Essays on Individual and Collective Decision Making

Thesis submitted to The Indian Statistical Institute for partial fulfillment of the requirements for the degree of Doctor of Philosophy

Saptarshi Mukherjee

February 2011

Indian Statistical Institute

Essays on Individual and Collective Decision Making

Thesis submitted to the Indian Statistical Institute in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Saptarshi Mukherjee

February 2011

Essays on Individual and Collective Decision Making

Saptarshi Mukherjee

February 2011

Thesis Supervisor: Prof. Arunava Sen

Thesis submitted to the Indian Statistical Institute in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Acknowledgements

This thesis is the output of research over a long period at Indian Statistical Institute, New Delhi. At this point of time I strongly feel the urge to express my gratitude towards some people without whose (direct or indirect) involvement, this thesis would not have been possibly written.

First, I am wholeheartedly grateful to my supervisor, Prof. Arunava Sen. It is him, who on the earth is fully responsible for developing my interest towards the area of social choice, mechanism design and implementation theory. On one hand, I have been simply amazed to feel his impeccable knowledge in the subject and on the other side I have been deeply influenced by the way he motivates a research problem. Similarly, I have noticed his careful attention to any mistake that I have made, in writing or in approaching a problem. Often I have been embarrassed to see his diligent effort to correct these mistakes and I have simply wished that I had not done them! I must also admit that, not only regarding other academic issues, I have got scope to interact with Prof. Sen frequently on other issues. In times, I have taken advice from him regarding different matters and have been benefitted. This helped me to refresh my mind and concentrate in my work. Once again, I feel very much indebted to him for all his efforts and patience.

There have been several other senior researchers who have been kind enough to extend their helping hands to me in the pursuit of this piece of research. I gratefully acknowledge inputs from Prof. Salvador Barberà, Prof. Dinko Dimitrov, Prof. Matthew O. Jackson, Prof. Francois Maniquet, Prof. Ariel Rubinstein, Prof. Hamid Sabourian, Prof. William Thomson. They have patiently listened to me and given many invaluable comments, advice and have showed paths for future research. I am really thankful to all them. I am deeply grateful to all the faculty members of ISI Delhi for their continual help and encouragement. In particular, I remember Prof. Chetan Ghate and Prof. Prabal Roy Chowdhury giving me great academic support at different points of time. I have desperately approached Prof. Debasis Mishra without any hesitation for help, particularly for LaTeX and he has always indulged me with smiling face. I am indebted to all them.

I have been lucky to get a great group of friends at ISI Delhi and outside, particularly Prof. Anirban Kar, Prof. Dipjyoti Majumdar, Prof. Meeta K. Mehra, Prof. Jaideep Roy and Prof. Abhijit Sengupta. I have continually received help, inputs and encouragement from them. I would also like to acknowledge specially the support I have obtained from Ms. Monica Dutta. Not only she is I could simultaneously discuss my research and goof off with, but also she did stand by me in times of troubles. Thanks for this!

My research has been benefitted from comments and feedbacks I received from various seminars and conference presentations. I wish to thank participants of Indian Statistical Institute, New Delhi, Jawaharlal Nehru University, New Delhi, Delhi School of Economics, South and South East Asia Econometric Society Meeting (2006) at Chennai, The Arne Ryde Symposium at Sweden (2008), European Symposium for Study in Economic Theory (CEPR) at Switzerland (2009), Social Choice and Welfare Meeting at Moscow (2010). I also gratefully acknowledge a Research Fellowship offered by Indian Statistical Institute and Ford Fellowship offered by Jawaharlal Nehru University, New Delhi.

Last, but definitely not the least, I am thankful to my parents, Mr. Sujit Kumar Mukherjee and Mrs. Mita Mukherjee for their immeasurable love, affection and for giving their *all*

for endowing me with the best possible education opportunities for ever! This work is a token of my respect for you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation: Chapter 1

Standard choice models generally assume that a decision maker (henceforth *DM*) chooses an alternative from a *set* of alternatives. It is often the case that a DM encounters the alternatives in a particular structure. For instance, a shopper has to choose from a set of items displayed on a shelf or a judge has to choose an winner from the contestants who appear one after another in a row. In both these cases, the alternatives appear sequentially to the DM, i.e. the set of alternatives appears in the form of a *list*. The structure of the set can be more complicated as illustrated in the following example: suppose the DM is purchasing a product online. For every feature of the product the DM gets to choose from a variety of options that appear as hyper-links in the webpage. After choosing an option, a new webpage opens up and provides options for another feature or attribute. Thus the DM has to choose sequentially and the whole set of alternatives appears in the form of a tree in which alternatives labeled at the terminal nodes are ordered, i.e. in the form of an *ordered-tree*. Some elections may also follow a sequential process which can be modeled as choosing from an ordered tree.

