IMPOSSIBILITY THEOREMS AND THE UNIVERSAL ALGEBRAIC TOOLKIT

BY YIXIN XU

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Mario Szegedy

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2015

ABSTRACT OF THE DISSERTATION

Impossibility Theorems and the Universal Algebraic Toolkit

by Yixin Xu

Dissertation Director: Mario Szegedy

In this dissertation, we elucidate a close connection between the theory of Evaluation Aggregation, and a subfield of universal algebra, that was recently applied to investigate constraint satisfaction problems. Our connection yields a full classification of non-binary evaluations into possibility and impossibility domains both under the idempotent and the supportive conditions. Prior to the current result E. Dokow and R. Holzman nearly classified non-binary evaluations in the supportive case, by combinatorial means. The algebraic approach gives us new insights to the easier binary case as well, which had been fully classified by the above authors. We give a classification theorem for the majoritarian aggregator and show how Sen's well known theorem follows from it. Our algebraic view lets us put forth a suggestion about a strengthening of the non-dictatorship criterion, that helps us avoid "outliers" like the affine subspace. Finally, we give upper bounds on the complexity of deciding if a domain is impossible or not (to our best knowledge no finite time bounds were given earlier).

Acknowledgements

First of all I would like to express my sincere gratitude to my advisor Mario Szegedy, for his offering me a wonderful opportunity which was precious for me when I was applying for admission despite of my poor English, for his steady support during the difficult times when I encountered, for his unselfish instructions which I could never have learned without him. There was a time that the most enjoyable thing everyday was to discuss with him. I thank him for sharing those enlightening ideas from which I could imagine the beauty of a mathematical world. I thank him for his patience as I was talking with a genius brain. I thank him for his willingness of spending his precious time in discussing. I still remember during our first meeting at Rutgers, I was told by him that there are three abilities that a PhD student needs to develop: paper reading, mathematical thinking and problem solving which I was always putting in my head and when I began to understand how to measure the world using Mathematics and a bicycle, it was already time to graduate.

I would like to thank Bill Steiger. I would like to thank him for his steady support during the difficult times when I encountered. Bill Steiger has always been willing to encourage me.

I would like to thank Mike Saks. During the years in joining in the reading group, Mike has always been willing to share his wonderful insights, his own experience in problem solving.

I would like to thank professors in our department: Eric Allender, Bahman Kalantari, Swastik Kopparty, Mike Saks, Bill Steiger, Mario Szegedy, Zheng Zhang for their help during my stay at Rutgers.

I would like to thank Mario Szegedy, Eric Allender, Mike Saks and Andrei Bulatov for spending time to serve on my dissertation committee. I would like to thank Eric Allender and Andrei Bulatov for giving detailed comments which greatly help to improve the dissertation. I would like to thank Well Chiu, Luke Friedman, Kashyap Kolipaka, Mario Szegedy and Chengu Wang for collaborating with me.

I would also like to thank the student theory group: Abdul Basit, Dev Desai, Luke Friedman, Edinah Gnang, Gabor Kun, Meng Li, Fengming Wang, Yan Zhu. The wonderful discussions with them made my life at Rutgers more pleasant.

Last but not least, I would like to thank Carol DiFrancesco and other staff in our department. Every time I came to their office, they would never hesitate to help my need.

Dedication

To my parents.

Table of Contents

Chapter 1

Introduction

1.1 Social choice theory

“Social choice theory or social choice is a theoretical framework for analysis of combining individual opinions, preferences, interests, or welfares to reach a collective decision or social welfare in some sense”[Sen08]. When talking about collective decision, one’s first instinctive idea is the majority rule. When the number of possible outcomes is two, for example in a poll to select someone from only two candidates, the majority rule works well. (Except when the number of voters is even and a tie occurs, if this happens we have to make an arbitrary choice.) Condorcet’s jury theorem [Con85], although perhaps way too simplified, provides a theoretical basis for this case. Let n be the number of voters and each voter has an independent probability $p > 1/2$ to make a correct decision, then the probability that the group decision is correct increases to 1 as n grows to infinity. May’s theorem states that “simple majority voting is the only anonymous, neutral, and positively responsive social choice function between two alternatives” [May52].

However, when the number possible of outcomes becomes larger, the plurality value has little meaning, except when a majority of voters agree that some single outcome prevails. There are different possible ways to get around this problem. An avenue, suggested by Nicolas de Condorcet (1743 - 1794), known as Condorcet method, is to associate with each possible opinion a binary vector, and aggregate the components separately with the majority function, thus reducing the problem to the yes-no case. For this to be meaningful, each component must be an important aspect of the opinion. But there is another, more problematic aspect: when we put the aggregated coordinates together, the string should encode a valid opinion.

Condorcet's paradox. Condorcet in [Con85] considered aggregating preference lists of three elements. In this problem each voter casts a vote on a linear order of three candidates, A, B, C (for instance $A < B < C$), and the society has to aggregate these linear orders into a single linear order. Condorcet has suggested to write down any linear order as a list of three yes-no opinions: opinions on the issues if $A < B$, $B < C$ and if $C < A$. An order now uniquely corresponds to a triplet in $X = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$. For instance, $A < B < C$ corresponds to $(1,1,0)$, because $A < B$, $B < C$ hold, but $C < A$ does not. Condorcet then showed that in the case of three voters his component-wise aggregation scheme does not work. We will denote by $\mathrm{MAJ}_3(x,y,z)$ the function that outputs one if there are at least two 1s in the input, and otherwise outputs zero. In Condorcet's example, when aggregating three voters' opinion with the majority function, we arrive at a nonsensical aggregate:

$$\begin{array}{llccccc}
 & & A<B? & B<C? & C<A? & & \\
\text{Opinion 1.} & A<B<C & 1 & 1 & 0 & \in & X \\
\text{Opinion 2.} & B<C<A & 0 & 1 & 1 & \in & X \\
\text{Opinion 3.} & C<A<B & 1 & 0 & 1 & \in & X \\
 & & \downarrow & \downarrow & \downarrow & & \\
 & & \mathrm{MAJ}_3 & \mathrm{MAJ}_3 & \mathrm{MAJ}_3 & & \\
 & & \downarrow & \downarrow & \downarrow & & \\
\multicolumn{2}{c}{\text{No corresponding order}} & 1 & 1 & 1 & \notin & X
\end{array}$$

Remark 1. Let the number of voters n be odd. Let $G(n,3)$ denote the probability that the output of the group decision under majority MAJ_n (similarly defined) is rational, i.e. in X, while each voter uniformly randomly selects an element in X. Then $\lim_{n\to\infty} G(n,3) = \frac{3}{4} + \frac{3}{2\pi}\arcsin(\frac{1}{3}) \approx 0.91226$ which is known as Guilbaud's formula (see [Kal02]).

Arrow's impossibility theorem. Trying to resolve Condorcet's paradox, but keeping the idea of decomposing opinions into a sequence of attributes, Kenneth J. Arrow has started to look for aggregators *other than the majority function.* When the i^{th}

component is aggregated by f_i, he only requested that each f_i satisfy the Idempotency and Non-dictatorship conditions (see Section 1.2 for details). (Arrow's original formulation was slightly different, but equivalent to what we use). MAJ_3 clearly satisfies these conditions, but many other functions also do. Arrow has famously shown, that even with his relaxed conditions opinions on preference lists cannot be aggregated for any number of voters greater than two.

Discursive dilemma. The next step was to look at domains other than preference lists. Consider the following example [SX15]: When amateur astronomer Ali Leo tuned into his favorite radio channel, he was lucky. Three experts were on, passionately discussing the possibility of life on the fourth largest moon of Jupiter, Europa. What Ali gathered from the experts were the following:

	There is a layer of liquid water in Europa	If there is water in liquid form in Europa then the moon harbors life	Europa harbors life
Expert 1	Definitely	Definitely	Definitely
Expert 2	Definitely	There is no such conclusion	Definitely not
Expert 3	Definitely not	Definitely	Definitely not

When trying to get a conclusion based on the opinions of the experts, Ali instinctively used the majority rule. He has arrived at: I. There is a layer of liquid water in Europa; II. If there is water in liquid form in Europa then the moon harbors life; III. Europa definitely does not harbor life. He immediately noticed the logical contradiction in this summary and feverishly started to search for a similar contradiction in the individual experts' opinion. To his greatest surprise there was none.

Judgment aggregation. The above paradox is known as the *Discursive dilemma*, which was first investigated at depth by philosopher Philip Pettit [Pet01]. It is a more general version of the so-called *Doctrinal paradox*, due to Kornhauser and Sager [KS86], which in turn had an impressive history going back to Vacca [Vac21], and even to Poisson [Poi37]. The Discursive dilemma belongs to the larger topic of Judgment aggregation [Lis12]. The setting in Judgment aggregation assumes k Boolean variables $y_1, \ldots, y_k$ and m Boolean predicates $\varphi_1(\vec{y}), \ldots, \varphi_m(\vec{y})$ where $\vec{y} = (y_1, \ldots, y_k)$. Each voter has to take a vote on all predicates whether they are true or false. An example of a vote is

$$(\neg\varphi_1(\vec{y}), \varphi_2(\vec{y}), \ldots, \neg\varphi_m(\vec{y}))$$

with the natural restriction that the conjunction $\neg\varphi_1(\vec{y}) \wedge \varphi_2(\vec{y}) \wedge \ldots \wedge \neg\varphi_m(\vec{y})$ of the components of the vote does not yield the identically false predicate.

Consider the previous Discursive dilemma for example: there are two Boolean variables y_1 and y_2 where:

y_1: There is a layer of liquid water in Europa,

y_2: Europa harbors life.

There are three predicates φ_1, φ_2 and φ_3 where $\varphi_1 = y_1$, $\varphi_2 = y_1 \to y_2$ and $\varphi_3 = y_2$. Now the set of all legal votes becomes $\{(\cdot, \cdot, \cdot), (\cdot, \neg, \neg), (\neg, \neg, \cdot), (\neg, \neg, \neg)\}$.

Evaluation Aggregation. The elegant general combinatorial framework, which serves as the framework of this dissertation, was laid down by E. Dokow and and R. Holzman in [DH10a, DH10b]. We have named it "Evaluation Aggregation" after their titles. In their two breakthrough results they make decisive advances towards classifying impossibility domains, i.e. those $X \subseteq D^m$ from which we cannot aggregate opinions. In particular, they completely settle the binary case (when $|D| = 2$). (See Section 1.2 for more details.)

1.2 Settings and Definitions

Let J be a finite set of issues and D be a finite set of possible positions/opinions (like 'yes', 'no'). Without loss of generality we assume that

$$J = [m] = \{1, \ldots, m\}.$$

An evaluation $(v_1, \ldots, v_m) \in D^m$ assigns a position in D to each $j \in [m]$. The binary case, when $D = \{0, 1\}$ has received special attention [Wil75, RF86, DH10a]. Our fundamental object is the *domain* $X \subseteq D^m$ of *feasible evaluations*, these are the evaluations (i.e. opinion-combinations) that we allow for the voters to choose from.

Example 2. Assume that during a murder trial the members of the jury have to vote on two issues: 1. the suspect had a knife; 2. the suspect was the murderer; with taking a position either 'yes' or 'no' on each of the issues. Each member must take a position on both issues, but the jury agrees that the position-combination (1: no, 2: yes) cannot be valid, and should not be taken by any member. Thus $X = \{(\text{no,no}), (\text{yes,no}), (\text{yes,yes})\}$.

Example 3. Judgment aggregation fits this framework, with $D = \{\text{negated, not negated}\}$ and $X = \{(\epsilon_1, \ldots, \epsilon_m) \mid \bigwedge_{j=1}^{m} \epsilon_j \varphi_j \text{ is not identically false}\}$, where $\epsilon_j \in D$ for $1 \leq j \leq m$.

Aggregators. When n members of a society take a position on all of the m issues, and each member's vote is from X, we get a *profile* vector

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

Our goal is to design a function $f : X^n \to X$ that takes profile vectors into single elements of X. Such functions are called *aggregators*. The aggregators we shall consider must satisfy three conditions that come directly from Arrow's famous conditions, that he has come up with while studying the special case of preference list aggregation. Before we describe them we remark that

$$x^{(i)} = (x_1^{(i)}, \ldots, x_m^{(i)}) \in D^m \qquad \text{for } i = 1, \cdots, n,$$

are vectors themselves: $x_j^{(i)}$ is the i^{th} voter position on the j^{th} issue. Thus the profile is a vector of vectors. The output of f is a vector in D^m, representing the aggregated positions on the m issues. The latter vector must also belong to X.

The first and key condition is that each issue ought to be aggregated independently from the others (also called point-wise aggregation or Issue by Issue Aggregation):

Issue-by-Issue Aggregation (IIA): There are functions $f_j : D^n \to D$ $(1 \le j \le m)$ such that for every $(x^{(1)}, \dots, x^{(n)}) \in X^n$:

$$f(x^{(1)}, \dots, x^{(n)}) = \left(f_1\left(x_1^{(1)}, \dots, x_1^{(n)}\right), \dots, f_m\left(x_m^{(1)}, \dots, x_m^{(n)}\right)\right)$$

There is a nice way to visualize the IIA property via the picture

$$\begin{array}{cccccc}
x_1^{(1)} & \cdots & x_m^{(1)} & \in & X \\
 & \vdots & & & \\
x_1^{(n)} & \cdots & x_m^{(n)} & \in & X \\
\hline
\downarrow_{f_1} & \cdots & \downarrow_{f_m} & & \\
x_1 & \cdots & x_m & \in & X
\end{array}$$

Above we aggregate column j (where $j \in [m]$ is an issue) by function f_j. The condition that f takes X^n to X is equivalent to saying, that if each row belongs to X then so does the aggregated row. The component aggregate functions should work in unison to accomplish this, so the IIA condition is not easy to satisfy. We have adopted the term "Issue by Issue Aggregation" coined by E. Dokow and R. Holzman [DH10a, DH10b], which is in this generalized context more fitting than the commonly used "Independence of Irrelevant Alternatives" expression, with the benefit that the acronym remains the same.

Uniqueness of the IIA decomposition. The representation of $f : X^n \to X$ as $(f_1, \dots, f_m)$ is clearly not unique for instance when D contains any element that does not occur as a constituent in any $x \in X$. In order to avoid non-uniqueness of the f_js we define

$$D_j = \mathrm{pr}_j X = \{u_j \mid (u_1, \dots, u_m) \in X\}.$$

We call D_j *the effective set of values* associated with the j^{th} component. If we define f_j on D_j^n instead of D^n, it is easy to see that f_j becomes unique. Throughout the paper we shall assume this.

In the sequel we will also call IIA aggregator(s) as aggregator(s) for simplification since we will not look at non-IIA aggregators in this dissertation. Next we describe the two other conditions (besides IIA) Arrow has imposed on an aggregator $f : X^n \to X$.

Idempotency (or Unanimity): $f(x, \ldots, x) = x$ for every $x \in X$.

Lemma 4. *An IIA aggregator $f = (f_1, \ldots, f_m)$ is idempotent if and only if every f_j is idempotent in the universal algebraic sense, i.e.*

$$\forall\, 1 \leq j \leq m \qquad \forall\, u \in D_j: \qquad f_j(u, \ldots, u) = u$$

Non-dictatorship: Aggregator $f : X^n \to X$ is a dictatorship if there is a $1 \leq k \leq n$ such that for every $(x^{(1)}, \ldots, x^{(n)}) \in X^n$ we have $f(x^{(1)}, \ldots, x^{(n)}) = x^{(k)}$. Otherwise the Non-dictatorship condition holds for f.

Lemma 5. *An IIA aggregator $f = (f_1, \ldots, f_m)$ is a dictatorship if and only if there is a $1 \leq k \leq m$ such that each f_j is a* projection *on the k^{th} coordinate in the universal algebraic sense:*

$$\forall\, 1 \leq j \leq m \qquad \forall\, u_1, \ldots, u_n \in D_j: \qquad f_j(u_1, \ldots, u_n) = u_k$$

Definition 6 (Impossibility/Possibility domains)**.** We call an $X \subseteq D^m$ a possibility domain (after Arrow) with respect to the IIA + Idempotency + Non-dictatorship conditions if for some $n \geq 2$ an aggregator for X exists that satisfies the conditions. Otherwise X is an impossibility domain.

In this dissertation we completely characterize impossibility domains with respect to the IIA + Idempotency + Non-dictatorship and also for the case when Idempotency is replaced with

Supportiveness: $f : X^n \to X$ is supportive if for every $x^{(1)}, \ldots, x^{(n)} \in X^n$ and every $1 \leq j \leq m$ we have that $f(x^{(1)}, \ldots, x^{(n)})_j \in \{x_j^{(1)}, \ldots, x_j^{(n)}\}$.

Supportiveness implies Idempotency, but not vice versa.

Prior to our result a full characterization of all impossible binary domains (see Definition 7) was obtained under IIA + Idempotency + Non-dictatorship in [DH10a]. They have extended their work to the non-binary case, but have obtained only a partial characterization, and only under the IIA + Supportiveness + Non-dictatorship conditions. To explain their results we need some definitions.

Definition 7. We call $X \subseteq D^m$ non-degenerate if $|\mathrm{pr}_j X| > 1$ for every $1 \leq j \leq m$. Since the issues where degeneration occurs can trivially be aggregated, in the aggregation problem without loss of generality we can assume that X is non-degenerate. We call $X \subseteq D^m$ binary[1] if $|\mathrm{pr}_j X| \leq 2$ for every $1 \leq j \leq m$. Otherwise call it non-binary.

Definition 8 (Affine Subspace)**.** A subspace S is a subset of $\{0,1\}^m$ (viewed as the m-dimensional vector space over the 2-element field $\mathbb{F}_2$) closed under linear combination $\lambda\vec{u}+\mu\vec{v}$ of its elements, where $\lambda, \mu \in \{0,1\}$ and $u, v \in S$. An affine subspace $X \subseteq \{0,1\}^m$ is then a subspace S shifted by some fixed vector $\vec{w} \in \{0,1\}^m$, i.e. $X = \vec{w}+S = \{\vec{w}+\vec{s} \mid s \in S\}$.

Note that there is a similar way to define affine subspace for larger fields other than $\mathbb{F}_2$. However, we only need the above definition for binary evaluations.

Definition 9 (Blockedness graph and MIPE)**.** The blockedness graph for domain $X \subseteq \{0,1\}^m$ is the following directed graph on the vertex set $V = [m] \times \{0,1\}$: There is a directed edge from $(k, \sigma) \in V$ to $(\ell, \rho) \in V$ where $k \neq \ell$ if and only if there are: (i.) a subset $S \subseteq [m]$ such that $k, \ell \in S$ and (ii.) a (partial-) evaluation $u : S \to \{0,1\}$ with $u_k = \sigma$ and $u_\ell = \neg\rho$ such that there is no extension of u to any full evaluation x in X, but if we flip any bit of u then the resulting partial evaluation extends to some element of X. The above partial assignment u is called a MIPE (minimally infeasible partial evaluation).

Definition 10 (Total blockedness)**.** Domain $X \subseteq \{0,1\}^m$ has the total blockedness condition if and only if the blockedness graph is strongly connected.

A result leading to [DH10a] was that of Nehring and Puppe [NP02]. They have obtained a complete classification of binary (see Definition 7) impossibility domains X, when a monotonicity condition is added to the usual conditions. An aggregate function

[1] Note that sometimes by abuse of notation we also call a relation or an aggregator of arity 2 binary, but this should happen only when it is clear from the context.

is said to be monotone[2] if for every issue j in any situation the aggregate position on issue j does not change if a voter decides to switch his/her position on the j^{th} issue to the current aggregate position.

Theorem 11 (Nehring and Puppe [NP02])**.** *A non-degenerate $X \subseteq \{0,1\}^m$ is an impossibility domain with respect to IIA + Idempotency + Monotonicity + Non-dictatorship if and only if X is totally blocked.*

The complete characterization of binary evaluations without the monotonicity condition was finally given by E. Dokow and R. Holzman:

Theorem 12 (E. Dokow, R. Holzman [DH10a])**.** *Let $X \subseteq \{0,1\}^m$, non degenerate. Then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship if and only if X is totally blocked and is not an affine subspace.*

E. Dokow and R. Holzman have also made significant progress for non-binary domains [DH10b]. We will need the following generalization of total blockedness for non-binary predicates:

Definition 13. Total blockedness for non-binary X is defined in [DH10b]. X is totally blocked if and only if the following directed graph on the vertex set $V = \{\sigma\sigma'_j = (j,\sigma,\sigma') |\ (j \in [m]) \ \wedge\ (\sigma,\sigma' \in \mathrm{pr}_j X) \ \wedge\ (\sigma \neq \sigma')\}$ is strongly connected. There is an edge from $\sigma\sigma'_k$ to $\rho\rho'_\ell$ where $k \neq \ell$ if and only if there are $B_1 \in \mathrm{pr}_1 X, \ldots, B_m \in \mathrm{pr}_m X$, such that each $|B_j| = 2$, $B_k = \{\sigma,\sigma'\}$ and $B_\ell = \{\rho,\rho'\}$, and if X_B denotes the relation $X \cap \prod_{j=1}^m B_j$, then when X_B is viewed as a binary relation (by identifying each B_i with $\{0,1\}$), (k,σ) is connected with (ℓ,ρ) in the blockedness graph of X_B (see Definition 9 for the definition of blockedness graphs for binary domains).

We also need to define a condition on a relation X, which is called 2-decomposability in the universal algebra literature. We have adopted this term rather than the "not multiply constrained" expression for the same concept in [DH10b].

