: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, ..., n\}$ such that: $(f_1, ..., f_m) = (pr_d^n, ..., pr_d^n)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \ldots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Definition

X is a possibility domain if, for some $n \in \mathbb{N}$, there exists a n-ary non-dictatorial aggregator for X.

Example: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- $(0, 1', \underline{1}, \hat{0}) \in X$.

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, \dots, n\}$ such that: $(f_1, \dots, f_m) = (pr^n_d, \dots, pr^n_d)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \dots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Definition

X is a possibility domain if, for some $n \in \mathbb{N}$, there exists a n-ary non-dictatorial aggregator for X.

Example: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- $(0, 1', \underline{1}, \hat{0}) \in X$.
- Non-dictatorial aggregators: $(pr^n_d, pr^n_d, pr^n_{d'}, pr^n_{d'})$

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, \ldots, n\}$ such that: $(f_1, \ldots, f_m) = (pr^n_d, \ldots, pr^n_d)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \ldots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Definition

X is a possibility domain if, for some $n \in \mathbb{N}$, there exists a n-ary non-dictatorial aggregator for X.

Example: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- $(0, 1', \underline{1}, \hat{0}) \in X$.
- Non-dictatorial aggregators: $(pr^n_d, pr^n_d, pr^n_{d'}, pr^n_{d'})$, $(\wedge, \wedge, pr^2_d, pr^2_d)$,$(pr^2_d, pr^2_d, \vee, \vee)$,

Possibility Domains

Definition

An n-ary aggregator $\bar{f}$ is called dictatorial for X if there is a $d \in \{1, \dots, n\}$ such that: $(f_1, \dots, f_m) = (pr_d^n, \dots, pr_d^n)$.

Example (non-dictatorial aggregators): Majority and minority aggregators, binary $\bar{f} = (f_1, \dots, f_m)$ s.t. there exists a binary $B_j \subseteq A_j$: $f_j \restriction_{B_j} = \wedge, \vee$.

Definition

X is a possibility domain if, for some $n \in \mathbb{N}$, there exists a n-ary non-dictatorial aggregator for X.

Example: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- $(0, 1', \underline{1}, \hat{0}) \in X$.
- Non-dictatorial aggregators: $(pr_d^n, pr_d^n, pr_{d'}^n, pr_{d'}^n)$, $(\wedge, \wedge, pr_d^2, pr_d^2)$,$(pr_d^2, pr_d^2, \vee, \vee)$,$(\wedge, \wedge, \vee, \vee)$,...

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components $(\wedge, \vee)$, when restricted to binary $B_j \subseteq A_j$.

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components ($\wedge, \vee$), when restricted to binary $B_j \subseteq A_j$.

Definition

X is a uniform possibility domain if it admits a uniform non-dict. aggr.

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components ($\wedge, \vee$), when restricted to binary $B_j \subseteq A_j$.

Definition

X is a uniform possibility domain if it admits a uniform non-dict. aggr.

Examples: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components $(\wedge, \vee)$, when restricted to binary $B_j \subseteq A_j$.

Definition

X is a uniform possibility domain if it admits a uniform non-dict. aggr.

Examples: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- Uniform non-dictatorial aggregator: $(\wedge, \wedge, \vee, \vee)$.

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \upharpoonright_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components $(\wedge, \vee)$, when restricted to binary $B_j \subseteq A_j$.

Definition

X is a uniform possibility domain if it admits a uniform non-dict. aggr.

Examples: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- Uniform non-dictatorial aggregator: $(\wedge, \wedge, \vee, \vee)$.

$X = \{(0, 0', \underline{1}), (0, 1', \underline{0}), (1, 0', \underline{0})\} \times \{\hat{0}, \hat{1}\}$.

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components $(\wedge, \vee)$, when restricted to binary $B_j \subseteq A_j$.

Definition

X is a uniform possibility domain if it admits a uniform non-dict. aggr.

Examples: $X = \{(0, 1'), (1, 0'), (0, 0')\} \times \{(\underline{0}, \hat{1}), (\underline{1}, \hat{0}), (\underline{1}, \hat{1})\}$.

- Uniform non-dictatorial aggregator: $(\wedge, \wedge, \vee, \vee)$.

$X = \{(0, 0', \underline{1}), (0, 1', \underline{0}), (1, 0', \underline{0})\} \times \{\hat{0}, \hat{1}\}$.

