COMPUTER-AIDED THEOREM DISCOVERY– A NEW ADVENTURE AND ITS APPLICATION TO ECONOMIC THEORY

by

Pingzhong Tang

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in Computer Science

May 2010, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Pingzhong Tang

17 May 2010

COMPUTER-AIDED THEOREM DISCOVERY– A NEW ADVENTURE AND ITS APPLICATION TO ECONOMIC THEORY

by

Pingzhong Tang

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. Fangzhen Lin, Thesis Supervisor

Prof. Mounir Hamdi, Head of Department

Department of Computer Science and Engineering
17 May 2010

Acknowledgments

I have been extremely lucky to learn from a number of advisors and colleagues during my Ph.D study.

First and foremost, I am deeply indebted to my PhD advisor Prof. Fangzhen Lin. Fangzhen is everything a Ph.D student can dream of; an insightful advisor, a technical co-worker and a generous friend. I will always go back to the time when I was struggling with the implementation of theorem discovery in game theory – that is, Chapter 4 of this dissertation, Fangzhen helped me to debug and optimize the PROLOG program and sent back the results to me at 2am with excitement. Just like that, every time I encountered a technical problem, Fangzhen was always there to turn to, from algorithm to implementation, so that I would not fear to explore new topics that I would otherwise not have the courage to think of. Every time I came up with a nice solution to a difficult problem, Fangzhen always congratulated me with a smile. It then becomes part of my motivation to work hard in order not to disappoint him. Every time I wrote a complicated solution, Fangzhen was also interested in simplifying it in some elegant fashion and sometimes pointing out errors during his rewriting. Besides research, Fangzhen is also a fun friend to talk to when it comes to politics, sports, wine and many others.

I would also like to thank Prof. Yoav Shoham, who hosted and supervised me during my visit to Stanford University in my 4th year of Ph.D study. Yoav deeply impressed me from his sense to research ideas, insightful views towards many scientific branches, to his beautiful and huge castle, and things in between. Special thanks also goes to Prof. Yiling Chen, who hosted and supervised me when I visited Harvard University in my final year of Ph.D. Yiling's guidance exposed me to several forefront research issues in electronic commerce.

I am very thankful to my external thesis reader, Prof. Johan van Benthem, for his detailed suggestions and insights in many perspectives that help improve this thesis. Johan, the world authority on logics and computational logics, introduced to me several related areas that I haven't thought of.

I am also grateful to a number of faculty members in HKUST. These include my thesis examination committee members Prof. Shiu-Yuen Cheng, Ke Yi, Nevin Zhang. Prof. Cheng who sets aside his busy schedule as a Vice President of HKUST to give me detailed advise on readings (including the impeccably lecture notes of Lloyd Shapley) in cooperative game theory and bargaining theory. These also include Prof. Chi-Keung Tang and Cunsheng Ding who generously give me advice and information on job hunting.

I owe lots of gratitudes to colleagues in AI, TCS and Networking Labs of HKUST, where I have spent a significant amount of time during the last five years. Special

thanks goes to Zhen Zhou, YajunWang, Shan Chen, Rong Pan, Junfeng Pan, Jialin Pan, Juncheng Jia, Jian Xia, Qin Zhang and Qi Wang – each time I proposed an interesting topic, either academic or not, these guys just suffocate me with all kinds of creative suggestions and solutions. I am also grateful that I have "passively" picked up lots of machine learning knowledge from members in AI lab, as the only non-machine-learning person. Ph.D life could be boring and frustrating without a healthy working environment.

Of all my other friends, I'll single out Tong Zhang, for her kind help and support.

Finally, I would like to give my deepest gratitudes to my parents, who made me who I am. I dedicate this dissertation to them.

Table of Contents

COMPUTER-AIDED THEOREM DISCOVERY– A NEW ADVENTURE AND ITS APPLICATION TO ECONOMIC THEORY

by

Pingzhong Tang

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

Abstract

"*Everything about him was old except his eyes and they were the same color as the sea and were cheerful and undefeated.*"

—Ernest Hemingway, *The old man and the sea.*

Theorem discovery, with the help of computer, presents at least two steps of challenges. The first concerns how to come up with reasonable conjectures automatically. This raises further challenges, such as how to represent these conjectures within the computers, what is the yardstick for reasonableness, etc. The second concerns how to prove or negate the conjectures automatically. However theorem proving, even for the best of human beings, is still an intelligence-demanding endeavor and sometimes even a nightmare.

Our starting point however, is a basic form of proof, namely proof by induction. The heuristic behind is extremely straightforward: We first formulate the problem domain in a proper language, say logic or other formal languages. We then enumerate the sentences (within certain length limit) in the underlying language that describe propositions in the domain. After that, we use a computer program to verify through these sentences to find those true in base cases, that is, where the problem size is small. The remaining sentences serves as conjectures, which can be extended, one way or other, to inductive cases.

It turns out that this methodology has been quite effective since we adopted it in economic theory. In particular, some of our programs on game theory have returned theorems that shed lights on the understanding of basic game forms such as zero-sum game, potential game and super-modular game. Some of them have helped us prove some Nobel Prize winning theorems such as Arrow's impossibility theorem and Sen's theorem on voting functions and discover new theorems that better characterize key concepts in social choice theory. Others also have helped us prove Nobel Prize winning theorems such as Maskin's theorem on Nash implementation as well as Gibbard-Satterthwaite theorem

on dominant strategy implementation. These proofs themselves also provide insights on discovering similar theorems.

This thesis reports all attempts that we have conducted in the past few years, in support of the general methodology of theorem discovery.

Chapter 1

Introduction

> "*The ultimate goal of mathematics is to eliminate any need for intelligent thought.*"
>
> —Alfred N. Whitehead.

> "Your vacuum cleaner roams around your home at night. Your rice cooker, toaster, and washing machine have their own minds. Your car parks itself; its transmission adapts itself to your driving preference, and it tells the dealership which parts it thinks it will need to have replaced three months from now. Your PDA knows your preferences and acts as your personal radio station, playing only music you like. You use a search engine that is capable of looking through billions of documents with new documents being added every millisecond. Semiautonomous rover are driving around on Mars. There is a virtual person on the phone, tirelessly trying to help you. You still cannot beat the AI in your kid's video game."

The above description, by Goker and Haigh from a special issue of AI Magazine [11], is one of many sweet dreams that AI researchers have had. However, using computers to automatically discover new, scientific theorems for us might go so far as to be beyond our sweetest dreams.

Theorem discovery, with the help of computer, presents at least two steps of challenges. The first concerns how to come up with reasonable conjectures automatically. This raises further challenges, such as how to represent these conjectures within the computers, what is the yardstick for reasonableness, etc. The second concerns how to prove or disprove the conjectures automatically. However theorem proving, even for the best of human beings, is still an intelligence-demanding endeavor and sometimes a nightmare.

Despite the implausibility as it seems, researchers have repeatedly challenged it with various attempts. In one pioneer work, Petkovsek et. al. [29] showed that to prove the following theorem,

"*The angle bisectors of every triangle intersect at one point.*",

it suffices to verify it in 64 non-isomorphic triangles, which can be automated by computers. In the same spirit, the authors went on to demonstrate that certain forms

of theorems concerning the close form of the sum of combinatorial sequences can be completely discovered by computers programs.

Langley [13] had briefly summarized the attempts of computer-aided discovery until 1998, ranging from mathematics to physics, chemistry as well as biology. Among those attempts, Lenat's AM system [14] and Fajtlowicz's Graffiti [7] are also remarkable progresses on theorem discovery. The AM system aims at finding new concepts and theorems based on existing concepts as well as a large amount of heuristic rules, which require extensive domain knowledge of the designers. Despite the complexity of design, the system managed to rediscover hundreds of common concepts as well as simple theorems. The Graffiti system, on the other hand, is more intuitive in design. First of all, the system itself does not attempt to prove anything. Alternatively, it aims at generating interesting conjectures in graph theory by guessing and testing some invariants, most of which are of forms $a \geq b$, $a = b$, and $\sum a_i \geq \sum b_i$, concerning two numerical features in a graph. It is worth some attention that Graffiti maintains the quality of the set of conjectures by filtering those implied by existing ones. In other words, the current set of conjectures are the strongest ones generated so far. This is similar to our approach in game theory, which will be introduced in detail later.

Our work follows from the ideas emerged in a previous line of work [15, 16]. Both papers try to simulate the pattern of proving a theorem by induction. In particular, the problem domain is formulated in a symbolic language. They then enumerate the sentences (within certain length limit) in the underlying language that describe propositions in the domain. After that, they use a computer program to test through these sentences to find those true in base cases, that is, where the problem domain size is small. The remaining sentences are then extended to inductive cases, automatically in [15] and manually in [16].

As mentioned, we choose economic theory as the domain to investigate. We do so for three reasons. For one thing, some of the existing theorems, especially those Nobel-prize winning ones in social choice theory, have very similar forms. Thus, we wonder if we can slightly vary them to get some other theorems of the same form. For another, the key concepts, such as pure Nash Equilibrium in game theory and preference in social choice theory, are ordinal, thus allowing for a concise formulation by a logical language. This property greatly facilitates our conjecture generation process. Last but not least, we are fascinated by the beauty of the interactions between computer science, game theory and social choice theory. It is the intense interest and curiosity that motivate us to explore the field. [1]

[1] Exemplified by our recent paper on Team Competition [41, 42], which characterizes the set of conditions for designing desirable forms of team competitions. Examples of such competitions include David Cup on tennis, Corbillon Cup on table tennis, etc.

We are then thrilled at the abundance of the theorems discovered by our programs.

- In our first project on game theory [39, 40], we discover several classes of games, known or unknown, that guarantee the uniqueness of pure Nash Equilibria (PNE). Similarly, we discover several classes of games that guarantee the existence of PNE as well as those guarantee pareto optimality of PNE. These discoveries also lead to our later more important findings of two equivalence relations, one about strictly competitiveness and uniqueness of PNE and the other about super-modular games and potential games.

- In our second project on social choice theory [17, 38], we prove three of the most important impossibility theorems in a unified computer-aided framework. It is worth mentioning that Arrow's impossibility theorem is the main citation of the author's Nobel prize. It is also worth mentioning that our proof is amongst the shortest and most straightforward proofs of these theorems. We also similarly discover several theorems that generalize Arrow's conditions as well as a new theorem that better interprets Arrow's *IIA* condition.

- In our third project on social choice theory [25], we compare several commonly used voting rules based on the degrees to which they violate certain desirable properties, when the number of voters and candidates are small. As a result, we also benefit from the intuitions returned by our programs that help to prove two general asymptotic theorems. [2]

- In our fourth project, we prove two of the most influential theorems in mechanism design theory, namely Maskin's theorem on Nash implementability, the main citation of Maskin's Nobel Prize, as well as Gibbard-Satterthwaite theorem on dominant strategy implementability. In fact, the way that we prove it is more or less the same as we did in proving these theorems in social choice theory. That is, using computer program to test the base case and extend it to inductive case. In this aspect, we unify proofs of almost all the important theorems in these fields.

It is also worth pointing out that these results promise a new angle in the currently developing area of *computational economics* (including *computational game theory* and *computational social-choice theory*, in particular) - an area that aims at advancing economic research via computational means.

The aim of this dissertation is to describe all of our findings, in support of the idea and methodology of computer-aided theorem discovery. We next introduce the preliminary

[2]The methodology for this part is somehow different from the previously described one. We decide to leave out this part and refer the readers to our paper.

knowledge for this manuscript in chapter 2, followed by our case study of our methodology on game theory, social choice theory and implementation theory in chapters 4, 5 and 6, respectively. For each of these chapters, we first introduce the background knowledge and the type of theorems that we want to discover, we then go through our procedures and report the theorems proved and discovered. We finally discuss, generalize and summarize these theorems. Finally in chapter 7, we conclude this dissertation.

Chapter 2

Preliminary

"*What we can speak about we must speak clearly. What we cannot speak about we must consign to silence.*"

—Ludwig Wittgenstein.

We now give notations and preliminary knowledge related to first order logic for the purpose of this thesis.

2.1 First order logic

In particular, we consider first-order language without function symbols, whose syntax consists of the following parts,

- Set of variables, denoted by x_i, y_i, ...
- Set of constants, denoted by a, b, ...
- Set of predicates, denoted by P_1, P_2, ... Predicates specify the relations between variables and constants.
- The logical connectives are as usual. That is, $\wedge$ and $\vee$ for conjunction and disjunction respectively, $\supset$ for implication and $\neg$ for negation.
- No function symbol.
- Quantifiers $\forall$, $\exists$.

An *atomic formula* is either a predicate that takes *terms* or a *term* is equal to another term: $term1 = term2$, where a term is a constant or variable.

A *literal* is either an atomic formula or the negation of an atomic formula.

A *formula* is a finite length of string that is connected of literals using logical connectives. (See any logic book for the formal definition for the so-called *well-formed formula*.)

An *interpretation* of this language is a set of instantiated atoms (on some domains). Thus an atomic formula F is true under some interpretation M iff $F \in M$.

An (atomic) formula is *satisfiable* iff it is true under at least one interpretation. We say the interpretation M is the *model* of the formula F, written as $\models_M F$.

The *satisfiability* relation of interpretation M and a compound formula is defined as follows:

- if F is a formula, $\models_M \neg F$ if $\not\models_M F$,
- if F_1, F_2 are formulas, $\models_M F_1 \vee F_2$ if $\models_M F_1$ or $\models_M F_2$,
- if F_1, F_2 are formulas, $\models_M F_1 \wedge F_2$ if $\models_M F_1$ and $\models_M F_2$,
- if F_1, F_2 are formulas, $\models_M F_1 \supset F_2$ if $\models_M \neg F_1 \vee F_2$,
- if F is a formula, $\models_M \exists x F$ if by substituting every appearance of x in F with some element a from the domain, we get a new formula F' and $\models_M F'$,
- $\models_M \forall x F$ iff $\models_M \neg\exists x \neg F$.

Suppose Σ is a set of formulas and F, we say $\Sigma \models F$ if all the models of Σ are models of F.

2.2 Finitely verifiable property of $\forall\exists$ formulas

We have the following nice properties (cf. [15]) in any first-order language without function symbols.

Theorem 2.1 *Let Q be a formula without quantifiers. If $\exists\vec{x}\forall\vec{y}Q$ is satisfiable, then it is satisfiable by an interpretation whose domain has $max\{1, |\vec{x}| + n\}$ elements, where n is the number of constants in Q.*

The theorem below then follows from a contrapositive argument on Theorem 2.1.

Theorem 2.2 *Let Q be a formula without quantifiers. If $\forall\vec{x}\exists\vec{y}Q$ is true by any interpretation whose domain has $max\{1, |\vec{x}| + n\}$ elements, where n is the number of constants in Q, then it is true by any interpretation. That is, it is valid.*

In other words, to verify the validity of a formula that is in the form $\forall\vec{x}\exists\vec{y}Q$, one only needs to exhaustively verify all the interpretations whose domain are within certain size limit. In short, we can just say this type of formulas are *finitely verifiable*. The significance of Theorem 2.2 lies in that it reduces the proof of this class of formulas to model checking within finite domains, which can be automated by computers. It serves as the theoretic foundation of our project on game theory.

It is worth mentioning that $\exists\forall$ formulas are also finitely verifiable. However, the underlying finite interpretation sets may be entirely different from the original interpretation sets (might be the interpretation of an entirely different language but still finite). For more on this subject, see [6].

Chapter 3

The methodology

> "*Any academic discipline must rely on a general methodology to provide a framework for inquiry and debate. Academic methodologies enable scholars to see connections that may be obscure to untrained layman. But scholars must also be aware that our expertise is diminished beyond the scope of our methodology, and we learn to stay within its boundaries.* "
>
> —Roger Myerson

In general, we can divide our discovery process into the following two steps:

- Step 1. Automated conjecture generation.
- Step 2. Automated theorem proving.

We briefly explain both steps in below. The actual procedures that we use to carry out these steps will become clear as we present our case-studies in subsequent chapters.

To begin with, we have in mind an existing theorem of the following form,

$$X_1 \text{ and } X_2 \ldots \text{ imply } Y,$$

where X_i's are a set of conditions and Y is an important or surprising property. Note that its form captures a class of theorems concerning sufficient conditions. Our goal is to find all the theorems of *similar* form.

To automate this goal, we begin by coming up with a formal language that represents the theorem above as one in the language. That is,

$$F_{X_1} \wedge F_{X_2} \ldots \Rightarrow F_Y,$$

where F_{X_i}'s and F_Y are formulas describing X_i's and Y, respectively.

We then start step 1, that is, to generate conjectures related to the theorem. We do so by exhaustive enumeration, that is, generating the set S of all the formulas within certain length in the language and replace X_1 by any $F' \in S$ in the theorem,

$$F' \wedge F_{X_2} \ldots \Rightarrow F_Y.$$

These formulas become conjectures for further consideration.

The set generated so far can be quite large and contains many obviously false conjectures. To refine it, we next verify all these conjectures on models of small sizes and return the survivors of this verification. Note that this procedure cannot fail by returning nothing, as long as $F_{X_1} \in S$, since it would at least return the theorem that we started with.

We then leap into step 2, theorem proving. From time to time, we are lucky to find that

$$F' \wedge F_{X_2} \ldots \Rightarrow F_Y.$$

satisfies finitely verifiable property, which suggests that conjectures surviving refinement are already guaranteed to be theorems. For others, we need to prove or negate it by other means.

Now we have a set of provably correct theorems. However, this set could still be large and impossible to be interpreted manually (especially given their logical representation). There are several ways to further refine this set. One typical way is to delete those implied by some other theorems, or those implied by conjunctions of several other theorems. In this sense, we only return the strongest ones.

Chapter 4

Discovering theorems in game theory

> "*I haven't played a chess match for several decades. At one point I lost most of my chess games. Then I realized many of my competitors were memorizing the best moves and I was unwilling to do this.*"
>
> —John Harsanyi

The target theorems that we are interested in are those concerning pure Nash Equilibrium, henceforth PNE, in ordinal games which merely consist of individual action sets as well as individual preferences over action profiles, as appear in [28]. This definition of games generalizes the persuasive definition using utility functions in the sense that every utility function can be reduced to a totally ordered preference relation but not vice versa, unless the preferences being von Neumann-Morgenstern (vNM) [27].

Traditional equilibrium analysis has been mostly focused on mixed equilibria. Part of the reasons for this bias is that such an equilibrium always exists and algorithms such as the one by Lemke-Howson are guaranteed to find one. Moreover, best response functions in games with mixed strategies are continuous and differentiable, allowing for standard calculus techniques to be applied.

However, the concept of mixed equilibria in not well defined in the ordinal games briefly mentioned above simply because the utility functions may not exist. And indeed, pure Nash equilibria (PNEs) are also of interest, and there is already much work about them. Examples here include the existence of PNEs in (ordinal) potential games [21], (quasi-)supermodular games [43] as well as games with dominant strategies, and uniqueness of PNE payoffs[1] in two-person strictly competitive games.

4.1 Two-person games

A two-person game in strategic form is a tuple $(A, B, \leq_1, \leq_2)$, where A and B are sets of strategies of players 1 and 2, respectively, and $\leq_1$ and $\leq_2$ are total orders on $A \times B$ called *preference relations* for players 1 and 2, respectively.

[1]Note that the uniqueness of PNE payoffs is also an ordinal property, which means all the PNEs in a game are equally preferred to all players. In particular, the notion of unique PNE payoffs degenerates to unique PNEs in strict games where the preference orders are linear.

Instead of two (vNM) preference relations, a two-person game can also be specified by two payoff functions, one for each player, which map profiles to numbers. The relationship between these two formulations are as follows: for any profiles s and s', $s \leq_i s'$ iff $u_i(s) \leq u_i(s')$, where u_i is the payoff function for player i. In the following, we shall use these two formulations interchangeably.

In the following, two profiles (a, b) and (a', b') are said to be *equivalent* if their payoff profiles are the same: $(u_1(a), u_2(b)) = (u_1(a'), u_2(b'))$. In terms of preference relations, (a, b) and (a', b') are equivalent iff

$$(x_1, y_1) \leq_i (x_2, y_2) \wedge (x_2, y_2) \leq_i (x_1, y_1),$$

for $i = 1, 2$.