[2] Note that this kind of monotonicity is different from the standard one (the greater the input, the greater the output). Say, $a < b < c$ and $f(x,y) = \min(x,y)$, except $f(a,c) = f(c,a) = b$. Then f is monotone in the usual sense, but not in the sense of this definition. However, for functions defined on $\{0,1\}^n$, these two notions of monotonicity coincide. We thank Andrei Bulatov for pointing out this.

Definition 14. For an m-ary relation X on D and for $1 \leq i < j \leq m$, let $\mathrm{pr}_{i,j}X = \{(u_i, u_j) \mid (u_1, \ldots, u_m) \in X\}$. A relation X is called 2-decomposable if, for any tuple $x = (x_1, \ldots, x_n) \in D^m$, we have $x \in X$ if and only if $(x_i, x_j) \in \mathrm{pr}_{i,j}X$ for all $1 \leq i < j \leq m$. In a similar way we can also define k-decomposable for any $k = 2, \cdots, m$.

Remark 15. Coincidentally, in [DH10b] the "2-decomposable" expression also occurs, but with a very different meaning.

Theorem 16 (E. Dokow, R. Holzman [DH10b])**.** *Let $X \subseteq D^m$ be non-degenerate and non-binary (see Definition 7). If X is totally blocked and not 2-decomposable (see Definition 14) then X is an impossibility domain with respect to IIA + Supportiveness + Non-dictatorship.*

Theorem 17 (E. Dokow, R. Holzman [DH10b])**.** *Let $X \subseteq D^m$. If X is non-degenerate and not totally blocked then X is a possibility domain with respect to IIA + Supportiveness + Non-dictatorship.*

1.3 Main results

In spite of the impressive advances due to E. Dokow and R. Holzman, important questions have remained open:

1. Complete the characterization of the Supportive case, when $|D| > 3$. (In [DH10b] the case $|D| = 3$ is resolved. See Theorem 3 in the paper.)
2. Settle the $|D| > 2$ case with the Idempotency condition.

We combine ideas from algebraic theory of constraint satisfaction problems (or more originally from universal algebra) with results from [DH10b] to get a full characterization of the non-binary case, with the IIA + Supportiveness + Non-dictatorship conditions. We state here the theorem in a simplified form for better understanding. The theorem that contains all the finer details is Theorem 80 in Section 4.1.

Theorem 18. *Let $X \subseteq D^m$ be non-degenerate. If X is totally blocked then X is an impossibility domain with respect to IIA + Supportiveness + Non-dictatorship if and only if there is no aggregator with at most three voters.*

Note that Theorem 18 is a generalization of Theorem 12 by E. Dokow and R. Holzman. It holds for both binary and non-binary domains. For binary case, the Supportiveness condition is equivalent to the Idempotency condition, while for non-binary case, the former implies the latter. If we insist on the Idempotency condition instead of the Supportiveness condition (question 2), we have the following characterization:

Theorem 19. *Let $X \subseteq D^m$ be non-degenerate and non-binary. If X is totally blocked then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship if and only if there is no aggregator with at most $|D|$ voters.*

The proof of the theorem with additional details is in Section 4.2.

Although the case of binary evaluations is completely settled by E. Dokow and R. Holzman, we revisit it in Chapter 3 and give a short proof, using results from universal algebra. We get a slightly refined form of their theorem:

Although the case of binary evaluations was completely settled by E. Dokow and R. Holzman, universal algebra gives a tad more refined form of their theorem (see Details in Chapter 3):

Theorem 20. *Let $X \subseteq \{0,1\}^m$ be non-degenerate. Then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship if and only if X is totally blocked and it is not an affine subspace. If X is not totally blocked then for all $1 \leq j \leq m$ one of the following holds:*

1. *there is an f such that f_j is the semi-lattice operation $u \vee v$ or $u \wedge v$,*

2. *there is an f such that f_j is the majority operation $(u \vee v) \wedge (v \vee w) \wedge (w \vee u)$,*

3. *there is an f such that f_j is the Mal'tsev operation $u - v + w \mod 2$,*

4. *f_j is a dictatorship for every f.*

Gadgets. How do the above characterizations help when we want to give a concise proof that a given X is an impossibility domain? It turns out that we can characterize impossibility domains in a dual way, in terms of a set of *gadgets.* Our new characterization, that provides *witnesses* to impossibility, has not been known earlier in the voting theory context. Gadgets (or *conjunctive queries*) are expressions used in reductions between constraint satisfaction problems when we translate instances locally (term by term). They are existentially quantified conjuncts of clauses, where each clause is a relation. The relations in this interpretation are viewed as Boolean-valued functions on not necessarily Boolean *variables*. The syntax of a gadget is:

$$R(\vec{x}) = \exists \vec{y}: \ S_1(\vec{x},\vec{y}) \wedge \ldots \wedge S_k(\vec{x},\vec{y}) \quad \text{(each } S_i \text{ in effect depends only on subsets of } \vec{x},\vec{y}\text{)}$$

Their purpose is to express new relations from a given set of relations.

A theorem of D. Geiger [Gei68] establishes a connection between the nonexistence of aggregators for a set Γ of relations and the existence of Γ-gadgets (i.e. in which all relations are from Γ or the '=' relation). This theorem serves as the backbone of the algebraic theory of constraint satisfaction problems developed by P. Jeavons, A. A. Bulatov, A. A. Krokhin, D. A. Cohen, M. Cooper, M. Gyssens [JCG97, Jea98, JCC98, BJK05] and several other researchers.

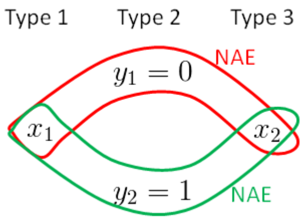


Figure 1.1: The multi-sorted Not All Equal (NAE) gadget to express the multi-sorted inequality relation between x_1 (type 1) and x_2 (type 3). This gadget is central in our proof of Arrow's theorem.

This is our starting point too. In universal algebra, (IIA) aggregators are called *polymorphisms*. They differ from the aggregators in our introduction in that they are single-sorted: $f_1 = f_2 = \ldots = f_m$. Multi-sorted polymorphisms, i.e. when the f_js can be different, have also been studied in the algebraic literature [BJ03, Bul11]. When dealing with multi-sorted polymorphisms, all relations and gadgets (gadgets also express relations) must be *multi-sorted* as well. In the multi-sorted world first we must declare a type for every variable. The typing of the variables serves the same purpose as in programming languages: when a function is called, the types of the called variables must match with the types in the function declaration. In the same spirit, every multi-sorted l-ary relation R we construct (or given to us) must come with a sequence of l (not necessarily different) types. We may call this type declaration. The typing must be consistent: the types of the variables involved in any occurrence of a relation in a gadget must match with the type declaration.

Our multi-sorted version of Geiger's theorem allows us to prove impossibility results simply by producing sets of appropriate gadgets. For illustration, we construct the multi-sorted gadget that leads to the proof of Arrow's theorem. In the algebraic language Arrow's theorem says that all idempotent aggregators of the multi-sorted NAE relation are dictatorships. The multi-sorted NAE is a binary relation that has three arguments: x with type 1, y with type 2 and z with type 3. NAE holds when the variables are not all zero and not all one. In Section 2.5 we show that Arrow's theorem follows if we can create a multi-sorted gadget that expresses the relation $x_1 \neq x_2$, where x_1 and x_2 are binary variables of types 1 and 3, respectively. (Because NAE is symmetric this also means that we can create the $x_1 \neq x_2$ relation for arbitrary two different types.) The basic building block for our gadget has to be the multi-sorted NAE relation, but because of the Idempotency condition we can also use assignment giving relations, like $y = 1$. Our gadget looks like (see also Fig. 1.1):

$$\exists y_1, y_2 : \; NAE(x_1, y_1, x_2) \wedge NAE(x_1, y_2, x_2) \wedge (y_1 = 0) \wedge (y_2 = 1) \qquad (1.1)$$

Any additional gadgets we need to build to prove Arrow's theorem can be easily constructed from (1.1) (see Section 2.5).

In this dissertation we prove a multi-sorted version of Geiger's theorem (see Section 2.3). This allows us to prove impossibility results simply by producing sets of gadgets. If we want to prove that $X \subseteq \prod_j^m D_j$ is an impossibility domain, we can construct X^+-gadgets for a certain "complete" set of (multi-sorted) relations. Here the '+' in the upper index refers to the permission to use assignment-giving relations (i.e. '$x = a$,' where $a \in D_j$, for type j variables) in our gadgets in addition to X. In all cases X must be viewed as a multi-sorted relation, all components (arguments) having different types. The Idempotency constraint is encoded in the '+' of X^+. Alternatively, the Supportiveness condition in the gadget-reformulation translates to the permission of using arbitrary unary relations in the gadgets. When we allow the latter, the gadget is an $X^{\mho}$-gadget.

Theorem 21. *For every D and m there is a fixed finite set $\mathcal{P} = \mathcal{P}(D, m)$ of multi-sorted relations such that $X \subseteq D^m$ is an impossibility domain with respect to IIA + Idempotency (Supportiveness) + Non-dictatorship if and only if we can express every member of $\mathcal{P}$ with a gadget whose conjunct has only (appropriately multi-sorted) X^+- ($X^{\mho}$)-clauses. In addition, the number of auxiliary variables is upper bounded by some explicit function $\phi(m, |D|)$ (single exponential in m).*

The theorem in more details is restated in Theorem 60 and proved in section 2.4.

Our characterizations allow us to decide if X is an impossibility domain (both in the Idempotent and in Supportive cases) in two different ways: either by checking aggregators up to a certain number of arguments (Theorem 18 and Theorem 19), or by checking gadgets up to a certain size (Theorem 21). If we do both searches in parallel, we also find a short certificate (compared to the search time) for either the possibility or the impossibility of X, depending on which holds.

One of the main applications of our gadget characterization is that we can prove the impossibility of a domain X by merely presenting a few gadgets (rather than trying to exclude a large set of aggregators). In some cases we can tailor the gadgets to the specific problem, exploiting symmetries. A combined approach where we exclude most aggregators by gadgets while the rest we treat directly is also possible. Using gadgets we can show that the Pairwise Distinctness relation defined by

$$\{(u_1, \ldots, u_m) \in D^m \mid u_k \neq u_\ell \, (1 \leq k < \ell \leq m)\}$$

is an impossibility domain when $|D| > m$ if $m = 2$ and when $|D| = m$ if $m \geq 3$ with respect to the IIA + Idempotency + Non-dictatorship conditions. In [DH10b] this is proven only under the IIA + Supportiveness + Non-dictatorship conditions, and [FF11] proves the above only when $|D| = m$ and $m \geq 3$.

Besides impossibility theorems, another avenue in social choice theory is the opposite. If some domain is a possibility domain, we are interested in what kind of aggregators such domain has. In particular, the majority aggregators have drawn great attention. This is the so called domain restriction problem. In Chapter 6 we study this traditional problem in social choice theory from an algebraic point of view.

The definitions of possibility (even with the Supportive restriction) is considered too generous by some researchers, and several further restrictions were studied [Kal02]. E. Dokow and R. Holzman, for instance, question if linear subspaces of $\{0,1\}^m$ should really be considered possibility domains [DH10a]. In Chapter 7 we define a new aggregator class that strengthens the notion of Non-dictatorship. Our new definition directly comes from the algebraic theory and has many desirable properties.

The rest of the dissertation is organized as follows. In Section 1.4, we introduce the algebraic theory of constraint satisfaction problems, which was originally developed to attack the dichotomy conjecture but later found many other applications. In Chapter 2, we present one characterization of impossibility domains, that is by gadgets. Two examples are also presented to illuminate this characterization: Arrow's impossibility theorem (Section 2.5) and Pairwise Distinctness (Section 2.6). In Chapter 3 we reprove previously known results for characterization of binary domains by E. Dokow and R. Holzman [DH10a] in a short way using the new tools developed in Chapter 2. In Chapter 4, we show another characterization of impossibility domains, that is by aggregators, for both the Supportiveness case and the Idempotency case. After these two characterizations are discussed, in Chapter 5, we show how to convert these characterizations to algorithms of determining if a domain is an impossibility domain or not. We also give upper bounds on the complexity of deciding if a domain is impossible or not (to our best knowledge no finite time bounds were given earlier). In Chapter 6, we study the domain restriction problem and give a classification theorem for the majoritarian aggregators and show how Sen's well known theorem follows from it. Last in Chapter 7, using our algebraic view we put forth a suggestion about a strengthening of the Non-dictatorship criterion, that helps us avoid "outliers" like the affine subspace.

1.4 Algebraic theory of Constraint Satisfaction Problems

It has to mention that the presentation of this dissertation still works well without referring any detailed definitions of constraint satisfaction problems (CSPs) and what is really relevant is universal algebra (see [Ros86, Qua95, JQ95, LP96, Csa05, FV09]), however, we still present universal algebra by means of its application to CSPs. Universal algebra applies to both CSPs and voting theory (as elucidated in this dissertation). Sometimes (but not always) terms in one area transform directly to the other. The following table lists some examples of similarities between these two different areas.

CSP(multi-sorted)	**Voting Theory**
Constraint relation	Domain
Multi-sorted polymorphism	IIA aggregator
Assignment-giving relation	Idempotency condition
Projection	Dictatorship
NP-hard	Impossibility domain
Tractable	Possibility domain
List constraints/conservative	Supportiveness condition

Definition 22 (CSP)**.** Let D be a finite set. Call a subset $R \subseteq D^r$ a *relation*, where $r \geq 1$ is the *arity.* Let Γ be a set of relations (possibly infinite) over D, an *instance* Φ of a *constraint satisfaction problem (CSP)* CSP(Γ) contains two parts:

1. $V = \{x_1, \cdots, x_n\}$ a set of n *variables*,
2. $\{(x_{i,1}, \cdots, x_{i,k_i}), R_i) | i = 1, \cdots, m\}$ a set of m *constraints* where each $R_i \in \Gamma$ has arity k_i and $x_{i,j} \in V$.

The question is to determine if there is a map $\varphi : V \to D$ such that for each $i = 1, \cdots, m$, $(\varphi(x_{i,1}), \cdots, \varphi(x_{i,k_i})) \in R_i$. Call φ a *solution* to Φ if there is such.

Remark 23. The constraint satisfaction problem can aslo be defined as homomorphism between relational structures, which turns out to be equivalent to Definition 22. In particular, if constraint relation is a graph H, we call this CSP as H-coloring problem. For example, the k-coloring problem can be viewed as when $H = K_k$.

Definition 24 (CSP as Homomorphism)**.** Call $(D; R_1, R_2, \cdots)$ a *relational structure* over D if D is a finite set and each R_i is a relation over D, i.e. there is some k_i such that $R_i \subseteq D^{k_i}$, and $(k_1, k_2, \cdots)$ is called *signature* of the relational structure. Two relational structures are *similar* if they have the same signature. A *homomorphism* between two similar relational structures $(D; R_1, R_2, \cdots)$ and $(D'; R'_1, R'_2, \cdots)$ is defined as a map $\varphi : D \to D'$ such that for each i and for each $(r_1, r_2, \cdots, r_{k_i}) \in R_i$, $(\varphi(r_1), \varphi(r_2), \cdots, \varphi(r_{k_i})) \in R'_{k_i}$.

Let Γ be a set of relations (possibly infinitely many) over D. Let $V = \{x_1, \cdots, x_n\}$ be a set of n *variables.* An *instance* Φ of a *constraint satisfaction problem (CSP)* CSP(Γ) is defined as to determine if there is a homomorphism between two similar relational structures $(V; V_1, \cdots, V_m)$ and $(D; R_1, \cdots, R_m)$ where $V_i = \{(x_{i,1}, \cdots, x_{i,k_i})\}$ and $R_i \in \Gamma$. If there is such, call it a *solution* to CSP(Φ).

Example 25. The following table lists some CSPs that mostly appear in computer science literature.

CSP Name	Domain	Constraint relation(s)
2-COLORING	$\{0, 1\}$	$\{(0, 1), (1, 0)\}$
2-SAT	$\{0, 1\}$	$\{0, 1\}^2 \setminus \{(0, 0)\}, \{0, 1\}^2 \setminus \{(0, 1)\}, \{0, 1\}^2 \setminus \{(1, 1)\}$
3-LIN	$\{0, 1\}$	$x_1 + x_2 + x_3 \equiv 0 \mod 2$, $x_1 + x_2 + x_3 \equiv 1 \mod 2$
HORN SAT	$\{0, 1\}$	$\forall k : x_1 \vee x_2 \vee \cdots \vee x_k$, $\bar{x}_1 \vee x_2 \vee \cdots \vee x_k$
DUAL-HORN SAT	$\{0, 1\}$	$\forall k : \bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k$, $x_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k$
NOT-ALL-EQUAL SAT	$\{0, 1\}$	$\{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$
3-SAT	$\{0, 1\}$	$x \vee y \vee z, x \vee y \vee \bar{z}, x \vee \bar{y} \vee \bar{z}, \bar{x} \vee \bar{y} \vee \bar{z}$
3-COLORING	$\{0, 1, 2\}$	$\{0, 1, 2\}^2 \setminus \{(0, 0), (1, 1), (2, 2)\}$

Example 26 (Sudoku)**.** The following is a typical Sudoku to be solved.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

In general, Sudoku can be written as a CSP. Let $D = \{1, 2, \cdots, 9\}$ be the domain. Let $D_{\neq} \subseteq D^2$ be a relation defined as $D_{\neq} = \{(x, y) \in D^2 | x \neq y\}$. Thus Sudoku can be viewed as a 9-coloring problem with singleton constraints, i.e. some vertices are assigned with predefined colors. The variable set for a Sudoku instance contains $x_{i,j} : 1 \leq i, j \leq 9$. Besides those singleton constraints for each particular Sudoku, the common constraints for all Sudoku contain $x_{i,j} \neq x_{k,l}$ where $(i, j) \neq (k, l)$ if (1) $i = k$ (row constraints); (2) $j = l$ (column constraints); (3) $\lfloor \frac{i-1}{3} \rfloor = \lfloor \frac{k-1}{3} \rfloor$ and $\lfloor \frac{j-1}{3} \rfloor = \lfloor \frac{l-1}{3} \rfloor$ (inner square constraints).

Multi-sorted CSPs (MCSP) differ from usual CSPs in that each variable in V has a type. Each type b has a designated set D_b in which variables of that type range. We denote the type of variable x by type(x). Without loss of generality we may assume that the set of permissible types is $[t]$ where t is a fixed positive integer. The corresponding ranges are $D_1, \ldots, D_t$. Let $X \subseteq \prod_{j=1}^{m} D_{\tau_j}$ where $\tau_j \in [t]$ for $1 \leq j \leq m$. We denote X with (X, τ), where $\tau = (\tau_1, \ldots, \tau_m)$, to indicate that it is multi-sorted and to indicate the types of its components. A typical choice in our assignment aggregation setting is $t = m$, $D_i = \mathrm{pr}_i X$ and the typing of X is $(X, (1, \ldots, m))$.

Multi-sorted CSPs are defined analogously to usual CSPs, except that all their variables are typed and their constraints are multi-sorted, using the same fixed type set $[t]$. The variable set V is $V_1 \dot{\cup} \cdots \dot{\cup} V_t$, where V_b is the set of variables of type b for every

$1 \leq b \leq t$. An (X, τ)-constraint $X(x_{\sigma_1}, \ldots, x_{\sigma_m})$ is an X-constraint with the additional requirement that $\text{type}(x_{\sigma_j}) = \tau_j$ for $1 \leq j \leq m$.

Remark 27. Single sorted (i.e. commonplace) CSPs can be viewed as multi-sorted ones, where $t = 1$. Thus multi-sorted CSPs are more general. Multi-sorted CSPs can be emulated with single-sorted ones on the expense of increasing the alphabet size and introducing type-constraints, and in Section 2.3 we shall exploit this.

Gadget. An important theme in computer science is to determine the complexity of CSP(Γ) for given set of constraint relations Γ. For example, it is well known that 2-SAT, 2-COLORING, 3-LIN, HORN SAT, and DUAL-HORN SAT are all in **P**, and NOT-ALL-EQUAL SAT, 3-SAT, 3-COLORING are all **NP**-complete. An important notion in determining the complexity of CSP is *gadget*.

Definition 28 (gadgets, single-sorted)**.** A relation $R \subseteq D^m$ *gadget reduces* to a set $\{X_1, \ldots, X_l\}$ of relations on D if

$$R(x_1, \ldots, x_m) = \exists\, y_1, \ldots, y_{m'} : \; R_1(z_{1,1}, \ldots, z_{1,m_1}) \wedge \ldots \wedge R_t(z_{t,1}, \ldots, z_{t,m_t})$$

where each R_i is one of $X_1, \ldots, X_l$ or the equality relation $x = y$. Furthermore

$$z_{i,j} \in \{x_1, \ldots, x_m, y_1, \ldots, y_{m'}\}.$$

Variables $\{x_1, \ldots, x_m\}$ are called *x-variables* and variables $\{y_1, \ldots, y_{m'}\}$ are called *auxiliary variables*. The number of auxiliary variables m' is defined as *the size of gadget* R. Let Γ, Γ' be two sets of relations. Say Γ gadget reduces to Γ' if for every relation $R \in \Gamma$, R gadget reduces to Γ'.

We rely on the following lemma with gadgets to reduce from one problem to another.