- Non-dictatorial aggregators: $(pr_d^n, pr_d^n, pr_d^n, pr_{d'}^n)$, $(pr_d^3, pr_d^3, pr_d^3, maj), \ldots$

Uniform Possibility Domains

Definition (Kirousis, Kolaitis, Livieratos 2017)

$\bar{f}$ is uniform non-dictatorial if for all j, B_j (binary) $f_j \restriction_{B_j}$ is not a projection.

★ Boolean framework: locally non-dictatorial (Nehring, Puppe 2010).

Examples: Majority/minority aggregators and binary aggregators with symmetric components $(\wedge, \vee)$, when restricted to binary $B_j \subseteq A_j$.

Definition

X is a uniform possibility domain if it admits a uniform non-dict. aggr.

Examples: $X = \{(0,1'),(1,0'),(0,0')\} \times \{(\underline{0},\hat{1}),(\underline{1},\hat{0}),(\underline{1},\hat{1})\}$.

- Uniform non-dictatorial aggregator: $(\wedge, \wedge, \vee, \vee)$.

$X = \{(0,0',\underline{1}),(0,1',\underline{0}),(1,0',\underline{0})\} \times \{\hat{0},\hat{1}\}$.

- Non-dictatorial aggregators: $(pr_d^n, pr_d^n, pr_d^n, pr_{d'}^n)$, $(pr_d^3, pr_d^3, pr_d^3, maj), \ldots$ **but not** any uniform ones.

Outline

Problems

Possibility domains

Can we "efficiently" decide if a domain X is a possibility domain?

Uniform Possibility domain

Can we "efficiently" decide if a domain X is a uniform possibility domain?

Problems

Possibility domains

Can we "efficiently" decide if a domain X is a possibility domain?

Uniform Possibility domain

Can we "efficiently" decide if a domain X is a uniform possibility domain?

Problems

Possibility domains

Can we "efficiently" decide if a domain X is a possibility domain?

Uniform Possibility domain

Can we "efficiently" decide if a domain X is a uniform possibility domain?

- Is there an algorithm that on input X, outputs the answer to the above questions, in polynomial time to the number of issues m and the size $|X|$ of X?
- In case X is a (uniform) possibility domain, can it also output an aggregator that witnesses this?

Problems

Possibility domains

Can we "efficiently" decide if a domain X is a possibility domain?

Uniform Possibility domain

Can we "efficiently" decide if a domain X is a uniform possibility domain?

- Is there an algorithm that on input X, outputs the answer to the above questions, in polynomial time to the number of issues m and the size $|X|$ of X?
- In case X is a (uniform) possibility domain, can it also output an aggregator that witnesses this?

An aggregator of any arity n will do! The search space it too big!

Outline

Avoiding dictators

Theorem (KKL, 2017)

X is a possibility domain if and only if X admits either:

- *a majority aggregator or*
- *a minority aggregator or*
- *a non-dictatorial binary aggregator.*

Avoiding dictators

Theorem (KKL, 2017)

X is a possibility domain if and only if X admits either:

- *a majority aggregator or*
- *a minority aggregator or*
- *a non-dictatorial binary aggregator.*

★ Characterization for both the Boolean and non-Boolean framework.

★ Provides explicit information about the nature of the components f_j.

Avoiding dictators

Theorem (KKL, 2017)

X is a possibility domain if and only if X admits either:

- *a majority aggregator or*
- *a minority aggregator or*
- *a non-dictatorial binary aggregator.*

★ Characterization for both the Boolean and non-Boolean framework.

★ Provides explicit information about the nature of the components f_j.

Theorem (KKL, 2017)

The following are equivalent:

- *X is a uniform possibility domain.*
- *For every $j, B_j \subseteq A_j$ (binary), either there is binary aggregator $\bar{f}$ s.t. $f_j{\restriction}_{B_j} \in \{\wedge, \vee\}$ or a ternary $\bar{f}$ s.t. $f_j{\restriction}_{B_j} \in \{\mathrm{maj}, \oplus\}$.*

Outline

Polymorphisms

- A finite set.
- $R \subseteq A^m$ m-ary relation.
- $f : A^n \mapsto A$ n-ary polymorphism of R if:

$$x^1, \ldots, x^n \in R \Rightarrow f(x^1, \ldots, x^n)$$
$$:= (f(x_1^1, \ldots, x_1^n), \ldots, f(x_m^1, \ldots, x_m^n)) \in R.$$

Polymorphisms

- A finite set.
- $R \subseteq A^m$ m-ary relation.
- $f : A^n \mapsto A$ n-ary polymorphism of R if:

$$x^1, \ldots, x^n \in R \Rightarrow f(x^1, \ldots, x^n)$$
$$:= (f(x_1^1, \ldots, x_1^n), \ldots, f(x_m^1, \ldots, x_m^n)) \in R.$$

▶ We will assume all polymorphisms are conservative/supportive (their output is part of the input).