For each $b \in B$, we define $B_1(b)$ to be the set of best responses by player 1 to the strategy b by player 2:

$$B_1(b) = \{a \mid a \in A, \text{ and for all } a' \in A, (a', b) \leq_1 (a, b)\}.$$

Similarly, for each $a \in A$, the set of best responses by player 2 is:

$$B_2(a) = \{b \mid b \in B, \text{ and for all } b' \in B, (a, b') \leq_2 (a, b)\}.$$

A profile $(a, b) \in A \times B$ is a *Pure Nash Equilibrium (PNE)* if both $a \in B_1(b)$ and $b \in B_2(a)$. A game can have exactly one, more than one, or no PNEs. We say that a game has a *unique* PNE payoff if all the PNEs are equivalent.

4.1.1 Strictly competitive games

Definition 4.1 *A game $(A, B, \leq_1, \leq_2)$ is strictly competitive [28] if for every pair of profiles s_1 and s_2 in $A \times B$, we have that $s_1 \leq_1 s_2$ iff $s_2 \leq_2 s_1$.*

Thus in strictly competitive games, the two players' preferences are exactly opposite. Strictly competitive games are also known as zero-sum games when the game is represented by utility functions instead of preference relations. We shall henceforth use two notations interchangeably.

Zero-sum describes a game in where one player's gain (may be negative) is exactly balanced by the loss of the other player. Examples are Chess, Go and sports competitions where there is only one winners. Zero-sum are generally thought of as constant-sum where the gains to all players sum to a constant value. Cake dividing is constant-sum since obviously taking a larger piece for me reduces the amount available for the other.

The following game is a simple zero-sum game:

Example 4.1.1 $A = \{a_1, a_2\}, B = \{a_1, a_2\}$

$1, -1$	$-1, 1$
$-1, 1$	$1, -1$

Strictly competitive game has many nice properties. If (a, b) and (a', b') are both Nash equilibria of a strictly competitive game, then (1) The equal payoff property: they are indistinguishable in sense that $(a, b) \leq_i (a', b')$ and $(a', b') \leq_i (a, b)$ for both $i = 1, 2$; (2)The Interchangeability property: they are interchangeable in the sense that (a', b) and (a, b') are also Nash equilibria. Thus if a strictly competitive game has Nash equilibria, then they are unique. Furthermore, the mini-max duality theorem holds, which means, the liner program that maximizes the minimum utility of the rows of player one and liner program that minimizes the maximum utility of the columns of player two have the same optimal value. Thus, solving a single player's utility optimization problem solves the game. This also means that computing all mixed equilibria in such games is polynomial.

4.1.2 Potential games and ordinal potential games

Definition 4.2 *For a game* Γ *of* (A, B, u_1, u_2)*, A function* $P : A \times B \to R$ *is an ordinal potential for* Γ*, if*

$$u_1(x, y) - u_1(z, y) > 0 \text{ iff } p(x, y) - p(z, y) > 0$$

for every $y \in B$*, every* $x, z \in A$ *and*

$$u_2(x, y) - u_2(x, w) > 0 \text{ iff } p(x, y) - p(x, w) > 0$$

for every $x \in A$*, every* $y, w \in B$*.*

Γ is called an ordinal potential game [21] if it admits an ordinal potential function.

Definition 4.3 *A function* $P : A \times B \to R$ *is an (exact) potential for* Γ*, if*

$$u_1(x, y) - u_1(z, y) = p(x, y) - p(z, y)$$

for every $y \in B$*, every* $x, z \in A$ *and*

$$u_2(x, y) - u_2(x, w) = p(x, y) - p(x, w)$$

for every $x \in A$*, every* $y, w \in B$*.*

Γ is called a potential game if it admits a potential function.

The first that uses potential functions in games was Rosenthal [32], where he defined the class of congestion games and proved, by explicitly constructing a potential function, that every game in this class possesses a pure-strategy equilibrium. The class of congestion games is, on the one hand, narrow, but on the other hand, widely applied in economics and computer science. The class of congestion games is proven to be best response equivalent to that of finite potential games.

Example 4.1.2 *The following game is a potential game where the utility matrices are as follows:*

$1,1$	$9,0$
$0,9$	$6,6$

and the following function is the potential:

4	3
3	0

The game above is also a special case of prisoner's dilemma. As we can see, it is also an ordinal potential game. Generally, any potential game is an ordinal potential game and for some ordinal potential game, it is not necessarily a potential game. If we change the above utility matrices as follows,

$2,1$	$9,0$
$0,9$	$6,6$

it is still a ordinal potential game, but it turns out there is no potential function for the above matrices.

Ordinal potential game also have many nice properties. For instances, every finite ordinal potential game possesses a pure-strategy Nash Equilibrium. Moreover, every finite ordinal potential game has the *finite improvement property*. That is, every path, which starts from any profiles and consists of a sequence of unilateral deviations that benefits the deviating player, is finite and ends in a Nash equilibrium.

4.1.3 Supermodular games and quasi-supermodular games

Before we introduce the definition of Supermodular game, preliminary knowledge on *supermodular* function and *monotone comparative statics* would be necessary. However, as a matter of fact, one can directly jump to the end of this subsubsection to see a simplified definition (definition 4.6) of quasi-supermodular game, which is all that we need for our project introduced in later sections. We include the precise definitions here for completeness.

Let R^n denote n-dimensional Euclidean space. Given $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ in R^n, denote by $x \vee y$ and $x \wedge y$ the coordinate-wise sup and inf of x and y,

$$x \vee y = (\max\{x_1, y_1\}, \ldots, \max\{x_n, y_n\})$$

and

$$x \wedge y = (\min\{x_1, y_1\}, \ldots, \min\{x_n, y_n\}).$$

More general, we extend the above notation to the lattice theory.

Let X be a partially ordered set, with the reflexive, antisymmetric and transitive binary relation $\geq$ Given elements x and z in X, denote by $x \vee z$ the least upper bound or join of x and z in X, provided it exists, and $x \wedge z$ the greatest lower bound or meet of x and z in X, provided it exists. A partially ordered set X that contains the join and the meet of each pair of its elements is called a lattice. A lattice in which each nonempty subset has a supremum and an infimum is complete. In particular, a finite lattice is complete. If Y is a subset of a lattice X and Y contains the join and the meet with respect to X of each pair of elements of Y , then is Y is a sublattice of X. A sublattice Y of a lattice X in which each nonempty subset has a supremum and an infimum with respect to X that are contained in Y is a subcomplete.

A function $F : A \to R$ is *supermodular* if

$$F(a \wedge a') + F(a \vee a') \geq F(a) + F(a'), \forall a, a' \in A.$$

The following increasing difference property is considered as a generalization of supermodularity on R^2.

Suppose $S \subset R$ and T is partially ordered. A function $F : S \times T \to R$ has *increasing differences* in (s, a) if

$$F(s', a') - F(s', a) \geq F(s, a') - F(s, a), \forall a' > a, s' > s.$$

This property does not discriminate between the two variables and the above condition

is equivalent to

$$F(s', a') - F(s, a') \geq F(s', a) - F(s, a), \forall a' > a, s' > s.$$

Moreover, if we restrict our attention on functions on R^2, the increasing differences property here is equivalent to the supermodularity property on R^2.

A subset of Euclidean space R^n is *compact* if it is *closed* and bounded. A set is *closed* if every limit point of the set is a point in the set.

A real valued function f is *upper semi-continuous* at a point x_0 if the function values for the argument near x_0 are either closed to $f(x_0)$ or less than $f(x_0)$.

If we replace the less than in the above definition by greater than, we get the definition of *lower semi-continuous*. A function is continuous if it is both upper and lower continuous.

Definition 4.4 *A two-person game (A, B, u_1, u_2) is supermodular if for $i \in 1, 2$:*
1. A, B is a compact subset of R;
2. u_i is upper-semi-continuous in s_i, continuous in s_{-i};
3. u_i satisfies the increasing differences property in (s_i, s_{-i}).

For the second condition, the utility functions are required to be u.s.c to ensure the best response function of each player is well-defined(i.e. the maximum is attained.).

For supermodular games, pure strategy Nash equilibria exist. Furthermore, the set of strategies that survive Iterated Strict Dominance (IEDS) (Iterated elimination of dominated strategies) has the greatest and least elements $\overline{s}$ and $\underline{s}$, which are PNEs.

Like the ordinal potential game, which is an ordinal version of potential game, can be represented by the preference relations rather than the requirement of the utility functions, the supermodular game also has its ordinal counterpart.

The following *single crossing property* is a weak (ordinal) version of increasing differences.

Suppose $S \subset R$ and T is partially ordered. A function $F : S \times T \to R$ has the *single crossing property* in $(x; t)$ if:

$$F(s', a) > F(s, a) \Rightarrow F(s', a') > F(s, a') \forall a' > a, s' > s$$

and

$$F(s', a) \geq F(s, a) \Rightarrow F(s', a') \geq F(s, a') \forall a' > a, s' > s.$$

The following *quasisupermodular* property is a ordinal approximation of *supermodular* property.

Given a lattice X, a function $F : X \to R$ is *quasisupermodular* if:

$$F(a) \geq F(a \wedge a') \Rightarrow F(a \vee a') \geq F(a')$$

and

$$F(a) > F(a \wedge a') \Rightarrow F(a \vee a') > F(a').$$

Definition 4.5 *A two-person game (A, B, u_1, u_2) is quasi-supermodular game if for $i \in \{1, 2\}$:*
1. A, B is a compact subset of R;
2. u_i is upper-semi-continuous in s_i when s_{-i} is fixed and continuous in s_{-i} when s_i is fixed;
3. u_i is quasisupermodular in s_i and satisfies the single crossing property in $(x_i; x_{-i})$.

A special case and yet widely used definition of quasi-supermodular game is simply given as follows,

Definition 4.6 *A finite game (A, B, u_1, u_2) is quasi-supermodular if there are two partial orders $<_1$ and $<_2$ on A and B, respectively, such that u_1, u_2 satisfies the single crossing property.*

Henceforth, by quasi-supermodular games, we refer to the games defined by definition 4.6.

It is also known that any quasi-supermodular game has a PNE. Also, quasi-supermodularity generalize supermodularity in a trivial way, as one can easily find a game that is quasi-supermodular but not supermodular.

4.2 Formulating two-person games in first-order logic

We consider a first-order language with two sorts α and β, equality, and two predicates $\leq_1$ and $\leq_2$. We use "$\wedge$" for conjunction, "$\vee$" for disjunction, "$\neg$" for negation, "$\supset$" for implication, and "$\equiv$" for equivalence. Negation has the highest precedence, followed by conjunction and disjunction, implication, and then equivalence. The rule of precedence can be overridden by a new line. For instance, the following expression

$$p \supset q \wedge$$
$$q \supset p$$

stands for the sentence $(p \supset q) \wedge (q \supset p)$.

In our language, sort α is for player 1's strategies, and β for player 2's strategies. In the following, we use variables $x, x_1, x_2, ..$ to range over α, and $y, y_1, y_2, ...$ to range over β.

The two predicates represent the two players' preference relations. In the following, as we have already done above, we write $\leq_i (x_1, y_1, x_2, y_2)$ in infix notation as $(x_1, y_1) \leq_i (x_2, y_2)$, $i = 1, 2$, and $(x_1, y_1) \simeq_i (x_2, y_2)$ as a shorthand for

$$(x_1, y_1) \leq_i (x_2, y_2) \wedge (x_2, y_2) \leq_i (x_1, y_1),$$

where $i = 1, 2$. We also write $(x_1, y_1) <_i (x_2, y_2)$ as a shorthand for

$$(x_1, y_1) \leq_i (x_2, y_2) \wedge \neg(x_2, y_2) \leq_i (x_1, y_1).$$

The two relations need to be total orders (in the rest of the chapter, unless otherwise stated, all free variables in a displayed formula are assumed to be universally quantified from outside):

$$(x, y) \leq_i (x, y), \tag{4.1}$$

$$(x_1, y_1) \leq_i (x_2, y_2) \vee (x_2, y_2) \leq_i (x_1, y_1), \tag{4.2}$$

$$(x_1, y_1) \leq_i (x_2, y_2) \wedge (x_2, y_2) \leq_i (x_3, y_3) \supset (x_1, y_1) \leq_i (x_3, y_3), \tag{4.3}$$

where $i = 1, 2$. In the following, we denote by Σ the set of the above sentences. Thus two-person games correspond to first-order models of Σ, and two-person finite games correspond to first-order finite models of Σ. This correspondence extends to other type of games as well. For instance, let Σ_s be the union of Σ with the following two axioms:

$$(x_1, y_1) \simeq_1 (x_2, y_2) \supset (x_1 = x_2 \wedge y_1 = y_2),$$
$$(x_1, y_1) \simeq_2 (x_2, y_2) \supset (x_1 = x_2 \wedge y_1 = y_2).$$

Then strict games and models of Σ_s are isomorphic.

We now show how some other notions in game theory can be formulated in first-order logic. The condition for a profile (ξ, ζ) to be a PNE is captured by the following formula:

$$\forall x.(x, \zeta) \leq_1 (\xi, \zeta) \wedge \forall y.(\xi, y) \leq_2 (\xi, \zeta). \tag{4.4}$$

In the following, we shall denote the above formula by $NE(\xi, \zeta)$.

The following sentence expresses the uniqueness of PNE payoff in a game:

$$NE(x_1, y_1) \wedge NE(x_2, y_2) \supset (x_1, y_1) \simeq_1 (x_2, y_2) \wedge (x_1, y_1) \simeq_2 (x_2, y_2). \tag{4.5}$$

A game is strictly competitive if it satisfies the following property:

$$(x_1, y_1) \leq_1 (x_2, y_2) \equiv (x_2, y_2) \leq_2 (x_1, y_1). \tag{4.6}$$

Thus it should follow that

$$\Sigma \models (4.6) \supset (4.5). \tag{4.7}$$

Notice that we have assumed that all free variables in a displayed formula are universally quantified from outside. Thus (4.6) is a sentence of the form $\forall x_1, x_2, y_1, y_2 \varphi$. Similarly for (4.5).

Theorems like (4.7) can actually be generated automatically using the following theorem.

Theorem 4.7 *Suppose Q is a formula without quantifiers, $\vec{x}_1$ and $\vec{x}_2$ tuples of variables of sort α, and $\vec{y}_1$ and $\vec{y}_2$ tuples of variables of sort β. We have that*

1. $\Sigma \models \exists \vec{x}_1 \exists \vec{y}_1 \forall \vec{x}_2 \forall \vec{y}_2 Q \supset (4.5)$
   iff for all model G of Σ such that $|A| \leq |\vec{x}_1| + 2$ and $|B| \leq |\vec{y}_1| + 2$, we have that
   $G \models \exists \vec{x}_1 \exists \vec{y}_1 \forall \vec{x}_2 \forall \vec{y}_2 Q \supset (4.5)$,
   where A is the domain of G for sort α, and B the domain of G for sort β.

2. $\Sigma \models \exists \vec{x}_1 \exists \vec{y}_1 \forall \vec{x}_2 \forall \vec{y}_2 Q \supset \neg \exists x, y. NE(x, y)$
   iff for all model G of Σ such that $|A| \leq |\vec{x}_1| + 1$ and $|B| \leq |\vec{y}_1| + 1$ we have that
   $G \models \exists \vec{x}_1 \exists \vec{y}_1 \forall \vec{x}_2 \forall \vec{y}_2 Q \supset \neg \exists x, y. NE(x, y)$,
   where A is the domain of G for sort α, and B the domain of G for sort β.

Proof: It follows directly from Theorem 2.2. ■

In other words, to prove that a sentence of the form $\exists \vec{x}_1 \exists \vec{y}_1 \forall \vec{x}_2 \forall \vec{y}_2 Q$ is a sufficient condition for the uniqueness of PNE payoff, it suffices to verify that this is the case for all games of sizes up to $(|\vec{x}_1|+2) \times (|\vec{y}_1|+2)$, and to prove that it is a sufficient condition for the non-existence of PNE, it suffices to verify this for games of sizes up to $(|\vec{x}_1|+1) \times (|\vec{y}_1|+1)$.

Theorem 4.7 holds for many specialized games as well. For instance, it holds for strict games as well.

Theorem 4.8 *Theorem 4.7 holds when Σ is replaced by Σ_s.*

In fact, Theorem 4.7 holds when Σ is replaced by any set of universally quantified sentences.

4.3 Uniqueness of PNE payoffs

In this section, we consider the possibility of using computers to discover new classes of two-person games that have unique PNE payoffs. Our starting point is the class of two-person strictly competitive games. We first formulate the notions of games, strictly

competitive games and PNEs in first-order logic. Under our formulation, a class of games corresponds to a first-order sentence. In particular, the sentence that corresponds to the class of strictly competitive games is a conjunction of two binary clauses with all variables universally quantified. So we implemented a program that examines all these universally quantified conjunctions of binary clauses to see if there is another such condition that also captures a class of games with unique PNE payoffs. We did not expect much as these conditions are rather simple, but to our surprise, despite the simple form, our program returned various theorems, known or mostly unknown, that shed light on our understanding of this property. Before we start to describe these findings, let us take a briefly look at our procedure and setup.

4.3.1 Theorem discovering

Since $p \equiv q$ is logically equivalent to $(\neg p \vee q) \wedge (p \vee \neg q)$, the condition (4.6) for strictly competitive games can be written as a conjunction of two binary clauses:

$$(l_1 \vee l_2) \wedge (l_3 \vee l_4), \tag{4.8}$$

where each l_i, $1 \leq i \leq 4$, is a literal, i.e. either an atom or the negation of an atom. As we mentioned, we want to know if there are other sentences of the form (4.8) that also capture classes of games with unique PNE payoffs. In the following, we say that a condition φ is a *uniqueness condition* if whenever a game satisfies this condition, it has unique PNE payoff, that is, if $\Sigma \models \varphi \supset$ (4.5).

Based on Theorem 4.7, a straightforward way of discovering uniqueness conditions of the form (4.8) is as follows: For each condition of the form (4.8), check that if a 2×2 game does not have unique PNE payoff, then it does not satisfy this condition. There are 810,000 such conditions, 1950 non-isomorphic 2×2 two-person games, and among them 709 games that do not have unique PNE payoffs. Thus this strategy can be implemented on a modern computer even by brute-force search.

The search space can also be pruned by noticing that the conditions of the form (4.8) are not independent. For instance, condition

$$(x_1, y_1) \leq_1 (x_2, y_2)$$

entails (is stronger than) condition

$$(x_1, y_1) \leq_1 (x_1, y_2).$$

Once we know that a condition C is a uniqueness condition, those that entail C are no longer interesting as they become special cases of C, thus can be pruned.

However, checking logical entailment is in general not decidable for first-order logic. But as a strategy for pruning search space, we can use a weaker notion called *subsumption* on conditions of the form (4.8): C subsumes C' if there is a substitution σ such that $C\sigma = C'$. For our language, subsumption can be checked efficiently, and the search tree can be designed in such a way that the condition associated with a node always subsumes the conditions associated with the ancestors of the node. Thus once a condition is found to be a uniqueness condition, the entire sub-tree under this condition can be pruned.

However, we still need a way to check for complete logical entailment under Σ for conditions of the form (4.8). This is because we want every condition returned by our program to be a most general, "weakest" uniqueness condition in the sense that it does not entail any other uniqueness condition of the form (4.8). Fortunately, this can be done using the following proposition.

Proposition 4.3.1 *To check whether condition $\forall \vec{x}_1 \vec{y}_1 Q_1$ entails condition $\forall \vec{x}_2 \vec{y}_2 Q_2$ for all two-person games, it suffices to check this for all games up to $max\{|\vec{x}_2|, 1\} \times max\{|\vec{y}_2|, 1\}$, where Q_1 and Q_2 are formulas without quantifiers. This result holds for strict games as well.*

Notice that what we have described applies to the task of discovering uniqueness conditions of the form (4.8) for strict two-person games as well.

We now report our experimental results, first for general two-person games, and then for strict two-person games.