Lemma 29. *Let* Γ, Γ' *be two sets of relations. If* Γ *gadget reduces to* Γ'*, then if* CSP(Γ) *is **NP**-complete, then* CSP(Γ′) *is **NP**-complete, if* CSP(Γ′) *is in **P**, then* CSP(Γ) *is also in **P**.*

Example 30. We give here a gadget reduction from 3-SAT to 1-in-3 SAT, thus give a proof of **NP**-hardness of CSP($\{(1,0,0),(0,1,0),(0,0,1)\}$). Denote by

$$T = \{(1,0,0),(0,1,0),(0,0,1)\}.$$

First we can create the following gadget:

$$\bar{x} \vee y \vee \bar{z} = \exists a, b, c, d : T(x, a, b) \wedge T(y, b, c) \wedge T(z, c, d).$$

Also we can create the following inequality gadget:

$$(x \neq y) = \exists a, b, c, d, e : T(x, y, a) \wedge T(a, b, c) \wedge T(b, d, e) \wedge (d = e).$$

Once we have these two gadgets, we can create the four constraint relations for 3-SAT. For example:

$$x \vee y \vee z = \exists a, b : (\bar{a} \vee y \vee \bar{b}) \wedge (a \neq x) \wedge (b \neq y).$$

Remark 31. There are different names for gadgets in literature. For example, in logic, they are often called primitive positive formula (pp-formula), a first order formula using existential quantifiers and conjunction only [BKJ01]. It is shown that a gadget R can also be created from Γ by taking permutation, extension, projection and intersection [Jea98], or Cartesian product, equality selection and projection [JCC98].

There is a voluminous literature on the satisfiability of CSPs and classification theorems constitute a significant part of this. Thomas J. Schaefer in his celebrated theorem of 1978 has classified the complexity of all binary CSPs (i.e. when $|D| = 2$):

Theorem 32 (Schaefer [Sch78])**.** *Let $D = \{0, 1\}$ and $X_1, \ldots, X_l$ be relations of arity $m_1, \ldots, m_l$ on D. Then* $\mathrm{CSP}(X_1, \ldots, X_l)$ *is in polynomial time if:*

1. $(\underbrace{0, \ldots, 0}_{m_i}) \in X_i$ *for all* $1 \leq i \leq l$;
2. $(\underbrace{1, \ldots, 1}_{m_i}) \in X_i$ *for all* $1 \leq i \leq l$;
3. X_i *is a 2CNF for every* $1 \leq i \leq l$;
4. X_i *is a Horn formula for every* $1 \leq i \leq l$;
5. X_i *is a dual-Horn formula for every* $1 \leq i \leq l$;
6. X_i *is the solution space of a linear system over* $\mathrm{GF}(2)$ *for every* $1 \leq i \leq l$.

Otherwise $\mathrm{CSP}(X_1, \ldots, X_l)$ *is NP hard.*

When H denotes a undirected graph, Pavol Hell and Jaroslav Nesetril in 1990 showed that CSP(H) is NP-complete if and only if H is non-bipartite, otherwise it is in P [HN90]. The famous dichotomy conjecture by Tomás Feder and Moshe Y. Vardi says that for any finite $|D| \geq 2$, for any relational structure Γ on D, CSP(H) is either P or NP-complete [FV93, FV98]. The algebraic theory of constraint satisfaction problems was developed in a sequence of papers by P. Jeavons, A. A. Bulatov, A. A. Krokhin, D. A. Cohen, M. Cooper, M. Gyssens [JCG97, Jea98, JCC98, BJK05]. A lot of progress has been made using the algebraic tools to attack this conjecture [Bul06, Bul11, BK09]. It has also found numerous other applications [BD03, ABI$^+$09, BK12].

For multi-sorted CSPs, we can define an analogue of gadget and reduction to typed-gadget and typed-reduction.

Definition 33 (gadgets, multi-sorted)**.** A multi-sorted relation (R, τ), where $R \subseteq D^k$ and $\tau = (\tau_1, \ldots, \tau_k) \in [t]^k$, multi-sorted gadget-reduces to a set $(X_1, \tau^1), \ldots, (X_l, \tau^l)$ of multi-sorted relations on D if there is a multi-sorted gadget expression

$$R(x_1, \ldots, x_k) = \exists\, y_1, \ldots, y_{k'} : \; R_1(z_{1,1}, \ldots, z_{1,k_1}) \wedge \ldots \wedge R_p(z_{p,1}, \ldots, z_{p,k_p})$$

where each R_i is one of $(X_1, \tau^1), \ldots, (X_l, \tau^l)$ or the multi-sorted equality relation $(x = y,(\text{a,a}))$ (meaning type(x) = type$(y) = a$) for some $a \in [t]$. It is important that we do not allow the free occurrence of the relation $(x = y, (a, b))$, where $a \neq b$.

In the algebraic theory of CSPs voting functions are called polymorphisms. They are very extensively studied and classified according to their algebraic properties. First we define polymorphisms in the single-sorted case.

Definition 34 (polymorphism)**.** Let $X \subseteq D^m$ be a relation. A function $g : D^n \to D$ is called a polymorphism of X if the tuple $(\underbrace{g, \ldots, g}_{m})$ is an IIA aggregator for relation X, i.e. it takes X^n into X. A function $g : D^n \to D$ is a polymorphism with respect to a set Γ of relations on D if it is a polymorphism for each relation in Γ. The set of all polymorphisms with respect to Γ is denoted by Pol(Γ).

Example 35 (Polymorphism examples)**.** Let $g : D^n \to D$ be an operation on D of arity n, (i.e. whose variables and output are D-valued), call it a D-function.

1. g is *idempotent* if for any $d \in D$, $g(d, \cdots, d) = d$.

2. g is *conservative* (or *supportive*) if for any $d_1, \cdots, d_n \in D$, $g(d_1, \cdots, d_n) \in \{d_1, \cdots, d_n\}$.

3. g is *essentially unary* if there exists some k, $1 \leq k \leq n$, and some $\phi : D \to D$, such that for any $d_1, \cdots, d_n \in D$, $g(d_1, \cdots, d_n) = \phi(d_k)$. In particular, if ϕ is an identical map, i.e. $\phi(d) = d$ for any $d \in D$, then g is said to be a *projection.*

4. if $n = 3$, g is said to be a *majority operation* if for any $d_1, d_2 \in D$ we have
$$g(d_1, d_2, d_2) = g(d_2, d_1, d_2) = g(d_2, d_2, d_1) = d_1.$$

5. if $n = 3$, g is said to be an *affine operation* if for any $d_1, d_2, d_3 \in D$ we have $g(d_1, d_2, d_3) = d_1 - d_2 + d_3$ where D is viewed as an Abelian group.

6. if for any $d_1, d_2 \in D$,
$$g(\underbrace{d_1, \cdots, d_1}_{n-1}, d_2) = g(\underbrace{d_1, \cdots, d_1}_{n-2}, d_2, d_1) = \cdots = g(d_2, \underbrace{d_1, \cdots, d_1}_{n-1}),$$
then g is said to be a *weak near unanimity (WNU) operation.* In particular, if
$$g(\underbrace{d_1, \cdots, d_1}_{n-1}, d_2) = g(\underbrace{d_1, \cdots, d_1}_{n-2}, d_2, d_1) = \cdots = g(d_2, \underbrace{d_1, \cdots, d_1}_{n-1}) = d_1,$$
then g is said to be a *near unanimity (NU) operation.*

In the type-less case we are forced to aggregate each issue with the same function, while multi-sorted CSPs give us the freedom to aggregate differently on different types. These are the polymorphisms we need to deal with in our evaluation aggregation setting.

Definition 36 (multi-sorted polymorphism)**.** Fix t and D_b for $1 \leq b \leq t$. Let (X, τ) be a multi-sorted relation with $X \subseteq \prod_{j=1}^{m} D_{\tau_j}$ where $\tau = (\tau_1, \ldots, \tau_m) \in [t]^m$. Fix $n \geq 1$. A collection $f_b : D_b^n \to D_b$ $(1 \leq b \leq t)$ of functions is said to be a multi-sorted polymorphism of (X, τ) if the tuple $(f_{\tau_1}, \ldots, f_{\tau_m})$ is an IIA aggregator for relation X, i.e. it takes X^n into X. More generally, a collection $f_b : D_b^n \to D_b$ $(1 \leq b \leq t)$ of functions is a multi-sorted polymorphism with respect to a set $\{(X_1, \tau^1), \ldots, (X_l, \tau^l)\}$ of multi-sorted relations (each relation is over the same fixed set $[t]$ of types) if $\{f_b\}_{b \in [t]}$ is a multi-sorted polymorphism for each relation.

Definition 37 (MPol)**.** The set of *all* multi-sorted polymorphisms for $\{(X_1, \tau^1), \ldots, (X_l, \tau^l)\}$ is denoted by $\mathrm{MPol}((X_1, \tau^1), \ldots, (X_l, \tau^l))$.

We are now going to state some invariance and composition properties, which are to a large extent responsible for the power of the algebraic approach. For simplicity we state them in the single-sorted setting, indicating whenever a statement has a natural extension to the multi-sorted case.

Lemma 38 (Gadgets preserve polymorphisms)**.** *If R gadget-reduces to $\{X_1, \ldots, X_l\}$ and $f \in \mathrm{Pol}(X_1, \ldots, X_l)$ then $f \in \mathrm{Pol}(R)$.* The statement generalizes to multi-sorted gadgets and multi-sorted polymorphisms.

Proof. Assume that $R \subseteq D^m$. Since R gadget-reduces to $\{X_1, \ldots, X_l\}$ by Definition 28 we have

$$R(x_1, \ldots, x_m) = \exists\, y_1, \ldots, y_{m'} :\ R_1(z_{1,1}, \ldots, z_{1,m_1}) \wedge \ldots \wedge R_t(z_{t,1}, \ldots, z_{t,m_t})$$

where each R_i is one of $X_1, \ldots, X_l$ or the equality relation $x = y$. Furthermore

$$z_{i,j} \in \{x_1, \ldots, x_m, y_1, \ldots, y_{m'}\}.$$

Since $f \in \mathrm{Pol}(X_1, \ldots, X_l)$, thus f keeps $R_1, \cdots, R_t$. Assume that the arity of f is k. For any k vectors $x^{(1)}, \cdots, x^{(k)}$ in R, we need to show that

$$f(x^{(1)}, \cdots, x^{(k)}) \in R.$$

That is, by Definition 28, we need to find some certificate $y_1, \ldots, y_{m'}$ for $f(x^{(1)}, \cdots, x^{(k)})$. For each single $x^{(i)} = (x_1^{(i)}, \cdots, x_m^{(i)}) \in R$, there is a certificate $(y_1^{(i)}, \cdots, y_{m'}^{(i)})$. Thus it can be verified that (by Definition 28)

$$(f(y_1^{(1)}, \cdots, y_1^{(k)}), \cdots, f(y_{m'}^{(1)}, \cdots, y_{m'}^{(k)}))$$

is the required certificate. □

An obvious consequence of Lemma 38 is that $\mathrm{Pol}(X_1, \ldots, X_l)$ and $\mathrm{Pol}(X_1, \ldots, X_l, R)$ are the same. The algebraic approach also fully exploits the following composition properties:

Lemma 39 (Gadgets compose)**.** *If R gadget reduces to $\{X_1, \ldots, X_l\}$ and S gadget reduces to $\{X_1, \ldots, X_l, R\}$ then S also gadget reduces to $\{X_1, \ldots, X_l\}$. Analogous statement holds for multi-sorted gadget reductions.*

Definition 40. For a set Γ of relations let us define the closure of Γ with respect to gadget creation:

$$\langle\Gamma\rangle = \{R \mid \ R \text{ gadget reduces to } \Gamma\}$$

The gadget composition lemma implies that $\langle\langle\Gamma\rangle\rangle = \langle\Gamma\rangle$.

Lemma 41 (Polymorphisms compose)**.** *Let $f, g_1, \ldots, g_n \in \mathrm{Pol}(\Gamma)$, where $f : D^n \to D$ (i.e. f has n variables) and $g_i : D^{n_i} \to D$. Then the $\sum_{i=1}^{n} n_i$ -argument composed function $f(g_1, \ldots, g_n)$ is in $\mathrm{Pol}(\Gamma)$. Analogous statement holds for multi-sorted polymorphisms: from a single n -variate multi-sorted polymorphism and from n arbitrary multi-sorted polymorphisms in $\mathrm{MPol}(\Gamma)$, where Γ is a set of multi-sorted relations, we get a new multi-sorted polymorphism, also in $\mathrm{MPol}(\Gamma)$, where we compose the component functions for each type $b \in [t]$ separately.*

In the above lemma we could have allowed $g_1, \ldots, g_n$ to share variables or even to have allowed repetition of the same variable inside any of the g_is. Note that the following lemma (Lemma 42) is a straightforward consequence of Lemma 41: compose with dictatorships (projections).

Lemma 42 (Identification of variables)**.** *Let $f \in \mathrm{Pol}(\Gamma)$, where $f : D^n \to D$, and $1 \leq i \neq i' \leq n$. Then the function $f' : D^{n-1} \to D$ where we identify the i^{th} and i'^{th} variables of f is also in $\mathrm{Pol}(\Gamma)$. Analogous statement holds for multi-sorted polymorphisms, where we can identify any two variables of the same type.*

Let us call a function of the type $f : D^n \to D$ (i.e. whose variables and output are D-valued) a D-function. A class of D-functions is called a clone if it contains all projections (dictatorships) and is closed under function composition and identification of variables.

For a set $\mathcal{F}$ of D-functions let us define the closure:

$$[\mathcal{F}] = \text{the smallest clone that contains } \mathcal{F}$$

Note that $[\mathcal{F}]$ could have been defined from "inside," i.e. as the set of functions that one can generate from $\mathcal{F}$ and the projections by compositions and identification of the variables. Either ways it follows that $[[\mathcal{F}]] = [\mathcal{F}]$. We also remark that because of the composition theorem for polymorphisms and because projections are always polymorphisms that $\text{Pol}(\Gamma)$ is a clone.

The algebraic theory of constraint satisfaction problems was developed in a sequence of papers by P. Jeavons, A. A. Bulatov, A. A. Krokhin, D. A. Cohen, M. Cooper, M. Gyssens [JCG97, Jea98, JCC98, BJK05]. The theory builds on a connection between relations on D and polymorphisms that aggregate them. For a (possibly infinite) set $\mathcal{F}$ of D-functions let us define

$$\text{Inv}(\mathcal{F}) = \{X \subseteq D^m \text{ for some positive integer } m \mid \mathcal{F} \subseteq \text{Pol}(X)\}$$

Thus $\text{Inv}(\mathcal{F})$ is the set of those relations that are kept by all elements of $\mathcal{F}$.

Theorem 43 (D. Geiger [Gei68])**.** *Fix D, and let Γ, Γ' be (possibly infinite) sets of relations on D and let $\mathcal{F}, \mathcal{F}'$ be (possibly infinite) sets of D-functions. Then*

1. $\text{Inv}(\text{Pol}(\Gamma)) = \langle \Gamma \rangle$

2. $\text{Pol}(\text{Inv}(\mathcal{F})) = [\mathcal{F}]$

3. $\Gamma \subseteq \Gamma' \implies \text{Pol}(\Gamma') \subseteq \text{Pol}(\Gamma)$

4. $\mathcal{F} \subseteq \mathcal{F}' \implies \text{Inv}(\mathcal{F}') \subseteq \text{Inv}(\mathcal{F})$

Definition 44 (Algebra)**.** An *algebra* is a tuple $\mathbf{A} = (A, t_0, t_1, \cdots)$ where A is a nonempty set (called *universe*) and each $t_i : A^{k_i} \to A$, is an operation on A with *arity* $k_i \geq 0$.

Definition 45 (Homomorphism between similar algebras)**.** Two algebras $\mathbf{A}$, $\mathbf{B}$ are *similar* if (they have the same number of operations and corresponding operations have the same arities.) A mapping $f : A \to B$ is a *homomorphism* between two similar algebras $\mathbf{A}$, $\mathbf{B}$ if for all $t_0, t_1, \cdots$ and for all $a_1, a_2, \cdots, a_{k_i} \in A$,

$$t_i^B(f(a_1), f(a_2), \cdots, f(a_{k_i})) = f(t_i^A(a_1, a_2, \cdots, a_{k_i})).$$

A bijective homomorphism is called an *isomorphism.*

Definition 46 (Subalgebra)**.** A set $B \subseteq A$ is a *subuniverse* of an algebra **A** if for all $i = 0, 1, \cdots,$ $t_i(\underbrace{B, \cdots, B}_{k_i}) \subseteq B$. For a nonempty subuniverse B of an algebra **A**, the algebra $\mathbf{B} = (B, t_0|_{B^{k_0}}, t_1|_{B^{k_1}}, \cdots)$ is called a *subalgebra* of **A**. A *term function* of an algebra is any function that can be obtained as a composition using the operations of the algebra together with all the projections. A set $C \subseteq A$ *generates a subuniverse* B in an algebra **A** if B is the smallest subuniverse containing C.

Definition 47 (Direct product, Subdirect product)**.** Given two similar algebras **A**, **B** of the same type, a *direct product* $\mathbf{A} \times \mathbf{B}$ of **A** and **B** is the algebra with universe $A \times B$, and operations are computed coordinatewise. The product of algebras $\mathbf{A}_i$, $i \in I$, is defined in a similar manner for any set I. A *subdirect product* of **A** and **B** is a subalgebra **C** of $\mathbf{A} \times \mathbf{B}$ such that the projections of C to A and B are full. For a set H, an *H-power* $\mathbf{A}^H$ of an algebra **A** has universe A^H (the set of mappings from H to A), and the operations are again computed coordinatewise.

Definition 48 (Congruence, Quotient algebra)**.** An equivalence relation $\sim$ on **A** is called a *congruence* of an algebra **A** if $\sim$ is a subalgebra of $\mathbf{A} \times \mathbf{A}$. An equivalence relation is a congruence if and only if it is the (kernel) of some homomorphism from **A**. If $\sim$ is a congruence of **A** one can form the *quotient algebra* $\mathbf{A}/\sim$: the universe of $\mathbf{A}/\sim$ is $A/\sim$ and the operations are derived from the operations on **A** by taking representatives of the congruence classes.

Definition 49 (Variety)**.** A *variety* is a class of algebras of the same type closed under forming subalgebras, products, and homomorphic images. The smallest variety containing an algebra **A** is called *the variety generated by* ***A***.

Definition 50 (Congruence lattice and type)**.** For an algebra **A**, let $\mathrm{Con}(A)$ denote the set of congruences of **A**. For any $\alpha, \beta \in \mathrm{Con}(A)$, the intersection of α and β is in

Con(A), denoted by $\alpha \wedge \beta$. The smallest congruence relation contains both α and β is also a congruence of $\mathbf{A}$, denoted by $\alpha \vee \beta$. Then $(\text{Con}(A), \wedge, \vee)$ forms a lattice, called the *congruence lattice* of $\mathbf{A}$. It is naturally ordered under inclusion, denoted by $\subseteq$. A pair (α, β) in Con(A) is prime quotient if there is no $\gamma \in \text{Con}(A)$ such that $\alpha \subseteq \gamma \subseteq \beta$. In 1988, Hobby and McKenzie [HM88] developed a theory, tame congruence theory, to "study a local structure of universal algebras through certain properties of prime quotients of the congruence lattice." There are five types of prime quotient for any congruence lattice of algebra $\mathbf{A}$. 1. Unary type, 2. affine type, 3. Boolean type, 4. lattice type, 5. semilattice type. The typeset of a variety is the union of typesets of its finite members.

1.5 Algebraic theory of CSPs and PCP theory

We show the connection developed in [KS09] between algebraic theory of CSPs and PCP theory in this section, in order to introduce the background for Chapter 7, in particular for Theorem 106.

Definition 51. Let D be a finite set. For two functions $f, g : D^n \to D$, the distance between f and g is defined as

$$\text{dist}(f, g) = \mathbb{P}_{x \sim D^n}[f(x) \neq g(x)]$$

where x is uniformly randomly chosen from D^n. Let F be a class of functions of arity n, $F \subseteq \{f : D^n \to D\}$. The distance between F and function $g : D^n \to D$ is defined as:

$$\text{dist}(g, F) = \min_{f \in F} \text{dist}(f, g).$$

In PCP (probabilistically checkable proof) theory, we are interested in testing whether a function $f : D^n \to D$ is in class F, or f is ϵ-far from F (defined as $\forall g \in F, \mathbb{P}_{x \sim D^n}(f(x) \neq g(x)) > \epsilon$). Linear test and long code test are two important examples.

Functions used in the linear test:

$$\{f : D^n \to D | f \text{ is linear over GF}(|D|)\}$$

assume $|D|$ is a prime power.

Functions used in the long code test:

$$\{f : D^n \to D | \exists k, \forall x_1, \cdots, x_n \in D, f(x_1, \cdots, x_n) = x_k\}.$$

For fixed class F, given a function f as input, the goal is to test whether f belongs to F or it is far from F, using a small number of black box queries to f.

1. If $f \in F$, the tester needs to accept with probability 1.
2. If the tester accepts f with probability at least ϵ, then there exists some δ that f is δ-close to F.

We use linear test (on $D = \{0, 1\}$) as an example to illustrate the connection (see [KS09] for more details).

Example 52 (Linear test)**.** In linear test we are interested in testing if $f : \{0, 1\}^n \to \{0, 1\}$ is a linear function or not, i.e. there is some $S \subseteq \{1, \cdots, n\}$ such that

$$f(x_1, \cdots, x_n) = \sum_{i \in S} x_i.$$

The algorithm goes as following [BLR90]: for uniformly randomly chosen $x, y \in \{0, 1\}^n$ test if $f(x \oplus y) = f(x) \oplus f(y)$.