A natural question that arises here is the following: does the structure of the set of alternatives affect the choice or in other words, does it matter how the alternatives are presented to the DM? We can identify several effects which suggest that the choice depends on the structure. We give some examples. When the DM chooses from a list, the first few alternatives may grab attention and become favorite. On the other hand, the DM may be more likely to remember the alternatives that come in the tail of the list. There can be other cases also. For instance, the DM may pay more attention to the alternative which is more distinct than the alternatives that surround it in the list. Similarly, while choosing from an ordered-tree, the DM can be naive, i.e. chooses a particular branch (say, the left-most branch) at each decision node. The DM may also choose a path that is shortest from the initial decision node, because it saves time.

A number of experimental and empirical findings indicate that the structure of set of alternatives affects decision. Rubinstein et al. (1996) consider a two-person game where one player hides a treasure in one of four places located in a row and the other player seeks it. They find that both players prefer middle positions to end-points. Attali and Bar-Hillel (2003) find that in an examination with multiple-choice questions, question-setters have tendency to keep the right answer in middle positions and students have tendency to seek it in middle positions. Christenfeld (1995) finds that while purchasing items from a shelf of grocery shop, buyers are naive towards middle positions. Online purchases of items like books, apparel and household items are very common nowadays. We often find that if we browse the same shopping site more than once, the order or the sequence in which items appear changes in every search. This suggests that online sellers take "order effects" into account. These examples suggest that presentation and the structure of the set of alternatives have an important effect on choice. It is therefore interesting to study choice rules in the presence of presentation effects.

Rubinstein and Salant (2006) axiomatically analyze choice functions from lists. Their paper proposes an independence axiom and characterizes choice functions that satisfy it. They show that the chosen alternative is maximal according to an ordering. If the maximal set is not a singleton then the choice is made using a tie-breaking rule which depends on the list. In Chapter 1 of this thesis we consider models where the DM faces a decision problem which is an *ordered-tree*. We conduct an axiomatic analysis of these problems in the same spirit as Rubinstein and Salant (2006). We summarize the main results of the chapter below.

1.2 Summary: Chapter 1

We consider choice functions that satisfy two reasonable axioms - Backward Consistency (BC) and Repetition Indifference (RI). The BC axiom requires the following: if we partition a tree t into a set of sub-trees so that t is a concatenation of these sub-trees, then the DM chooses the same alternative in t whether he chooses from t as a whole, or considers the sub-trees of t, chooses from each sub-tree and then again chooses from the reduced form tree. The axiom is defined recursively. Informally, one can start from the "end" of a tree choosing subsequently from the sub-trees which are concatenated backward to form the tree. The axiom is similar in spirit to the backward induction algorithm in game theory.

The RI axiom requires that the choice from a tree should remain the same when a non-chosen alternative is replaced by any other alternative from its "partner set". Two alternatives belong to the same partner set in a tree if every sub-tree of the tree which contains one of the alternatives also contains the other. The axiom is in the same spirit as the various Independence of Irrelevant Alternatives (IIA) axioms used in social choice theory, e.g. Sen's Property (Sen (1993)).

We first characterize choice in binary trees. We show that the chosen alternative is maximal according to a weak ordering over the set of alternatives. If the maximal set contains more than one alternative, there exists a tie-breaking rule. The rule chooses the left-most (or the right-most, depending on the set of alternatives appearing in the tree) alternative from the maximal set. Thus it depends on the structure of the tree. Next we investigate choice in ternary and higher order trees. We show that there exists a weak ordering where indifference is allowed only for bottom-ranked alternatives. The choice function chooses the maximal alternative according to this ordering. If there is indifference, the choice function uses a tie–breaking rule. The tie-breaking rule also depends on the cardinality of the set of alternatives that are indifferent (and are bottom-ranked). We note that the weak order characterization of choice rules in binary trees is similar to the weak order characterizing choice from lists shown in Rubinstein and Salant (2006). However the result for the binary trees is qualitatively different from that in the ternary trees or higher order trees. In the latter, the ordering is more restrictive and the tie-breaking rules are different and more complicated. We also extend the analysis to the general class of trees.