4.3.2 General games

For two-person general games, our program returns the following seven uniqueness conditions for 2x2 games.

$$(x_1, y) \leq_1 (x_2, y) \supset (x_2, y) \leq_2 (x_1, y) \wedge$$
$$(x, y_1) \leq_2 (x, y_2) \supset (x, y_2) \leq_1 (x, y_1) \tag{4.9}$$
$$(x_1, y) \leq_1 (x_2, y) \supset (x_1, y) \leq_2 (x_2, y) \wedge$$
$$(x, y_1) \leq_2 (x, y_2) \supset (x, y_2) \leq_1 (x, y_1) \tag{4.10}$$
$$(x_1, y) \leq_1 (x_2, y) \supset (x_2, y) \leq_2 (x_1, y) \wedge$$
$$(x, y_1) \leq_2 (x, y_2) \supset (x, y_1) \leq_1 (x, y_2) \tag{4.11}$$
$$(x_1, y_1) \leq_1 (x_2, y_1) \supset (x_1, y_2) \leq_2 (x_2, y_2) \wedge$$
$$(x, y_1) \leq_2 (x, y_2) \supset (x, y_1) \leq_1 (x, y_2) \tag{4.12}$$
$$(x_1, y) \leq_1 (x_2, y) \supset (x_1, y) \leq_2 (x_2, y) \wedge$$
$$(x_1, y_1) \leq_2 (x_1, y_2) \supset (x_2, y_1) \leq_1 (x_2, y_2) \tag{4.13}$$
$$(x_1, y_1) \leq_1 (x_2, y_2) \supset (x_1, y_1) \leq_2 (x_2, y_1) \wedge$$
$$(x_1, y_1) \leq_2 (x_2, y_2) \supset (x_2, y_1) \leq_1 (x_2, y_2) \tag{4.14}$$
$$(x_1, y_1) \leq_1 (x_2, y_2) \supset (x_1, y_2) \leq_2 (x_2, y_2) \wedge$$
$$(x_1, y_1) \leq_2 (x_2, y_2) \supset (x_1, y_1) \leq_1 (x_1, y_2). \tag{4.15}$$

By Theorem 4.7, these are also uniqueness conditions for all two-person games. Furthermore, since these are the only conditions returned by our program, for any sentence C of the form (4.8), if it is a uniqueness condition, then it must entail one of the above conditions under Σ. In other words, the above seven conditions are the weakest (most general) uniqueness conditions of the form (4.8).

Notice that condition (4.10) and condition (4.11) are symmetric in the sense that one can be obtained from the other by swapping the roles of the two players. So are (4.12) and (4.13), and (4.14) and (4.15). On the other hand, (4.9) is symmetric to itself. It is easy to see that if two conditions are symmetric, then one is a uniqueness condition iff the other is.

Condition (4.9) looks like condition (4.6) for strictly competitive games, except that the strategy of one of the players is fixed in each implication. As it turned out, it captures exactly the class of two-person games that are *weakly unilaterally competitive* [12]:

> "a game belongs to this class if a unilateral move by one player which results in an increase in that player's payoff also causes a (weak) decline in the payoffs of all other players. Furthermore, if that move causes no change in the mover's payoff then all other players' payoffs remain unchanged."

Clearly, if a game is strictly competitive, then it is also weakly unilaterally competitive, but the converse is not true in general. Kats and Thisse [12] showed that if a game is

weakly unilaterally competitive, then it has unique PNE payoff. For us, for two-person games, this follows directly from our computer output and Theorem 4.7.

Condition (4.10) can be given a similar interpretation:

> A two-person game satisfies this condition if a unilateral move by player 1 which results in a (weak) increase in his payoff also causes a (weak) increase in the payoff of player 2, but a unilateral move by player 2 which results in a (weak) increase in his payoff will causes a (weak) decline in the payoff of player 1.

Thus in this class of games, the two players are not equal, and it clearly favors player 2. The game may be competitive for player 1, but not for player 2.

Proposition 4.3.2 *Given a game that satisfies (4.10), if player 2's payoff is maximal at* (a, b)*, i.e.* $(a', b') \leq_2 (a, b)$ *for all* a', b'*, then there is a strategy* a^* *such that* (a^*, b) *is a PNE and* $(a^*, b) \simeq_2 (a, b)$*.*

Thus for the class of games that satisfy condition (4.10), the optimal strategy for player 2 is to do the strategy for which there is a strategy by the other player that will give him the maximum payoff. The following is an example of such games (as usual, player 1 is the row player, and player 2 the column player; the first number in a cell is the payoff of the row player, the second the column player):

3, 6	4, 5	5, 1
2, 3	1, 4	6, 2

It has a unique equilibrium $(3, 6)$.

As we mentioned, condition (4.11) is symmetric to condition (4.10), with the roles of the two players swapped. For the classes of games corresponding to the other conditions, (4.12) - (4.15), both players can obtain their maximal payoffs.

Proposition 4.3.3 *Given a game that satisfies one of the conditions (4.12) - (4.15), if player 1's (player 2's) payoff at* (a, b) *maximal, then there is a strategy* b^* *(*a^**) such that* (a, b^*) *(*(a^*, b)*) is a PNE where both players receive the maximum payoffs.*

Thus, from these two propositions, we see that the classes of games represented by the conditions (4.10) - (4.15) are not really "competitive" games. We can then conclude that among the classes of games that can be represented by a conjunction (4.8) of two binary clauses, the class of weakly unilaterally competitive games is the most general class of "competitive" and "fair" games that have unique PNE. As we mentioned above, by this we do not mean that other types of games are not interesting. In real life, unfair games like those described by (4.10) may well arise.

4.3.3 Strict games

We now describe our experimental results for strict games. Recall that these are games where for each player, different profiles have different payoffs. Thus uniqueness of PNE payoff simply means uniqueness of PNE in strict games.

Games with dominant strategies

We first consider conditions that mention only $\leq_1$:

$$s_1 \leq_1 s_2 \vee s_3 \leq_1 s_4.$$

For this class of conditions, our program outputs the following six uniqueness conditions on 2x2 strict games:

$$\begin{aligned}
&(x_1, y_1) \leq_1 (x_2, y_1) \vee (x_2, y_1) \leq_1 (x_1, y_2),\\
&(x_1, y_1) \leq_1 (x_2, y_1) \vee (x_2, y_2) \leq_1 (x_1, y_1),\\
&(x_1, y_1) \leq_1 (x_2, y_1) \vee (x_2, y_2) \leq_1 (x_1, y_2),\\
&(x_1, y_1) \leq_1 (x_2, y_2) \vee (x_2, y_1) \leq_1 (x_1, y_1),\\
&(x_1, y_1) \leq_1 (x_2, y_2) \vee (x_2, y_2) \leq_1 (x_1, y_2).
\end{aligned}$$

By Theorem 4.8, these are also uniqueness conditions for all strict two-person games. Notice that these conditions do not mention $\leq_2$. This means that if player 1's preference relation satisfies any of the above conditions, then the game has a unique PNE, no matter what the other player's preference relation is.

For instance, the first condition can be written as

$$\neg(x_1, y_1) \leq_1 (x_2, y_1) \supset (x_2, y_1) \leq_1 (x_1, y_2).$$

For strict games, this is equivalent to

$$(x_2, y_1) <_1 (x_1, y_1) \supset (x_2, y_1) \leq_1 (x_1, y_2)$$

as $\neg(x_1, y_1) \leq_1 (x_2, y_1)$ iff $(x_2, y_1) <_1 (x_1, y_1)$. It is not hard to see that the above condition implies the following condition:

$$\exists x \forall x', y.(x', y) \leq_1 (x, y),$$

meaning that no matter what player 2 does, the best response for player 1 is always the same. For strict games, this means that player 1 has a *strictly dominant strategy*: a strategy x is a strictly dominant strategy if for all other strategy x' of player 1, and any strategy y of player 2, $(x', y) <_1 (x, y)$. As it turned out, this is also the case for the other five conditions above, as the following proposition shows.

Proposition 4.3.4 *A strict game $G = (A, B, \leq_1, \leq_2)$ has a strictly dominant strategy for player 1 if and only if for any preference relation $\leq_2'$ for player 2, the game $G' = (A, B, \leq_1, \leq_2')$ has exactly one PNE.*

Given this result, there is no need to consider any condition of the form (4.8) that mentions only one player's preference.

It is interesting to note that for the prisoner's dilemma

4, 4	0, 5
5, 0	1, 1

each player has a strictly dominant strategy, thus should play this strategy. The dilemma is that each player can get a higher payoff by a unilateral move away from his dominant strategy.

Weakly unilaterally competitive games for individual players

For other conditions of the form (4.8), our program returns 16 uniqueness conditions for strict games. However, each of them has a symmetric one when the roles of the two players are swapped. Thus there are really only eight such conditions, given below:

$$(x_1, y) \leq_1 (x_2, y) \vee (x_1, y) \leq_2 (x_2, y), \tag{4.16}$$

$$(x_1, y_1) \leq_1 (x_1, y_2) \vee (x_1, y_2) \leq_2 (x_2, y_1), \tag{4.17}$$

$$(x_1, y_1) \leq_1 (x_1, y_2) \vee (x_2, y_2) \leq_2 (x_1, y_1), \tag{4.18}$$

$$(x_1, y_1) \leq_1 (x_1, y_2) \vee (x_2, y_2) \leq_2 (x_2, y_1), \tag{4.19}$$

$$(x_1, y_1) \leq_1 (x_2, y_2) \vee (x_1, y_2) \leq_2 (x_1, y_1), \tag{4.20}$$

$$(x_1, y_1) \leq_1 (x_2, y_2) \vee (x_2, y_2) \leq_2 (x_1, y_2), \tag{4.21}$$

$$(x_1, y_1) \leq_1 (x_1, y_2) \vee (x_1, y_1) \leq_2 (x_2, y_1), \tag{4.22}$$

$$(x_1, y_1) \leq_1 (x_2, y_1) \vee (x_2, y_2) \leq_2 (x_2, y_1). \tag{4.23}$$

In particular, we found that for strict games, a conjunction $C_1 \wedge C_2$ of two binary clauses is a uniqueness condition iff either C_1 or C_2 is a uniqueness condition.

The first condition is equivalent to

$$(x_2, y) \leq_1 (x_1, y) \supset (x_1, y) \leq_2 (x_2, y) \tag{4.24}$$

as in strict games, $s_1 \leq_1 s_2$ iff $s_1 <_1 s_2 \vee s_1 = s_2$. This is exactly one of the two conjuncts in the condition (4.9) for weakly unilaterally competitive games.

Now swap the roles of the two players in (4.24), we get the following condition

$$(x, y_1) \leq_2 (x, y_2) \supset (x, y_2) \leq_1 (x, y_1), \tag{4.25}$$

which is exactly the other conjunct in the condition (4.9).

In the following, we call a game that satisfies (4.24) a *weakly unilaterally competitive for player 1*, and a game that satisfies (4.25) a *weakly unilaterally competitive for player 2*. Thus a game is weakly unilaterally competitive if it is weakly unilaterally competitive for both players. The following example shows that a game can be weakly unilaterally competitive for player 1 but not for player 2.

2, 1	3, 4
1, 2	4, 3

This example also shows that a weakly unilaterally competitive game for player 1 may not be *almost strictly competitive* [3]: a game is almost strictly competitive if

1. the set of payoff vectors of the PNEs is the same as the set of payoff vectors of the *twisted equilibria*; and

2. there is a PNE that is also a twisted equilibrium,

where (a, b) is a twisted equilibrium if no player can decrease the payoff of the other player by a unilateral change of his strategy: for every $a' \in A$ ($b' \in B$), $(a, b) \leq_2 (a', b)$ ($(a, b) \leq_1 (a, b')$). For this example, it is easy to see that the only equilibrium of the game, $(4, 3)$, is not a twisted equilibrium.

As it turns out, (4.24) and (4.25) are the only non-trivial conditions. The last two conditions (4.22) and (4.23) can never be satisfied by games larger or equal to 3x3. The remaining five conditions (4.17) - (4.21) are games with dominant strategies.

Proposition 4.3.5 *If G is a strict game and satisfies one of the conditions (4.17) - (4.21), then one of the players has a strictly dominant strategy in G.*

4.3.4 Generalization of the experimental results

To summarize, for strict games, the only interesting uniqueness conditions that can be expressed by a conjunction of two binary clauses and include games that do not have dominant strategies are weakly unilaterally competitive conditions for individual players, (4.24) and (4.25). This led us to wonder if these two conditions are also necessary conditions for a strict game to have a unique PNE. However, it is easy to see that this is not

the case. In fact, a universal condition like (4.8) can never be both a necessary and a sufficient condition for a game to have unique PNE. This is because for any given game, no matter how many PNEs it has, we can always extend it by one more strategy for each player, and make it into a game with a unique PNE by assigning payoffs large enough to a profile made of the two new strategies. However, if a universal condition is satisfied by a game, it is also satisfied by any of its sub-games.

This led us to consider not individual games, but classes of games under certain equivalence relation.

Two games $G_1 = (A, B, \leq_1, \leq_2)$ and $G_2 = (A', B', \leq'_1, \leq'_2)$ are *unilaterally order equivalent*[2] if

- $A = A'$, and $B = B'$.
- For every $a \in A$, $b, b' \in B$, $(a, b) \leq_2 (a, b')$ iff $(a, b) \leq'_2 (a, b')$.
- For every $b \in B$, $a, a' \in A$, $(a, b) \leq_1 (a', b)$ iff $(a, b) \leq'_1 (a', b)$.

They are *best-response equivalent* [31] if for all $a \in A$, $B_2(a)$ in G_1 and G_2 are the same, and for all $b \in B$, $B_1(b)$ in G_1 and G_2 are the same. Clearly, if G_1 and G_2 are unilaterally order equivalent, then they are also best-response equivalent, but the converse is not true in general. Both notions of equivalence preserve PNEs.

We have the following result.

Theorem 4.9 *A strict game has at most one PNE iff it is best-response equivalent to a strictly competitive game.*

To prove this theorem, for any given game $G = (A, B, u_1, u_2)$, we associate with it a direct graph R whose vertices are profiles of the game and there is an arc from s to s' if (s, s') is in the following set:

$$\{((a, b), (a', b)) \mid a' \in B_1(b), a \notin B_1(b)\} \cup$$
$$\{((a, b), (a, b')) \mid b \in B_2(a), b' \notin B_2(a)\}.$$

The theorem then follows from the following two lemmas about R.

Lemma 4.10 *A 2-person game $G = (A, B, u_1, u_2)$ is best-response equivalent to a strictly competitive game iff R has no cycle.*

Proof: If R has a cycle, then G cannot be equivalent to a strictly competitive game because if $G' = (A, B, u'_1, u'_2)$ is such a game, then for any profiles s_1 and s_2, if there is an

[2]We call it unilaterally order equivalence to distinguish it from *order equivalence* [31] that requires both the row and column orders in the two games to be the same for both players.

arc from s_1 to s_2 in R, then $u_1'(s_1) < u_1'(s_2)$. So along the cycle, there must be a sequence $u_1'(t_1) < u_1'(t_2) < \ldots < u_1'(t_k) < u_1'(t_1)$, which is a contradiction.

Now if R has no cycle, construct a game $G' = (A, B, u_1', u_2')$ where u_1' and u_2' are defined as follows:

- $R_0 = \{(s, s') |$ there is an arc from s to s' in $R\}$, $S_0 = \{s \mid s \in A \times B$, there is no s' such that $(s', s) \in R_0\}$.
- Suppose that R_k and S_k is defined, let

$$R_{k+1} = \{(s, s') \mid (s, s') \in R_k, \text{ and } s, s' \notin S_k\},$$
$$S_{k+1} = \{s \mid \text{ for some } s', (s, s') \in R_k \text{ but there is no } s' \text{ such that } (s', s) \in R_{k+1}\}.$$

- Since R has no cycles, there is a finite number n such that $R_k = S_k = \emptyset$, and $A \times B = S_0 \cup \cdots \cup S_n$.
- Let u_1' be a one-to-one function from $A \times B$ to the set of positive integers such that if $i < j$, then for any $s \in S_i$ and $s' \in S_j$, $u_1'(s) < u_1'(s')$.
- Let $u_2' = -u_1'$.

Clearly, G' is strictly competitive, and best-response equivalent to G. ■

Lemma 4.11 *If G is a strict 2-person game, and its graph R has a cycle, then G has more than one Nash equilibria.*

Proof: Suppose $s_1, ..., s_k, s_{k+1}$ is a cycle in R. Suppose $s_1 = (a, b)$. Then either $s_2 = (a', b)$ for some $a' \neq a$ or $s_2 = (a, b')$ for some $b' \neq b$. Suppose it is the first case, $s_2 = (a', b)$. The proof for the second case is similar. Then by our construction of R, $s_3 = (a', b')$ for some $b' \neq b$, $s_4 = (a'', b')$ for some $a'' \neq a'$. We show that s_2 and s_4 are both Nash equilibria of G. Because there is an arc from (a, b) to (a', b) in R, $a' \in B_1(b)$. Because there is an arc from (a', b) to (a', b') in R, $b \in B_2(a')$. Thus $s_2 = (a', b)$ is a Nash equilibrium. Similarly, Because there is arc from (a', b') to (a'', b') and an arc from (a'', b') to (a'', b'') in R for some b'' (it is possible that $a = a''$ and $b = b''$) $a'' \in B_1(b')$ and $b' \in B_2(a'')$. Thus $s_4 = (a'', b')$ is a Nash equilibrium as well. Since G is a strict game, $s_2 \neq s_4$ implies $u_1(s_2) \neq u_1(s_4)$, thus G has at least two Nash equilibria with different payoffs. ■

Theorem 4.9 does not hold for general two-person games. For instance, the following game

1, 1	2, 2
2, 2	1, 1

has a unique equilibrium $(2, 2)$ but is not best-response equivalent to any strictly competitive games.

4.4 Existence of PNE

The same approach can be applied to discover the sufficient conditions for the existence of PNE.

4.4.1 Theorem discovering

As mentioned, we conducted experiments on discovering the sufficient conditions of the existence of PNE, using the same setting as in that of uniqueness PNE payoffs. We have made the following observations concerning the results.

- Failure of the finite verifiable property. As one might notice, the formula that describe the existence of PNE,

$$\exists x, yNE(x, y), \tag{4.26}$$

  violates the format in Theorem 2.2, no matter the format of the sufficient condition.

- The results returned by our program (that is, the ones pass the verification on small domains) are extremely likely to be true in general. In fact, we haven't found any counter example that passes the small domain tests (up to 3×3 games) but fails in general. Thus, these conditions serve as very good conjectures for the sufficient conditions.

- The conditions returned by our program implies either,

  1. Each player possesses a dominant strategy, or;
  2. Potential game, or;
  3. Supermodular game.

- Besides the three classes of games described above, there still exist other games possessing a PNE.

4.4.2 Generalization of the experimental results

Although the last item reveals the sad fact that the program cannot return the complete set of conditions that characterize the existence of PNE, we do notice that, two important classes of games among those conditions, namely potential games and supermodular games, overlap extremely frequently. It lets us wonder if we can similarly generalize a result up to best-response equivalence, as we did in the uniqueness case, that connect these two classes of games, and the answer is surely affirmative.

Given a 2-person game, a best-response path of the game is a sequence of profiles $s_1, ..., s_n$ such that for each $1 \leq i < n$, s_i and s_{i+1} differ on exactly one coordinate with the deviating player moving to a best response: if $s_i = (a, b)$ and $s_{i+1} = (a', b)$, then a' is a best response of player 1 to the action b by player 2, similarly if s_i and s_{i+1} differ on the second coordinate.

Voorneveld [44] showed that a game with countable strategy sets is best response equivalent to an ordinal potential game iff it has no best response cycle. We prove here that a 2-person strict finite game is best response equivalent to a quasi-supermodular game iff it has no best response cycle. Thus, a finite strict 2-person game is best response equivalent to an ordinal potential game iff it is best response equivalent to a quasi-supermodular game.

Theorem 4.12 *A strict game is best response equivalent to a quasi-supermodular game iff it has no best response cycle.*

Corollary 4.13 *A 2-person strict game is best-response equivalent to a quasi-supermodular game iff it is best-response equivalent to an ordinal potential game.*

Notice that best-response paths model Cournot dynamics, thus Theorem 4.12 also implies that if Cournot dynamics does not cycle, then it must be best-response equivalent to a quasi-supermodular game.