Condition 1. says that if f is linear then the tester needs to accept with probability 1. This is equivalent to saying that f is a polymorphism (see Definition 34) for relation

$$R = \{(x, y, z) | x + y + z = 0 \mod 2\}$$

over $\{0, 1\}$. The following picture depicts the above claim in a more intuitive way. Note that the last row:

$$(f(x), f(y), f(x \oplus y)) \in R$$

is equivalent to $f(x \oplus y) = f(x) \oplus f(y)$.

x	y	$x \oplus y$	
0	1	1	$\in R$
1	0	1	$\in R$
$\vdots$	$\vdots$	$\vdots$	
0	0	0	$\in R$
$f(x)$	$f(y)$	$f(x \oplus y)$	$\in R$

Condition 2. is equivalent to saying that for some randomly chosen $(x^{(1)}, \cdots, x^{(n)}) \in R^n$, if probability that $f(x^{(1)}, \cdots, x^{(n)}) \in R$ is at least ϵ then there exist some δ and some linear function that is δ-close to f. Theorem 53 ensures that Condition 2. is validated.

Theorem 53 (Linear test, [BLR90])**.** *Let $f : \{0,1\}^n \to \{0,1\}$. If there exists some $\epsilon > \frac{1}{2}$ such that*

$$\mathbb{P}_{x,y \sim \{0,1\}^n}[f(x \oplus y) = f(x) \oplus f(y)] \geq \epsilon$$

then f is ϵ close to some linear function.

Chapter 2

Gadgets characterize impossibility

The Galois connection of Geiger sends the message that the more gadgets we can create from a set of relations (or from a single relation), the smaller set of aggregators this set of relations has. An impossibility domain X does not have any non-trivial aggregator, therefore X is expected to generate all relations.[1] Of course, what excites us more is the converse. If we can write all relations as gadgets made from X and some simple relations then X must be an impossibility domain. This chapter makes the above line of ideas precise. The details involve:

1. We will need a multi-sorted version of Geiger's theorem since the polymorphisms we deal with are multi-sorted.
2. We need to understand the role of the Idempotency (Supportiveness) conditions.
3. We want to find a minimal (for algorithmic reasons, but also for convenience) gadget set that already implies the impossibility of X.

2.1 The Idempotency and Supportiveness conditions

The Idempotency and Supportiveness conditions correspond to adding extra relations to our base set of relations from which we build the gadgets that prove the impossibility of X. Our base set is originally $\{X\}$. For the Idempotency condition we add all *Assignment-giving relations* on D and for the Supportiveness condition we add all *Unary relations* on D (see definitions below). This comes from Geiger duality: when the aggregator is required to satisfy extra conditions, the set $\mathcal{F}$ of good aggregators is

[1] To be strict, we should say X^{+} or $X^{\mho}$ instead of X. See Definition 54 and 55 for details.

smaller, so $\mathrm{Inv}(\mathcal{F})$ is larger, therefore it has to contain other generators than X. The details are as follows.

Definition 54 (Unary relations, $X^{\mho}$, $\Gamma^{\mho}$)**.** A unary relation on D is a nonempty subset D' of D. If X is a relation on D, then $X^{\mho}$ denotes the set of relations that consists of X and all the unary relations on D. If Γ is a set of relations on D, then $\Gamma^{\mho}$ is the set of relations that besides the elements of Γ also contains all unary relations on D. In the multi-sorted case, we add the unary relations for all types i.e. all (D', τ), where D' is any unary relation on D_τ and τ is any element of $[t]$.

Definition 55 (Assignment-giving relations, X^+, Γ^+)**.** There are special unary relations, called assignment-giving-relations, of the form $x = v$, where v is some element of D. Expressed as a set, $x = v$ is simply $\{v\}$. The binary ($D = \{0, 1\}$) assignment giving relations are $x = 0$ and $x = 1$. If X is a relation on D then we will denote with X^+ the set of relations that includes X and all assignment giving relations on D. Similarly, if Γ is a set of relations on D then Γ^+ denotes the set of all relations that we obtain from Γ by adding all assignment giving relations on D to it. Multi-sorted assignment-giving relations are unary relations of the form $(x = v, (a))$, where $v \in D_a$. If X is a multi-sorted relation, then X^+ is the set of relations that includes X and all assignment giving relations for the type set $[t]$, $\{D_b\}_{b\in[t]}$ on which X is defined. Similarly we can define Γ^+ for a set of multi-sorted relations Γ.

Lemma 56. *A function $g : D^n \to D$ is idempotent if and only if it aggregates all assignment giving relations on D. A function $g : D^n \to D$ is supportive if and only if it aggregates all unary relations on D. The above generalizes to the multi-sorted case: Fix positive integer n. A multi-sorted function $g = (g_1, \ldots, g_t)$, where $g_b : D_b^n \to D_b$ is an aggregator for type b, is idempotent, if it aggregates all assignment giving relations for all types. A multi-sorted function $g = (g_1, \ldots, g_t)$, where $g_b : D_b^n \to D_b$ is an aggregator for type b, is supportive, if it aggregates all unary relations for all types.*

We omit the simple proof of this statement. Assume now that we have a multi-sorted relation $X = (X, \tau)$ and we would like to describe the set $\mathcal{F}$ of all idempotent multi-sorted polymorphisms for X. By Lemma 56, $\mathrm{Inv}(\mathcal{F})$, besides X, has to include all assignment giving relations on D, and this condition is also sufficient. Similar thing holds in the supportive case with "assignment giving" replaced with "unary". Therefore $\mathcal{F}$ is characterized as the largest set of functions whose Inv contains X^+ (or $X^{\mho}$ in the case of Supportiveness condition).

2.2 Characterization of Impossibility Domains in terms of gadgets

Let $\mathcal{F}$ be any family of multi-sorted functions for some fixed type set $[t]$. The "Non-dictatorship" condition for $\mathcal{F}$ i.e. that $\mathcal{F}$ contains at least one non-dictator function, turns out to be equivalent (see next two sections) to that at least one of the members of a certain prescribed finite set of relations is not present in $\mathrm{Inv}(\mathcal{F})$. In particular, if $\mathcal{F} = \mathrm{MPol(X^+)}$ (or $\mathrm{MPol(X^{\mho})}$), our multi-sorted Geiger's theorem will give that as long as X is an impossibility domain with respect to the Idempotency (Supportiveness) conditions, we must be able to generate the elements of the above crucial set of relations from X^+ ($X^{\mho}$, respectively) as gadgets. As a first step to this, in the next section we will prove the following:

Lemma 57. *Domain $X \subseteq D^m$ is an impossibility domain with respect to IIA + Idempotency (Supportiveness) + Non-dictatorship if and only if* all *multi-sorted relations can be generated as multi-sorted X^+ ($X^{\mho}$) -gadgets. Here $t = m$, $D_j = \mathrm{pr}_j X$ for $1 \leq j \leq m$ and the typing of X is $(X, (1, \dots, m))$.*

The set of all (multi-sorted) relations is however infinitely large! Luckily, we can select a finite subset (actually, in many different ways) that generate all relations, and it is sufficient to consider only those.

Definition 58 (The Non-Binary OR relation)**.** We define the multi-sorted relation $R^{u,v}_{k,\ell}$ which is unsatisfied only when $x = u$ and $y = v$ simultaneously hold (x has type k and v has type ℓ). In formula:

$$R^{u,v}_{k,\ell}(x, y) = (\neg(x = u \wedge y = v), (k, \ell)).$$

Definition 59 (The multi-sorted Not-All-Equal relation)**.** The multi-sorted NAE relation on types a, b, c is defined as

$$\mathrm{NAE}_{a,b,c} = (\{(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0)\},(a,b,c)).$$

We are now ready to state our characterization of impossibility domains X, which we shall prove in the subsequent two sections from the multi-sorted Geiger's theorem.

Theorem 60. *Let $X \subseteq D^m$ be non-degenerate. Let $t = m$ and $\tau = (1, \ldots, m)$, and the effective set of values (see Section 1.2) for type j is $D_j = \mathrm{pr}_j X$. Then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship if and only if there are $(X,\tau)^+$-gadgets expressing $R^{u,v}_{k,\ell}$ for every $1 \le k, \ell \le m$; $u \in \mathrm{pr}_k X$, $v \in \mathrm{pr}_\ell X$. Furthermore, if $|D_j| = 2$ for some $1 \le j \le m$, we also need to add the multi-sorted* NAE *gadget on types (j, j, j). The analogous statement, when we replace "Idempotency" with "Supportiveness", requires to replace $(X,\tau)^+$ with $(X,\tau)^{\mho}$.*

2.3 A multi-sorted version of Geiger's theorem

In the proof of Theorem 60 we will need a multi-sorted version of Geiger's theorem. The goal of this section is to extend Geiger's theorem to the multi-sorted setting. For what follows we fix a type set $[t]$. For a set Γ of multi-sorted relations (with all types from $[t]$) we define:

$$\langle\Gamma\rangle_M = \{(R,\tau) \mid \ (R,\tau) \text{ multi-sorted gadget reduces to } \Gamma\}$$

Here 'M' in $\langle\Gamma\rangle_M$ means "multi-sorted". We will omit the subscript when this is clear from the context. The notion MPol was defined in Definition 37. The notion MInv is defined analogously to Inv in the multi-sorted setting. If $\mathcal{F}$ is a (possibly infinite) set of multi-sorted D-functions (i.e. set of $f : D^n \to D$) then MInv($\mathcal{F}$) is the set of all multi-sorted relations that are kept by all elements of $\mathcal{F}$.

Now we can state our multi-sorted Geiger theorem:

Theorem 61. *Fix $[t]$ and $\{D_b\}_{b\in[t]}$, and let Γ, Γ' be (possibly infinite) sets of multi-sorted relations and let $\mathcal{F}, \mathcal{F}'$ be (possibly infinite) sets of multi-sorted aggregators. Then*

1. $\mathrm{MInv}(\mathrm{MPol}(\Gamma)) = \langle\Gamma\rangle$

2. $\Gamma \subseteq \Gamma' \implies \mathrm{MPol}(\Gamma') \subseteq \mathrm{MPol}(\Gamma)$

3. $\mathcal{F} \subseteq \mathcal{F}' \implies \mathrm{MInv}(\mathcal{F}') \subseteq \mathrm{MInv}(\mathcal{F})$

Proof. In order to make the translation of our multi-sorted version to the single-sorted Geiger theorem (Theorem 43) we first define

$$D = D_1 \dot{\cup} D_2 \dot{\cup} \cdots \dot{\cup} D_t$$

where D_b is the range for type b. If originally the D_bs are not disjoint, we make them disjoint without the loss of generality. A non-empty multi-sorted relation $X \subseteq D_{\tau_1} \times \cdots \times D_{\tau_m}$ can be now interpreted as the single-sorted relation $X_D \subseteq D^m$. We remark that viewing them as sets, X and X_D are exactly the same. The index D in X_D is only a reminder that we view X_D as a single sorted relation over domain D, while we view X as (X, τ). Since the D_bs are disjoint, from any such $X_D \neq \emptyset$ we can recover the types of every coordinate (the components of a single element of X_D already give this information). If Γ is a set of multi-sorted relations over a fixed type-set $[t]$, let Γ_D be the set of those X_Ds that $X \in \Gamma$.

For $b \in [t]$ we introduce the unary relation T_b on D (in the single sorted world):

$$T_b(u) \longleftrightarrow u \in D_b$$

In other words, $T_b(u)$ expresses that "u has type b in the multi-sorted world." For the set $\{T_1, \ldots, T_t\}$ of relations we introduce the notation Θ.

Definition 62. Let Δ be any set of relations on D such that $\Theta \subseteq \Delta$. Then for any polymorphism $f : D^n \to D$ of Δ and any $b \in [t]$ we can define $f_b : D_b^n \to D_b$ as $f_b(x) = f(x)$ on D_b^n.

Note. We know that f on D_b^n takes value from D_b since it is an aggregator of $T_b \in \Delta$.

We have now:

Lemma 63. *Let Γ be any set of multi-sorted relations over a fixed type-set $[t]$ and with the notations as before. Then the following are equivalent:*

1. *$(f_1, \ldots, f_t)$ is a multi-sorted polymorphism for Γ;*

2. *the sequence $f_1, \ldots, f_t$ arises, as in Definition 62, from some polymorphism f of $\Delta = \Gamma_D \cup \Theta$.*

We do not prove this easy lemma. Returning to the proof of Theorem 61, the only challenge is to prove 1. since 2. and 3. are obvious. It is also follows from known composition lemmas (polymorphisms compose, as do gadgets) that

$$\mathrm{MInv}(\mathrm{MPol}(\Gamma)) \supseteq \langle\Gamma\rangle$$

We have to show the containment in the other direction. Theorem 43 gives that

$$\mathrm{Inv}(\mathrm{Pol}(\Gamma_D \cup \Theta)) = \langle\Gamma_D \cup \Theta\rangle_D$$

Here the subscript in $\langle\cdot\rangle_D$ refers to that gadget generation is taken in the single sorted world. Our proof scheme is to relate $\langle\Gamma\rangle$ to $\langle\Gamma_D \cup \Theta\rangle_D$ and $\mathrm{MInv}(\mathrm{MPol}(\Gamma))$ to $\mathrm{Inv}(\mathrm{Pol}(\Gamma_D \cup \Theta))$.

Which non-empty relations can we generate from $\Gamma_D \cup \Theta$, i.e. what are the elements of $\langle\Gamma_D \cup \Theta\rangle$? A gadget with this set of generators is of the form

$$R(x_1, \ldots, x_k) = \exists\, y_1, \ldots, y_{k'} :\ R_1(z_{1,1}, \ldots, z_{1,k_1}) \wedge \ldots \wedge R_p(z_{p,1}, \ldots, z_{p,k_p}) \qquad (2.1)$$

where each R_i is either from Γ_D or from Θ or the $=$ relation.

Definition 64. We say that (an x- or y-) variable z in (2.1) has type b if whenever the conjunct holds for an assignment (without the existential quantifier), z has a value in D_b. In different words, adding $T_b(z)$ to the conjunction would not eliminate any of the satisfying assignments of the conjunct.

Since a $\Gamma_D \cup \Theta$ gadget (recall, this is a single-sorted gadget) has variables only over D, there is no a priori type restriction (other than the entire D) on any variable.

Nevertheless, if a variable z is involved in a relation S_D, where S is a relation from Γ, then S gives a type b to that variable: S_D never holds if $z \notin D_b$, so we might as well restrict z to D_b. Assume now that there is a variable z' such that the right hand side of (2.1) contains the relation $z = z'$ with an already restricted z. Then z' must also be from D_b. Of course, a chain of such equations also enforces a type on the variable in the end of the chain. Finally, any relation $T_b(z)$ enforces a type b on z. In summary, we can assign type b to variable z if

1. $T_b(z)$ occurs in the gadget;
2. z occurs in a constraint from Γ_D with type b;
3. There is a chain of equality relations that starts form any variable restricted to type b (by 1. or 2.) that leads to z.

We note that a variable cannot have two different (i.e. contradicting) types defined this way. Any contradiction in types would make R unsatisfiable (i.e. the empty relation). Therefore R is equivalent to a direct product of a "typed part" and an "untyped part":

$$R = R_{\text{typed}} \times \underbrace{(w_{1,1} = \cdots = w_{1,s_1}) \times \cdots \times (w_{l,1} = \cdots = w_{l,s_l})}_{\text{untyped part}}$$

The $w_{i,j}$s are variables and the untyped (or typed) part might not be there. Point 1.-3. also give that R_{typed} is a syntactically recognizable part of R that arises by deleting some variables and terms from the right hand side. Although R_{typed} is a Γ_D-gadget (recall that Γ_D is an *untyped* set of relations constructed from the *typed* set of relations, Γ), because of its syntax we can also read it as a (typed) Γ-gadget, and in addition with the exact same semantics (meaning that R_{typed} as a set is the exact same set as when we read the formula as a Γ-gadget). So R_{typed} [viewed as a multi-sorted relation] $\in \langle\Gamma\rangle$.

Next we claim that

$$R \in \text{MInv}(\text{MPol}(\Gamma)) \quad \longrightarrow \quad R_D \in \text{Inv}(\text{Pol}(\Gamma_D \cup \Theta)).$$

The left hand side reads that R is kept by all multi-sorted polymorphism $(f_1, ..., f_t) \in$ MPol(Γ)). By Lemma 63 we have that $f \in \text{Pol}(\Gamma_D \cup \Theta)$ if and only if its component-sequence (as in Definition 62) $(f_1, ..., f_t) \in \text{MPol}(\Gamma)$. So R_D is kept by all $f \in \text{Pol}(\Gamma_D \cup$

Θ), proving our claim.

By the single sorted Geiger theorem (applied to $\Gamma_D \cup \Theta$) we have then that since $R_D \in \mathrm{Inv}(\mathrm{Pol}(\Gamma_D \cup \Theta))$, we also have that $R_D \in \langle \Gamma_D \cup \Theta \rangle$. Two paragraphs earlier we have seen, that then R_D must be of the form $R_{\mathrm{typed}} \times (\alpha_{1,1} = \cdots = \alpha_{1,s_1}) \times \cdots \times (\alpha_{l,1} = \cdots = \alpha_{l,s_l})$. But since all component of R_D are typed, only the typed part is there ($R_D = R_{\mathrm{typed}} = R$ as sets). So $R \in \langle \Gamma \rangle$. □

2.4 Special gadgets and their powers

In this section we make some observations about the power of special gadgets and also finish proving Theorem 60. In the previous sections we have proven that the more gadgets we can create from X^+ or $X^{\mho}$ the more restrictions must hold for $\mathrm{MPol}(X^+)$ and $\mathrm{MPol}(X^{\mho})$ respectively. In particular, it is sufficient to create all gadgets required by Theorem 60 in order to conclude that $\mathrm{MPol}(X^+)$ and $\mathrm{MPol}(X^{\mho})$ have only dictatorships. But even just a single gadget present in $\langle X^+ \rangle$ or $\langle X^{\mho} \rangle$ can have strong consequences. For instance, if we can construct the equality gadget $(x = y, (k, \ell))$ between two different types k and ℓ (with a common alphabet), any idempotent (or supportive, if our gadgets use relations from $\langle X^{\mho} \rangle$) multi-sorted polymorphism $(f_1, \ldots, f_t)$ that aggregates X must have the same k and ℓ components.

Once we start to create gadgets we can use them as "subroutines" to create even more gadgets. The following lemma says that for any fixed $1 \le k, \ell < t$ from $R^{u,v}_{k,\ell} \in \langle \Gamma \rangle$ one can construct *all* multi-sorted binary relation with type (k, ℓ) in $\langle \Gamma \rangle$.

Lemma 65. *Let* $1 \le k, \ell \le t$ *and* $(S, (k, \ell))$ *be any multi-sorted relation contained in* $\mathrm{pr}_k X \times \mathrm{pr}_\ell X$. *Then* $(S, (k, \ell)) = \wedge_{(u,v) \notin S} R^{u,v}_{k,\ell}$.

We omit the straightforward proof. Note that nothing stops us setting $k = \ell$ in the lemma, in which case we get all binary relations ($R \subseteq D_k^2$). What does the presence of these relations in $\langle \Gamma \rangle$ say about the (multi-sorted) polymorphisms for Γ? They say a lot. In fact, if D_k has size at least three, already the not-equal relation, $(x \neq y, (k, k))$, alone excludes all idempotent (and so all supportive) polymorphisms other than the dictatorships *for that type*:

Lemma 66 (Not-equal gadget lemma)**.** *Assume that the multi-sorted non-equality relation* $((x \neq y), (k, k)) \subseteq \langle\Gamma\rangle$*. Then for every* $f = (f_1, \cdots, f_t) \subseteq \mathrm{MPol}(\Gamma)$ *it must hold that* f_k *is a dictatorship on* $\mathrm{pr}_k X$*.*

The above lemma with Lemma 65 imply

Lemma 67. *Assume that* $|\mathrm{pr}_k X| \geq 3$ *and the conditions of Theorem 60 hold, so we can create all* $R^{u,v}_{k,\ell}$ *from* $(X, \tau)^+$ *(resp. from* $X, \tau)^{\mho}$*). Then the* k^{th} *component of any idempotent (supportive) aggregator* $f = (f_1, \cdots, f_t)$ *of* X *must be a dictatorship.*

What if $|\mathrm{pr}_k X| = 2$? Then the NAE gadget can be used to take care of the same thing.

Lemma 68 (Not-all-equal gadget lemma)**.** *Assume that the multi-sorted not-all-equal relation* $(|x, y, z| > 1 \ \wedge \ x, y, z \in \mathrm{pr}_k X, \ (k, k, k))$ *gadget-reduces to a set* Γ *of multi-sorted relations. Then for every* $f = (f_1, \cdots, f_t) \in \mathrm{MPol}(\Gamma)$ *it must hold that* f_k *is a dictatorship on* $\mathrm{pr}_k X$*.*

Among the conditions of Theorem 60 one explicitly states that in the case when $|\mathrm{pr}_k X| = 2$ the $(|x, y, z| > 1 \wedge x, y, z \in \mathrm{pr}_k X, \ (k, k, k))$ gadget can be constructed from $(X, \tau)^+$ (resp. from $X, \tau)^{\mho}$) to make Lemma 68 applicable.

Putting Lemmas 67 and 68 together, we get that if the gadgets promised by Theorem 60 are present, then all components of every multi-sorted polymorphism $(f_1, \ldots, f_m)$ are dictatorships on the respective $\mathrm{pr}_j X$s (not necessarily to the same coordinate).