Polymorphisms as aggregators

In order to use polymorphisms in our framework:

- Mark the elements of each A_j in a different way (color, number, etc.).
- Pass from X to $\tilde{X} \subseteq \bigcup_{j=1}^{m} A_j$.

Polymorphisms as aggregators

In order to use polymorphisms in our framework:

- Mark the elements of each A_j in a different way (color, number, etc.).
- Pass from X to $\tilde{X} \subseteq \bigcup_{j=1}^{m} A_j$.
- ★ Any aggregator for X can be translated to a polymorphism of $\tilde{X}$ and vice versa.

Outline

Theorem 1

Theorem (KKL, 2018)

There is a polynomial-time algorithm that decides if a domain X is a possibility domain, and if it is, that produces a non-dictatorial aggregator of arity at most three.

★ We assume that X is extensively part of the input.

Proof Outline

1. There are known algorithms to check if $\tilde{X}$ has a majority or a minority polymorphism (Bressiére & Carbonnel 2013, 2016).

Proof Outline

1. There are known algorithms to check if $\tilde{X}$ has a majority or a minority polymorphism (Bressiére & Carbonnel 2013, 2016).
2. For the binary aggregators, define H_X with vertices uu'_j, where $u, u' \in X_j$ s.t. $u \neq u'$ and edges $uu'_k \to vv'_l$, where:
   - $k \neq l$,
   - there are $z, z' \in X$ extending $(u, v), (u', v')$ and
   - there is no $w \in X$ extending (u, v') s.t. $w_j \in \{z_j, z'_j\}$.

Proof Outline

1. There are known algorithms to check if $\tilde{X}$ has a majority or a minority polymorphism (Bressiére & Carbonnel 2013, 2016).
2. For the binary aggregators, define H_X with vertices uu'_j, where $u, u' \in X_j$ s.t. $u \neq u'$ and edges $uu'_k \to vv'_l$, where:
   - $k \neq l$,
   - there are $z, z' \in X$ extending $(u, v), (u', v')$ and
   - there is no $w \in X$ extending (u, v') s.t. $w_j \in \{z_j, z'_j\}$.

★ Intuitively, $uu'_k \to vv'_l$ means that, for any binary aggregator $\bar{f}$: if $f_k(u, u') = u$, then $f_l(v, v') = v$.

Proof Outline

1. There are known algorithms to check if $\tilde{X}$ has a majority or a minority polymorphism (Bressiére & Carbonnel 2013, 2016).
2. For the binary aggregators, define H_X with vertices uu'_j, where $u, u' \in X_j$ s.t. $u \neq u'$ and edges $uu'_k \to vv'_l$, where:
   - $k \neq l$,
   - there are $z, z' \in X$ extending $(u, v), (u', v')$ and
   - there is no $w \in X$ extending (u, v') s.t. $w_j \in \{z_j, z'_j\}$.

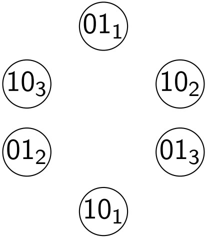
★ Intuitively, $uu'_k \to vv'_l$ means that, for any binary aggregator $\bar{f}$: if $f_k(u, u') = u$, then $f_l(v, v') = v$.

3. X has a binary non-dictatorial aggregator if and only if H_X is not strongly connected.

★ H_X can be constructed in polynomial time and there are known polynomial algorithms (e.g. Kosaraju's and Tarjan's algorithms), that check the strong connectedness of a graph.