1.3 Motivation: Chapter 2

Implementation theory deals with group decision-making processes under various information structures. The objective of the theory is to structure the strategic interactions of the agents in a group so that their actions lead to a socially desirable outcome in each "state of the world". The group's collective objectives are specified by a social choice correspondence (SCC) that selects a set of alternatives from the available set in every state of the world. A classic example is the one of building a public good or project by a public authority. The authority needs to compare its cost to its social benefit . For this the authority needs to know agents' valuations for the project. But these valuations are private information of the agents and unknown to the authority. Implementation theory attempts to design a game-form such that in every state equilibrium actions of agents according to a pre-specified equilibrium notion leads to a socially desirable outcome in that state i.e. they belong to the image-set of the SCC in that state. Clearly, information available to the agents but unknown to the planner will affect the socially desirable outcome. The formulation of the problem must use game-theoretic solution concepts that are appropriate for agent behavior and consistent with informational assumptions.

The literature on implementation considers various equilibrium notions. In the "complete information model", it is assumed that all agents know the state while the mechanism designer does not. A natural notion of equilibrium in this context is Nash equilibrium (Maskin (1999)). Other notions that are consistent with this information setting and have been studied include the iterated eliminated of weakly undominated strategies (Moulin (1979)), sub-game perfect Nash equilibrium and various other Nash equilibrium refinements.[1]

In a private information setting each agent has private information about her type and a state of the world is a collection of types for all agents. Suppose an agent's type is her *preference ordering* over a finite set of alternatives. Thus each agent knows her own preferences but is not aware of the preferences of others. The mechanism designer does not have information regarding the state. Equilibrium notions such as the iterated elimination of dominated strategies or Nash equilibria are inappropriate in this information structure. For instance, iterated eliminations of dominant strategies is inappropriate because an agent will be unable to predict the strategies that will be eliminated by other agents because that will depend on the types of the other agents which are unobservable.

Some natural notions of equilibria in such an information setting are dominant strategies and Bayes-Nash equilibrium. If the dominant strategy notion is used there exists a message in

[1]See Corchòn (2009), Jackson (2001) and Serrano (2004) for surveys of this literature.

each state that (weakly)-dominates all other messages. On the other hand, a strategy-tuple is a Bayes-Nash equilibrium if unilateral deviations are not profitable in terms of expected pay-offs where these expectations are computed from prior beliefs of the other agents' types. A significant advantage of using dominant strategies is that the mechanism designed does not depend on the prior beliefs of the agents. However dominant strategy implementation is a strong requirement. The most important result in this area is the Gibbard-Satterthwaite (Gibbard (1973), Satterthwaite (1975)) Theorem. According to the Theorem, the only social choice functions (SCFs) that can be implemented in a complete domain of preferences are *dictatorial* provided the SCF has a range of at least three alternatives.

Another prior-free notion of equilibrium that is consistent with the private information setting is the *single round elimination of weakly dominated strategies*. We will refer to implementation in this solution concept as implementation in undominated strategies. Implementation in undominated strategies has many of the strengths of dominant strategy implementation. However as we shall see it is not as restrictive as dominant strategy implementation provided we consider SCCs rather SCFs.

Implementation in undominated strategies was first introduced and studied in an important paper by Jackson (1992). The paper proved a surprising and powerful result: *all* SCCs can be implemented in undominated strategies. The mechanism constructed involves infinite strings of strategies each of which dominates the earlier one. The paper interpreted this as a weakness in the solution concept and proposed the following restriction on admissible mechanisms: for each state each weakly dominated strategy must be dominated by an undominated strategy. The paper refers to such mechanisms as *bounded* mechanisms.

A fundamental question is the following: what is the class of social choice correspondences that can be implemented in undominated strategies by bounded mechanisms? This appears to be an exceedingly difficult question. The only available general result is in Jackson

(1992) and shows that a social choice function is implementable over a complete domain of preferences if and only if it is dictatorial. In Chapter 2 we investigate some special aspects of the implementation of SCCs in undominated strategies. Throughout the chapter we will consider the mechanisms in which message spaces are finite. Note that such mechanisms are bounded.

1.4 Summary: Chapter 2

We first consider the case where there is a single agent and three alternatives. We provide a complete characterization of all implementable SCCs in this environment. We provide a necessary and sufficient condition called the "Neighborhood Flip (NF) condition" for implementation. This condition can be thought of as a generalization of the standard monotonicity condition that is necessary for implementation in dominant strategies. We also highlight the differences with implementation in dominant strategies by showing that a wider class of SCCs can be implemented.