We now prove Theorem 4.12 through two lemmas. The first one relates quasi-supermodular games to the complementarity of best response functions, and the second the complementarity of best-response function to the acyclicity of best-response paths. While the "only if" part of Theorem 4.12 is already entailed by the current results, these two lemmas also provide an interesting alternative proof.

Given a 2-person game (A, B, u_1, u_2), and two linear orderings $<_1$ and $<_2$ of A and B, respectively, we say that the best-response function B_1 for player 1 is *non-decreasing* with respect to $<_1$ and $<_2$, if for each pair $b_i <_2 b_j$ $a_i = B_1(b_i)$ and $a_j = B_1(b_j)$, we have $a_i \leq_1 a_j$ and similarly for player 2, the best-response function B_2 is non-decreasing with respect to $<_1$ and $<_2$, if for each pair $a_i < a_j$ $1 \leq i < j \leq m$, $b_i = B_2(a_i)$ and $s'_j = (a_j, b_j)$, we have $b_i \leq_2 b_j$

Essentially, if both players' best-response functions are non-decreasing, then they are complementary.

Lemma 4.14 *A strict game is best-response equivalent to a quasi-supermodular game iff there exist two linear orderings under which the best-response functions for both players are non-decreasing.*

Non-decreasing best-response functions for quasi-supermodular game is a well known intuition (the only if part), although we still prove it as follows since it is simple.

Proof: Let $G = (A, B, u_1, u_2)$ be a strict game.

$\Rightarrow$: Suppose G is best-response equivalent to a quasi-supermodular game G'. Clearly, the best-response functions of G and G' are the same under any linear orderings. Now let $G' = (A, B, u_1', u_2')$. Since G' is quasi-supermodular, there are two linear orderings $<_1$ of A and $<_2$ of B such that u_1' and u_2' satisfy the single crossing properties. Then in G', the best-response functions under $<_1$ and $<_2$ are non-decreasing. For otherwise, suppose that B_1 is not non-decreasing in G'. Then there are two profiles (a_1, b_1) and (a_2, b_2) such that

$$a_1 \in B_1(b_1), \quad a_2 \in B_1(b_2), \quad b_1 <_2 b_2,$$
$$a_2 <_1 a_1.$$

Thus $u_1'(a_1, b_1) > u_1'(a_2, b_1)$ and $u_1'(a_2, b_2) > u_1'(a_1, b_2)$, which violate the single crossing conditions.

$\Leftarrow$: Suppose the best-response functions of the two players in G are non-decreasing under $<_1$ and $<_2$. We denote in this part that for $a_i, a_j \in A$, $a_i <_1 a_j$ if $i < j$. Similar for $b_i, b_j \in B$. Now consider the following game $G' = (A, B, u_1', u_2')$:

- if $s < t$, then $u_1'(a_i, b_s) < u_1'(a_j, b_t)$ for any a_i and a_j in A;
- for any $b_s \in B$, if $B_1(b_s) = \{a_k\}$ in G, then
  - if $i \neq a_k$, then $u_1'(a_i, b_s) < u_1'(a_k, b_s)$;
  - if $i > j > k$, then $u_1'(a_i, b_s) < u_1'(a_j, b_s)$;
  - if $i < j < k$, then $u_1'(a_j, b_s) > u_1'(a_i, b_s)$;
  - if $i < k < j$, then $u_1'(a_j, b_s) > u_1'(a_i, b)$;
- similarly for u_2'.

Clearly, one can find two such functions u_1' and u_2', and that G' is best-response equivalent to G. We show that G' is quasi-supermodular on $<_1$ and $<_2$. Suppose $i < j$, $s < t$, and $u_1'(a_j, b_s) > u_1'(a_i, b_s)$. We show that $u_1'(a_j, b_t) > u_1'(a_i, b_t)$. Let $B_1(b_s) = \{a_k\}$ and $B_1(b_t) = \{a_l\}$. Then $k \leq l$ according to the non-decreasing property of player one's best-response function. Given $u_1'(a_j, b_s) > u_1'(a_i, b_s)$, there are three cases:

1. $j = k$. We have $i < j \leq l$, thus $u_1'(a_j, b_t) > u_1'(a_i, b_t)$.

2. $i < j < k$. We have $i < j < k \leq l$, thus $u_1'(a_j, b_t) > u_1'(a_i, b_t)$.

3. $i < k < j$. We have either $i < l \leq j$ or $i < j < l$. Either way, we have $u_1'(a_j, b_t) > u_1'(a_i, b_t)$ by our construction.

Thus u_1' satisfies the single crossing condition. Similarly, u_2' satisfies the single crossing condition. So G is a quasi-supermodular game. ■

Lemma 4.15 *A strict 2-person game has no best-response cycle iff there are two linear orderings under which both players' best response functions are non-decreasing.*

Proof: Let $G = (A, B, u_1, u_2)$ be a strict game.

$\Leftarrow$: Let $<_1$ and $<_2$ be two linear orderings of A and B, respectively, under which the best-response functions B_1 and B_2 for player 1 and 2, respectively, are non-decreasing. We show that G has no best-response cycle. We begin with a profile (a_1, b_1) where, with loss of generality, $\{a_1\} = B_1(b_1)$. Let then player 2 and player 1 deviates to each own best-response alternatively in the following rounds. If in the first round, player 2 deviates to an action b_2, there are three cases

- Case 1: $b_1 <_2 b_2$, then according to the non-decreasing property of B_1, we have $a_1 \leq_1 a_2$ where $\{a_2\} = B_1(b_2)$. If $a_1 = a_2$, it means that (a_1, b_2) is a PNE, where the best response path terminates. Otherwise $a_1 <_1 a_2$, this process will continue generating new profiles that alternatively increase in each player's action until it meets a PNE and terminates. This shows that the best-response path generated this way can not return to where it started, therefore in this case there is no best-response cycle.

- Case 2: $b_2 <_2 b_1$. It is not hard to see that this process will continue generating new profiles that strictly increases in one of its players rank until it meets a PNE. This shows again that there is no best-response cycle.

- Case 3: $b_1 = b_2$, this means that (a_1, b_1) is a PNE, where the best-response path terminates. No cycle exists.

We have proved the $\Leftarrow$ part of the lemma.

$\Rightarrow$: We prove this part constructively. Suppose there is no best-response cycle in G. Our following procedure will produce two ranking functions, $rank_A$ that maps each $a \in A$ to an integer between 1 and $|A|$, and $rank_B$ that maps each $b \in B$ to an integer between

1 and $|B|$. From these two rankings, we get two linear orderings on A and B: $a <_1 a'$ if $rank_A(a) > rank_A(a')$, and $b <_2 b'$ if $rank_B(b) > rank_B(b')$. We show that under these two linear orderings, the best-response functions for both players are non-decreasing.

1. Initially, let $A_r = A$, $B_r = B$, $I_A = I_B = 1$, $A_c = B_c = []$ (the empty list), $rank_A(a) = 0$ for any $a \in A$, and $rank_B(b) = 0$ for any $b \in B$.

2. **while** $A_r \neq \emptyset$ or $B_r \neq \emptyset$ **do**

   2.1. Let $(a, b) \in A_r \times B_r$ be a Nash equilibrium of G (there must be such a Nash equilibrium as we show below), and let $A_c = [a]$, $B_c = [b]$.

   2.2. **while** $A_c \neq []$ or $B_c \neq []$ **do**

      i. **If** $A_c \neq []$ **then**

         (a) Let a^* be the first element in A_c.

         (b) $rank_A(a^*) = I_A$; $I_A = I_A + 1$.

         (c) Delete a^* from A_c.

         (d) Delete a^* from A_r.

         (e) For each $b' \in B$ such that $B_1(b') = \{a^*\}$, add b' to the end of B_c. (If there are more than one such b', the order by which they are added to B_c does not matter.)

      ii. **If** $B_c \neq []$ **then**

         (a) Let b^* be the first element in B_c.

         (b) $rank_B(b^*) = I_B$; $I_B = I_B + 1$.

         (c) Delete b^* from B_c.

         (d) Delete b^* from B_r.

         (e) For each $a' \in A$ such that $B_2(a') = \{b^*\}$, add a' to the end of A_c. (If there are more than one such a', the order by which they are added to A_c does not matter.)

We now show the correctness of step 2.1 and that the best-response sequences for both players under the orderings output by the procedure are non-decreasing.

Let A_{r_i} and B_{r_i} be the A_r and B_r at the beginning of the ith loop and let $G_i = (A_{r_i}, B_{r_i}, u_1, u_2)$, we have the following properties of our procedure that guarantee the precondition of 2.1 can always be satisfied.

Proposition 4.4.1 *For each $G_i = (A_{r_i}, B_{r_i}, u_1, u_2)$, we have*

1. *For all $b_j \in B_{r_i}$, $B_1(b_j) \subseteq A_{r_i}$.*

2. *For all* $a_j \in A_{r_i}$, $B_2(a_j) \subseteq B_{r_i}$.

3. G_i *has no best-response cycle*

4. *Every Nash equilibrium of* G_i *is also one of the original game* G.

Proof: We prove this proposition by induction on i.

- Base case: It is easy to verify that $G_1 = G$ satisfies 1-4.
- Inductive case: Suppose G_i satisfies 1-4, we now verify 1-4 for G_{i+1}.

  1. Suppose otherwise, there exists $b_j \in B_{r_{i+1}}$ such that $B_1(b_j) = \{a\} \subseteq A_{r_i} \setminus A_{r_{i+1}}$ According to our procedure, a will be deleted during this loop, which means b_j will been added to B_c and deleted during this loop because B_c will be empty at the end of the loop. This means $b_j \in B_{r_i} \setminus B_{r_{i+1}}$, a contradiction.
  2. Similar to above.
  3. 1-2 tells us the best-responses of G_{i+1} are also best-responses of G. Suppose G_{i+1} has a best-response cycle, this cycle would still be one in G, a contradiction.
  4. This part also follows directly from 1-2.

■

Since G_i has no best-response cycle $\Leftrightarrow$ G_i is best-response equivalent to an ordinal potential game $\Rightarrow$ G_i has a Nash equilibrium. According to 1 and 4 of the above proposition, G_i always has a Nash equilibrium. Up to now, we have proved that the precondition of 2.1 in our procedure can always be satisfied.

Proposition 4.4.2 *The best-response functions under the orderings generated by our procedure are non-decreasing.*

Proof: For B_2 and for all $a_1, a_2 \in A$ with $a_2 <_1 a_1$, let $B_2(a_1) = \{b_1\}$ and $B_2(a_2) = \{b_2\}$, we show in the following that $b_2 \leq_1 b_1$.

If $b_1 = b_2$, we get to the conclusion immediately.

Now consider $b_1 \neq b_2$, when b_1 is the first element of B_c, a_1 is added to the end of A_c because $B_2(a_1) = \{b_1\}$. Similarly, when b_2 is the first element of B_c, a_2 is added to the end of A_c because $B_2(a_2) = \{b_2\}$. Since $a_2 <_1 a_1$, we must have a_2 is added to A_c later than a_1. This means b_2 appears as the first element of B_c later than b_1. So we get $b_2 <_2 b_1$.

Similar for player B_1. ■

■

4.5 Summary and discussion

To sum up, we have conducted theorem discovery on two types of theorems, the conditions that imply the unique PNE payoffs as well as those imply the existence of PNE.

For the uniqueness part, our program returned a condition that is more general than the strict competitiveness condition. As it turned out, it exactly corresponds to Kats and Thisse's [12] class of *weakly unilaterally competitive* two-person games. Our program also returned some other conditions. Two of them capture a class of "unfair" games where one player has advantage over the other. The remaining ones capture games where everyone gets what he wants - each receives his maximum payoff in every equilibrium state, thus there is no real competition among the players. Thus one conclusion that we can draw from this experiment is that among all classes of games that can be expressed by a conjunction of two binary clauses, the class of weakly unilaterally competitive games is the most general class of "competitive" and "fair" games that have unique PNE payoffs. Of course, this does not mean that the other conditions are not worth investigating. For instance, sometimes one may be forced to play an unfair game.

For the same set of conditions, we also consider strict two-person games where different profiles have different payoffs for each player. Among the results returned by our program, two of them are exactly the two conjuncts in Kats and Thisse's weakly unilaterally competitive condition, but the others all turn out to be special cases of games with dominant strategies. Motivated by these results, we consider certain equivalent classes of games, and show that a strict game has a unique PNE iff it is best-response equivalent [31] to a strictly competitive game.

For the existence part, the program outputs per se are not so exciting, as they are among the special cases of either games with dominant strategies, potential games or supermodular games. However, a closer look at them gives us very good intuition that helps to prove a rather surprising result, which says that in strict games, potential games and supermodular games are best response equivalent.

We want to advocate that this methodology be applied to more theorems. For instance, one can also prove that if we replace the formula that we describe the unique PNE payoff by the pareto optimality of PNE, the finitely verifiable property still holds. This amounts to say that, to prove such a condition is a sufficient condition for pareto optimality of PNE, it also suffices to verify it for all the 2×2 games. Moreover, as we mentioned, even for some theorem format that may not pertain finitely verifiable property, as in the existence case, going through our routine test still provides valuable conjectures and intuitions which can aid later manual discovery.

Chapter 5

Proving and discovering theorems in social choice theory

"*Social choice theory, a science of the impossible.*"

—*Handbook of social choices and welfare*

Arrow's Impossibility Theorem is one of the landmark results in social choice theory. Over the years since the theorem was proved in 1950, quite a few alternative proofs have been put forward. In this chapter, we propose yet another alternative proof of the theorem. The basic idea is to use induction to reduce the theorem to the base case with 3 alternatives and 2 agents and then use computers to verify the base case. This turns out to be an effective approach for proving other impossibility theorems such as Muller-Satterthwaite and Sen's theorems as well. Motivated by the insights of the proof, we discover a new theorem with the help of computer programs. We believe this new proof opens an exciting prospect of using computers to discover similar impossibility or even possibility results.

5.1 Social choice theory and impossibility theorems

The particular theorems that we are interested in are the impossibility theorems such as those by Arrow [2], Sen [35], and Muller and Satterthwaite [23] in social choice theory [1], an area concerning about how individual preferences can be aggregated to form a collective preference in a society. Social choice theory has sometimes been called "a science of the impossible" because of the many famous impossibility theorems that have been proved in it. Among them, Arrow's theorem [2] on the non-existence of rational social welfare function is without doubt the most famous one. It shows the non-existence of the collective social preference (called social welfare function) even when some minimal standards such as Pareto efficiency and non-dictatorship are imposed. Arrow's original proof of this result is relatively complex, and over the years, quite a few alternative proofs have been advanced (see e.g [8, 4, 37, 9]).

We propose yet another alternative proof of this result, with the help of computers. Briefly, Arrow's theorem says that in a society with at least three possible outcomes (alternatives) for each agent, it is impossible to have a social welfare function that satisfies

the following three conditions: unanimity (Pareto efficiency), independent of irrelevant alternatives (IIA), and non-dictatorship. We shall show by induction that this result holds if and only if it holds for the base case when there are exactly two agents and three alternatives (the single agent case is trivial). For the base case, we verify it using computers in two ways. One views the problem as a constraint satisfaction problem (CSP), and uses a depth-first search algorithm to generate all social welfare functions that satisfy the first two conditions, and then verifies that all of them are dictatorial. The other translates these conditions to a logical theory and uses a SAT solver to verify that the resulting logical theory is not satisfiable. Either way, it took less than one second on an AMD Opteron-based server (with 4 1.8GHz CPUs and 8GB RAM) for the base case to be verified.

As it turns out, this strategy works not just for proving Arrow's theorem. The same inductive proof can be adapted almost directly for proving other impossibility results such as Sen's and Muller-Satterthwaite theorems. We have used it to prove Gibbard-Satterthwaite theorem [10, 34] as well, but we leave its proof to the next chapter.

As a byproduct of our proof of Arrow's theorem, the social welfare functions that satisfies IIA only in the base case can all be generated by our program. To our surprise, the number of such functions is so small that we are able to look at them one by one. By doing so, we form an interesting conjecture and then prove it using the same techniques as in the previous proofs. We then demonstrate the powerfulness of the newly proved theorem by showing that it subsumes both Arrow's and Wilson's theorems.

These proofs suggest that many of the impossibility results in social choice theory are all rooted in some small base cases. Thus an interesting thing to do is to use computers to explore these small base cases to try to come up with new conjectures automatically, and to understand the boundary between impossibility and possibility results. This is what we think the long term implication of our new proofs of Arrow's and other impossibility theorems lies, and the main reason why we want to formulate the conditions in these theorems in a logical language and use a SAT solver to check their consistency.

5.2 Arrow's Theorem

A voting model is a tuple (N, O), where N is a finite set of individuals (agents) and O a finite set of outcomes (alternatives). An agent's preference ordering is a linear ordering of O, and a preference profile $>$ of (N, O) is a tuple $(>_1, ..., >_n)$, where $>_i$ is agent i's preference ordering, and n the size of N. In the following, when N is clear from the context, we also call $>$ a preference profile of O. Similarly, when O is clear from the context, we also call it a preference profile of N.

Definition 5.1 *Given a voting model (N, O), a* social welfare function *is a function $W : L^n \to L$, where L is the set of linear ordering of O, and n the size of N.*

A social welfare function defines a social ordering for each preference profile. If we consider the social ordering given by a social welfare function as the aggregates of the preference orderings of the individuals in the society, it is natural to impose some conditions on it. For instance, it should not be dictatorial in that the aggregated societal preference ordering always is the same as a particular individual's preference. Arrow showed that a seemingly minimal set of such conditions turns out to be inconsistent.

In the following, given a preference profile $> = (>_1, ..., >_n)$, we sometimes write $>_W$ for $W(>)$. Thus both $a >_W b$ and $a\ W(>)\ b$ mean the same thing: the alternative a is preferred over the alternative b according to the societal preference ordering $W(>)$.

Definition 5.2 *A social welfare function W is* unanimous *(Pareto efficient) if for all alternatives a_1 and a_2, we have that if $a_1 >_i a_2$ for every agent i, then $a_1 >_W a_2$*

In words, if everyone ranks alternative a_1 above a_2, then a_1 must be ranked above a_2 socially.

Definition 5.3 *A social welfare function W is* independent of irrelevant alternatives *(IIA) if for all alternatives a_1 and a_2, and all preference profiles $>'$ and $>''$, we have that $\forall i\ a_1 >'_i a_2$ iff $a_1 >''_i a_2$ implies that $a_1 >'_W a_2$ iff $a_1 >''_W a_2$.*

Literally, IIA means that the relative social ordering of two alternatives depends only on their relative orderings given by each agent and has nothing to do with other alternatives.

Definition 5.4 *An agent i is a dictator in a social welfare function W if for all alternatives a_1 and a_2, $a_1 >_W a_2$ iff $a_1 >_i a_2$. If there is a dictator in W, then it is said to be dictatorial. Otherwise, W is said to be non-dictatorial.*

It is easy to see that if there are at least two alternatives, then there can be at most one dictator in any social welfare function.

Theorem 5.5 ***(Arrow's theorem [2])*** *For any voting model (N, O), if $|O| \geq 3$, then any social welfare function that is unanimous and IIA is also dictatorial.*

Arrow's original proof of this result is somewhat complicated, and there are several alternative proofs by others, e.g [8, 4, 9]. We now give yet another one using induction.

5.3 An inductive proof of Arrow's Theorem

For ease of presentation, we assume the following notations.