To finish the "if" part of Theorem 60 all we need to show is that these dictatorships are controlled by the same index K (the dictator). Assume this is not the case, and let k and ℓ be components such that f_k is dictated by the K^{th} voter and f_ℓ is dictated by voter $L \neq K$. Because X is non-degenerate (and so $\mathrm{pr}_k X$ and $\mathrm{pr}_\ell X$ have size at least two) and because of König's theorem (or simply by basic combinatorics) there ought to be $u, v \in \mathrm{pr}_k X$ and $u', v' \in \mathrm{pr}_\ell X$ such that

1. $u \neq v$ and $u' \neq v'$;

2. There is an element U of X that takes u on k and u' on ℓ;

3. There is an element V of X that takes v on k and v' on ℓ.

Let us now aggregate a set of votes, all from $\{U, V\}$, but the K^{th} vote is U and the L^{th} vote is V. Then $(f_1, \ldots, f_m)$ aggregates this input to Z such that the k^{th} issue aggregates to u and the ℓ^{th} issue aggregates to v'. Notice now that $R_{k,\ell}^{u,v'}$ holds for U and V, but not for Z, which is a contradiction, since every aggregator must keep all gadgets constructible from X^+ $(X^{\mho})$, in particular $R_{k,\ell}^{u,v'}$. This concludes the "if" (non-trivial) part of the proof of Theorem 60.

In the rest of the section we assume that for two types $a, b \in [t]$ we have $D_a = D_b$ (equivalently, a 1-1 correspondence '=' between D_a, D_b). We show that the ability to construct the '=' gadget between types a and b from a set Γ of (multi-sorted) relations implies that for every polymorphism $(f_1, \ldots, f_t)$ of Γ we have $f_a = f_b$. This can be useful, because the algebraic theory is developed mainly for single-sorted polymorphisms.

Lemma 69 (Equality gadget lemma)**.** *Let Γ be a set of multi-sorted relations with type set $[t]$. Assume that for types a and b we have $D_a = D_b$, and that the multi-sorted equality relation $(x = y,\ (a, b))$ gadget-reduces to Γ. Then for every $f = (f_1, \cdots, f_t) \in$ $\mathrm{MPol}(\Gamma)$ it must hold that f_a is identical to f_b.*

Proof. We need to prove that for every $u = (u^1, \ldots, u^n) \in D_a^n$ $(= D_b^n)$ it holds that $f_a(u) = f_b(u)$. Consider an arbitrary $u \in D_a^n$. f is a multi-sorted polymorphism of $(x = y,\ (a, b))$. This follows from the facts that $f = (f_1, \cdots, f_t)$ is a polymorphism of Γ and that $(x = y,\ (a, b))$ gadget-reduces to Γ. Therefore, since each line of the table

$$\begin{array}{ccc}
\text{type } a & & \text{type } b \\
 & & \\
u_1 & = & u_1 \\
u_2 & = & u_2 \\
 & \vdots & \\
u_n & = & u_n \\
\hline
f_a(u) & = & f_b(u)
\end{array}$$

above the solid horizontal line satisfies the $(x = y,\, (a, b))$ relation, we can apply polymorphism f for the two columns of the table. Now, as discussed, f must keep the relation $(x = y,\, (a, b))$, so $f_a(u) = f_b(u)$. □

2.5 Example: Arrow's Theorem

Let $\mathcal{A} = \{A_1, \ldots, A_k\}$ denote a set of k items and let S_k be the domain that corresponds to the set of $k!$ different linear orders on $\mathcal{A}$ in a way we describe below. Each voter must vote for some linear order and the votes have to be aggregated into a single linear order, the "choice of the society."

Instead of thinking of a linear order as is (like $A_2 < A_3 < A_4 < A_1$) we rather represent it as a sequence of $\binom{k}{2}$ binary positions corresponding to questions of the form

$$A_1 < A_2? \ , \ A_1 < A_3? \ , \ \ldots \ , \ A_{n-1} < A_n?$$

(see also Section 1.1). When '$<$' is a linear order on $\mathcal{A}$, the answers to these questions (0 = no; 1 = yes) uniquely (and even redundantly) encodes '$<$' in the form of a valid *evaluation.* The set of all valid evaluations (these are binary vectors of length $\binom{n}{2}$) is exactly the relation $S_k \subseteq \{0,1\}^{\binom{n}{2}}$. Arrow famously shows:

Theorem 70 (Arrow [Arr50]). *When $k \geq 3$, there is no aggregator $f : S_k^n \to S_k$ for any $n \geq 2$ that satisfies IIA + Idempotency + Non-dictatorship.*

It is easy to see that the impossibility of S_3 implies the impossibility of S_k for $k \geq 3$. Below we are going to prove Theorem 70 when $k = 3$, using our method of gadgets. In the $A_1 < A_2$? $A_2 < A_3$? $A_3 < A_1$? basis (replacing '$A_1 < A_3$?' with '$A_3 < A_1$?' does not change the problem) we have that:

$$S_3 = \{001, 010, 011, 100, 101, 110\} = \text{NAE} \qquad \text{(Not All Equal)}$$

Since Arrow allows to aggregate different coordinates with different aggregators, we view S_3 as a multi-sorted relation with type-set [3]. We need to show that the only polymorphisms of the multi-sorted relation set S_3^+ are projections (dictatorships). In section 1.3 we have created a gadget for the relation $(\neg(x = y), (1, 3))$. Let us denote it by R_{13}. By symmetry, S_3^+-gadgets also exist for $R_{12} = (\neg(x = y), (1, 2))$ and $R_{23} = (\neg(x = y), (2, 3))$. Then the gadget

$$\exists x_2 : \; R_{12}(x_1, x_2) \wedge R_{23}(x_2, x_3)$$

expresses $(x_1 = x_3, (1, 3))$ (see Fig. 2.1). We can similarly express any $(x_a = x_b, (a, b))$ for $1 \leq a < b \leq 3$. Once we have generated these relations, by Lemma 69 we conclude that every multi-sorted polymorphism of S_3^+ must be a single sorted polymorphism i.e. of the form (f_1, f_1, f_1), and it is well known that all single-sorted idempotent polymorphisms of NAE are dictatorships.

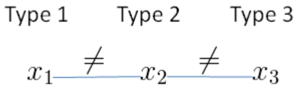


Figure 2.1: The gadget expressing $x_1 = x_3$.

2.6 Example: Pairwise distinctness

We continue to illustrate how one can use hand-made gadgets to prove impossibility theorems for specific domains:

Theorem 71. *Let D and m such that $|D| > m$ when $m = 2$ or $|D| = m$ when $m \geq 3$. Define*

$$X = \{(x_1, \ldots, x_m) \in D^m \mid x_1, \ldots, x_m \text{ are pairwise distinct } \}$$

Then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship conditions.

In [DH10b], this theorem is proven under the IIA + Supportiveness + Non-dictatorship conditions. A special case of this theorem, i.e. when $|D| = m \geq 3$, can be derived from results in [FF11]. We give a gadget proof for it when $m = 2$ and $|D| = 3$, which is in a sense the hardest setting of the parameters. In this case the relation $X \subseteq [3]^2$

is a binary relation of type $(1, 2)$, and we have to show that MPol(X^+) contains only projections (dictatorships).

We remark that the problem in this case is essentially equivalent to showing that three coloring of bipartite graphs is NP-hard as long as assignment giving constraints are also allowed, i.e. we are allowed to specify that "vertex v has color c". If we drop the bipartite condition, then the problem is well-known to be NP hard. The bipartite condition comes from the multi-sorted nature of the problem: relation X can connect only type 1 variables with type 2 variables.

First we create a gadget for the relation $(\neg(x = y = 2), (1, 1))$. Clearly, by symmetry, then gadgets also exist for any $(\neg(x = y = a), (b, b))$, where $1 \leq a \leq 3$ and $1 \leq b \leq 2$.

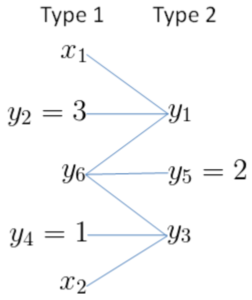


Figure 2.2: The inequality gadget

The gadget in Fig. 2.2 corresponds to the following formula:

$$R(x_1, x_2) \quad = \quad \exists y_1, \ldots, y_6 : \; X(x_1, y_1) \wedge X(x_2, y_3) \wedge X(y_2, y_1) \wedge X(y_4, y_3) \wedge X(y_6, y_1) \wedge$$
$$\wedge X(y_6, y_3) \wedge X(y_6, y_5) \wedge (y_2 = 3) \wedge (y_4 = 1) \wedge (y_5 = 2)$$

It is not hard to check that R implements the $(\neg(x = y = 2), (1, 1))$ relation. Let us now denote the relation $(\neg(x = y = a), (b, b))$ by R_a^b (i.e. the relation $R_{b,b}^{a,a}$ defined in Definition 58). Then $R_1^b(x, y) \wedge R_2^b(x, y) \wedge R_3^b(x, y)$ expresses the $(x \neq y, (b, b))$ relation. Finally, the gadget

$$T(x_1, x_2) = \exists y_1, y_2 : \; (x_1 \neq y_1) \wedge (x_1 \neq y_2) \wedge (y_1 \neq y_2) \wedge X(y_1, x_2) \wedge X(y_2, x_2)$$

expresses $(x_1 = x_2, (1, 2))$ (see Fig. 2.3). Once we have generated this relation, by Lemma 69 we conclude that every multi-sorted polymorphism of X^+ must be a single sorted polymorphism i.e. of the form (f_0, f_0). It is well known that the only idempotent polymorphisms of the $x \neq y$ (single-sorted) relation are the dictatorships.

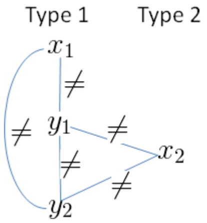


Figure 2.3: The equality gadget

Chapter 3

Binary evaluations

In this chapter we first translate the total blockedness condition to the algebraic language, then using it revisit the case of binary evaluations, give an alternative proof to the classification Theorem of E. Dokow, R. Holzman, and also indicate the proof of Theorem 20.

3.1 Total blockedness and its consequences

Lemma 72. *Let $X \subseteq \{0,1\}^m$ be totally blocked (see Definition 10) and non-degenerate. Let (X,τ) be any typing of the variables of X from a type-set $[t]$, $D_a = \{0,1\}$ for $1 \leq a \leq t$. We also assume that all types are used. Then for all $1 \leq a, b \leq t$: $(x = y,\ (a,b))$ (multi-sorted) gadget-reduces to X^+.*

Proof. Recall that the total blockedness condition means that on the vertex set $V = [m] \times \{0,1\}$ we have a strongly connected graph defined as follows: There is a directed edge from $(k,\epsilon) \in V$ to $(\ell,\epsilon') \in V$ where $k \neq \ell$ if and only if there are: (i.) a subset $S \subseteq [m]$ such that $k, \ell \in S$ and (ii.) a (partial-)evaluation $u : S \to \{0,1\}$ with $u_k = \epsilon$ and $u_\ell = 1 - \epsilon'$ such that there is no extension of u to any full evaluation x in X, but if we flip any bit of u then the resulting partial evaluation extends to some element of X. Let us focus on a directed edge $((k,\epsilon),(\ell,\epsilon'))$ as above, with $S = \{k, \ell, s_1, \ldots, s_q\}$ (we fix this S for $k, \ell, \epsilon, \epsilon'$), and create the gadget

$$E_{k,\ell,\epsilon,\epsilon'}(x_k, x_\ell) = \exists y_{s_1}, \ldots, y_{s_q}, \vec{y} :\ X(x_k, x_\ell, y_{s_1}, \ldots, y_{s_q}, \vec{y}) \wedge (y_{s_1} = u_{s_1}) \wedge \ldots \wedge (y_{s_q} = u_{s_q})$$

Here with some abuse of the notation we tried to indicate, via the indices, the variables' positions in X. In particular, $\vec{y}$ collects the $m - 2 - q$ variables of X that are not in

$S = \{k, \ell, s_1, \dots, s_q\}$. We remark that the type of a variable is uniquely determined by its position in X. Then we have

$$E_{k,\ell,\epsilon,\epsilon'}(\epsilon, \epsilon') = 1, \quad E_{k,\ell,\epsilon,\epsilon'}(\epsilon, 1-\epsilon') = 0, \quad E_{k,\ell,\epsilon,\epsilon'}(1-\epsilon, 1-\epsilon') = 1$$

All three equations follow from the fact that $u : S \to \{0, 1\}$ with $u_k = \epsilon$ and $u_\ell = 1 - \epsilon'$ was a minimally unsatisfying partial assignment, thus if we change the value of exactly one of the u_k, u_ℓ, the assignment becomes satisfying. Consider now a chain

$$((k^0, \epsilon^0), (k^1, \epsilon^1)),\ ((k^1, \epsilon^1), (k^2, \epsilon^2)),\ \dots, ((k^{t-1}, \epsilon^{t-1}), (k^t, \epsilon^t))$$

of edges in the blockedness graph, for which we have generated relations

$$E_{k^0,k^1,\epsilon^0,\epsilon^1}, \dots, E_{k^{t-1},k^t,\epsilon^{t-1},\epsilon^t}$$

as above. Create the gadget

$$R(x_{k^0}, x_{k^t}) = \exists y_{k^1}, \dots, y_{k^{t-1}} :\ E_{k^0,k^1,\epsilon^0,\epsilon^1}(x_{k^0}, y_{k^1}) \wedge \dots \wedge E_{k^{t-1},k^t,\epsilon^{t-1},\epsilon^t}(y_{k^{t-1}}, x_{k^t})$$

It is easy to see that the typing is consistent. If we set $x_{k^0} = \epsilon^0$ then inductively all y_{k_i} variables are forced to take ϵ^i, eventually forcing $x_{k^t} = \epsilon^t$. On the other hand $R(1-\epsilon^0, 1-\epsilon^t) = 1$. To show this it is sufficient to set all y_{k_i} variables to $1 - \epsilon^i$ in the right hand side of the above formula. Since the blockedness graph is strongly connected, for any $a, b \in [t]$ with $\text{type}(x_k) = a$, $\text{type}(x_\ell) = b$ for some $1 \le k, \ell \le m$ and for any $\epsilon_k, \epsilon_\ell \in \{0, 1\}$ we can build now gadget $R_{a,b,\epsilon,\epsilon'}$ that forces $x_\ell = \epsilon_\ell$ as long as $x_k = \epsilon_k$ and also permits the $x_k = 1 - \epsilon_k$, $x_\ell = 1 - \epsilon_\ell$ assignment, and a gadget $R'_{a,b,1-\epsilon,1-\epsilon'}$ that forces $x_\ell = 1 - \epsilon_\ell$ as long as $x_k = 1 - \epsilon_k$ and also permits the $x_k = \epsilon_k$, $x_\ell = \epsilon_\ell$ assignment. Then $R_{a,b,\epsilon,\epsilon'} \wedge R'_{a,b,1-\epsilon,1-\epsilon'}$ implements $x_k + \epsilon_k = x_\ell + \epsilon_\ell \mod 2$. In particular, by choosing $\epsilon_k = \epsilon_\ell = 0$ we have implemented the $(x = y, (a, b))$ relation. □

Example 73. Let $X = \{000, 011, 101, 110\} = \{xyz \in \{0,1\}^3 \mid x + y + z = 0 \mod 2\}$. We also assume that $[t] = 3$ and the i^{th} coordinate has type i. Then the minimally infeasible partial evaluations (MIPEs) are all those $xyz \in \{0,1\}^3$ for which $x + y + z = 1 \mod 2$. Then the blockedness graph is the directed complete graph (i.e. directed edges are drawn both ways for every edge). For $k = 1,\ \ell = 2,\ \epsilon_k = 0, \epsilon_\ell = 1$ we can create the gadget (based on $S = \{1, 2, 3\}$, $u = 001$):

$$E_{1,2,0,1}(x_1, x_2) = \exists y :\ X(x_1, x_2, y) \wedge (y = 1)$$

This together with $E_{1,2,1,0}$ created from $S = \{1, 2, 3\}$, $u = 111$ (accidentally, the two gadgets turn out to be the same, so the conjunction remains $E_{1,2,0,1}(x_1, x_2)$) gives the $(x \neq y, (1, 2))$ relation, as one can check it directly.

Lemma 74. *Let $X \subseteq \{0, 1\}^m$ be totally blocked. Then every aggregator $f = (f_1, \ldots, f_m)$ which is IIA + Idempotent satisfies the condition that all f_j's are identical.*

Proof. Let $t = m$ and view X as the multi-sorted relation $(X, (1, 2, \ldots, m))$. Then $(x = y,\ (k, \ell))$ $(1 \leq k, \ell \leq m)$ gadget-reduces to X^+ by Lemma 72. Since f is an idempotent IIA aggregator of X we have that $f \in \mathrm{MPol}(X^+)$. Combining the above two things the statement then follows from Lemma 69. □

We remark (although do not use it in the sequel) that total blockedness also generates all non-equal relations:

Lemma 75. *Let $X \subseteq \{0, 1\}^m$, totally blocked. and X has type $(X, (1, \ldots, m))$. Then X (as a multi-sorted relations) generates all relations of the form $(x \neq y, (a, b))$, where $a, b \in [t]$.*

3.2 A new proof of Theorem 12

We give a new proof of the interesting part of Theorem 12 (E. Dokow and R. Holzman, [DH10a]), i.e. that when X is totally blocked, it is a possibility domain (with respect to IIA+Idempotency+No Dictatorship) if and only if it is an affine subspace. The new proof uses results from the algebraic theory of constraint satisfaction problems.

Lemma 74 gives that when X is totally blocked, all idempotent multi-sorted polymorphisms of X are single-sorted and we can use Schaefer's theorem[1], or more precisely

[1] Historically, it is a consequence of E. Post [Pos41]. We thank Andrei Bulatov for pointing out this.

algebraic version of it (Hubie Chen [Che09]) to determine the types of functions in $\mathrm{Pol}(X)$:

Theorem 76 (Schaefer, algebraic version)**.** *Let $D = \{0, 1\}$ and Γ be a set (single-sorted) relations on D. Then Γ^+ either has one of the following four operations as a polymorphism:*

1. *The binary AND operation $\wedge$;*

2. *The binary OR operation $\vee$;*

3. *The ternary majority operation $\mathrm{Maj}_3(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$;*

4. *The Mal'tsev operation $u - v + w \mod 2$.*

Otherwise $\mathrm{Pol}(\Gamma^+)$ contains only projections (dictatorships).

Remark 77. For us this "deep" version of Schaefer's theorem is more useful than Theorem 32. The two versions are related as follows: The constant unary operations (which are excluded because of the Idempotency condition) are polymorphisms of $(0, \cdots, 0)$ or $(1, \cdots, 1)$ (or any relation that contains either of these). The AND and OR operations are polymorphisms of the Horn and dual-Horn clauses, respectively. The Maj_3 operation is a polymorphism of any 2CNF and the Mal'tsev operation is a polymorphism of affine subspaces.

Theorem 76 gives that when X is totally blocked and X is not an impossibility domain then one of the cases 1.-4. must hold. But [DH10a] proves more, it shows that when X is totally blocked, only Case 4. and the default case (i.e. no non-trivial polymorphisms) may occur. We provide a brief proof of this. We exclude Cases 1.-3. (from Theorem 76) as follows:

Excluding 1. and 2: We show that if $\vee \in \mathrm{Pol}(X)$, then in the blockedness graph, no node of the form $(k, 1)$ has a directed edge to any node of the form $(\ell, 0)$, so the blockedness graph cannot be strongly (or anyhow) connected. For this it is sufficient to show that every MIPE has at most one variable set to 1. Suppose that $x_k = x_\ell = 1$

is part of a MIPE with $k \neq \ell$ and the rest of MIPE evaluates to α. By the definition we have assignments:

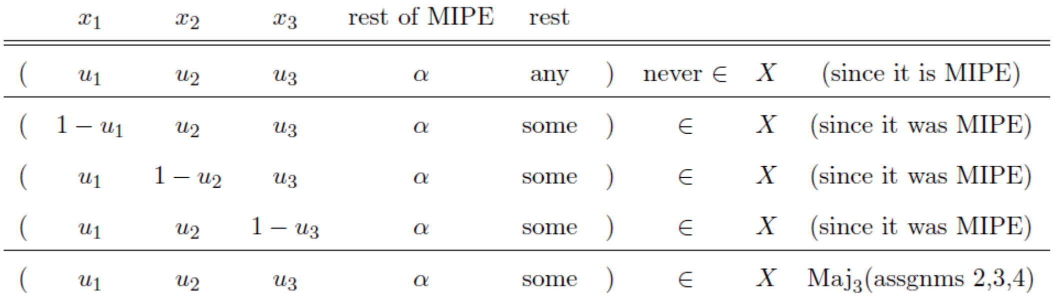
	x_k	x_ℓ	rest of MIPE	rest				
(	1	1	α	any	)	never $\in$	X	(since it is MIPE)
(	0	1	α	some	)	$\in$	X	(since it was MIPE)
(	1	0	α	some	)	$\in$	X	(since it was MIPE)
(	$0 \vee 1$	$1 \vee 0$	α	some	)	$\in$	X	(assgnm 2 $\vee$ assgnm 3)

Then the first and fourth lines of the table contradict to each other. An analogous proof shows that when $\wedge \in \mathrm{Pol}(X)$ the blockedness graph is not strongly connected.