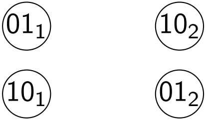
H_X

H_{X_1}



$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$

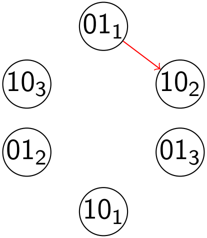
H_{X_2}



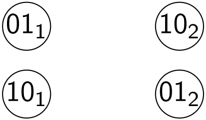
$X_2 = \{(0,1),(1,0)\}$

H_X

H_{X_1}



H_{X_2}



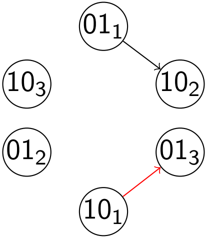
$X_2 = \{(0,1),(1,0)\}$

$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$

$(0,1,0),(1,0,0) \in X_1$, $(0,0,0) \notin X_1$

H_X

H_{X_1}

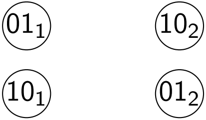


$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$

$(0,0,1),(1,0,0) \in X_1,$

$(1,*,1) \notin X_1$

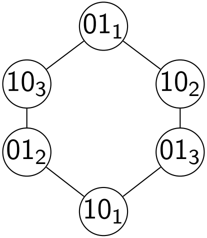
H_{X_2}



$X_2 = \{(0,1),(1,0)\}$

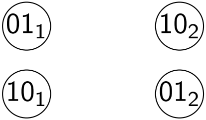
H_X

H_{X_1}



$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$

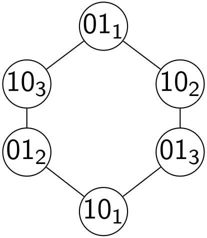
H_{X_2}



$X_2 = \{(0,1),(1,0)\}$

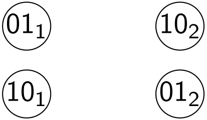
H_X

H_{X_1}



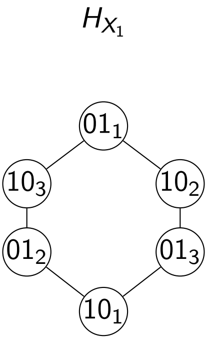
$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$

H_{X_2}

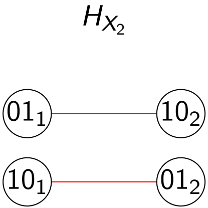


$X_2 = \{(0,1),(1,0)\}$

H_X



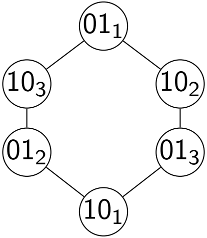
$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$



$(0,1),(1,0) \in X_2$,
$(0,0),(1,1) \notin X_2$.

H_X

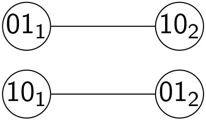
H_{X_1}



$X_1 = \{(0,0,1),(0,1,0),(1,0,0)\}$

Strongly Connected

H_{X_2}



$(0,1),(1,0) \in X_2$,
$(0,0),(1,1) \notin X_2$.

Not Strongly connected

Outline

Theorem 2

Theorem (KKL, 2018)

There is a polynomial-time algorithm that decides if a domain X is a uniform possibility domain, and if it is, that produces a uniform non-dictatorial aggregator of arity at most three.

Theorem 2

Theorem (KKL, 2018)

There is a polynomial-time algorithm that decides if a domain X is a uniform possibility domain, and if it is, that produces a uniform non-dictatorial aggregator of arity at most three.

Proof:

- (Carbonnel, 2016): There is a polynomial-time algorithm that decides if for all binary $B \subseteq A$ there is a polymorphism f of a constraint language Γ (set of relations over A) such that f is either binary and $f\restriction_B \in \{\wedge, \vee\}$, or f is ternary and $f\restriction_B \in \{maj, \oplus\}$ and that, if there is, produces one.
- This translates to the aggregator of the characterization theorem for uniform possibility domains!

Future Work

Open problem: What happens when X is given implicitly?

Future Work

Open problem: What happens when X is given implicitly?

- Integrity constraint (Grandi, Endriss 2010): $X_\phi \subseteq \{0,1\}^m$: set of satisfying assignments of a propositional formula ϕ.

Future Work

Open problem: What happens when X is given implicitly?

- Integrity constraint (Grandi, Endriss 2010): $X_\phi \subseteq \{0,1\}^m$: set of satisfying assignments of a propositional formula ϕ.
- Agenda: $\bar{\phi} = (\phi_1, \ldots, \phi_m)$, where each individual decides if each ϕ_j is true or false and $X_{\bar{\phi}}$ contains all consistent such judgments.

Thank you!