Next, we investigate implementation (for an arbitrary number of players and alternatives) in a more restricted class of mechanisms which we call "covered mechanisms". These are mechanisms where every message of an agent is undominated for at least at one preference ordering of the agent. In general a mechanism that implements a SCC may contain a message for an agent that is always dominated. We show that if a SCC is implementable, then there exists a covered mechanism that weakly implements the SCC i.e. implements a sub-correspondence of the SCC. We identify a condition which we call "General Monotonicity (GM)" that is necessary for implementation by covered mechanisms. The GM condition builds on the NF condition and also implies the strategy-resistance condition that Jackson (1992) shows is necessary for implementation. We show that the Pareto correspondence for two agents and three alternatives does not satisfy the GM condition and hence cannot be implemented by a covered mechanism. But we design a complicated uncovered mechanism that implements it in the two agent three alternatives case.

It is well-known that SCCs which pick unions of top-ranked alternatives of agents at every preference profile, are implementable.[2] A drawback of these SCCs is that they overlook "compromise" alternatives. Consider a case where there are two agents and a hundred alternatives. Alternative a is ranked first by agent 1 but hundredth by agent 2. Alternative b is ranked first by agent 2 but hundredth by agent 1. On the other hand, alternative c is ranked second by both agents. It seems very natural for a SCC to pick c at this profile while the union of top-ranked alternatives SCC will pick the set $\{a, b\}$. Borgers (1991) investigates the implementability of SCCs that pick compromise alternatives. In particular an alternative is a compromise at a profile if is not first-ranked by an agent but is (Pareto) efficient and a SCC satisfies the *Compromise Axiom* if there exists a preference profile where the SCC picks *only* compromises. The paper proves that if there are either two agents or three alternatives, then there doest not exist an efficient, implementable SCC satisfying the compromise axiom.

We extend and refine the Borgers' result (Borgers (1991)) in a number of ways by proving a number of possibility as well as impossibility results. In particular we show the following:

- There does not exist an *efficient*, implementable SCC for any number of agents and alternatives satisfying the compromise axiom at preference profile that are "near unanimous", i.e. preference profiles where all but one agents agree on the top-most alternative.

- There does not exist a *neutral*, implementable SCC for two agents and an arbitrary

[2] "Integer game" mechanisms of the type used in complete information implementation theory (Maskin (1999)) can be used for the purpose.

number of alternatives satisfying the compromise axiom.

- There does not exist a *unanimous*, implementable SCC for two agents and an arbitrary number of alternatives satisfying the compromise axiom and the additional axiom of *minimality*. Each alternative in the image set of a minimal SCC at a profile is the maximal element in that set for some agent at that profile.

- There exists a *unanimous*, implementable SCC for two agents and three alternatives satisfying the compromise axiom.

- There exists an efficient, implementable SCC satisfying the compromise axiom for an arbitrary number of agents and alternatives in the special case where *a single agent* has only *one* preference ordering (i.e. is of a single "type").

Chapter 2

Choice in ordered-tree based decision making problems

2.1 Introduction

Our model deals with situations where a decision maker (henceforth *DM*) faces a decision problem which has a sequential structure. It is often the case that the DM encounters the alternatives in a particular structure, for instance in the form of a list or in the form of an ordered-tree (henceforth by *tree* we mean *ordered tree* only). Casual observations indicate that choice in these situations depends on the structure in which the alternatives are presented to the DM. While purchasing a product online, the alternatives are listed from left to right or top to bottom. In this case, the choice depends on the list the DM faces while browsing the internet. The order in which alternatives are listed might influence her choice. Similarly alternatives can appear in the form of a tree-where decisions are taken sequentially. We give some examples.

1. The DM is planning to purchase a product (e.g. television) online. Suppose she visits one of the online shopping websites. She is first asked to choose the type of television (by clicking on the hyperlink provided on the webpage) whether it is a Flat (F) or an LCD (L) or an Ordinary (O) television. Suppose she chooses an F type. She then has to choose a screen size, for instance between a 14″, 22″ or 29″ model. If she chooses an L or O type, she has to make a similar decision about screen size. We represent this problem in the following figure.