Excluding 3: Assume that X is totally blocked and $\mathrm{Maj}_3 \in \mathrm{Pol}(X)$. First we show:

Lemma 78. *If* $\mathrm{Maj}_3 \in \mathrm{Pol}(X)$ *then every MIPE for* X *has length at most two.*

Proof. Consider a MIPE S, which contrary to our assumption has at least three elements, x_1, x_2, x_3. Assume that x_i evaluates to u_i for $1 \leq i \leq 3$, while the rest of the MIPE evaluates to α. Then the first and fifth lines of the following table together give a contradiction:



	x_1	x_2	x_3	rest of MIPE	rest				
(	u_1	u_2	u_3	α	any	)	never $\in$	X	(since it is MIPE)
(	$1 - u_1$	u_2	u_3	α	some	)	$\in$	X	(since it was MIPE)
(	u_1	$1 - u_2$	u_3	α	some	)	$\in$	X	(since it was MIPE)
(	u_1	u_2	$1 - u_3$	α	some	)	$\in$	X	(since it was MIPE)
(	u_1	u_2	u_3	α	some	)	$\in$	X	Maj_3(assgnms 2,3,4)

□

Assume now that we have an edge from (k, ϵ) to (ℓ, ϵ') in the blockedness graph. Since the MIPE creating this edge, by Lemma 78 has length two (cannot have length one), we conclude that $x_k = \epsilon$ forces $x_\ell = \epsilon'$. Consider any $x \in X$, by our assumption the total blockedness graph is strongly connected so there is a path from $(1, x_1)$ to

$(1, 1-x_1)$, which in the light of the above argument means that $(1, x_1)$ forces $(1, 1-x_1)$ through a sequence of edges, which is an obvious contradiction. Thus X is empty.

Schaefer's theorem now tells us that X must be either an impossibility domain or an affine subspace (see Definition 8). So what about the case when X is not totally blocked? Then we use the following deep theorem of A. Bulatov and P. Jeavons:

Theorem 79 (Bulatov and Jeavons [BJ03])**.** *Let $D = \{0, 1\}$ and Γ be a set of multi-sorted relations on D with type set $[t]$. Then for every type $j \in [t]$ either for every $\vec{f} = (f_1, \cdots, f_t) \in \mathrm{MPol}(\Gamma^+)$ the j^{th} component is a dictatorship or there is an $\vec{f} = (f_1, \cdots, f_t) \in \mathrm{MPol}(\Gamma^+)$ such that f_j is one of*

1. *the semi-lattice operation $u \vee v$ or $u \wedge v$,*
2. *the majority operation $(u \vee v) \wedge (v \vee w) \wedge (w \vee u)$,*
3. *the Mal'tsev operation $u - v + w \mod 2$.*

From this Theorem 20 easily follows.

Chapter 4

Non-binary evaluations

In this chapter we complete the classification theorem of E. Dokow and R. Holzman [DH10b] for non-binary evaluations. We treat for both the Supportive case and the Idempotent case (E. Dokow and R. Holzman have treated only the Supportive case).

4.1 Supportive non-binary evaluations

We restate here Theorem 18:

Theorem 80. *Let $X \subseteq D^m$ be non-degenerate and non-binary. If X is totally blocked then X is an impossibility domain with respect to IIA + Supportiveness + Non-dictatorship if and only if X does not have IIA + Supportiveness + Non-dictatorship aggregators of arity 3.*

Remark 81. The theorem is a classification in the sense that it gives us an algorithm to determine if a given domain X is an impossibility domain with respect to IIA + Supportiveness + Non-dictatorship or not: just check all potential aggregators with at most three arguments (see Chapter 5 for details).

Proof. The 'only if' part is straightforward since if X is an impossibility domain with respect to IIA + Supportiveness + Non-dictatorship then every supportive IIA aggregator $f = (f_1, \ldots, f_m)$ of X is a dictatorship, not only for arity $n \leq 3$. For the 'if' part we need the following results of E. Dokow and R. Holzman. For completeness, we still write down the sketched proof of Lemma 83 and 85.

Definition 82 (E. Dokow and R. Holzman [DH10b])**.** Let $f = (f_1, \ldots, f_m)$ be a supportive IIA aggregator of arity n for $X \subseteq D^m$. For an issue j and an ordered pair of distinct positions $u, v \in D_j$ we translate $f_j|_{\{u,v\}} : \{u, v\}^n \to \{u, v\}$ to a function $W_j^{uv} : \{0, 1\}^n \to \{0, 1\}$ under $0 \leftrightarrow v,\ 1 \leftrightarrow u$. By abuse of notation, we denote the set

$$\{S \subseteq [n] \mid f_j(x_1, \cdots , x_n) = u \text{ where } x_i = u \text{ if } i \in S,\ x_i = v \text{ if } i \notin S\}$$

to be W_j^{uv} as well.

Lemma 83 (E. Dokow and R. Holzman [DH10b], Propostion 1)**.** *If X is totally blocked, then all W_j^{uv}s are the same.*

Proof. (of Lemma 83) In the blockedness graph G_X formulated by X (see Definition 13), if there is an edge $uu'_k \to vv'_\ell$, we show that $W_k^{uu'} \subseteq W_\ell^{vv'}$. By contradiction assume that there is a subset $S \subseteq [n]$ which is in $W_k^{uu'}$ but not in $W_\ell^{vv'}$. For the edge $uu'_k \to vv'_\ell$ by Definition 13, there are $B_1, \cdots , B_m$ such that $|B_i| = 2$, $B_i \subseteq D_i$, $i = 1, \cdots , m$ and $K \subseteq N$ and a MIPE (see Definition 9) $x = (x_i)_{i\in K} \in X_B = X \cap \prod_{i=1}^m B_i$. Since $x = (x_i)_{i\in K}$ is a MIPE, there exists $y = (y_1, \cdots , y_m) \in X_B$ such that $y_k = u$, $y_\ell = v$ and $y_i = x_i$, $i \in K \setminus \{k, \ell\}$, and there exists $z = (z_1, \cdots , z_m) \in X_B$ such that $z_k = u'$, $z_\ell = v'$ and $z_i = x_i$, $i \in K \setminus \{k, \ell\}$, and there exists no $w = (w_1, \cdots , w_m) \in X_B$ such that $w_k = u$, $w_\ell = v'$ and $w_i = x_i$, $i \in K \setminus \{k, \ell\}$. We construct the following table.

		x_k	x_ℓ	rest of MIPE	rest			
S	(	u	v	$(x_i)_{i\in K\setminus\{k,\ell\}}$	some	)	$\in$	X
$N \setminus S$	(	u'	v'	$(x_i)_{i\in K\setminus\{k,\ell\}}$	some	)	$\in$	X
	(	u	v'	$(x_i)_{i\in K\setminus\{k,\ell\}}$	some	)	*never* $\in$	X

In the table, the k-th column is aggregated as u since the assumption that S is in $W_k^{uu'}$. The ℓ-th column is aggregated as v' since the assumption that S is not in $W_k^{vv'}$ and the aggregator is supportive. It is a contradiction that the last row in the table is never in X.

Since X is totally blocked, by definition, we know that $W_k^{uu'} \subseteq W_\ell^{vv'}$ holds for any u, u', v, v', k, ℓ, thus the conclusion in the lemma follows.

□

Definition 84 (E. Dokow and R. Holzman [DH10b])**.** We call f *2-dictatorial* if all W_j^{uv}s ($j \in [m]$, $u, v \in D_j$) are dictatorships with respect to the same coordinate d.

Lemma 85 (E. Dokow and R. Holzman [DH10b], Proposition 5)**.** *If X is totally blocked and f is 2-dictatorial then f is a dictatorship.*

Proof. (of Lemma 85) Since f is 2-dictatorial, there exists some $d \in [n]$ such that for all $j \in [m]$, $u, v \in D_j$, $W_j^{uv} = \{S \subseteq [n] | d \in S\}$. By contradiction assume f is not a dictatorship, i.e. there exist some $j \in N$ and $x_j^{(1)}, \cdots, x_j^{(n)} \in D_j$ such that $f(x_j^{(1)}, \cdots, x_j^{(n)}) \neq x_j^{(d)}$. Denote the distinct elements in $x_j^{(1)}, \cdots, x_j^{(n)}$ as $w^{(1)}, \cdots, w^{(q)}$ such that $f(x_j^{(1)}, \cdots, x_j^{(n)}) = w^{(1)}$ and $w^{(2)} = x_j^{(d)}$. Wlog assume that q is minimal among all possible $j \in N$ and $x_j^{(1)}, \cdots, x_j^{(n)} \in D_j$. Since f is 2-dictatorial, by Definition 84 we have $q \geq 3$. Let $S^{(h)} = \{i | w^{(h)} = x_j^{(i)}\}$, where $h = 1, \cdots, q$, be a partition of N. For any $k \in N$ and any $y^{(1)}, \cdots, y^{(q)} \in D_k$, denote $f_k(y^{(1)}, \cdots, y^{(q)})$ (by a slight abuse of notation since originally f_k has arity n) to be the value of f_k on n elements where positions in S_h have value $y^{(h)}$ for $h = 1, \cdots, q$. We now claim that:

Claim 85.1 (E. Dokow and R. Holzman [DH10b], Lemma 4)**.** *Let $a^{(h)} = (a_1^{(h)}, \cdots, a_m^{(h)}) \in X$, $h = 3, \cdots, m$ be $(q-2)$ fixed evaluations. If there is an edge $uu'_k \to vv'_\ell$ in the blockedness graph G_X formulated by X (see Definition 13) and $f_k(u, u', a_k^{(3)}, \cdots, a_k^{(q)}) = u$ then $f_\ell(v, v', a_\ell^{(3)}, \cdots, a_\ell^{(q)}) = v$.*

Recall that there exists some $j \in N$ such that $w^{(1)}, \cdots, w^{(q)} \in D_j$ and $f_j(w^{(1)}, \cdots, w^{(q)}) = w^{(1)}$. Choose $(q-2)$ evaluations $(y_1^{(h)}, \cdots, y_m^{(h)}) \in X$, $h = 3, \cdots, q$ such that $w^{(h)} = y_j^{(h)}$, $h = 3, \cdots, q$. By repeatedly applying the above Claim 85.1 along a path in the blockedness graph G_X from $w^{(1)}w_j^{(2)}$ to $w^{(1)}w_j^{(3)}$, since $f_j(w^{(1)}, \cdots, w^{(q)}) = w^{(1)}$, we have $f_j(w^{(1)}, w^{(3)}, w^{(3)}, \cdots, w^{(q)}) = w^{(1)}$. By the minimality assumption of q, this contradicts with $f_j(w^{(1)}, w^{(3)}, w^{(3)}, \cdots, w^{(q)}) = w^{(3)}$. □

We still have to prove the above Claim.

Proof of Claim 85.1. Since there is an edge $uu'_k \to vv'_\ell$ in the blockedness graph G_X, by Definition 13, there are $B_1, \cdots, B_m$ such that $|B_i| = 2$, $B_i \subseteq D_i$, $i = 1, \cdots, m$

and $K \subseteq N$ and a MIPE (see Definition 9) $x = (x_i)_{i \in K} \in X_B = X \cap \prod_{i=1}^{m} B_i$. Since $x = (x_i)_{i \in K}$ is a MIPE, there exists $y = (y_1, \cdots, y_m) \in X_B$ such that $y_k = u$, $y_\ell = v$ and $y_i = x_i$, $i \in K \setminus \{k, \ell\}$, and there exists $z = (z_1, \cdots, z_m) \in X_B$ such that $z_k = u'$, $z_\ell = v'$ and $z_i = x_i$, $i \in K \setminus \{k, \ell\}$. Wlog assume that $k = 1$, $\ell = 2$ and $K = \{1, \cdots, r\}$. Construct the following table:

	1	2	3	$\cdots$	r	$r+1$	$\cdots$	m
S_1	u	v	x_3	$\cdots$	x_r	y_{r+1}	$\cdots$	y_m
S_2	u'	v'	x_3	$\cdots$	x_r	z_{r+1}	$\cdots$	z_m
S_3	a_1^3	a_2^3	a_3^3	$\cdots$	a_r^3	a_{r+1}^3	$\cdots$	a_m^3
$\vdots$								
S_q	a_1^q	a_2^q	a_3^q	$\cdots$	a_r^q	a_{r+1}^q	$\cdots$	a_m^q
	u	s_2	x_3	$\cdots$	x_r	s_{r+1}	$\cdots$	s_m

The first column is aggregated as u by the previous assumption in the statement of the Claim. The i-th column is aggregated as x_i for $i = 3, \cdots, r$ since there are duplicated elements in each column. Assume that the rest columns are aggregated as s_j, $j = 2, r+1, \cdots, m$. Then the conclusion follows from $s_2 = v$. Assume by contradiction that $s_2 \neq v$. Construct the following table:

	1	2	3	$\cdots$	r	$r+1$	$\cdots$	m
S_1	u	s_2	x_3	$\cdots$	x_r	s_{r+1}	$\cdots$	s_m
S_2	u'	v'	x_3	$\cdots$	x_r	z_{r+1}	$\cdots$	z_m
S_3	a_1^3	a_2^3	a_3^3	$\cdots$	a_r^3	a_{r+1}^3	$\cdots$	a_m^3
$\vdots$								
S_q	a_1^q	a_2^q	a_3^q	$\cdots$	a_r^q	a_{r+1}^q	$\cdots$	a_m^q
	u	v'	x_3	$\cdots$	x_r	t_{r+1}	$\cdots$	t_m

The first column is again aggregated as u by the assumption. The second column is aggregated as v' since $s_2 \neq v$ but $s_2 \in \{v', a_3^{(3)}, \cdots, a_q^{(3)}\}$ thus the second column in the table contains duplicated elements, and the aggregator is supportive. But this is a contradiction since the aggregated row in the table is not in X as $x = (x_i)_{i \in K}$ is a

MIPE, i.e. there exists no $w = (w_1, \cdots, w_m) \in X_B$ such that $w_k = u$, $w_\ell = v'$ and $w_i = x_i$, $i \in K \setminus \{k, \ell\}$.

□

Let us try to put the above two lemmas together! The total blockedness is a condition in the theorem whose 'if' part we want to prove, so what is missing is that under the theorem's conditions the W_j^{uv}s are not simply the same, but they are all dictatorships. We rely on the following lemma which essentially uses similar idea as proving Rosenberg's Classification Theorem [Ros86] (see also [JQ95, Swi61, Csa05]) and algebraic proof of Schaefer's dichotomy theorem (see [Che09]).

Lemma 86. *Let Γ be a set of multi-sorted relations over D with type set $[t]$. If Γ does not have IIA + Supportiveness + Non-dictatorship aggregators of arity 3, then for any $f = (f_1, \cdots, f_t) \in \mathrm{MPol}(\Gamma^{\mho})$, $a \in [t]$, $u, v \in D_a$ with $u \neq v$ the restriction $f_a|_{\{u,v\}}$ is a dictatorship.*

Proof. (of Lemma 86.) Assume the contrary, namely that Γ does not have IIA + Supportiveness + Non-dictatorship aggregators with $n \leq 3$, but it has some aggregator $f = (f_1, \cdots, f_t) \in \mathrm{MPol}(\Gamma^{\mho})$, such that there exist $a \in [t]$ and $u, v \in D_a$, $u \neq v$ with the property that $f_a|_{\{u,v\}}$ is not a dictatorship, where n is the arity of f. Then n must be at least 4. Let f be such a counter-example with minimal n. In particular, any $f' = f(x^{(1)}, \ldots, \underbrace{x^{(i)}}_{i^{\text{th}}\text{ argument}}, \ldots, \underbrace{x^{(i)}}_{i'^{\text{th}}\text{ argument}} \ldots, x^{(n)})$, i.e. when we identify two inputs, must be a two-dictatorship, because f' aggregates $n-1$ inputs.

Denote $f_a|_{\{u,v\}}$ by g, which is presumably not a dictatorship. We will arrive at a contradiction by showing that g is a dictatorship, i.e. there exists one $k \in [n]$ such that for any $x_1, \cdots, x_n \in D'$, $g(x_1, \cdots, x_n) = x_k$. Since any identification of the variables of g arises by first identifying these variables in f, and then restricting the resulting type $= a$ component to the binary set $\{u, v\}$, we have, that any identification of variables of g, must result in a dictatorship. If $\underline{\text{all}}$ of these identifications $x^{(i)} = x^{(i')}$ result in a dictatorship that projects to coordinate i (as opposed to some coordinate $i'' \notin \{i, i'\}$), we get a contradiction by setting $\{i, i'\}$ first to $\{1, 2\}$ then to $\{3, 4\}$:

$$u = f(\underbrace{u, u}, v, v, \ldots) = f(u, u, \underbrace{v, v}, \ldots) = v. \quad \text{(Used that } n \geq 4.)$$

Thus there exist two coordinates where identifying the corresponding variables will result in a dictatorship function that projects to <u>some other</u> (i.e. not i, i') coordinate. Wlog assume that

$$g(\underbrace{x_1, x_1}, x_3, x_4, \cdots, x_n) = x_4$$

Then $g(x_1, x_1, x_1, x_4, \cdots, x_n) = x_4$. We show that this implies $g(x_1, x_2, x_1, x_4, \cdots, x_n) = x_4$. If $g|_{x_3 = x_1}$ was a dictator x_i other than x_4, then setting $x_2 = x_1$ and letting x_4 vary we would get a contradiction. Similar reasoning gives that $g(x_1, x_3, x_3, x_4, \cdots, x_n) = x_4$. Thus whenever there is a duplication among the values of x_1, x_2, x_3, the output of g is always x_4. But duplication always occurs since $|D'| = 2$, thus g is a dictatorship, a contradiction. □

We are now ready to prove the 'if' part of Theorem 80. Assume that when $n \leq 3$ there are no other supportive aggregators for X than dictatorships (and the other conditions: X is non-degenerate, totally blocked) also hold. Consider an aggregator f for X with $n \geq 4$. By Lemma 86, for any $j \subseteq [m]$ and for any $u, v \in D_j$, $u \neq v$, we have W_j^{uv} is a dictatorship. By Lemma 83, since X is totally blocked, all W_j^{uv} are the same. Lemma 85 then implies that f is a dictatorship.

□

4.2 General idempotent non-binary evaluations

In this section we finish the proof of Theorem 19: a characterization for general Idempotency case instead of the Supportive case discussed in Section 4.1.

Lemma 87. *For a given domain $X \subseteq D^m$, non-degenerate and non-binary, i.e. $|D| = d \geq 3$, and totally blocked, if X does not have any IIA + Idempotency + Non-dictatorship aggregator of arity d, then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship.*

Proof. Assume by contradiction that X satisfies the condition of the lemma and X is a possibility domain with respect to IIA + Idempotency + Non-dictatorship. Then there is an idempotent IIA aggregator $f = (f_1, \cdots, f_m)$ which is not a dictatorship. By the hypothesis of the lemma, f has arity $n \geq d+1$. Assume that f is of minimal arity, i.e. no idempotent non-dictatorial aggregator with smaller arity exists. We first show that f is supportive, i.e.

$$\forall\, x^{(1)}, \cdots, x^{(n)} \in X \qquad \forall\, 1 \leq j \leq m \ : \qquad\qquad f_j(x_j^{(1)}, \cdots, x_j^{(n)}) \in \{x_j^{(1)}, \cdots, x_j^{(n)}\}.$$

We are done if we can show that for any fixed $x^{(1)}, \cdots, x^{(n)}$ and j we have $f_j(x_j^{(1)}, \cdots, x_j^{(n)}) \in \{x_j^{(1)}, \cdots, x_j^{(n)}\}$. We use the pigeon hole principle. Since $n \geq d+1$, among $x_j^{(1)}, \cdots, x_j^{(n)}$ there must be at least two elements which are the same. For notational conveniences they are $x_j^{(1)}$ and $x_j^{(2)}$.

Remark 88. The collision does not happen at the same pair of indices for all inputs and js, but this does not affect us, since we are setting the input fixed.

Since the collision is at indices 1 and 2, we are going to examine the aggregator of $n-1$ elements g that we get from f by identifying the first two inputs of f. Since g is also an idempotent IIA aggregator of X, by our minimality assumption g must be a dictatorship. Thus g_j is also a dictatorship and therefore supportive. In particular,

$$u = g_j(x_j^{(2)}, \cdots, x_j^{(n)}) \in \{x_j^{(2)}, \cdots, x_j^{(n)}\}.$$

But u is also the value that f_j takes on $x_j^{(1)}, x_j^{(2)} \cdots, x_j^{(n)}$ (since by our assumption $x_j^{(1)} = x_j^{(2)}$). This concludes the proof of the fact that f if supportive.

Then, since we have found a non-dictatorial supportive aggregator for X (on some number of inputs), by Theorem 80 there must also be a non-dictatorial supportive aggregator on three inputs. Since supportive aggregators are also idempotent we get into a contradiction with the Lemma's assumption that the smallest non-dictatorial idempotent aggregator is on more than $|D| \geq 3$ inputs. □

This lemma resolves the *idempotent case*, which was the remaining part of Theorem 19.

Chapter 5

Algorithms to determine impossibility

In this chapter, we turn the previous characterization theorems (Theorem 60, 80, 19) into algorithms. When we try to determine if $X \subseteq D^m$ is an impossibility domain or not with respect to IIA + Idempotency (or Supportiveness) + Non-dictatorship, we can rely on two different types of characterization theorems: 1. by gadgets (Theorem 60) and 2. by aggregators (Theorem 80 and 19). Both types lead to algorithmic solutions, and we can use both of them as alternatives.