- For any set S, we use S_{-a} to denote $S \setminus \{a\}$, i.e. the result of deleting a in S.
- We extend the above notation to tuples as well: if $t = (t_1, ..., t_n)$, then we use t_{-i} to denote the tuple $(t_1, ..., t_{i-1}, t_{i+1}, ..., t_n)$. Furthermore, we use (t_{-i}, s) to denote the result of replacing ith item in t by s: $(t_{-i}, s) = (t_1, ..., t_{i-1}, s, t_{i+1}, ..., t_n)$. We use $t_{-\{i,j\}}$ to denote $(t_{-i})_{-j}$.
- If $>$ is a linear ordering of O, and $a \in O$, then we let $>_{-a}$ be the restriction of $>$ on O_{-a}: for any $a', a'' \in O_{-a}$, $a' >_{-a} a''$ iff $a' > a''$. On the other hand, if $>$ is a linear ordering of O_{-a} for some $a \in O$, then we let $>^{+a}$ be the extension of $>$ to O such that for any $a' \in O_{-a}$, $a' >^{+a} a$. Similarly, we let $>^{a+}$ to be the extension of $>$ to O such that for any $a' \in O_{-a}$, $a >^{a+} a'$. Thus if $>$ is a linear ordering of O, and $a \in O$, then $>_{-a}^{+a}$ is $(>_{-a})^{+a}$, i.e. the result of moving a to the bottom of the ordering. These notations extend to tuples of orderings. Thus if $>$ is a preference profile of (N, O_{-a}), then

  $$>^{+a} = (>_1, ..., >_n)^{+a} = (>_1^{+a}, ..., >_n^{+a}),$$

  which will be a preference profile of (N, O). Similarly for $>^{a+}$

Like any inductive proof, there are two cases for our proof, the inductive case and the base case.

5.3.1 The inductive case

Lemma 5.6 *If there is a social welfare function for n individuals and $m+1$ alternatives that is unanimous, IIA and non-dictatorial, then there is a social welfare function for n individuals and m alternatives that satisfies these three conditions as well, for all $n \geq 2, m \geq 3$.*

Proof: Let $N = \{1, ..., n\}$ be a set of n agents, O a set of $m+1$ alternatives, and W a social welfare function for (N, O) that satisfies the three conditions in the lemma. We show that there is an $a \in O$ such that the "restriction" of W on O_{-a} also satisfies these three conditions.

For any $a \in O$, we define the restriction of W on O_{-a}, written W_a, to be the following function: for any preference profile $> = (>_1, ..., >_n)$ of O_{-a}, $W_a(>) = W(>^{+a})_{-a}$. In other words, $W_a(>)$ is the result of applying W to the preference profile $>^{+a}$ of O, and

then projecting it on O_{-a}. The key property of this welfare function is that for any a' and a'' in O_{-a}, and any preference profile $>$ of O_{-a}, $a'\ W_a(>)\ a''$ iff $a'\ W(>^{+a})\ a''$.

We show that W_a is unanimous and IIA:

- Suppose $a', a'' \in O_{-a}$ and $a' >_i a''$ for all i. By our definition $a' >_i^{+a} a''$ for all i as well. Since W is unanimous, $a'\ W(>^{+a})\ a''$. Thus $a'\ W_a(>)\ a''$. This shows that W_a is unanimous.

- Let $a', a'' \in O_{-a}$ and $>', >''$ be two preference profiles of O_{-a} such that $\forall i\ \ a' >_i' a''$ iff $a' >_i'' a''$. Thus $\forall i\ \ a' >_i'^{+a} a''$ iff $a' >_i''^{+a} a''$ as well. Since W is IIA, $a'\ W(>'^{+a})\ a''$ iff $a'\ W(>''^{+a})\ a''$. Hence $a'\ W_a(>')\ a''$ iff $a'\ W_a(>'')\ a''$. This shows that W_a is also IIA.

We now show that there is an $a \in O$ such that W_a is not dictatorial. First for any $a \in O$ and any $a', a'' \in O_{-a}$, and any profile $>$ of O, we have

$$a'\ >_W\ a'' \text{ iff } a'\ W(>_{-a}^{+a})\ a''. \tag{5.1}$$

This follows because W is IIA and $a', a'' \in O_{-a}$.

Now let b be any alternative in O. Suppose W_b has a dictator, say agent 1 in it. Since W is not dictatorial, there must be a preference profile $>$ of O and some $c, d \in O$ such that $c >_1 d$ but $d >_W c$. Since $|O| = m+1 > 3$, we can find an alternative $e \in O_{-b} \setminus \{c, d\}$. We now show that W_e is not dictatorial. Suppose otherwise. There are two cases:

- Agent 1 is again the dictator in W_e. Then $W_e(>_{-e})$ and $>_1$ agree on c and d. Thus $c\ W_e(>_{-e})\ d$. By our definition of W_e, this means that $c\ [W(>_{-e}^{+e})]_{-e}\ d$. Since $c, d \in O_{-e}$, this means that $c\ W(>_{-e}^{+e})\ d$. By (5.1), we have $c\ >_W\ d$, a contradiction with our assumption that $d\ >_W\ c$.

- Another agent, say agent 2 is the dictator in W_e. Let $a_1 \neq a_2$ be any two alternatives in $O \setminus \{b, e\}$. This is possible since $|O| > 3$. Let $>'$ be a preference profile of O such that $a_1 >_1' a_2$ but $a_2 >_2' a_1$. From $a_1 >_1' a_2$, $\{a_1, a_2\} \subseteq O_{-b}$, and that agent 1 is the dictator in W_b, we can conclude $a_1\ >_W'\ a_2$ as we have done in the previous case. Similarly, from $a_2 >_2' a_1$, $\{a_,a_2\} \subseteq O_{-e}$, and that agent 2 is the dictator in W_e, we can conclude $a_2\ >_W'\ a_1$, a contradiction.

Thus we have shown that W_e cannot have a dictator. ■

Note that it is essential for our proof that $m \geq 3$. Notice also that we only use the assumptions that W is IIA and non-dictatorial in our proof that W_a is not dictatorial for some $a \in O$. The assumption that W is unanimous is used only in showing that W_a is also unanimous.

Lemma 5.7 *If there is a social welfare function for $n+1$ individuals and m alternatives that is unanimous, IIA and non-dictatorial, there will also be a social welfare function for n individual and m alternatives that satisfies these three conditions as well, for all $n \geq 2, m \geq 3$.*

Proof: Let $N = \{1, ..., n, n+1\}$ be a set of agents, and O a set of m alternatives, and W a social welfare function for (N, O) that satisfies the three conditions in the lemma. For any $i \neq j \in N$, we define $W_{i,j}$ to be the following social welfare function for (N_{-i}, O): for any preference profile $>$ of (N, O), $W_{i,j}(>_{-i}) = W(>_{-i}, >_j)$, where $(>_{-i}, >_j)$, as we defined earlier, is the result of replacing $>_i$ in $>$ by $>_j$. Thus the social welfare function $W_{i,j}$ is defined through W by making agent i and agent j always agreeing with each other. Clearly, for any i, j, $W_{i,j}$ is unanimous and IIA because W satisfies these two conditions. We now show that we can find two distinct agents i and j such that $W_{i,j}$ is not dictatorial. Suppose otherwise, for every pair $i > j \in N$, $W_{i,j}$ is dictatorial. Now consider three distinct agents $i_1 < i_2 < i_3$ in N. This is possible because $|N| = n+1 \geq 3$. Suppose i is the dictator in W_{i_1,i_2}, j the dictator in W_{i_1,i_3}, and k the dictator in W_{i_2,i_3}. There are two cases:

- Case 1: $i = j = k$. Since W is not dictatorial, there is a profile $>$ of (N, O) and two alternatives a_1 and a_2 such that $>_W$ and $>_i$ disagree on a_1 and a_2, say $a_1 >_i a_2$ but $a_2 >_W a_1$. Now at least two players from $\{i_1, i_2, i_3\}$ must agree on a_1, a_2. Let these two players be j_1 and j_2, and without loss of generality, suppose $j_1 < j_2$. Now consider the profile $(>_{-j_1}, >_{j_2})$. Since W is IIA, and because $>_{j_1}$ and $>_{j_2}$ agree on a_1 and a_2, $>_W$ and $W(>_{-j_1}, >_{j_2})$ must agree on a_1 and a_2. So $a_2\ W(>_{-j_1}, >_{j_2})\ a_1$. But i is the dictator in W_{j_1,j_2}, $W_{j_1,j_2}(>_{-j_1})$ must agree with $>_i$. Since $W_{j_1,j_2}(>_{-j_1})$ is defined to be $W(>_{-j_1}, >_{j_2})$, thus $W(>_{-j_1}, >_{j_2})$ agrees with $>_i$, so $a_1\ W(>_{-j_1}, >_{j_2})\ a_2$, a contradiction.

- Case 2: $i \neq j$ or $i \neq k$ or $j \neq k$. First, by our definition of $W_{x,y}$, and our assumption that agents i, j, and k are dictators in W_{i_1,i_2}, W_{i_1,i_3}, and W_{i_2,i_3}, respectively, for any preference profile $>$ of (N, O), if $>_{i_1} = >_{i_2} = >_{i_3}$, then $>_W = >_i$, $>_W = >_j$, and $>_W = >_k$. Since two of $\{i, j, k\}$ must be distinct, this means that $\{i, j, k\} \subseteq \{i_1, i_2, i_3\}$. Since i must be in N_{-i_1}, so $i \neq i_1$, thus $i \in \{i_2, i_3\}$. Similarly, $j \in \{i_2, i_3\}$ and $k \in \{i_1, i_3\}$. This leads to eight possible combinations for i, j, and k. Each of

them will lead to a contradiction, using the following table:

(i,j,k)	$>_{i_1}$	$>_{i_2}$	$>_{i_3}$
(i_2,i_2,i_1)	$c>a>b$	$a>b>c$	$a>c>b$
(i_2,i_2,i_3)	case 1		
(i_2,i_3,i_1)	case 1		
(i_2,i_3,i_3)	$c>a>b$	$b>c>a$	$a>b>c$
(i_3,i_2,i_1)	$c>a>b$	$a>b>c$	$b>c>a$
(i_3,i_2,i_3)	$b>a>c$	$b>c>a$	$a>b>c$
(i_3,i_3,i_1)	$b>c>a$	$b>a>c$	$a>b>c$
(i_3,i_3,i_3)	case 1		

Each row in the above table either gives a preference profile that will lead to a contradiction or point to "case 1", meaning a contradiction can be derived similar to case 1. For instance, consider the row $(i,j,k)=(i_2,i_3,i_1)$, which says "case 1". This case can be reduced to "case 1" as follows. Since $i=i_2$ is the dictator in W_{i_1,i_2}, i_1 is the dictator in W_{i_2,i_1}. Similarly, i_1 is the dictator in W_{i_3,i_1} because $i_3=k$ is the dictator in W_{i_1,i_3}. Thus i_1 is the dictator in W_{i_2,i_1}, W_{i_3,i_1}, and W_{i_2,i_3}, and the same reasoning in case 1 will lead to a contradiction here.

Now consider the first row $(i,j,k)=(i_2,i_2,i_1)$, and the preference profile $>$ given in the row:

$$c >_{i_1} a >_{i_1} b, \quad a >_{i_2} b >_{i_2} c, \quad a >_{i_3} c >_{i_3} b.$$

Because $i_2=j$ is the dictator in W_{i_1,i_3}, $W(>_{-i_1},>_{i_3})=>_{i_2}$. But $>_{i_1}$ and $>_{i_3}$ agree on b and c, thus by IIA:

$$b >_W c \text{ iff } b\ W(>_{-i_1},>_{i_3})\ c \text{ iff } b >_{i_2} c.$$

So

$$b >_W c. \tag{5.2}$$

Similarly, $>_{i_1}$ and $>_{i_2}$ agree on a and b, and i_2 is the dictator in W_{i_1,i_2}, thus $a >_W b$ iff $a >_{i_2} b$. So

$$a >_W b. \tag{5.3}$$

Now $>_{i_2}$ and $>_{i_3}$ agree on a and c, and i_1 is the dictator in W_{i_2,i_3}, thus $a >_W c$ iff $a >_{i_1} c$. So $c >_W a$, which contradicts with (5.2) and (5.3). The other cases are similar.

This means that there must be some $i\neq j\in N$ such that $W_{i,j}$ is not dictatorial. ■

Again notice that it is essential for our proof that $|N| = n + 1 \geq 3$, and that the existence of a non-dictatorial $W_{i,j}$ depends only on the assumptions that W is IIA and non-dictatorial.

By these two lemmas, we see that Arrow's theorem holds iff it holds for the case when there are exactly two agents and three possible outcomes.[1]

5.3.2 The base case

We now turn to the proof of the base case, and as we mentioned earlier, we use computer programs to do that.

The base case says that when $|N| = 2$ and $|O| = 3$, there is no social welfare function on (N, O) that is unanimous, IIA, and non-dictatorial. A straightforward way of verifying this is to generate all possible social welfare functions in (N, O) and check all of them one by one for these three conditions. However, there are too many such functions for this to be feasible on current computers: there are $3! = 6$ number of linear orderings of O, resulting in $6 \times 6 = 36$ total number of preference profiles of (N, O), and 6^{36} possible social welfare functions.

Thus one should not attempt to explicitly generate all possible social welfare functions. What we did instead is to generate explicitly all social welfare functions that satisfy the conditions of unanimity and IIA, and then check if any of them is non-dictatorial.

We treat the problem of generating all social welfare functions that satisfy the conditions of unanimity and IIA as a constraint satisfaction problem (CSP). A CSP is a triple (V, D, C), where V is a set of variables, and D a set of domains, one for each variable in V, and C a set of constraints on V (see, e.g. [33]). An assignment of the CSP is a function that maps each variable in V to a value in its domain. A solution to the CSP is an assignment that satisfies all constraints in C.

Now consider the voting model $(\{1, 2\}, \{a, b, c\})$ in our base case. We define a CSP for it by introducing 36 variables $x_1, ..., x_{36}$, one for each preference profile of the voting model. The domain of these variables is the set of 6 linear orderings of $\{a, b, c\}$, and the constraints are the instantiations of the unanimity and IIA conditions on the voting model. As can be easily seen, there is a one-to-one correspondence between the social welfare functions of the voting model and the assignments of the CSP. Furthermore, a solution to the CSP corresponds to a social welfare function that satisfies the unanimity and IIA conditions, and vice versa.

To solve this CSP, we use a depth-first search that backtracks whenever the current partial assignment violates the constraints, and implemented it in SWI-Prolog. As we

[1]Technically speaking, we also need to consider the case when $|N| = 1$, but this is a trivial case.

mentioned earlier, when run on our AMD server machine, our Prolog program returned in less than one second two solutions, one corresponds to the social welfare function where agent 1 is the dictator, and the other agent 2 the dictator.

This verifies the base case of our inductive proof of Arrow's theorem, thus completes our proof. As mentioned in the introduction, we also verified the base case using a SAT solver. This requires a logical language to encode postulates in social choice theory, and will be described in a separate section below.

At last, it is worth noting that Suzumura [37] also provided, in his presidential address to the Japanese Economic Association, a specific backwards induction proof that reduces Arrow's theorem to two agents (but n alternatives) base case and then proves the base case with almost the same amount of efforts as the inductive case. In contrast, our further reduction to three alternatives case makes the computational verification possible and as we will show, our reduction to two agents case is more general and can be used to prove other impossibility theorems.

5.4 Muller-Satterthwaite Theorem

As mentioned before, the same strategy that we used for proving Arrow's theorem can be used to prove other impossibility theorems. In fact, we have modified the above proof for proving Sen's and Muller-Satterthwaite theorems. We prove in the following Muller-Satterthwaite theorem (cf. eg. [18]).

Arrow's theorem is about the social welfare function which maps a preference profile to a preference ordering. In comparison, Muller-Satterthwaite theorem concerns about *social choice function* which maps a preference profile to an outcome which is supposed to be the "winner" of the election (as represented by the preference profile).

Definition 5.8 *Given a voting model (N, O), a social choice function is a function $C : L^n \to O$, where L is the set of linear orders on O, and n the number of agents in N.*

Instead of the conditions of unanimity, IIA, and non-dictatorship in Arrow's theorem, Muller and Satterthwaite considered the following three corresponding conditions.

Definition 5.9 *A social choice function C is* weakly unanimous *if for every preference profile $>$, if there is a pair of alternatives a_1, a_2 such that $a_1 >_i a_2$ for every agent i, then $C(>) \neq a_2$.*

Thus according to this condition, an alternative that is dominated by another should never be selected.

Definition 5.10 *A social choice function C is* monotonic *if, for every preference profile $>$ such that $C(>) = a$, if $>'$ is another profile such that $a >'_i a'$ whenever $a >_i a'$ for every agent i and every alternative a', then $C(>') = a$ as well.*

In words, monotonicity means that if a choice function selects an outcome for a preference profile, then it will also select this outcome for any other preference profile that does not decrease the ranking of this outcome.

Definition 5.11 *An agent i is a dictator in a social choice function C if C always selects i's top choice: for every preference profile $>$, $C(>) = a$ iff for all $a' \in O$ that is different from a, $a >_i a'$. C is* non-dictatorial *if it has no dictator.*

Theorem 5.12 ***(Muller-Satterthwaite Theorem [23])*** *For any voting model (N, O) such that $|O| \geq 3$, any social choice function that is weakly unanimous and monotonic is also dictatorial.*

Like our proof of Arrow's theorem, we prove this theorem by induction. The inductive step is again by two lemmas similar to the ones for Arrow's theorem.

Lemma 5.13 *If there is a social choice function for n individuals and $m+1$ alternatives that is weakly unanimous, monotonic and non-dictatorial, then there is also a social choice function for n individuals and m alternatives that satisfies these three conditions, for all $n \geq 2, m \geq 3$.*

Proof: Let (N, O) be a voting model such that $|N| = n$ and $|O| = m + 1$, and C a social choice function that satisfies the three conditions in the lemma. Just like our proof of the corresponding Lemma 6.7, for any $a \in O$, we define C_a to be a social choice function that is the "restriction" of C on O_{-a}: for any preference profile $>$ of O_{-a}, $C_a(>) = C(>^{+a})$. Again it can be easily seen that for any $a \in O$, C_a is weakly unanimous and monotonic. Now we show that there is one such a such that C_a is non-dictatorial.

Suppose otherwise: for any a, C_a is dictatorial. We start by assuming C_b has a dictator i. Since C is non-dictatorial, we can find a profile $>\in O$ such that $C(>) = c \neq d$, where d is top ranked outcome according to $>_i$. Since there are $|m + 1| \geq 4$ outcomes, we can find another outcome e that is distinct from b, c, d. Now we consider C_e, there are two cases:

- C_e still has agent i as its dictator. We have $d = C_e((>)_{-e}) = C((>)^{+e}_{-e})$, but according to monotonicity, we have $C((>)^{+e}_{-e}) = C(>) = c$, which leads to a contradiction since $c \neq d$.

- C_e has a dictator $j \neq i$. For any preference profile $>' \in O$ such that f is ranked top according to $>'_i$, g is ranked top according to $>'_j$, $f \neq g$ and f, g are distinct from b, e, we consider the following two preference profiles $>''= ((>)^{+b}_{-b})^{+e}_{-e}$ and $>'''= ((>)^{+e}_{-e})^{+b}_{-b}$. Clearly, we have $C(>'') = g$ and $C(>''') = f$. However, according to monotonicity, we have $C(>'') = C(>''')$. This leads to a contradiction.

Therefore, C_e cannot have a dictator. So we have prove that there is always a outcome a so that C_a is non-dictatorial.

■

Lemma 5.14 *If there is a social choice function for $n+1$ individuals and m alternatives that is weakly unanimous, monotonic and non-dictatorial, then there is also a social choice function for n individuals and m alternatives that satisfies these three conditions, for all $n \geq 2, m \geq 3$.*

Proof: Let (N, O) be a voting model such that $|N| = n + 1$ and $|O| = m$, and C a social choice function that satisfies the three conditions in the lemma. Just like our proof of Lemma 6.8, for any pair of agents $i \neq j \in N$, we define $C_{i,j}$ to be the following social welfare function for (N_{-i}, O): for any preference profile $>$ of (N, O), $C_{i,j}(>_{-i}) = C(>_{-i}, >_j)$. Again it can be easily seen that for any pair of agents $i \neq j$, $C_{i,j}$ is weakly unanimous and monotonic.

We prove in the following that we can find two distinct agents i, j such that $C_{i,j}$ is non-dictatorial. Suppose not, then for every pair of agents i, j, there is an agent $d_{i,j}$ that is a dictator of $C_{i,j}$. We first show that $d_{i,j} = j$ for any i, j. Suppose otherwise, $d_{i,j} = k \neq j$. Since C is non-dictatorial, we can find a profile $>$ such that $a = C(>) \neq b$ where b is on top of $>_k$. We then still have $C(>_{-\{i,j\}}, (>_j)^{a+}_{-a}, (>_j)^{a+}_{-a}) = a$ according to monotonicity of C. But according to the dictatorship of $C_{i,j}$, we have $C(>_{-\{i,j\}}, (>_j)^{a+}_{-a}, (>_j)^{a+}_{-a}) = b$, a contradiction. Therefore, we have $d_{i,j} = j$ for any i, j.