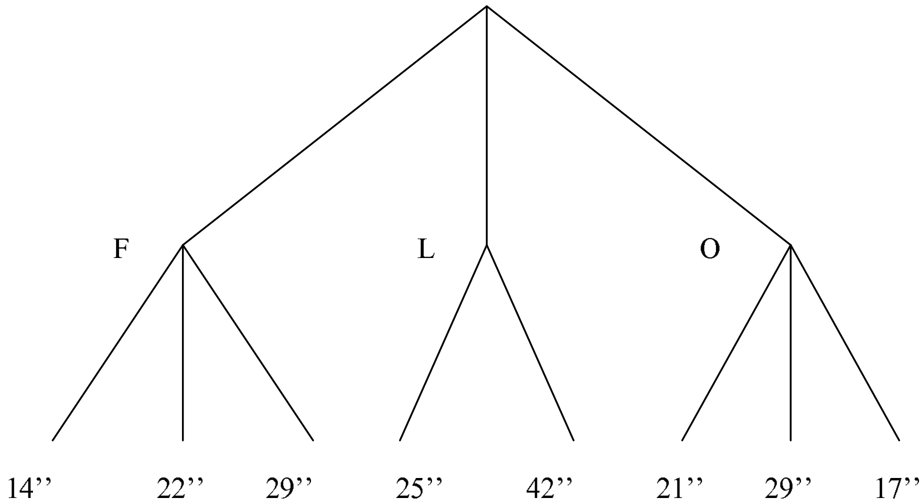

Figure 2.1: Tree showing the choice problem for online TV shopping

2. Amendment Agendas (Ordeshook and Schwartz (1987)). Such agendas work as follows: a sequence of alternatives is given, and a vote is taken between the first two, after which the winner fights with the third alternative. Then the winner faces the fourth alternative. For instance consider the motions on the floor of a senate regarding a bill. The issues are : a bill, an amendment to the bill, and an amendment to the amendment. Then there are four possible outcomes: (i) the status quo, (ii) the unchanged bill, (iii) the bill amended and (iv) the bill changed by the amended amendment. The procedure requires sequential voting where the first vote is between (iii) and (iv), i.e. whether to amend the amendment. Next vote is taken between the winner and (ii), i.e. decision regarding whether to amend the bill.

Finally the vote is taken between the winner and (i). Thus the last decision is whether to pass the final form of the bill. We note that in this example the DM is the legislature.

We can represent this agenda by the tree in following figure:

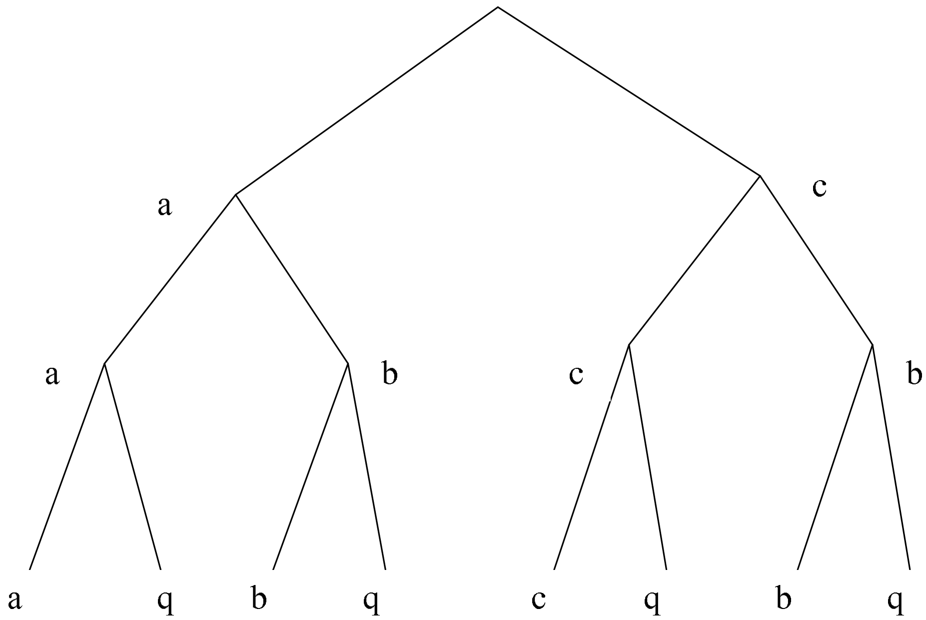

Figure 2.2: Tree showing choice problem in Amendment Agendas

Here

q: the status quo

b: unchanged bill

a: bill changed by original amendment

c: the bill changed by the amended amendment

Here the decision making process takes place sequentially when the set of alternatives appear in the structure of a tree.