5.1 Algorithms from the characterization by gadgets

As we have seen in the previous sections, to determine if X is an impossibility domain or not with respect to IIA + Idempotency + Non-dictatorship, according to Theorem 60, we need to check if all $R^{u,v}_{k,\ell}$ relations can be expressed as X^+-gadgets (and whenever $|D_j| = 2$, we also need to express the NAE relation on type j). The same theorem gives that if we replace Idempotency with Supportiveness, we just replace X^+ with $X^{\mho}$. The task is therefore to solve the following type of problem:

Find Gadget: Given a set Γ of relations over D and a relation R over D determine if there is a Γ-gadget for R. In the multi-sorted version the relations are over a type set $[t]$ with associated domains $D_1, \ldots, D_t$.

Theorem 89. *There is an algorithm for the Find Gadget problem that in the multi-sorted setting runs in time $\sum_{\gamma\in\Gamma}\prod_{b=1}^{t}|D_b|^{|R|c_{b,\gamma}+|D_b|^{|R|}}$, where for any $\gamma \in \Gamma$ and $b \in [t]$ the number of variables of type b in γ is denoted by $c_{b,\gamma}$.*

Proof. Let us call the number of auxiliary variables in a gadget its *size*. First we argue about the single-sorted case, following arguments of Geiger [Gei68], Jeavons [Jea98] and

Trevisan et. al. [TSSW96].

Lemma 90. *If there is a Γ-gadget for R then there is also a Γ-gadget with size at most $|D|^{|R|}$, with the additional property that its x-variables (see Definition 28) are only involved in the '$=$' relations.*

Proof. Assume there is a Γ-gadget $\exists y\ G(x,y)$ for R (G is a conjunct where all terms in G are from Γ or the relation '$=$', see Definition 28), where the number, $|y|$, of auxiliary variables is ν. For any $x \in R$ let $y(x) \in D^{\nu}$ be any string of length ν such that $G(x, y(x)) = 1$. Let $1 \leq i, i' \leq \nu$. We say that i and i' are equivalent if for every $x \in R$ we have $y(x)_i = y(x)_{i'}$. To create the new gadget that expresses the same relation as $\exists y\ G(x,y)$ but with at most $|D|^{|R|}$ auxiliary variables we identify variables y_i and $y_{i'}$ whenever i and i' are equivalent. Examining the equivalence classes we notice that each class corresponds to a function $\phi : R \to D$ that for every $x \in R$ tells the value of those y-variables in $y(x)$ that are members of the class. Thus the number of classes is at most the number of such functions ϕ, i.e. at most $|D|^{|R|}$. For every class we introduce a single variable that represents the entire class. The list of these newly created variables form the new witness $\tilde{y}$ of length that equals the number of equivalence classes. The terms in the newly constructed gadget correspond to the original terms in G. More precisely, every term in the new gadget, $\tilde{G}$, comes from a term of the old one, but instead of a variable in y we plug in its representative from $\tilde{y}$. The identification of the variables with their representatives might cause different terms in G becoming the same term in $\tilde{G}$, and we keep only one copy of every term. We note that the x variables remain unchanged under our transformation. It is easy to see that the new gadget still accepts all elements of R (this is because for every $x \in R$ the string $y(x)$ is identically represented by $\tilde{y}(x)$). Furthermore, the new gadget is not going to accept any $x \notin R$. To see this assume that $\tilde{G}(x, \tilde{y}) = 1$. Define $y \in D^{\nu}$ by giving the same value to the variables in every equivalence class of variables as their representative receives in $\tilde{y}$. Then $G(x,y) = 1$, implying $x \in R$, a contradiction. Finally, we modify this gadget a little bit so that the x-variables are only involved in the '$=$' relations as follows. First equate any variable x_j with the $\tilde{y}$-variable (using the '$=$' relation) that corresponds to the $\phi : R \to D$ function that upon $x \in X$ returns x_j (if the class is not present we add it). Then we always use this $\tilde{y}$-variable instead of x_j. □

Lemma 91. *In the multi-sorted setting, with type set $[t]$ and ranges $D_1, \ldots, D_t$ if there is a Γ-gadget for multi-sorted relation R then there is also a Γ-gadget with total size at most $\sum_{b=1}^{t} |D_b|^{|R|}$. In fact, the number of variables of type b is at most $|D_b|^{|R|}$. Again, the x-variables are involved only in '$=$' relations with an identical type of y variables.*

Proof. The proof goes along the exact same lines as that of the single-sorted version. The only difference is that we need to define the equivalence relation of the previous proof for every type separately. □

The above proof does not only give an upper bound on the size of the smallest Γ-gadget for R, but also gives us a $\tilde{y}(x)$ for all $x \in R$. We can construct the $\tilde{y}$ variables without knowing G (or any Γ-gadget for R) in advance as follows. Imagine a matrix, whose columns correspond to the x-variables (altogether m of them) and whose rows correspond to all assignments in R. We augment this $|R| \times m$ matrix with $\sum_{b=1}^{t} |D_b|^{|R|}$ columns (all of length $|R|$) corresponding to all potential $\tilde{y}$ variables. In particular, for type b we have $|D_b|^{|R|}$ columns that correspond to all functions $\phi : R \to D_b$. Consider terms (i.e. constraints) on the y-variables that express relations from Γ. We say that a term $T(\tilde{y})$ *respects* R if for every $x \in R$ we have $T(\tilde{y}(x)) = 1$.

Remark 92. Perhaps strangely, whether a term respects R has very little to do with R itself. The rows of the matrix that describe the $\tilde{y}$ variables are labeled by R and certain designated columns in the $\tilde{y}$ matrix are equated to the x variables – this is the connection. However, if $|R| = |R'|$ and all the types range in the same sets, then the Γ gadgets expressing R and R' respectively will only differ (as we will see) in how the x variables are equated with the $\tilde{y}$ variables.

We define G_{all}, which is the conjunction of *all* terms T such that:

1. $T \in \Gamma$

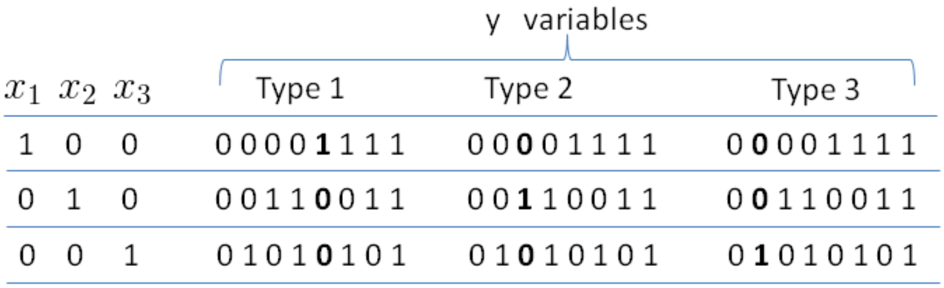
2. T respects R

We show that G_{all} together with the equations that equate the x and the corresponding $\tilde{y}$ variables is a Γ-gadget of R as long as any Γ-gadget for R exists at all. Indeed, by Lemma 91 if a Γ-gadget for R exists at all then a gadget $\tilde{G}$ must exist on the $\tilde{y}$ variables. From the fact that among the terms G_{all} all terms of $\tilde{G}$ must occur, and the fact that G_{all} itself must respect R (since all of its terms do) we are done.

The algorithm is now to cycle through all Γ terms and add the current term to G_{all} if it respects R. If the gadget we constructed this way computes R then we output 'yes'. The other possibility is that the gadget we have constructed also accepts assignments that do not belong to R. In this case we output 'no'. The running time, aside from a $\prod_{b=1}^{t} |D_b|^{|D_b|^{|R|}}$ factor (in worst case we have to solve a multi-sorted CSP with $\sum_{b=1}^{t} |D_b|^{|R|}$ variables) is determined by the number of iterations of the main cycle for every term in Γ. For every $\gamma \in \Gamma$ the number of potential ways to plug in y variables into γ is $\prod_{b=1}^{t} |D_b|^{|R|c_{b,\gamma}}$. Thus the Find Gadget problem in the multi-sorted setting runs in time $\sum_{\gamma\in\Gamma} \prod_{b=1}^{t} |D_b|^{|R|c_{b,\gamma}+|D_b|^{|R|}}$.

□

We use the following example to illustrate ideas in the above proof. For the multi-sorted relation $(R = \{(1,0,0),(0,1,0),(0,0,1)\},(1,2,3))$, we want to determine if it can be gadget reduced from multi-sorted NAE^{+} relations(see Definition 59). We create the following matrix (in Fig. 5.1).



			y variables		
x_1	x_2	x_3	Type 1	Type 2	Type 3
1	0	0	0 0 0 0 **1** 1 1 1	0 0 **0** 0 1 1 1 1	0 **0** 0 0 1 1 1 1
0	1	0	0 0 1 1 **0** 0 1 1	0 0 **1** 1 0 0 1 1	0 **0** 1 1 0 0 1 1
0	0	1	0 1 0 1 **0** 1 0 1	0 1 **0** 1 0 1 0 1	0 **1** 0 1 0 1 0 1

Figure 5.1: A matrix expressing auxiliary variables in gadget

It has $|R| = 3$ rows. The first three columns correspond to x-variables: x_1, x_2, x_3. The remaining 24 columns correspond to y-variables: y_i, $i = 1, 2, \cdots, 24$. In gadget

$\exists y\ G_{\text{all}}(x,y)$, besides those NAE$^+$ constraints, we also have '=' constraints: $x_1 = y_5$, $x_2 = y_{11}$ and $x_3 = y_{18}$. (As denoted in boldface in Fig. 5.1.) Actually $(R,(1,2,3))$ is indeed an NAE$^+$ gadget since G_{all} contains at least: which already expresses $(R,(1,2,3))$.

$$\text{NAE}(y_5,y_{11},y_{18})\wedge\text{NAE}(y_5,y_{11},y_{24})\wedge\text{NAE}(y_5,y_{16},y_{18})\wedge\text{NAE}(y_8,y_{11},y_{18})\wedge y_8 = 1\wedge y_{16} = 1\wedge y_{24} = 1$$

As illustrated in Fig. 5.2, neither can (y_5, y_{11}, y_{18}) have at least two 1s nor can it be $(0,0,0)$. By adding more constraints, since G_{all} respects R, thus $R(x) = \exists y\ G_{\text{all}}(x,y)$.

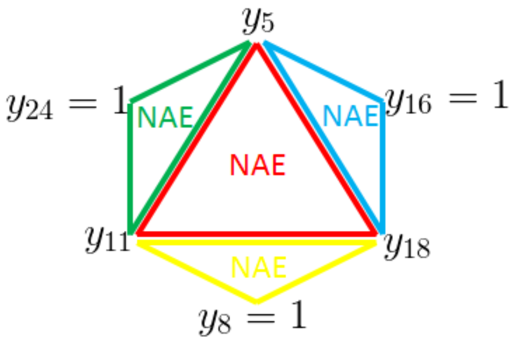


Figure 5.2: A multi-sorted NAE$^+$ gadget express $(R,(1,2,3))$

Now we can apply Theorem 89 and combine it with Theorem 60 to give an upper bound of the running time to determine for a given domain $X \subseteq D^m$ if it is an impossibility domain or not with respect to IIA + Idempotency + Non-dictatorship, or IIA + Supportiveness + Non-dictatorship.

Theorem 93. *For a given non-degenerate domain $X \subseteq D^m$, the running time to determine if it is an impossibility domain or not with respect to IIA + Idempotency + Non-dictatorship can be upper bounded by $|D|^{O(m|D|^{|D|^2})}$, and for IIA + Supportiveness + Non-dictatorship the running time can also be upper bounded by $|D|^{O(m|D|^{|D|^2})}$.*

Proof. As in Theorem 60, let $t = m$ and $\tau = (1,\dots,m)$, and the possible values for type j is $D_j = \text{pr}_j X$. Then X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship if and only there are $(X,\tau)^+$-gadgets $R^{u,v}_{k,\ell}$ for every $1 \le k,\ell \le m$; $u \in \text{pr}_k X$, $v \in \text{pr}_\ell X$. Furthermore, if $|D_j| = 2$ for some $1 \le j \le m$, we also need to add the multi-sorted NAE gadget on types (j, j, j). By Theorem 89, for each $R^{u,v}_{k,\ell}$, the running time to determine if it is an $(X, \tau)^+$-gadget or not is upper bounded by

$$\begin{aligned} & \sum_{\gamma \in (X,\tau)^+} \prod_{b=1}^{m} |D_b|^{|R^{u,v}_{k,\ell}| c_{b,\gamma} + |D_b|^{|R^{u,v}_{k,\ell}|}} \\ \leq \quad & (1 + \sum_{b=1}^{m} |D_b|) \prod_{b=1}^{m} |D_b|^{|D_k| \cdot |D_\ell| + |D_b|^{|D_k| \cdot |D_\ell|}} \\ \leq \quad & |D|^{O(m|D|^{|D|^2})}. \end{aligned}$$

If $|D_j| = 2$ in this case $|R| = 6 = O(|D|^2)$ thus it is still bounded by $|D|^{O(m|D|^{|D|^2})}$. Thus the total running time to determine if X is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship is upper bounded by $|D|^{O(m|D|^{|D|^2})} \cdot m^2 \cdot |D|^2 \leq |D|^{O(m|D|^{|D|^2})}$.

The analogous statement, when we replace "Idempotency" with "Supportiveness", requires to replace $(X, \tau)^+$ with $(X, \tau)^{\mho}$. In this case we need to replace $|(X, \tau)^+| = 1 + \sum_{b=1}^{m} |D_b|$ with $|(X, \tau)^{\mho}| = 1 + \sum_{k=1}^{m} (2^{|D|_k} - 1)$, but this does not change the main term in the upper bound thus the running time is still bounded by $|D|^{O(m|D|^{|D|^2})}$.

□

5.2 Algorithms from the characterization by aggregators

Theorem 80 and 19 give us a complete classification for non-binary case by providing an algorithm to determine if a domain X is impossible or not. Here is an algorithm to determine a given non-degenerate and non-binary $X \subseteq D^m$ whether it is impossible or not with respect to IIA + Supportiveness + Non-dictatorship:

1. Check if X is totally blocked (see Definition 13). If X is not totally blocked, then X is a possibility domain; otherwise
2. FOR each possible idempotent $f = (f_1, \ldots, f_m)$ where $f : X^3 \to X$, check if f satisfies the IIA + Non-dictatorship criteria. If none satisfies them then X is an impossibility domain, otherwise it is a possibility domain.

Now we determine upper bound of time complexity for the above algorithm. The running time is dominated by the number of possible fs in step 2. Since for each i, f_i is a function from D_i^3 to D_i, the number of fs is upper bounded by $|D|^{|D|^3}$. Thus we have proved:

Theorem 94. *The running time of the algorithm to determine a given domain $X \subseteq D^m$ is an impossibility domain with respect to IIA + Supportiveness + Non-dictatorship is upper bounded by $O(|X|^3 \cdot m \cdot |D|^{m \cdot |D|^3})$.*

Completely similarly we get the running time of the algorithm to determine a given domain $X \subseteq D^m$ is an impossibility domain with respect to IIA + Idempotency + Non-dictatorship is upper bounded by $O(|X|^{|D|} \cdot m \cdot |D|^{m \cdot |D|^{|D|}})$.

Chapter 6

Majority aggregators and algebraic results

The basic goal of this chapter is to characterize those domains $X \subseteq D^m$ for which majority is an IIA aggregator. We will first discuss some basic theory in universal algebra on algebras with a majority polymorphism in Section 6.1. Then we will focus on majority aggregator for binary evaluations and prove characterization theorems in Section 6.2. We will then use these characterizations to show how one can deduce results in domain restriction for the preference aggregation problem in Section 6.3.

6.1 Algebras with a majority polymorphism

Definition 95. Let D be a finite set, $k \geq 3$. Then we define the function family MAJ_k^D as $\{f : D^k \to D | f(u_1, \dots, u_k) = u$ if u occurs more than $k/2$ times in the input$\}$. If $|D| = 2$ we denote MAJ_k^D with MAJ_k, and when k is odd, it is unique.

Call $f : D^3 \to D$ a majority polymorphism, if for every $x, y \in D$, we have

$$f(x, x, y) = f(x, y, x) = f(y, x, x) = x,$$

i.e. f is chosen from the function family MAJ_3^D. Call algebra $\mathbf{A} = (D; f)$ a majority algebra if f is a majority polymorphism. Another polymorphism similar to majority is near unanimity (NU). Call f a k-NU operation if $f : D^k \to D$ satisfies: for every $x, y \in D$, we have

$$f(\underbrace{x, \cdots, x}_{k-1}, y) = f(\underbrace{x, \cdots, x}_{k-2}, y, x) = \cdots = f(y, \underbrace{x, \cdots, x}_{k-1}) = x.$$

Note that 2-NU is the same as majority. In [JCC98], the following theorem is given.

Theorem 96 (P. Jeavons, D. Cohen and M. Cooper [JCC98])**.** *For any set of relations Γ, over a finite set D, and any $k \geq 2$, the following conditions are equivalent:*

1. *Every relation R in Γ is closed under a $(k+1)$-NU f.*

2. *Every relation R in $\langle\Gamma\rangle$ is k-decomposable (see Definition 14).*

In [BP75], the following theorem is given which can be viewed as an abstract version of Theorem 96.

Theorem 97 (K. Baker and A. Pixley [BP75])**.** *For a variety $\boldsymbol{V}$ and integer $k \geq 2$, the following conditions are equivalent:*

1. *$\boldsymbol{V}$ has a $(k+1)$-NU operation in each algebra of $\boldsymbol{V}$.*

2. *In $\boldsymbol{V}$ if $\boldsymbol{A}$ is a subalgebra of a direct product $P = C_1 \times C_2 \times \cdots \times C_m (m \geq k)$, then $\boldsymbol{A}$ can be uniquely determined from a knowledge of its k-fold projections.*

3. *In any algebra $\boldsymbol{A} \in \boldsymbol{V}$, if m congruences $x \equiv a_i \mod \theta_i$, $1 \leq i \leq m$, $k \leq m$, are solvable k at a time, then they are solvable simutaneously.*

The following results are deep results from universal algebra.

Recall that a lattice $\mathbf{L}$ is distributive if $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ for all $a, b, c \in \mathbf{L}$ (or equivalently $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$). A variety V is called congruence distributive if all the algebras in V have distributive congruence lattices.

Definition 98. A sequence $t_0, t_1, \cdots, t_s$ of ternary operations on a set A is called a Jónsson chain if for every $a, b, c \in A$

1. $t_0(a, b, c) = a$,

2. $t_s(a, b, c) = c$,

3. $t_r(a, b, a) = a$ for all $r \leq s$,

4. $t_r(a, a, b) = t_{r+1}(a, a, b)$ for all even $r < s$,

5. $t_r(a, b, b) = t_{r+1}(a, b, b)$ for all odd $r < s$.

An algebra $\mathbf{A} = (A, t_0, t_1, \cdots, t_s)$, where $t_0, t_1, \cdots, t_s$ is a Jónsson chain, will be called a CD(s) algebra.

Note that an algebra is in CD(1) if and only if it has size 1 and is in CD(2) if and only if it has a majority term operation. The following theorem connects the existence of Jónsson terms with the congruence distributivity of the algebras in a variety.

Theorem 99. *An algebra $\boldsymbol{A}$ has a Jónsson chain of term functions if and only if the variety generated by $\boldsymbol{A}$ is congruence distributive.*

6.2 Majority aggregators for binary evaluations

One may ask the existence of majoritarian aggregators for an arbitrary domain $X \subseteq \{0,1\}^m$. We have an answer to that, which is taken off the shelve [BP75, JCC98] from Theorem 96 and 97. Note that we consider only IIA aggregators with components all the same, in other words we do not consider sorts here in this and subsequent section. We leave IIA aggregators with different components chosen from MAJ_k^D for future research.

Theorem 100. *If $X \subseteq \{0,1\}^m$ is 2-decomposable (see Definition 14) and k is odd, then MAJ_k is an IIA aggregator for X. For an arbitrary D if $X \subseteq D^m$ is not 2-decomposable then MAJ_3^D is not an IIA aggregator for X.*

In Section 6.3 we show how Theorem 100 specializes to Sen's result when $X \subseteq S_k$. From an early result of [BP75, JCC98] we have the following Theorem 101. It is also an immediate consequence of E. Post [Pos41].

Theorem 101. *If $X \subseteq \{0,1\}^m$ is 2-decomposable (see Definition 14) then MAJ_3 is an IIA aggregator for X. If $X \subseteq D^m$ is not 2-decomposable then MAJ_3^D is not an IIA aggregator for X.*

Proof. If a binary relation $X \subseteq \{0,1\}^m$ is 2-decomposable we want to show that X is closed under MAJ_3, that is to say for any $x^{(1)}, x^{(2)}, x^{(3)} \in X$ we have $\mathrm{MAJ}_3(x^{(1)}, x^{(2)}, x^{(3)}) \in X$. Let $x = \mathrm{MAJ}_3(x^{(1)}, x^{(2)}, x^{(3)})$, if we can show that for any $1 \leq i < j \leq m$, $(x_i, x_j) = pr_{i,j}x \in pr_{i,j}X$ then we are done. Here is the table how we obtain (x_i, x_j) from MAJ_3.

	i	j	other positions	
$x^{(1)}$:	$x_i^{(1)}$	$x_j^{(1)}$	some	$\in X$
$x^{(2)}$:	$x_i^{(2)}$	$x_j^{(2)}$	some	$\in X$
$x^{(3)}$:	$x_i^{(3)}$	$x_j^{(3)}$	some	$\in X$
$\mathrm{MAJ}_3(x^{(1)}, x^{(2)}, x^{(3)})$:	x_i	x_j	some	

Since $D = \{0, 1\}$ is binary, we know that there are at least two of $x_i^{(1)}, x_i^{(2)}, x_i^{(3)}$ which are equal to x_i, and there are at least two of $x_j^{(1)}, x_j^{(2)}, x_j^{(3)}$ which are equal to x_j, thus there is some $1 \leq k \leq 3$ such that $(x_i^{(k)}, x_j^{(k)}) = (x_i, x_j)$. As a result we have shown for any $1 \leq i < j \leq m$, $(x_i, x_j) = pr_{i,j}x \in pr_{i,j}X$.