Now consider a profile any $>$ on $(N + 1, O)$, any triple of agents i, j, k and any triple of alternatives (this is possible since $|N + 1| \geq 3$, $|O| \geq 3$) where

- $a >_i c >_i b >_i$ ... for $>_i$
- $c >_j b >_j a >_j$ ... for $>_j$
- $b >_k a >_k c >_k$ ... for $>_k$

Notice that $>_i, >_j, >_k$ only differ in $\{a, b, c\}$. There are the following cases:

1. $C(>) = a$, then we change $>_j$ to $>_k$ and denote the new profile $>'$. By monotonicity, we still have $C(>') = a$. This leads to the contradiction that $d_{j,k} \neq k$.

2. Other cases where $C(>) = b, c$ or other alternatives are similar to the case above.

Therefore, we conclude that there are two distinct agents i, j such that $C_{i,j}$ is non-dictatorial. ■

For the base case again notice that the case for $N = 1$ is trivial, thus we need only to consider the case when there are two agents and three alternatives. Again the number of all possible social choice functions is too large to enumerate explicitly, but both our methods for verifying the base case in Arrow's theorem can be adapted here. For the depth-first search method, our program similarly reported that there are exactly two social choice functions that are weakly unanimous and monotonic, and both of them are dictatorial.

One additional interesting thing to note is that it is also extremely fast to generate all the social choice functions that satisfy monotonicity only. There are 17 functions returned in total: 2 are dictatorships, 3 are constant and the remaining 12 are all functions whose ranges contain 2 elements. Since a generalization of Muller Satterthwaite theorem [26] says that the condition weak unanimity can be weakened by only requiring that the range contains at least 3 elements, these 12 functions are the only interesting ones to look at when one wants to completely generalize the monotonicity condition.

Notice that our proof outlined above parallels our earlier proof of Arrow's theorem but does not make use of Arrow's theorem. In contrast, the existing proofs such as those in [23, 18, 26] are more complicated and [23, 18] rely on Arrow's theorem.

5.5 Sen's Theorem

We show in the following that our proof can also be copied to prove the impossibility theorem by Sen [35].

Definition 5.15 *A* collective choice rule *is a functional relationship* $F : L^n \to R$ *that specifies one and only one social preference relation* r *for any preference profile.*

The set R of preference relations includes all the possible binary relations. Particularly, the members of R are not necessarily transitive or complete. However, Sen focused only on *social decision functions*, a subset of collective choice rules with certain restriction on R.

Definition 5.16 *A* social decision function *is a collective choice rule* $C : L^n \to R$ *such that for each* $r \in R$, r *should generate a choice function.*

A preference relation r should generate a "choice function" if according to r, there exists a best alternative in every subset of alternatives. In other words, there exists an alternative that is at least as preferred as any other alternative in that subset.

Sen then suggested three conditions which should be satisfied by any rational social decision function, namely unrestricted domain (condition U), unanimity (condition P, named after Pareto principle) and liberalism (condition L).

The first two conditions are mentioned explicitly or implicitly in Arrow's framework: unrestricted domain says that all the possible preference profiles should be included in the domain of a social decision function while unanimity is exactly the same one as in Arrow's theorem.

The third condition, liberalism, is somewhat debatable. The intuitive justification behind is that each individual has the freedom to determine at least one social choice. For example, I should feel free to have my own garden planted lily rather than rose.

Definition 5.17 Liberalism*: For each individual i, there is at least one pair of alternatives, say (a_1, a_2), such that this individual is decisive for (a_1, a_2)*[2].

Theorem 5.18 ***(Sen' Theorem [35])*** *There is no social decision function that can simultaneously satisfy U, P and L.*

Sen further weakened the condition L to be the following form L^*,

Definition 5.19 Liberalism* *There are at least two individuals such that for each of them there is at least one pair of alternatives over which he is decisive.*

In other words, condition L^* only guarantees the freedom for two individuals instead of everyone in the society, as required by condition L. The following theorem subsumes Theorem 5.18.

Theorem 5.20 ***(Sen' Theorem [35])*** *There is no social decision function that can simultaneously satisfy Conditions U, P, and L^*, for any voting model with $|N| \geq 2$ and $|O| \geq 3$.*

We prove in the following Theorem 5.20. The inductive step consists of the following two lemmas

Lemma 5.21 *If there is a social decision function for $m+1$ alternatives and n outcomes that satisfies U, P and L^*, then there is a social decision function for m outcomes and n individuals that satisfies these three conditions as well, for all $m \geq 4$.*

[2] i is decisive for (a_1, a_2) if i prefers a_1 to a_2 implies that a_1 is preferred to a_2 according to the social preference relation returned by the decision function.

Proof: Let (N, O) be a voting model such that $|N| = n$ and $|O| = m + 1$, and C a social decision function that satisfies the three conditions in the lemma. For any $a \in O$, we define C_a to be a function that is the "restriction" of C on O_{-a}: for any preference profile $>$ of O_{-a}, $C_a(>) = C(>^{+a})_{-a}$.

- C_a is still a social decision function. Since C is a social decision function, so the range of C is the set of preferences that can generate a choice function. That is, for any subset of outcomes, there is a best outcome. This outcome will still be the best after we restrict on C_a since a is less preferred than any other outcome by unanimity.

- The property of U and P of C_a follows directly from that of C.

- Since C satisfies L^*, we can always find two individuals and their decisive pairs (a_1, a_2) and (a_3, a_4) respectively. Since $|m + 1| \geq 5$, we can find an element a_5 that is not in $\{a_1, a_2, a_3, a_4\}$. Now we can see that C_{a_5} still satisfies L^* because the two decisive individuals are still decisive for their pairs of alternatives (a_1, a_2) and (a_3, a_4).

■

Lemma 5.22 *If there is a social decision function for m alternatives and $n+1$ outcomes that satisfies U, P and L^*, then there is a social decision function for m outcomes and n individuals that satisfies these three conditions as well, for all $n \geq 2$.*

Proof: By the property L^* of C, we have two individuals j, k that are decisive for their own pair of outcomes. We can also find another distinct agent i, since there are at least $2+1$ three individuals for C. We now define $C_{i,j}$ to be the following social welfare function for (N_{-i}, O): for any preference profile $>$ of (N, O), $C_{i,j}(>_{-i}) = C(>_{-i}, >_j)$. Then $C_{i,j}$ is still a social decision function and all the three properties follows directly from that of C. ■

Notice that we have $m \geq 4$ in lemma 5.21, so the base case for Sen's theorem is $|N| = 2$ and $|O| = 3, 4$. We can still check it by our depth-first search algorithm, which we do not want to repeat here.

5.6 Discovering new theorems

We have been advocating a methodology of theorem discovering using computers [15, 39]. The basic idea is to look for conjectures that are true in small domains using computers.

Once we find such a conjecture, we then hope it to be true in general. In the following, we present a new theorem discovered this way.

5.6.1 An observation in small domain

Recall that in our CSP formulation of the base case of Arrow's theorem, constraints are the instantiations of both IIA and Unanimity conditions. Using the same algorithm, we can generate all the functions that satisfies IIA by restricting the constraints to be the instantiations of IIA only.

To our surprise (not so surprised if one is familiar with Wilson's theorem [45], which will be introduced later in this section), among the total 6^{36} social welfare functions, they are only 94 of them satisfying IIA. This seems to suggest that the impossibility in Arrow's theorem is actually not caused by the conflict between unanimity and IIA but mostly by IIA, which is too strong for a social welfare function to satisfy.

Among these 94 functions, 2 of them are dictatorial, 2 of them are inversely dictatorial which means the social order of the function is always opposite to someone's individual order, and each of the remaining 90 functions has at most two values in the range. Of course among the 90 functions, there are 6 constant functions, each of which has exactly one value. Moreover, for any of the remaining 84 functions which have two different values, the distance between these values is at most one pair of outcomes. For example, if one value is $a_1 >_W a_2 >_W a_3$, then the other value can only be $a_2 >_W a_1 >_W a_3$ or $a_1 >_W a_3 >_W a_2$.

Definition 5.23 *An agent i is a* inverse dictator *in W if for all alternatives a_1 and a_2, $a_1 >_W a_2$ iff $a_2 >_i a_1$. If there is a inverse dictator in W, then it is said to be* inversely dictatorial.

Definition 5.24 *The* (Kendall tau) distance *of two orderings on O is the number of pairs of outcomes where two orderings disagree.*

When IIA holds for a function W, we can define from it a social welfare function W_Y: $L_Y^n \to L_Y$, the restriction of W on an arbitrary non-empty subset Y of O, where L_Y is the restriction of L on Y and for any profile $>' \in L_Y^n$, $W_Y(>') = W(>)_Y$, for any $> \in L^n$ such that $>_Y = >'$.

We then generalize the above observation in small domain into the following theorem.

Theorem 5.25 *If a social welfare function W on (N, O) satisfies IIA, then for every subset Y of O such that $|Y| = 3$,*

1. *W_Y is dictatorial, or*

2. *W_Y is inversely dictatorial, or*

3. *The range of W_Y has at most 2 elements, whose the distance is at most 1.*

Notice that 1-3 are pairwise disjoint. Fortunately, by observation we have already proved the base case for theorem 5.25.

Lemma 5.26 *If a social welfare function W on (N, O) where $|N| = 2, |O| = 3$ satisfies IIA,*

1. *W is dictatorial, or*

2. *W is inversely dictatorial, or*

3. *The range of W has at most 2 elements, whose distance is at most 1.*

We show in the following the inductive step hold for this theorem too.

5.6.2 The inductive step

We first prove the following lemma that translates dictatorship to unanimity and translates inverse dictatorship to inverse unanimity under IIA.

Definition 5.27 *A social welfare function W is* inversely unanimous *(inversely Pareto efficient) if for all alternatives a_1 and a_2, we have that if $a_1 >_i a_2$ for all agent i, then $a_2 >_W a_1$*

Lemma 5.28 *If a social welfare function W on (N, O) where $|O| \geq 3$ satisfies IIA, then*

1. *W is dictatorial iff W is unanimous;*

2. *W is inversely dictatorial iff W is inversely unanimous.*

Proof: Assuming IIA,

1. if W is unanimous, by Arrow's theorem, it is dictatorial; if W is dictatorial, by the definition of unanimity, it is unanimous.

2. Now if W is inversely dictatorial, but W is not inversely unanimous, we can construct a new function W' such that $a >_W b$ iff $b >_{W'} a$ for any (a, b) and any preference profile $>$. We can see that W' satisfies IIA and dictatorial, but not unanimity. This contradicts to what we have proved above; similarly, if W is inversely unanimous but not inversely dictatorial, we can construct the same W' that satisfies IIA and unanimity but not dictatorial, violating Arrow's theorem.

■

By the following lemma, together with Lemma 5.28, we can extend Lemma 5.26 to voting models with any number of agents.

Lemma 5.29 *If there is a social welfare function for $n+1$ individuals and 3 outcomes that is IIA, but not unanimous or inversely unanimous and its range has two elements whose distance is at least 2, then there is a social welfare function for n individuals and 3 outcomes that is IIA, but not unanimous or inversely unanimous and its range has two elements whose distance is at least 2 as well.*

Proof: Let $N = \{1, ..., n, n+1\}$ be a set of agents, and $O = \{a, b, c\}$ a set of 3 alternatives, and W a social welfare function for (N, O) that satisfies the four conditions in the lemma. The same as before, for any $i \neq j \in N$, we define $W_{i,j}$ to be the following social welfare function for (N_{-i}, O): for any preference profile $>$ of (N, O), $W_{i,j}(>_{-i}) = W(>_{-i}, >_j)$, where $(>_{-i}, >_j)$ is the result of replacing $>_i$ in $>$ by $>_j$. Clearly, for any i, j, $W_{i,j}$ is IIA, not unanimous or inversely unanimous because W satisfies these three conditions. We now show that we can find two distinct agent i, j such that the range of $W_{i,j}$ has two elements whose distance is at least 2.

Since there exist two preference profiles $>$ and $>'$ such that $W(>)$ differs from $W(>')$ in at least two pair of outcomes, say (a, b) and (a, c). Since W is IIA, its restrictions $W_{\{a,b\}}$ and $W_{\{a,c\}}$ are well defined. Now we consider $W_{\{a,b\}}(a > b, \ldots, a > b)$, $W_{\{a,b\}}(b > a, \ldots, b > a)$, $W_{\{a,c\}}(a > c, \ldots, a > c)$ and $W_{\{a,c\}}(c > a, \ldots, c > a)$. There are four cases as follows:

1. $W_{\{a,b\}}(a > b, \ldots, a > b) \neq W_{\{a,b\}}(b > a, \ldots, b > a)$ and $W_{\{a,c\}}(a > c, \ldots, a > c) \neq W_{\{a,c\}}(c > a, \ldots, c > a)$. Without loss of generality, we suppose $W(>)$ agrees with $W_{\{a,b\}}(a > b, \ldots, a > b)$ in (a, b) and agrees with $W_{\{a,c\}}(a > c, \ldots, a > c)$ in (a, c), therefore $W(>')$ agrees with $W_{\{a,b\}}(b > a, \ldots, b > a)$ in (a, b) and agrees with $W_{\{a,c\}}(c > a, \ldots, c > a)$ in (a, c). Now we consider a profile $>''= (a > b > c, \ldots, a > b > c)$, clearly $W(>)$ agrees with $W(>'')$ in $(a, b), (a, c)$; similarly, for $>'''= (c > b > a, \ldots, c > b > a)$, $W(>')$ agrees with $W(>''')$ in $(a, b), (a, c)$. So $W(>'')$ and $W(>''')$ differ in $(a, b), (a, c)$. For two profiles $>''_{-i}$, $>'''_{-i}$, their values of $W_{i,j}$ differ in $(a, b), (a, c)$ for any j.

2. $W_{\{a,b\}}(a > b, \ldots, a > b) = W_{\{a,b\}}(b > a, \ldots, b > a)$ and $W_{\{a,c\}}(a > c, \ldots, a > c) \neq W_{\{a,c\}}(c > a, \ldots, c > a)$. Without loss of generality, we suppose $W(>)$ agrees with $W_{\{a,b\}}(a > b, \ldots, a > b)$ in (a, b) and agrees with $W_{\{a,c\}}(a > c, \ldots, a > c)$ in (a, c), therefore $W(>')$ agrees with $W(c > a, \ldots, c > a)$ in (a, c). Now we consider a profile $>''= (a > b > c, \ldots, a > b > c)$, clearly $W(>)$ agrees with $W(>'')$ in $(a, b), (a, c)$;

we can also construct another profile $>'''$ such that $>'''$ has $c > a$ for each agent and $>'''$ with $>'$ on (a, b) for each agent. So $W(>'')$ and $W(>''')$ differ in $(a, b), (a, c)$. Now we look at the relation of (a, b) in $>'''$, since there are at least 3 agents, we can always find two agents, say i, j that agree on (a, b). For profiles $>''_{-i}$, $>'''_{-i}$, their values of $W_{i,j}$ differ in $(a, b), (a, c)$.

3. $W_{\{a,b\}}(a > b, \ldots, a > b) \neq W_{\{a,b\}}(b > a, \ldots, b > a)$ and $W_{\{a,c\}}(a > c, \ldots, a > c) = W_{\{a,c\}}(c > a, \ldots, c > a)$. This case is similar to case 2 above.

4. $W_{\{a,b\}}(a > b, \ldots, a > b) = W_{\{a,b\}}(b > a, \ldots, b > a)$ and $W_{\{a,c\}}(a > c, \ldots, a > c) = W_{\{a,c\}}(c > a, \ldots, c > a)$. In this case, we first show that there exist two profiles $>^1$ and $>^2$ such that $W(>^1)$ and $W(>^2)$ differ in (b, c). We construct $>^1$ in such a way that $>^1$ agrees with either $>$ or $>'$ in (a, b) for each player so that $b >^1_W a$ and $>^1$ agrees with either $>$ or $>'$ in (a, c) for each player so that $a >^1_W c$. Therefore, we have $b >^1_W a >^1_W c$. Similarly, we can construct $>^2$ so that $c >^2_W a >^2_W b$. In this way, $W(>^1)$ and $W(>^2)$ differ in $(a, b), (b, c), (a, c)$. Now we consider further $W_{\{b,c\}}(c > b, \ldots, c > b)$ and $W_{\{b,c\}}(b > c, \ldots, b > c)$. There are two cases:

   - $W_{\{b,c\}}(c > b, \ldots, c > b) \neq W_{\{b,c\}}(b > c, \ldots, b > c)$, then this case will still be case 2 by considering $(a, b), (b, c)$ instead.
   - $W_{\{b,c\}}(c > b, \ldots, c > b) = W_{\{b,c\}}(b > c, \ldots, b > c)$. We prove in the following that this case is impossible. Suppose $W(a > b > c, \ldots, a > b > c) = o_1 > o_2 > o_3$ where (o_1, o_2, o_3) is a permutation of (a, b, c). Now we construct a new profile $>^*$ where o_2 is always on top of each agent's preference and $>^*$ agrees with either $>^1$ or $>^2$ in (o_1, o_3) for each player so that $o_3 W(>^*) o_1$. But since o_2 is always on top such that we have $o_1 W(>^*) o_2$ and $o_2 W(>^*) o_3$ because $W_{\{a,b\}}(a > b, \ldots, a > b) = W_{\{a,b\}}(b > a, \ldots, b > a)$, $W_{\{a,c\}}(a > c, \ldots, a > c) = W_{\{a,c\}}(c > a, \ldots, c > a)$ and $W_{\{b,c\}}(c > b, \ldots, c > b) = W_{\{b,c\}}(b > c, \ldots, b > c)$. By transitivity, we have $o_1 W(>^*) o_3$, which is a contradiction.

■

Since W is IIA, its restriction on any non-empty subset Y of $|O|$ is still IIA. Therefore our Theorem 5.25 follows from Lemma 5.26, 5.28 and 5.29.

5.6.3 The implication of the new theorem

We show in the following how to use Theorem 5.25 to prove two existing theorems.

A brief proof of Arrow's Theorem

One immediate implication of theorem 5.25 is Arrow's theorem. Given that a social welfare function W on (O, N) is IIA, by applying theorem 5.25, we know W_Y is either of the three cases when $Y \subseteq |O|, |Y| = 3$. If W is further unanimous assumed by Arrow's theorem, so is W_Y. Clearly W_Y can only be case 1 for any Y. In other words, The restriction of W on any three-element subset is dictatorial. Now we arbitrarily choose such a $Y = \{a_1, a_2, a_3\}$, suppose the dictator in W_Y is i. Then i will still be a dictator in W_{Y^1}, where $Y^1 = \{a_1, a_2, a_4\}$ for any $a_4 \in O \setminus Y$, since there can only be one agent that is decisive for the pair (a_1, a_2). Similarly, i is still the dictator for W_{Y^2}, where $Y^2 = \{a_1, a_3, a_4\}$ or $\{a_2, a_3, a_4\}$, $\{a_1, a_4, a_5\}$ or $\{a_4, a_5, a_6\}$ for any distinct a_5, a_6. Therefore, we prove that all the restrictions of W on three-elements subset have a common dictator i. Since i is decisive for any pair in O^2, i is a dictator in W.

A brief proof of Wilson's Theorem

There have been fruitful researches on relaxing the unanimity condition in Arrow's framework. In other words, these researches also aim at finding the implication of IIA condition. One of the most famous one is Wilson's Theorem [45]. It states that even with a condition called nonimposition that is much weaker than unanimity, IIA can already imply dictatorship or inverse dictatorship.