We address the question of choosing in such situations such as those described above, in the spirit of the analysis in Rubinstein and Salant (2006). In particular, we consider choice functions which satisfy certain plausible axioms. First we consider choice in binary trees. We show that the chosen alternative is maximal according to a weak ordering over the set of alternatives. If the maximal set is not singleton, there exists a tie-breaking rule. The rule chooses the left-most (or the right-most, depending on the set of alternatives appearing in the tree) alternative from the maximal set. Thus it depends on the structure of the tree. Next we consider choice in ternary trees or trees of higher order. We show that there exists a weak ordering with indifference allowed only for alternatives which are bottom-ranked. The choice function chooses the maximal alternative from a tree according to this ordering. If there is indifference, the choice function uses a tie–breaking rule. It is important to appreciate that the result for the binary trees is qualitatively different from that in the ternary trees or higher order trees- in the latter, the ordering is more restrictive and the tie-breaking rules are different.

Our chapter is closely related to and strongly influenced by Rubinstein and Salant (2006). They characterize choice functions from lists using an independence axiom. They show that the chosen alternative is maximal according to an ordering. If the maximal set is not a singleton then the choice is made using a tie-breaking rule which depends on the list. Our result in the binary tree case is similar to Rubinstein and Salant (2006) result, but the ternary trees or higher order cases are different. We discuss the exact relationship between our axioms and results in the body of the chapter.

In Section 2 we discuss axioms with some examples. In Section 3 we characterize choice functions from binary trees. In section 4 we consider the ternary and higher-order case. In Section 5 we consider the most general case, i.e. choices from all possible trees, including mixed trees. In section 6, we conclude.

2.2 The Model

We consider a finite set of alternatives X with $|X| = n$. A tree is a graph with no cycles (for a formal definition of tree see Harary (1969)) and an ordered-tree is a tree where alternatives labeled at the terminal nodes are ordered. Henceforth by "tree" we mean "ordered-tree" only. Throughout this chapter we consider finite trees, i.e. trees with a number of terminal nodes. The initial node is called *"origin"* and the terminal nodes are labeled with alternatives from X. The *length of a path* from any terminal node to the origin is the number of nodes that appear in that path. The *elementary tree of order* k is the tree with a single vertex and k offsprings. Each offspring ends in a terminal node which is labeled with an alternative of X. A tree with equal number of choices, say k ($k = 2, 3, ...$) at each node is called a *k-ary tree.* Trees with $k = 2$ and $k = 3$ are called binary and ternary trees respectively. In general we say that k is the order of a k-ary tree. Trees which allow different number of offsprings at different nodes are called *mixed trees* or (MT).

In next two figures we present an elementary binary and an MT.

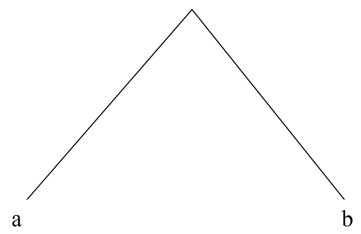

Figure 2.3: An elementary Binary Tree

A tree is symmetric if the length of the path from the origin to any terminal node is the same. We give an example in figure 2.5.

Let Γ^k denote the set of all k-ary trees. Thus Γ^2 is the set of all binary trees. Also let the set of all possible trees be denoted by Γ, so that $\cup_{k\geq 2}\Gamma^k \subset \Gamma$. For any $t \in \Gamma$, let