For arbitrary D and $X \subseteq D^m$, for a vector $x = (x_1, x_2, \cdots, x_m)$, if for any $1 \leq i < j \leq m$, there exists an $x' \in X$ such that $pr_{i,j}x = pr_{i,j}x'$, then we want to show that $x \in X$ provided that X is closed under a MAJ_3^D. We show this by induction. First if $m = 2$, this definitely holds. Assume it is true for any relation of arity less than m. Now for a relation X of arity m which is closed under a MAJ_3^D, define $X_{[l]}$ to be $X_{[l]} = pr_{\{1,2,\cdots,m\}\backslash\{l\}}R$. For any $1 \leq l \leq m$ we know that $X_{[l]}$ is also closed under MAJ_3^D, by the inductive hypothesis we have $pr_{\{1,2,\cdots,m\}\backslash\{l\}}x \in X_{[l]}$, thus by definition of $X_{[l]}$ there exists an $x_{[l]} \in X$ such that $pr_{\{1,2,\cdots,m\}\backslash\{l\}}x_{[l]} = pr_{\{1,2,\cdots,m\}\backslash\{l\}}x$. Thus $x = \mathrm{MAJ}_3^D(x_{[1]}, x_{[2]}, x_{[3]}) \in X$.

□

In order to obtain the characterization of Theorem 100 it is sufficient to show that MAJ_3^D generates some MAJ_k^D function for $k \geq 3$ by composition (since polymorphisms compose). Note that the following result (Lemma 102) is not ours. It is a folklore result.

Lemma 102. *The* MAJ_3^D *function generates some* MAJ_k^D *function for any odd* $k \geq 3$.

Proof. Let us call the function to be built $m_k(x_1, \cdots, x_k)$. We will define m_k by building a function with 3^l inputs (for some l) and then identifying input gates within k random groups with each other. (Thus it is going to be a probabilistic construction.) Note that composition and identification preserves polymorphisms. We build a 3-ary tree of height l (where l is to be determined later) and label each leaf of the tree with a randomly chosen variables from $\{x_1, \cdots, x_k\}$. The intermediate and top nodes of the tree are MAJ_3^D "gates." We claim that if l is large enough then *some* random choice (i.e. a random assignment of variables to leaf gates) will satisfy our condition. (Here we just follow Erdős's random method.) Let $I \subseteq D^k$ be the set of inputs for which there is a majority variable, i.e. a one that occurs greater than $k/2$ times. Fix such an input $x \in I$ with majority value u. Since the number of occurrences of u must be an integer, if we randomly pick a component of x, the chance that it will be u is at least $0.5 + \frac{1}{2k}$. We compute the probability that on x our tree outputs u. The probability that a gate outputs u increases as we go up in the circuit. At the leaf gates this probability is (at least) $p = 0.5 + \frac{1}{2k}$. At the first level this probability is at least $\varphi(p) = p^3 + 3p^2(1-p)$. Because of symmetry we can iterate this getting $\varphi(\varphi(p))$, $\varphi(\varphi(\varphi(p)))$, etc. as we go up. It is easy to show that this sequence converges to one. Let l be large enough that the l^{th} iterate is at least $1 - \frac{1}{2|I|}$. Then the probability that the top gate fails to output the majority value for x is at most $\frac{1}{2|I|}$, Since this holds for all $x \in I$, we have a positive probability that our circuit works well for *all* $x \in I$. Thus a random circuit with the desirable properties exists and we pick this for $m_k(x_1, \cdots, x_k)$. □

6.3 The domain restriction problem for the preference aggregation problem

The preference aggregation problem is when each voter votes for some order on k prescribed items (such as k candidates) and the task is to aggregate these votes in a consistent manner. Let $\mathcal{A}$ denote the set of all items ($|\mathcal{A}| = k$) and let S_k denote the set of $k!$ linear orders of these items. The issues we want the aggregator to keep invariant are of the form $A < B$, where $A, B \in \mathcal{A}$. For $\{A, B\}$, $A \neq B$ the voter can choose between two positions, $A < B$ or $B < A$, which s/he implicitly does by voting for a linear order of all items. In notation we pick one of the two positions, say $A < B$ as the default position, which we denote by 1, and we denote the other position ($B < A$) by 0.

(This notation, which depends on the choice of the default position, has the (perhaps confusing) advantage that, for instance, in the table below we do not have repeat our argument twice by flipping ones and zeroes.) Every element of S_k (i.e. a linear order on $\mathcal{A}$) induces a *position-vector* of length $\binom{n}{2}$ whose coordinates correspond to the (binary) issues. This is how S_k can be viewed as a domain: $S_k \subseteq \{0,1\}^{\binom{k}{2}}$. As mentioned in the introduction, aggregating preference lists with the majority function is not possible for any length greater than two. Researchers have considered different approaches to resolve the problem. There are two of these approaches that have received a huge attention and have given rise to new research directions. We have already discussed one such approach started by Arrow. In this section we discuss the other one.

For an impossibility domain $X \subseteq D^m$ (in our setting here, $X = S_k$, $D = \{0,1\}$ and $m = \binom{k}{2}$), it is natural to consider a subset $Y \subseteq X$ such that aggregation function on Y is possible *relative to* X, i.e. there is a non-dictatorial Idempotent function f such that $f : Y^n \to X$. A significant effort has been spent on the case when X is S_k in the *preference aggregation problem* and f is the majority function. While insisting on aggregating with the majority function, we require that all voters pick their vote from some carefully chosen subset Y of domain X, but the aggregate can be anywhere from X. Black's median voter theorem [Bla48] is an example for this. Black gives a subset of the set S_k of all linearly ordered lists of k elements $A_1, \ldots, A_k$ that can be successfully aggregated with respect to the issues $A_i < A_j$. Black's subset contains 2^{k-1} lists (out of the $k!$ possible). It can be shown that Black's subset is not maximal in size of $|Y|$, and it is still an interesting problem to estimate the maximum size of $|Y|$ for arbitrary large k.

After intermediate results in 1966 Amartya Sen in a breakthrough result has given a beautiful characterization called *triplewise value restriction* [Sen66, SP69] of those subsets Y of S_k that have majoritarian aggregator. A subset of $Y \subseteq S_k$ satisfies the triple-wise value restriction property if for every $A, B, C \in \mathcal{A}$ either A or B or C is either never first or second or third in any element of $y \in Y$, restricted to $\{A, B, C\}$. In Lemma 103 and Lemma 104 we show how our Theorem 100 which addresses majoritarian aggregation for general domains specializes to (and so proves) Sen's result. Although in the above setting aggregate objects from Y can be anywhere in X, it can be shown that if Y is maximal with this property, the majority function is an IIA aggregator for Y i.e. the aggregate stays in Y. Sen's result states that if $Y \subseteq S_k$ has the triple-wise value restriction property, the majority function aggregates Y in an IIA manner.

Lemma 103. *Let $Y \subseteq S_k$. If Y is 2-decomposable, then Y satisfies the triple-wise value restriction property.*

Proof. By contradition assume that Y does not satisfy the triplewise value restriction property. It is easy to see that then there are linear orders $y^1, y^2, y^3 \in Y$ and $A, B, C \in \mathcal{A}$ such that the position-vectors of these linear orders are described by the following table:

	$A > B$	$B > C$	$C > A$	rest
y^1 :	0	0	1	rest
y^2 :	0	1	0	rest
y^3 :	1	0	0	rest

Then since Y is 2-decomposable, $y = \mathrm{MAJ}_3(y^1, y^2, y^3) \in Y$, where MAJ_3 is applied component-wise. But in y we have $A < B, B < C, C < A$, which contradicts to $Y \subseteq S_k$. □

Lemma 104. *Let $Y \subseteq S_k$ and assume that Y satisfies the triple-wise value restriction property. Define $Y' = \{y \mid \forall 1 \leq i, j \leq \binom{n}{2} : \mathrm{pr}_{i,j}\, y \in \mathrm{pr}_{i,j}\, Y\}$. Then Y' satisfies the following properties:*

1. *Y' is 2-decomposable,*
2. *$Y \subseteq Y' \subseteq X$,*
3. *Y' satisfies the triplewise value restriction property.*

Proof. 1. Since $\mathrm{pr}_{i,j}\, Y'$ remains the same as $\mathrm{pr}_{i,j}\, Y$ for all $1 \leq i, j \leq \binom{n}{2}$, but every y that is consistent with $\mathrm{pr}_{i,j}\, Y$ is now included, Y' is 2-decomposable.

2. $Y \subseteq Y'$ is trivial. To prove $Y' \subseteq X$ it is sufficient to show that for any $A, B, C \in \mathcal{A}$, there is no $(A > B, B > C, C > A) = (1, 1, 1)$ in Y' (it is well known that every transitive tournament is consistent with some linear order so it is sufficient to prove the lack of three-cycles). Assume the opposite. Then, by the definition of Y' there must be a $y_1 \in Y$ consistent with $A > B, B > C$, a $y_2 \in Y$ consistent with $A > B, C > A$ and a $y_3 \in Y$ consistent with $B > C, C > A$. But then contrary to our assumption $y_1, y_2, y_3 \in Y$ do not satisfy the triple-wise value restriction property,

3. By 1. and 2., and by the previous lemma. □

To summarize the lemmas, we have proven that maximal subsets of S_k that satisfy the triple-wise value restriction property are exactly the maximal 2-decomposable subsets of S_k. Conversely, for a subset of S_k the triple-wise value restriction property holds if and only if it is a subset of a 2-decomposable subset of S_k (since the the triple-wise value restriction property is closed downwards).

Chapter 7

Degrees of democracy

Although the Non-dictatorship condition represents a minimal criterion for democracy, there are many functions that pass the Non-dictatorship test, but can barely be called democratic. Consider for instance the Boolean function that takes the majority value if the first voter votes zero, and takes the value one otherwise. Although this is not dictatorship the first voter has an overwhelming way in the outcome. What voting functions should we consider democratic? Scenarios taken from real life, such as the American electoral system (iterated majority function), show that the majority vote is not the only one that can be viewed as truly democratic. The answer is non-trivial.

Different criteria for democracy have been formulated such as Anonymity (invariant under S_n) or Symmetricity, by Kalai [Kal02] (invariant under a transitive permutation group acting on $[n]$), that are somewhere on the scale in between the majoritarian and dictatorial voting schemes. In this dissertation we would like to introduce StrongDem, with deep algebraic motivation.

StrongDem: Let f be an aggregator for $X \subseteq D^m$ for $n \geq 2$ voters that satisfies the IIA condition, so it is of the form $f = (f_1, \ldots, f_m)$. We say that f is StrongDem if for every $1 \leq j \leq m$, $1 \leq i \leq n$ there are $c_1, \ldots, c_n \in D$ such that $f_j(c_1, \ldots, c_{i-1}, x, c_{i+1}, \ldots, c_n)$ does not depend on $x \in D_j$. We further require that this property holds not only for D_j but when we replace D_j with any $D'_j \subseteq D_j$ such that the operation f preserves (respects) D'_j. (In the replacement the independence from x must hold only when x is also from D'_j.)

The majority function on three or more arguments is StrongDem: take any $D' \subseteq D$ and set all votes except the vote of voter i on the j^{th} issue to some (arbitrary) $c \in D'$.

Then the outcome will be c no matter what position the i^{th} voter takes. Since any StrongDem aggregator is clearly Non-dictatorship, we have the containment:

$$\text{Non-dictatorship} \supset \text{StrongDem} \supset \text{Majority voting}$$

All containments are strict in the following strong sense: For any two of the above conditions we can find X which is a possibility domain with respect to the larger class, but an impossibility domain for the smaller (IIA + Idempotency are assumed). An important example for an X which admits an f with the Non-dictatorship condition, but has no StrongDem voting scheme is the affine subspace. Let $D = \{0, 1\}$, and $X = \{(x_1, x_2, x_3) \in D^3 \mid x_1 + x_2 + x_3 = 1 \mod 2\}$. There is a non-dictatorial voting scheme when n is odd: Let $f_j : (u_1, \ldots, u_n) \to \sum_{i=1}^{n} u_i \mod 2$ for $1 \leq j \leq 3$. It is easy to see that $f = (f_1, f_2, f_3)$ has the Non-dictatorship property. It can be shown that this is the *only* non-dictatorial aggregator for X, and it is not StrongDem.

Majority, Anonymity, Symmetricity	The scheme treats all voters exactly in the same way
StrongDem	When votes of all others are appropriately fixed, a single voter cannot change the outcome
Non-dictatorship	There is not a single voter who exclusively controls the outcome.

Figure 7.1: Conditions on democracy and their informal meaning

The StrongDem condition is equivalent to the aggregator falling into a well-researched class of universal algebraic operations. This class contains operations with "no ability to count". In contrast, functions like the parity function that have the ability to count are very input sensitive: their value changes even when the count changes only by one. In a far-reaching part of the algebraic theory the "no ability to count" class of operations generate algebras that "avoid types **1** and **2**" [HM88] congruences. This class of algebras has been recently characterized in terms of the local consistency checking algorithm, which was a breakthrough [BK09]. What makes the notion of StrongDem particularly attractive is that when viewing its minimalistic definition, it seems a *necessary* condition for democracy, but it also has equivalent formulations, that are strong enough to accept it as a *sufficient* condition.

Definition 105 (Strong resilience)**.** Let D be a finite domain and μ be a probability measure on D. The influence $\mathrm{Inf}_{i,\mu}(f)$ of the i-th variable of $f : D^n \to D$ is $\mathbb{P}_{\mu^{n+1}}(f(x) \neq f(x'))$, where x, x' run through all random input pairs that differ only in the i-th coordinate (μ^{n+1} gives a natural measure on such pairs). The maximal influence $\max \inf_\mu(f)$ is $\max_i \mathrm{Inf}_{i,\mu}(f)$. A function $f : D^n \to D$ is strongly resilient if for every measure μ on D: $\max \inf_\mu(f^k) \to 0$ when $k \to \infty$ where f^k is defined recursively by composition $f^k = f(f^{k-1}, \ldots, f^{k-1})$.

Theorem 106 (G. Kun and M. Szegedy [KS09])**.** *The following are equivalent:*

1. *f is StrongDem.*

2. *There is a strongly resilient operation in $[\{f\}]$.*

References

[ABI$^+$09] Eric Allender, Michael Bauland, Neil Immerman, Henning Schnoor, and Heribert Vollmer. The complexity of satisfiability problems: Refining Schaefer's theorem. *J. Comput. Syst. Sci.*, 75(4):245–254, 2009.

[Arr50] Kenneth J. Arrow. A Difficulty in the Concept of Social Welfare. *Journal of Political Economy*, 58(4):328–346, August 1950.

[BD03] Andrei A. Bulatov and Víctor Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 562–571, 2003.

[BJ03] Andrei A. Bulatov and Peter Jeavons. An Algebraic Approach to Multisorted Constraints. In *CP*, pages 183–198, 2003.

[BJK05] Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the Complexity of Constraints Using Finite Algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

[BK09] L. Barto and M. Kozik. Constraint Satisfaction Problems of Bounded Width. In *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on*, pages 595–603, Oct 2009.

[BK12] Libor Barto and Marcin Kozik. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 931–940, 2012.

[BKJ01] Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. The complexity of maximal constraint languages. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 667–674, 2001.

[Bla48] Duncan Black. The Decisions of a Committee Using a Special Majority. *Econometrica*, 16(3):245–261, 1948.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 73–83, New York, NY, USA, 1990. ACM.

[BP75] KirbyA. Baker and AldenF. Pixley. Polynomial interpolation and the Chinese Remainder Theorem for algebraic systems. *Mathematische Zeitschrift*, 143(2):165–174, 1975.

[Bul06] Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.

[Bul11] Andrei A Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic (TOCL)*, 12(4):24, 2011.

[Che09] Hubie Chen. A Rendezvous of Logic, Complexity, and Algebra. *ACM Comput. Surv.*, 42(1):2:1–2:32, December 2009.

[Con85] Marquis de Condorcet. *Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix.* Paris: Imprimerie Royale, 1785.

[Csa05] Bela Csakany. Minimal clones – a minicourse. *Algebra Univers.*, 54(1):73–89, 2005.

[DH10a] Elad Dokow and Ron Holzman. Aggregation of binary evaluations. *Journal of Economic Theory*, 145(2):495–511, 2010.

[DH10b] Elad Dokow and Ron Holzman. Aggregation of non-binary evaluations. *Advances in Applied Mathematics*, 45(4):487–504, 2010.

[FF11] Dvir Falik and Ehud Friedgut. An algebraic proof of a robust social choice impossibility theorem. In *FOCS*, pages 413–422, 2011.

[FV93] Tomás Feder and Moshe Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 612–622, 1993.

[FV98] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[FV09] Ralph Freese and Matthew A. Valeriote. On the complexity of some Maltsev conditions. *International Journal of Algebra and Computation*, 19(01):41–77, 2009.

[Gei68] D. Geiger. Closed Systems of Functions and Predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.

[HM88] D. Hobby and R. McKenzie. The Structure of Finite Algebras. *Contemporary Mathematics*, 76, 1988.

[HN90] Pavol Hell and Jaroslav Nesetril. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990.

[JCC98] Peter Jeavons, David A. Cohen, and Martin C. Cooper. Constraints, Consistency and Closure. *Artif. Intell.*, 101(1-2):251–265, 1998.

[JCG97] Peter Jeavons, David A. Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.

[Jea98] Peter Jeavons. On the Algebraic Structure of Combinatorial Problems. *Theor. Comput. Sci.*, 200(1-2):185–204, 1998.

[JQ95] J. Jezek and R. Quackenbush. Minimal clones of conservative functions. *International Journal of Algebra and Computation*, 5(06):615–630, 1995.

[Kal02] Gil Kalai. A Fourier-theoretic perspective on the Condorcet paradox and Arrow's theorem. *Advances in Applied Mathematics*, 29(3):412 – 426, 2002.

[KS86] Lewis A. Kornhauser, , and Lawrence G. Sager. Unpacking the Court. *Yale Law Journal*, 96(1):82–117, 1986.

[KS09] Gábor Kun and Mario Szegedy. A new line of attack on the dichotomy conjecture. In *STOC*, pages 725–734, 2009.

[Lis12] Christian List. The theory of judgment aggregation: an introductory review. *Synthese*, 187(1):179–207, 2012.

[LP96] L. Levai and P. Palfy. On binary minimal clones. *Acta Cybern.*, 12:279–294, 1996.

[May52] Kenneth O. May. A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decision. *Econometrica*, 20(4):680–684, 1952.

[NP02] K Nehring and Clemens Puppe. Strategy-proof social choice on single-peaked domains: possibility, impossibility and the space between. *Unpublished manuscript, Department of Economics, University of California at Davis*, 2002.

[Pet01] Philip Pettit. Deliberative Democracy and the Discursive Dilemma. *Philosophical Issues*, 11(1):268–299, 2001.

[Poi37] S. D. Poisson. *Recherches sur la probabilité des jugements en matière criminelle et en matière civile: précédées des règles générales du calcul des probabilités.* 1837.

[Pos41] Emil L. Post. The Two-Valued Iterative Systems of Mathematical Logic. *Annals of Mathematics Studies*, 5:122, 1941.

[Qua95] R. W. Quackenbush. A survey of minimal clones. *Aequationes Math.*, 50:3–16, 1995.

[RF86] Ariel Rubinstein and Peter C Fishburn. Algebraic Aggregation Theory. *Journal of Economic Theory*, 38(1):63–77, 1986.

[Ros86] I. G. Rosenberg. Minimal clones I: The five types. *Lectures in Universal Algebra (Proc. Conf. Szeged 1983)*, 43:405–427, 1986.

[Sch78] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pages 216–226, 1978.

[Sen66] Amartya K. Sen. A Possibility Theorem on Majority Decisions. *Econometrica*, 34(2):pp. 491–499, 1966.

[Sen08] Amartya Sen. Social Choice. In Steven N. Durlauf and Lawrence E. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, Basingstoke, 2008.

[SP69] Amartya Sen and Prasanta K. Pattanaik. Necessary and sufficient conditions for rational choice under majority decision. *Journal of Economic Theory*, 1(2):178–202, August 1969.

[Swi61] S. Swierczkowski. Algebras which are independently generated by every n elements. *Fund. Math.*, 49:93104, 1960/61.

[SX15] Mario Szegedy and Yixin Xu. Impossibility Theorems and the Universal Algebraic Toolkit. *in preparation*, 2015.

[TSSW96] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, Approximation, and Linear Programming. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 617–626, Oct 1996.

[Vac21] R. Vacca. Opinioni Individuali e Deliberazioni Collettive. *Rivista Internazionale di Filosofia del Diritto*, pages 52–59, 1921.

[Wil75] Robert Wilson. On the Theory of Aggregation. *Journal of Economic Theory*, 10(1):89–99, 1975.