Definition 5.30 *A social welfare function W is* nonimposition *if for all distinct alternatives a_1 and a_2, there exists a preference profile $>$ such that $a_1 >_W a_2$*

Theorem 5.31 ***(Wilson's theorem [45])*** *For any voting model (N, O), if $|O| \geq 3$, then any social welfare function that satisfies nonimposition and IIA is either dictatorial or inversely dictatorial.*

Theorem 5.25 also implies Wilson' theorem as well. Given that a social welfare function W on (O, N) is IIA, by applying Theorem 5.25, we know W_Y is either of the three cases when $Y \subseteq |O|, |Y| = 3$. If W is further nonimposition assumed by Wilson's theorem, so is W_Y. Therefore W_Y can only be dictatorial or inversely dictatorial since case 3 in Theorem 5.25 obviously violates nonimposition. Dictatorship or inverse dictatorship then follows from similar arguments to those of Arrow's theorem above.

5.7 A logical language for social choice theory

As we mentioned earlier, we are not only interested in alternative proofs of existing theorems or even the manual discovery of new theorem like what we did in section 5.25. Our long term goal is to automate the discovery of theorems in social choice theory, game theory, and others [15, 39]. One insight of our new proofs is that these known impossibility results are all rooted in some small base cases. Thus by experimenting with other

conditions in small cases, we could discover some new results. To fully automate the enumeration and verification process of these conditions, we propose a logical language for social choice theory.

This language is a variant of the situation calculus [20], one of the best known languages in AI. For representing Arrow's theorem, we use two predicates: $p(x, a, b, s)$ (in the situation s, agent x prefers a over b) and $w(a, b, s)$ (in the situation s, a is preferred over b according to the social welfare function). The intuition is that in each situation, there is a preference ordering for each player (represented by predicate p), and a social welfare function for the society (predicate w). The requirement that the preferences be linear corresponds to the following axioms:

$$p(x, a, b, s) \vee p(x, b, a, s) \vee a = b, \tag{5.4}$$

$$\neg p(x, a, a, s) \wedge \neg w(a, a, s), \tag{5.5}$$

$$p(x, a, b, s) \wedge p(x, b, c, s) \supset p(x, a, c, s), \tag{5.6}$$

$$w(a, b, s) \vee w(b, a, s) \vee a = b, \tag{5.7}$$

$$w(a, b, s) \wedge w(b, c, s) \supset w(a, c, s), \tag{5.8}$$

where "$\supset$" is the logical implication operator. We have used the convention that all free variables in a formula are implicitly universally quantified from outside unless stated otherwise. So the full sentence for the first axiom above is:

$$\forall x, a, b, s.p(x, a, b, s) \vee p(x, b, a, s) \vee a = b.$$

We also need an axiom which says that the predicate w indeed represents a function that aggregates individual preferences:

$$\begin{aligned}[\forall x, a, b.p(x, a, b, s_1) \equiv p(x, a, b, s_2)] \supset \\ [\forall a, b.w(a, b, s_1) \equiv w(a, b, s_2)].\end{aligned} \tag{5.9}$$

The unanimity condition corresponds to the following axiom:

$$\forall a, b, s.[\forall x\ p(x, a, b, s)] \supset w(a, b, s), \tag{5.10}$$

the non-dictatorship condition the following axiom:

$$\neg\exists x \forall s, a, b.p(x, a, b, s) \equiv w(a, b, s), \tag{5.11}$$

and the IIA condition the following one:

$$\begin{aligned}\forall a, b, s_1, s_2.[\forall x.p(x, a, b, s_1) \equiv p(x, a, b, s_2)] \supset \\ [w(a, b, s_1) \equiv w(a, b, s_2)],\end{aligned} \tag{5.12}$$

Furthermore, we need to say that each preference profile is represented by some situation (the assumption of unrestricted domain). One way to do it is to introduce an action $swap(x, a, b)$ which when performed will swap the positions of a and b in agent x's preference ordering.

$$p(x, a, b, do(swap(x, a, b), s)) \equiv p(x, b, a, s),$$

where in general, $do(A, s)$ denotes the situation resulting from doing action A in s. We also need other axioms to say that in the new situation, agent x prefers a' over b iff she prefers a' over a before, she prefers a' over a iff she prefers a' over b before, that this action has no effects on the orderings of other pairs of alternatives, and no effect on the preference orderings of other agents. All these can be conveniently specified using Reiter's successor state axioms:

$$\begin{aligned}
&p(x, a, b, do(swap(y, a_1, b_1), s)) \equiv \\
&\quad p(x, a, b, s) \wedge [x \neq y \vee (a \neq a_1 \wedge a \neq b_1 \wedge \\
&\qquad\qquad b \neq a_1 \wedge b \neq b_1)] \vee \\
&\quad x = y \wedge a = a_1 \wedge b = b_1 \wedge p(x, b, a, s) \vee \\
&\quad x = y \wedge a = a_1 \wedge b \neq b_1 \wedge b \neq a \wedge p(x, b_1, b, s) \vee \\
&\quad x = y \wedge b = b_1 \wedge a \neq a_1 \wedge b \neq a \wedge p(x, a, a_1, s).
\end{aligned}$$

This way, given an initial situation S_0 that encodes any preference profile, we can get any other preference profile by performing a sequence of swapping actions in S_0.

However, if we are given a specific voting model, we can name each preference profile explicitly by a situation constant. For instance, for the voting model $(\{1, 2\}, \{a, b, c\})$ corresponding to the base case in our proof of Arrow's theorem, there are 36 different profiles, so we introduce 36 situation constants $S_1, ..., S_{36}$, and add axioms like the following ones to define them:

$$\begin{aligned}
&p(1, a, b, S_1) \wedge p(1, a, c, S_1) \wedge p(1, b, c, S_1), \\
&p(2, a, b, S_1) \wedge p(2, a, c, S_1) \wedge p(2, b, c, S_1).
\end{aligned}$$

In fact, this is what we did for using a SAT solver to verify the base case in our inductive proof of Arrow's theorem. We instantiated the axioms (5.10) – (5.12) as well as the general axioms about p and w on $(\{1, 2\}, \{a, b, c\})$, and converted them as well as the axioms like the above ones for the 36 situation constants to clauses. The resulting set of clauses has 35973 variables and 106354 clauses, and we were surprised that the SAT solver Chaff2 [22] returned in less than 1 second when run on our AMD server machine and confirmed that the set of clauses has no models.

5.8 Summary and discussion

We have given a new proof of Arrow's theorem. The basic idea is extremely simple: use induction to reduce it to the base case which is then verified using computers. One remarkable thing about it is that it appears to be a very general approach for proving other theorems in the area. In fact, we have adapted it almost straightforwardly to proving two other well-known theorems of the same nature, one by Muller and Satterthwaite and the other by Sen.

One insight we have obtained from the proof is that theorems that are verified to be true in the small base cases are extremely likely to be true in general. That is how we have discovered and proved our new theorem in section 5.6.

If all these axioms in social choice theory can be checked in base case as fast as those in Arrow's theorem, an interesting future work is to verify all the possible combinations of these candidate axioms using a computer program and then try to extend the survivors to general case using the "two-lemma trick" introduced in the inductive step. To facilitate the above systematical generation and verification process, it becomes nature to describe these axioms in a logical language that is easy in syntax and semantics as well as allows for fast implementation. That is why we have proposed a new logical formalism for social choice theory despite the rich literature. In fact, we did discover this way two theorems, as described in [17]. It is pity that both theorems can be implied immediately by existing theorems. We are still exploring this territory to see if we could come up with something new.

Chapter 6

Proving theorems in implementation theory

> *Implementation theory, which, given a social goal, characterizes when we can design a mechanism whose predicted outcomes (i.e., the set of equilibrium outcomes) coincide with the desirable outcomes, according to that goal.*
>
> —Eric Maskin

The Gibbard-Satterthwaite Theorem is a landmark result in both social choice theory and implementation theory, as it bridges normative and strategic analysis of voting problems. The theorem states that any social choice function that is strategy-proof and onto is also dictatorial. In this chapter, we provide a computer-aided inductive proof for the theorem. We first show that this result holds if and only if it holds for the base case where there are exactly 2 agents and 3 alternatives. We then verify the base case using a computer program. Following the same strategy, we prove Maskin's theorem on Nash implementation as well. These proofs successfully generalize this general methodology from social choice theory to implementation theory.

6.1 Gibbard-Satterthwaite Theorem

As introduced before, a voting model is a tuple (N, O), where N is a finite set of individuals (agents) and O a finite set of outcomes (alternatives). An agent's preference ordering is a linear ordering of O, and a preference profile $>$ of (N, O) is a tuple $(>_1, ..., >_n)$, where $>_i$ is agent i's preference ordering, and n the size of N. In the following, when N is clear from the context, we also call $>$ a preference profile of O. Similarly, when O is clear from the context, we also call it a preference profile of N.

Definition 6.1 *Given a voting model (N, O), a* social choice function *is a function $W : L^n \to O$, where L is the set of linear ordering of O, and n the size of N.*

The first assumption about a social choice function is that of *onto*.

Definition 6.2 *A social choice function C is an* onto *if for each $a \in O$, there exists a $> \in L^n$ such that $C(>) = a$.*

In other words, this assumption guarantees that every alternative has a chance to win. Before we get to the second assumption called *strategy-proof*, we need another concept called *manipulation*.

Definition 6.3 *A social choice function C is* manipulable *at profile $>$ by individual i via $>'$ if $C(>_{-i}, >'_i) >_i C(>)$, where $(>_{-i}, >'_i)$ is the profile resulting from replacing $>_i$ with $>'_i$ in $>$.*

Manipulability is a typical feature of strategic voting in contrast with normative voting where agents always report the truth. If a social choice function C is manipulable, there is always a state where some agent is better off by lying about his true preference, thus the resulting outcome may not truthfully represent a social choice. Manipulability is not a desirable property and should be precluded from the picture of a rational social choice function.

Definition 6.4 *A social choice function is* strategy-proof *if it is not manipulable by any individual at any profile.*

Strategy-proof is sometimes said to be *dominant strategy incentive compatible*, because if a choice function is strategy-proof, then every agent always has the incentive to report his true preference in the sense that no matter what the other agents report, he is better off to report his true preference.

Definition 6.5 *An agent i is a* dictator *in a social choice function C if C always selects i's top choice: for every preference profile $>$, $C(>) = a$ iff for all $a' \in O$ that is different from a, $a >_i a'$. C is* non-dictatorial *if it has no dictator.*

Theorem 6.6 ***(Gibbard-Satterthwaite Theorem)[10, 34]*** *For any voting model (N, O) such that $|O| \geq 3$, any social choice function that is strategy-proof and onto is also dictatorial.*

6.2 An inductive proof of Gibbard-Satterthwaite Theorem

For ease of presentation, we assume the following notations.

- For any set S, we use S_{-a} to denote $S \setminus \{a\}$, i.e. the result of deleting a in S.
- We extend the above notation to tuples as well: if $t = (t_1, ..., t_n)$, then we use t_{-i} denotes the tuple $(t_1, ..., t_{i-1}, t_{i+1}, ..., t_n)$. Furthermore, we use (t_{-i}, s) to denote the result of replacing ith item in t by s: $(t_{-i}, s) = (t_1, ..., t_{i-1}, s, t_{i+1}, ..., t_n)$. We use $t_{-\{i,j\}}$ to denote $(t_{-i})_{-j}$.

- If $>$ is a linear ordering of O, and $a \in O$, then we let $>_{-a}$ be the restriction of $>$ on O_{-a}: for any $a', a'' \in O_{-a}$, $a' >_{-a} a''$ iff $a' > a''$. On the other hand, if $>$ is a linear ordering of O_{-a} for some $a \in O$, then we let $>^{+a}$ to be the extension of $>$ to O such that for any $a' \in O_{-a}$, $a' >^{+a} a$. Similarly, we let $>^{a+}$ to be the extension of $>$ to O such that for any $a' \in O_{-a}$, $a >^{a+} a'$. Thus if $>$ is a linear ordering of O, and $a \in O$, then $>^{+a}_{-a}$ is $(>_{-a})^{+a}$, i.e. the result of moving a to the bottom of the ordering. These notations extend to tuples of orderings. Thus if $>$ is a preference profile of (N, O_{-a}), then

$$>^{+a} = (>_1, ..., >_n)^{+a} = (>_1^{+a}, ..., >_n^{+a}),$$

which will be a preference profile of (N, O). Similarly for $>^{a+}$

6.2.1 The inductive step

The inductive step consists of the following two lemmas.

Lemma 6.7 *If there is a social choice function for n individuals and $m+1$ alternatives that is onto, strategy-proof and non-dictatorial, then there is a social choice function for n individuals and m alternatives that satisfies these three conditions as well, for all $n \geq 2, m \geq 3$.*

Proof: Let (N, O) be a voting model such that $|N| = n$ and $|O| = m+1$, and C a social choice function that satisfies the three conditions in the lemma. For any $a \in O$, we define C_a to be a social choice function that is the "restriction" of C on O_{-a}: for any preference profile $>$ of O_{-a}, $C_a(>) = C(>_{+a})$.

- We first show that C_a is well defined. That is, $C_a(>) \neq a$ for any $>$ on O_{-a}. Suppose $C_a(>) = C(>_{+a}) = a$ for some $>$. We have $C(>_{-i}, >'_i) = a$ for any agent i and any ordering $>'_i$, otherwise i can manipulate at $>$ via $>'_i$. Similarly, we have $C(>_{-\{i,j\}}, >'_i, >'_j) = a$, otherwise j can manipulate at $(>_{-i}, >'_i)$ via $>'_j$. We continue the above argument until we get $C(>') = a$ for any $>'$, which leads to the contradiction that C is an onto.

- We now show that C_a is strategy-proof. Suppose otherwise: some agent i can manipulate at $>$ via $>'_i$ in C_a. This is equivalent of saying i can manipulate at $>^{+a}$ via $>'^{+a}$, which leads to the contradiction that C is not strategy-proof.

- Instead of showing C_a is an onto, we prove C and C_a are unanimous, which implies onto by letting each alternative on top of everyone's preference. Since C is an onto, we can find some $>$ for each b such that $C(>) = b$. We have $C(>_{-i}, (>'_i)^{b+}_{-b}) = b$.

In words, by moving b to the top of any $>_i'$, the resulting choice will still be b. Otherwise, i can manipulate at $(>_{-i}, (>_i')^{b+}_{-b})$ via $>_i$. We continue the argument until we get $C((>')^{b+}_{-b}) = b$. Therefore C is unanimous and so is C_a by definition.

- We finally show that an alternative a can be chosen properly so that C_a is non-dictatorial. We proceed in two steps:

  1. We now prove that for any two distinct alternative c, d, C_c and C_d have the same dictator, if they have one. Suppose otherwise, that is, C_c has dictator i, C_d has dictator j and $i \neq j$. We consider the following profile $>$, where we have

     - $a >_s b >_s ... >_s d >_s c, for \ \ all \ \ s \neq j$
     - $b >_j a >_j ... >_j d >_j c, for >_j$

     Obviously, $C(>) = a$ because C_c has dictator i. Now we change $>_i$ into $a >_i' b >_i' ... >_i' c >_i' d$, then we still have $C(>_{-i}, >_i') = a$. Otherwise i can manipulate at $(>_{-i}, >_i')$ via $>_i$. We continue the same argument until we get $C(>'_{-j}, >_j) = a$. However, there is a manipulate for j at $(>'_{-j}, >_j)$ via $>_j'$: $\ b >_j' a >_j' ... >_j' c >_j' d$. So contradiction occurs.

  2. Now suppose C_a has a dictator for each $a \in O$, by (1) above, $\forall a \in O \ \ C_a$ has the same dictator i. Since C is non-dictatorial, there exists some $>$ so that $C(>) = a_1$ is different from the topmost alternative a_2 of $>_i$. Now consider $>' = (>^{a_1+}_{-a_1})^{+a_3}_{-a_3}$, where a_3 is distinct from a_1 and a_3. We can continue to change $>_s$ to $>_s'$ for all $s \neq i$ without changing the social value until we get $C(>'_{-i}, >_i) = a_1$. Now we have a manipulation for agent i at $(>'_{-i}, >_i)$ via $(>_i)^{+a_3}_{-a_3}$, which contradicts to the strategy-proof property of C.

  Therefore, there exists some a such that C_a is non-dictatorial.

■

Lemma 6.8 *If there is a social choice function for $n+1$ individuals and m alternatives that is onto, strategy-proof and non-dictatorial, then there is a social choice function for n individuals and m alternatives that satisfies these three conditions as well, for all $n \geq 2, m \geq 3$.*

Proof: Let (N, O) be a voting model such that $|N| = n+1$ and $|O| = m$, and C a social choice function that satisfies the three conditions in the lemma. For any pair of agents $i \neq j \in N$, we define $C_{i,j}$ to be the following social choice function for (N_{-i}, O): for any

preference profile $>$ of (N, O), $C_{i,j}(>_{-i}) = C(>_{-i}, >_j)$. That is, let agent i always agree with agent j.

- Again the onto property of $C_{i,j}$ follows from the unanimity of C as shown in the proof of lemma 6.7.

- For the strategy-proofness of $C_{i,j}$, suppose otherwise that $C_{i,j}(>_{-k}, >'_k) >_k C_{i,j}(>)$ for some $>$, $>'_k$ and k. There are two cases:

  1. If $k \neq j$, then C is not strategy-proof since $C_{i,j}(>_{-k}, >'_k) = C(>_{-k}, >'_k, >_j) >_k C(>, >_j) = C_{i,j}(>)$[1]. A contradiction.
  2. If $k = j$, by definition of $C_{i,j}$, we have $a_1 = C(>_{-j}, >'_j, >'_j) >_j C(>_{-j}, >_j, >_j) = a_2$. Now consider $C(>_{-j}, >_j, >'_j) = a_3$.
     - If $a_1 >_j a_3$, then j can manipulate at $(>_{-j}, >_j, >'_j)$ via $>'_j$
     - If $a_3 \geq_j a_1$, then i can manipulate at $(>_{-j}, >_j, >_j)$ via $>'_j$ (since i, j have the same preference in this profile).

     Either way, there is a contradiction to the strategy-proofness of C.

  So $C_{i,j}$ is strategy-proof.

- We prove in the following that we can find two distinct agents i, j such that $C_{i,j}$ is non-dictatorial. Suppose not, then for every pair of agents i, j, there is an agent $d_{i,j}$ that is a dictator of $C_{i,j}$. We first show that $d_{i,j} = j$ for any i, j. Suppose otherwise, $d_{i,j} = k \neq j$. Since C is non-dictatorial, we can find a profile $>$ such that $a = C(>) \neq b$ where b is on top of $>_k$. We then still have $C(>_{-i}, (>_i)^{a+}_{-a}) = a$, otherwise there would be a manipulation for i at $>' = (>_{-i}, (>_i)^{a+}_{-a})$ via $>_i$. Similarly, we have $C(>'_{-j}, >'_i) = a$. Now we have a contradiction, since according to $d_{i,j} = k$, we would have $C(>'_{-j}, >'_i) = b$.

  Now consider a profile any $>$ on $(N+1, O)$, any triple of agents i, j, k and any triple of alternatives (this is possible since $|N+1| \geq 3$, $|O| \geq 3$) where

  - $a >_i c >_i b >_i$ ... for $>_i$
  - $c >_j b >_j a >_j$ ... for $>_j$
  - $b >_k a >_k c >_k$ ... for $>_k$

  Notice that $>_i, >_j, >_k$ only differ in $\{a, b, c\}$. There are the following cases:

  - $C(>) = a$, then we change $>_j$ to $>_k$ and denote the new profile $>'$. Then,

[1] According to the definition of $C_{i,j}$, $(>_{-k}, >'_k, >_j)$ is a preference profile on $|N| = n + 1$ by adding $>_j$ as agent i's preference to $(>_{-k}, >'_k)$, which is on $N \setminus \{i\}$; similarly for $(>, >_j)$ and others.

1. If $C(>') \neq b$, then $d_{j,k} \neq k$
2. If $C(>') = b$, then j can manipulate at $>$ via $>'_j$

– Other cases where $C(>) = b, c$ or other alternatives are similar to the case above.

Therefore, we conclude that there are two distinct agents i, j such that $C_{i,j}$ is non-dictatorial.

■

6.2.2 The base step

We use computer programs to verify this part. The base case says that when $|N| = 2$ and $|O| = 3$, there is no social choice function on (N, O) that is onto, strategy-proof, and non-dictatorial. One might start to wonder if it is possible to generate all possible social choice functions in (N, O) and check all of them one by one for these three conditions. However, there are too many such functions for this to be feasible on current computers: there are $3! = 6$ number of linear orderings of O, resulting in $6 \times 6 = 36$ total number of preference profiles of (N, O), and 3^{36} possible social choice functions.