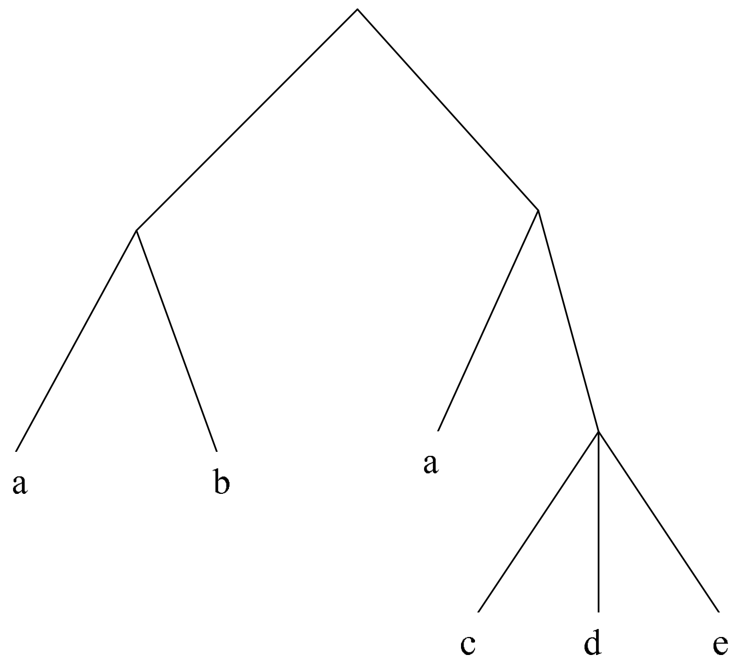

Figure 2.4: A Mixed Tree

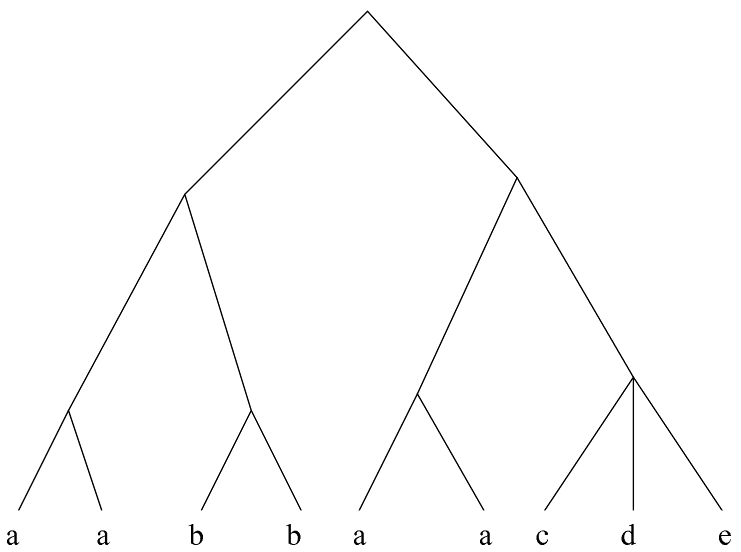

Figure 2.5: A Symmetric Tree

$X(t) \subset X$ be the set of alternatives which appear as terminal nodes in t. Suppose $\Gamma^k(B)$ denotes the set of all k-ary trees formed with all alternatives from the set B, $B \subset X$. Thus

$\Gamma^k(B) = \{t \in \Gamma^k : X(t) = B\}$. Similarly $\Gamma^k_e(B)$ denotes the set of all elementary k-ary trees formed with all alternatives from the set B, i.e. $\Gamma^k_e(B) = \{t \in \Gamma^k, t \text{ elementary} : X(t) = B\}$. Here $B \subset X$. For any tree $t \in \Gamma^k(B)$, $X(t) = B$.

A tree can be represented as an ordered sequence $t \equiv (X^1, X^2, \ldots, X^M)$ where each $X^j \in \underbrace{X \times X \times \ldots \times X}_{r^j}$, $j = 1, 2, \ldots, M$. Thus each X^j is a collection of r^j alternatives of X. Note that X^j may contain repetitions. The interpretation here is that $X^1 \cup X^2 \ldots \cup X^M$ constitutes the set of terminal nodes in t. All nodes represented in X^j, $j = 1, \ldots, M$ are successors of the same non-terminal node. For example consider the tree t in figure 2.4. Here t can be equivalently represented as $t \equiv (X^1, X^2, X^3)$, where $X^1 = \{a, b\}$, $X^2 = \{a\}$ and $X^3 = \{c, d, e\}$.

Note that the sets $X^1, \ldots, X^M$ are ordered. Thus we refer to X^j as the j-th set from the left. Similarly if $X^j \equiv \{x^j_1, \ldots x^j_k, \ldots x^j_r\}$, then x^j_k is the k-th alternative from the left in X^j. In general we can also say without ambiguity that x^k_j is to the left of $x^{k'}_{j'}$ if $j < j'$ or $j = j'$ and $k < k'$. It follows from this discussion that any tree $t \equiv (X^1, X^2, \ldots, X^M)$ can be written as $t \equiv (\{x^1_1, x^1_2, \ldots, x^1_{r^1}\}, \ldots, \{x^M_1, x^M_2, \ldots, x^M_{r^M}\})$ where $X_i = \{x^i_1, x^i_2, \ldots, x^i_{r^i}\}$.