Our approach here is similar to what we did to verify the base case of Arrow;s theorem in [41]: we generate explicitly all social welfare functions that satisfy the conditions of strategy-proof and onto, and then check if any of them is non-dictatorial.

To achieve this, we formulate the above function generation problem as a constraint-satisfaction problem (CSP). A CSP is a triple (V, D, C), where V is a set of variables, and D a set of domains, one for each variable in V, and C a set of constraints on V (see, e.g. [33]). An assignment of the CSP is a function that maps each variable in V to a value in its domain. A solution to the CSP is an assignment that satisfies all constraints in C.

Now consider the voting model $(\{1, 2\}, \{a, b, c\})$ in our base case. We define a CSP for it by introducing 36 variables $x_1, ..., x_{36}$, one for each preference profile of the voting model. The domain of these variables is the set of 3 elements in $\{a, b, c\}$, and the constraints are the instantiations of the strategy-proof and onto conditions on the voting model. Apparently, there is a one-to-one correspondence between the social welfare functions of the voting model and the assignments of the CSP. Furthermore, a solution to the CSP corresponds to a social welfare function that satisfies the strategy-proof and onto conditions, and vice versa.

To tackle this CSP, we use the standard depth-first search that backtracks whenever the current partial assignment violates the constraints. We implement the above idea SWI-Prolog. Our program returns in less 1 second on an old laptop with 2.0GHZ CPU

and 512MB RAM. There are 17 strategy-proof social choice functions when $|N| = 2$ and $|O| = 3$. Among these 17 functions, 3 of them are constant functions which choose a single outcome for all the preference profiles, 12 of them have the ranges of size 2, the remaining 2 of them are onto which correspond to the dictatorships of two agents. In fact, all these 17 functions are all dictatorial on their images (p256, [1]).

This verifies the base case, thus completes our inductive proof of Gibbard-Satterthwaite Theorem.

6.2.3 Related work

There are several existing proofs of the Gibbard-Satterthwaite theorem. Gibbard [10] finds a connection between a social welfare function that satisfies Arrow's condition and a social choice function, thus by proving the strategy-proofness of the underlying social choice function, dictatorship follows from that of Arrow's impossibility theorem. Muller and Satterthwaite [24] find an equivalent relation between strategy-proofness and *strong monotonicity* and generalize Gibbard-Satterthwaite theorem to their celebrated Muller-Satterthwaite theorem. Others [5, 30] prove them directly without using Arrow's theorem. Particularly, Sen [36] also provides a semi-inductive proof in which he reduces the theorem to a base case where there are n agents and 3 outcomes, which makes proving the base case intellectually demanding as well. In our proof, we completely reduce the theorem to the base case where there are exactly 2 agents and 3 outcomes so that we can verify it using a simple computer program. Our proof follows the same structure of proof given for Arrow's theorem and other impossibility theorems [38].

6.3 Maskin's Theorem

As we mentioned, Gibbard-Satterthwaite theorem not only states the sad fact that designing strategy-proof voting rule is impossible (unless being dictatorial) but also describes a more general paradox that implementation in dominant strategy is also impossible (again, unless begin dictatorial)[2].

Definition 6.9 *A* mechanism M *is a tuple* $\langle S, g \rangle$*, where*

- $S = \prod S_i$*, is the product set of each agent's action space* S_i*.*
- $g : S \to O$*, is the outcome function that maps each action profile to an outcome.*

[2]This conclusion can be derived directly from the theorem as well as revelation principle, which states that a social choice function is implementable in dominant strategy equilibrium or Nash equilibrium if and only if it is implementable in a mechanism where each agent reports their truthful preference as an equilibrium strategy. We will make use of this principle when we prove Maskin's theorem.

Maskin's theorem gives a necessary condition for a social choice function to be *Nash Implementable*.

Definition 6.10 *A mechanism M* implements *a social choice function C in* Nash equilibrium *if for each preference profile $>$ and each Nash equilibrium s in the game induced by M and $>$, we have $g(s) = C(>)$. Such a C is called* Nash implementable.

A condition of a social choice function that is closely related to the Nash implementability is called *monotonicity*.

Definition 6.11 *A social choice function C is* monotonic *if, for every preference profile $>$ such that $C(>) = a$, if $>'$ is another profile such that $a >_i' a'$ whenever $a >_i a'$ for every agent i and every alternative a', then $C(>') = a$ as well.*

In other words, monotonicity means that if a choice function selects an outcome for a preference profile, then it will also select this outcome for any other preference profile that does not decrease the ranking of this outcome.

Theorem 6.12 ***Maskin's Theorem [19]***. *If a social choice function C is Nash implementable, then C is monotonic.*

As we mentioned, the above theorem gives a tight necessary condition to characterize the Nash implementatbility as Maskin [19] later showed that together with another extremely weak condition called *no veto power*, monotonicity suffices for Nash implementatbility. These findings gave Maskin the honor of Nobel Prize in Economics in 2007. We prove Theorem 6.12 in the following again using our computer-aided approach.

6.4 An inductive proof of Maskin's Theorem

For consistency, we keep the notation we used for the proof of Gibbard-Satterthwaite theorem.

6.4.1 The inductive step

Again, the inductive step consists of two lemmas.

Lemma 6.13 *If there is a social choice function for n individuals and $m+1$ alternatives that is Nash-implementable and non-monotonic, then there is a social choice function for n individuals and m alternatives that satisfies these two conditions as well, for all $n \geq 2, m \geq 2$.*

Proof: Let (N, O) be a voting model such that $|N| = n$ and $|O| = m + 1$, and C a social choice function that satisfies the two conditions in the lemma. We define C_a to be the same one as we defined in the previous proof. That is, for any preference profile $>$ of O_{-a}, $C_a(>) = C(>_{+a})$. We now prove separately that,

- C_a is well defined. That is, $C_a(>) \neq a$. Suppose otherwise that $C_a(>) = C(>_{+a}) = a$. Since C is Nash-implementable, there exists a mechanism $M = \langle S, g\rangle$ and an action profile s such that s is a Nash equilibrium in the game induced by $>_{+a}$ and $g(s) = a$. We must have $g(s_{-i}, s'_i) = a$ for any $s'_i \in S_i$ since otherwise will contradict to the fact that s is a Nash equilibrium in $>_{+a}$. In fact, this further implies that s is a Nash equilibrium in the game induced by any preference profile $>' \in O$. By the definition of Nash implementation, we must have $C(>') = a$ for any $>'$. Notice that such a C is monotonic in a trivial sense, contradicting to our assumption that C is non-monotonic.

- C_a is Nash-implementable. We can use the mechanism that implements C to implement C_a.

- We can find a $o \in O$ such that C_o is non-monotonic. Since C is non-monotonic, there exist $C(>) = a$ and $C(>') = b \neq a$ such that a improves its ranking in $>'$ with respect to $>$. Let $o \neq a, b$ and we prove that $C_o(>_{-o}) = a$ and $C_o(>'_{-o}) = b$. Since $C(>) = a$, there exists a mechanism $M = \langle S, g\rangle$ and an action profile s such that s is a Nash equilibrium in the game induced by $>$ and $g(s) = a$. s is still a Nash equilibrium in the game induced by $>_{-o}$. By the definition of Nash implementation, we have $C_o(>_{-o}) = a$. Similarly, we have $C_o(>'_{-o}) = b$. We also have a improves its ranking in $>'_{-o}$ with respect to $>_{-o}$. Therefore, we find a monotonic violation in C_o too. In other words, C_o is non-monotonic.

■

Lemma 6.14 *If there is a social choice function for $n+1$ individuals and m alternatives that is Nash-implementable and non-monotonic, then there is a social choice function for n individuals and m alternatives that satisfies these two conditions as well, for all $n \geq 2, m \geq 2$.*

Proof: Let (N, O) be a voting model such that $|N| = n + 1$ and $|O| = m$, and C a social choice function that satisfies the two conditions in the lemma. For any pair of agents $i \neq j \in N$, we similarly define $C_{i,j}$ on (N_{-i}, O) such that for any preference profile $>$ of (N, O), $C_{i,j}(>_{-i}) = C(>_{-i}, >_j)$. That is, let agent i always agree with agent j. We now show that,

- $C_{i,j}$ is Nash Implementable. Suppose C is implemented by $\langle S, g\rangle$ and we define $g_{i,j}(s_{-i}) = g(s_{-i}, s_j)$, the same way as we define $C_{i,j}$. Suppose $C_{i,j}(>_{-i}) = C(>_{-i}, >_j)$ is implemented by any Nash Equilibrium s in the game induced by $>_{-i}, >_j$ and g. We now show that s_{-i} is still a Nash Equilibrium in game induced by $>_{-i}$ and $g_{i,j}$ (and all the Nash Equilibria in it can be generated this way). It follows immediately that, for any agent $k \neq i, j$, his unilateral deviation cannot lead to a better outcome. The only exception is agent j, whose unilateral deviation in $g_{i,j}$ now leads to the deviation of both i and j in g. We now show that this deviation cannot lead to a better outcome either. By revelation principle, we can restrict our attention on direct revelation mechanism, i.e, $C = g$. Suppose otherwise, that is, $C_{i,j}(>) = C(>, >_j) = a$ and $C_{i,j}(>_{-j}, >'_j) = C(>_{-j}, >'_j, >'_j) = b$ with $b >_j a$. On one hand, since we have $C(>, >_j) = a$, we still have $C(>, >''_j) = a$ where $>''_j = \{c > a > \ldots, \}$. This is because the Nash equilibrium that implements $>, >_j$ is still a Nash equilibrium that implements $(>, >''_j)$. Similarly, we have $C(>_{-j}, >''_j, >''_j) = a$; on the other hand, for the same reason, since $C(>_{-j}, >'_j, >'_j) = b$, we also have $C(>_{-j}, >''_j, >''_j) = b \neq a$. This leads to a contradiction. To sum up, we have proved that $g_{i,j}$ implements $C_{i,j}$.

- We can find i, j such that $C_{i,j}$ is non-monotonic. Again, by revelation principle, we consider $C = g$ only. Since C is non-monotonic, there exist $C(>) = a$ and $C(>') = b \neq a$ such that a improves its ranking in $>'$ with respect to $>$. There are two cases,

  - **Case 1.** We can find two agents i, j such that $a > b$ in both $>_i$ and $>_j$. For the same reason we argued in the previous part, we have $C(>_{-i,-j}, >''_j, >''_j) = a = C_{i,j}(>_{-i,-j}, >''_j)$ where $>''_j = \{a > b > \ldots, \}$. Similarly, since a improves its ranking in $>'$ with respect to $>$, we have $a > b$ in both $>'_i$ and $>'_j$. Similarly, we have $C(>'_{-i,-j}, >''_j, >''_j) = b = C_{i,j(>'_{-i,-j}, >''_j)}$. As one can see, a still improves its ranking in $(>'_{-i,-j}, >''_j)$ with respect to $(>_{-i,-j}, >''_j)$. Thus, we find a monotonic violation in $C_{i,j}$
  - **Case 2.** Otherwise (since there are at least 3 agents), we can find two agents i, j such that $b > a$ in both $>_i$ and $>_j$. The remaining follows similar arguments to Case 1.

■

6.4.2 The base step

To verify the base case where there are two agents and two outcomes, notice again that we could apply the revelation principle and verify only if there is any social choice function that can be truthfully implemented by a direct revelation mechanism. The technical details, which we omit here, are similar to those of Gibbard-Satterthwaite theorem. Up till now, we have finished the proof of Maskin's theorem.

6.5 Summary and discussion

In this chapter, we follow the methodology we used earlier for proving Arrow's theorem and successfully prove Gibbard-Satterthwaite theorem. We first show that GS theorem holds if and only if it holds for the base case where there are exactly 2 agents and 3 alternatives. We then verify the base case using a computer program. Following similar strategy, we prove Maskin's theorem on Nash implementation as well. These proofs successfully generalize our methodology from social choice theory to implementation theory.

Chapter 7

Concluding remarks

"*That's one small step for a man, one giant leap for mankind.*"

—Neil Armstrong

From the beginning of this thesis, we have been advocating a methodology for discovering and proving theorems using computers. We now restate it as follows.

- Start from an existing theorem describing a sufficient condition of a property.
- Formulate the theorem and underlying theory in a logical language.
- Substitute the sufficient condition in the theorem with any logical sentence (within certain length) in the language and all such substitutions form a set of conjectures.
- Base step: Model-check these conjectures on small domains.
- Inductive step: Extend these survivors by finite verifiable property or any other means.
- Post-processing: delete those subsumed by others and return the remainings.

We have also seen some initial applications of this approach to economic theory. Some highlights are,

1. Game Theory
    - Starting from strictly competitiveness, we find a set of conditions that are also sufficient for uniqueness of PNE payoffs in two-person games. Among these conditions, we re-discover unilateral competitiveness and discover several new conditions. For strictly games, among others, we discover that unilateral competitiveness for individual player is suffice.
    - Strictly competitiveness is best-response equivalent to Uniqueness of PNE in strict two-person games.
    - In strict two-person games, a game is best-response equivalent to ordinal potential game iff it is best response equivalent to a quasi-supermodular game.
2. Social-Choice Theory

- We reprove Arrow's, Muller-Satterthwaite and Sen's impossibility Theorems.
- We discover two generalizations of Arrow's theorem.
- We discover a characterization theorem for Arrow's IIA condition.

3. Implementation Theory.
   - We reprove Gibbard-Satterthwaite and Maskin's Theorems.

We plan to expand our methodology in at least two directions.

1. One direction is to seek other fields where the methodology is applicable. One potential field we wish to explore is the classic bargaining theory. Nash's axiomatic approach reveals a sufficient condition for the optimal solution. One straightforward idea is to weaken this condition and search for a condition that is both sufficient and necessary. To achieve this, we first need to formulate the bargaining problem in a concise language and then apply our computer-aided approach to search for formulas that describe various sufficient conditions. Other potential fields include impossibility theorems in auction as well as those in mechanism design theory.

2. The other direction is to precisely define an algorithmic procedure that can be carried out given a target theory. This presents challenges for knowledge representation, that is, how to precisely represent a target theory.

When hopping on the moon, Neil Armstrong said that,

> "It suddenly struck me that that tiny pea, pretty and blue, was the Earth. I put up my thumb and shut one eye, and my thumb blotted out the planet Earth. I didn't feel like a giant. I felt very, very small."

That's also how we feel in front of the skyscraper of science. As Lenat [14] pointed out, theorem discovery is a science that requires the cooperation of domain experts. A nice theorem may be overlooked without appreciation of experts in the field. That is why we would hope a call to arms for this great adventure. After all, we have nothing to lose, because at least we give a computer-aided proof to the theorem we started with.

Bibliography

[1] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1 of *Handbook of Social Choice and Welfare*. Elsevier, 2002.

[2] K.J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, pages 328–246, 1950.

[3] Robert Aumann. Almost strictly competitive games. *Journal of the SIAM*, (6), 1962.

[4] Salvador Barbera. Pivotal voters : A new proof of arrow's theorem. *Economics Letters*, 6(1):13–16, 1980.

[5] Salvador Barbera. Strategy-proofness and pivotal voters: A direct proof of the gibbard-satterthwaite theorem. *International Economic Review*, 24(2):413–17, June 1983.

[6] Heinz-Dieter Ebbinghaus and Jrg Flum. *Finite Model Theory*. New York : Springer, 1995.

[7] S. Fajtlowicz. On conjectures of graffiti. *Discrete Math.*, 72(1-3):113–118, 1988.

[8] Peter C. Fishburn. Arrow's impossibility theorem: Concise proof and infinite voters. *Journal of Economic Theory*, 2(1):103–106, March 1970.

[9] John Geanakoplos. Three brief proofs of arrow's impossibility theorem. *Economic Theory*, (26):211–215, April 2005.

[10] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, July 1973.

[11] Mehmet H. Goker and Karen Zita Haigh. Introduction to the special issue on innovative applications of artificial intelligence. *AI Magazine*, 30(2):15–16, 2009.

[12] Amoz Kats and Jacques-Francois Thisse. Unilaterally competitive games. *International Journal of Game Theory*, 21(3):291–99, 1992.

[13] Pat Langley. The computer-aided discovery of scientific knowledge. In *In Proceedings of the first international conference on discovery science*, pages 25–39. Springer, 1998.

[14] Douglas B. Lenat. Automated theory formation in mathematics. In *IJCAI*, pages 833–842, 1977.

[15] Fangzhen Lin. Discovering state invariants. In *KR'04*, pages 536–544, 2004.

[16] Fangzhen Lin and Yin Chen. Discovering classes of strongly equivalent logic programs. *Journal of Artificial Intelligence Research*, 28:431–451, 2007.

[17] Fangzhen Lin and Pingzhong Tang. Computer aided proofs of arrows and other impossibility theorems. In *AAAI'08*, 2008.

[18] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, June 1995.

[19] Eric Maskin and Tomas Sjostrom. *Implementation theory*, volume 1 of *Handbook of Social Choice and Welfare*. Elsevier, March 2002.

[20] John McCarthy. Situations, actions and causal laws. *Semantic Information Processing*, pages 410–417, 1968.

[21] D. Monderer and L.S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[22] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, 2001.

[23] Eitan Muller and Mark A. Satterthwaite. The equivalence of strong positive association and strategy-proofness. *Journal of Economic Theory*, 14(2):412–418, April 1977.

[24] Eitan Muller and Mark A. Satterthwaite. The equivalence of strong positive association and strategy-proofness. *Journal of Economic Theory*, 14(2):412–418, April 1977.

[25] Michael Munie, Pingzhong Tang, and Yoav Shoham. A framework for quantitative evaluation of voting rules. In *Logic, Game Theory and Social Choice 6*, 2009.

[26] Roger B. Myerson. Fundamentals of social choice theory. Discussion Papers 1162, Northwestern University, Center for Mathematical Studies in Economics and Management Science, September 1996.

[27] Martin J. Osborne. *An introduction to game theory*. Oxford Univ. Press, 2004.

[28] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[29] Marko Petkovsek, Herbert Wilf, and Doron Zeilberger. *A=B*. A K Peters, Ltd, 1996.

[30] Philip J. Reny. Arrow's theorem and the gibbard-satterthwaite theorem: a unified approach. *Economics Letters*, 70(1):99–105, January 2001.

[31] Robert W. Rosenthal. Correlated equilibria in some classes of two-person games. *International Journal of Game Theory*, (3):119 – 128, 1974.

[32] R.W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[33] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

[34] Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, April 1975.

[35] Amartya Sen. The impossibility of a paretian liberal. *Journal of Political Economy*, 78(1):152–57, Jan.-Feb. 1970.

[36] Arunava Sen. Another direct proof of the gibbard-satterthwaite theorem. *Economics Letters*, 70(3):381–385, March 2001.

[37] Kotaro Suzumura. Welfare economics beyond welfarist-consequentialism. Discussion Paper Series a382, Institute of Economic Research, Hitotsubashi University, October 1999.

[38] Pingzhong Tang and Fangzhen Lin. Computer aided proofs of arrows and other impossibility theorems. *Artificial Intelligence*, 173:1041–1053, 2009.

[39] Pingzhong Tang and Fangzhen Lin. Discovering theorems in game theory: Two-person games with unique nash equilibria payoffs. In *IJCAI'09*, 2009.

[40] Pingzhong Tang and Fangzhen Lin. Two equivalence results for two-person strict games. *Games and Economic Behavior, to appear*, 2010.

[41] Pingzhong Tang, Yoav Shoham, and Fangzhen Lin. Team competition. In *Proceedings of AAMAS '09*, pages 241–248, 2009.

[42] Pingzhong Tang, Yoav Shoham, and Fangzhen Lin. Team competition. *Artificial Intelligence, to appear*, 2010.

[43] Donald Topkis. *Supermodularity and Complementarity*. Princeton University Press, New Jersey, 1998.

[44] Mark Voorneveld. Best-response potential games. *Economics Letters*, 66(3):289–295, March 2000.

[45] Robert Wilson. Social choice theory without the pareto principle. *Journal of Economic Theory*, 5(3):478–486, December 1972.