Let $x(r;t)$ denotes the alternative in set $X(t)$ that can be reached in the tree t by picking the r-th branch from the left at every non-terminal node, starting from the initial node. For any k-ary tree, $1 \leq r \leq k$. For any mixed tree, $x(r;t)$ is the alternative chosen in the following way: pick the r-th branch from the left at every non-terminal node, starting from initial node; if at some node, the right-most branch available is the m-th branch, $m < r$, then the m-th branch at that node is picked. For instance, for the tree in figure 2.5, $x(3;t) = e$. Here e is obtained by picking the second branch (because there is no third branch at these nodes) at the first two nodes (starting from the initial node) and picking the third branch in the next node. Thus for any tree t in Γ, $x(r;t)$ is the alternative that can be reached in the tree t, by always picking $Min\ (m, r)$-th branch from the left at every non-terminal node starting from the initial node where m is the number of offsprings available at that non-terminal node.

Definition 1 A choice function C is a map $C : \Gamma \to X$, such that $C(t) \in X(t)$, for all $t \in \Gamma$.

We have already made the assumption about representing non-symmetric trees as symmetric trees. This extension of non-symmetric trees to symmetric trees is consistent with the structure of our problem and our goal in this chapter. Repetitions at the terminal nodes make no change in the choice problem. Thus it is trivial that our domain contains only symmetric trees. Choice functions for k-ary and elementary trees are defined in a natural way. A choice function C from k-ary trees is a map $C : \Gamma^k \to X$, such that $C(t) \in X(t)$, $\forall t \in \Gamma^k$. A Choice Function from elementary trees is a choice function that chooses an alternative from $X(t)$ for every elementary tree t.

We give examples of some salient choice functions below.

Example 1 A choice function $C : \Gamma' \to X$, $\Gamma' \subset \Gamma$, is *Preference Based* or PREF, if there exists a strict ordering $\succ$[1] over X such that for all trees $t \in \Gamma'$, $C(t) = (M(X(t), \succ))$ [Given a set of alternatives X, and any ordering R over X, we define the *maximal set*, as $M(X, R) = \{x | xRy \text{ for all } y \in X\}$]. Let $X = \{a, b, c, d, e\}$ and let $\succ$ be a strict ordering over X such that $a \succ b \succ c \succ d \succ e$. Let C be $PREF$ which chooses $M(X(t), \succ)$ for any $t \in \Gamma$. Let $t \equiv (\{c, b, b\}, \{b, c, d\}, \{d, d, b\})$. Here $M(X(t), \succ) = b$ and $C(t) = b$. This choice function is similar to the Rational Choice Function defined in Rubinstein and Salant (2006).

Example 2 A choice function $C : \Gamma' \to X$, $\Gamma' \subset \Gamma$, is *Procedural* or PROC, if there exists a positive integer r such that for all $t \in \Gamma'$, $C(t) = x(r; t)$. Recall that $x(r; t)$ denotes the

[1] A binary relation $\succ$ is called *strict ordering* if it satisfies complete, antisymmetric and transitive.

alternative in set $X(t)$ that can be reached in the tree t by picking the r-th branch from the left at every non-terminal node, starting from the initial node. For instance, we could have $C(t) = x(2; t)$ for any $t \in \Gamma^3$. Let $t \equiv (\{c, b, b\}, \{b, c, d\}, \{d, d, b\})$. Here $C(t) = c$.

The choice function defined above is called "Procedural" because whenever a DM faces a choice, she uses a rule-of-thumb based on the structure of the tree: for instance in a binary tree, she always chooses left when faced with a choice; in a ternary tree, she always chooses middle and so on.

A more general version of the PROC is the following: the DM chooses k_1^{th} branch [2] at first decision node, the k_2^{th} branch at the second node reached and k_i^{th} branch at the i^{th} node reached by this procedure and so on until a terminal node is reached. We shall call this choice function G-PROC.

Example 3 A choice function $C : \Gamma' \to X$, $\Gamma' \subset \Gamma$, is *Satisficing* or SAT if there exists a strict ordering $\succ$ over X and a threshold $a^* \in X$ such that for any $t \in \Gamma'$, $C(t)$ is the first alternative in the tree that is not inferior to a^* (recall that t can be written as an ordered sequence $t \equiv (X^1, X^2, \ldots, X^M)$, where X_i, $i = 1, 2, \ldots, M$ are ordered sets); if there is none then the last alternative in the tree is chosen. This rule follows from Simon (1955) and has also been mentioned in Rubinstein and Salant (2006).

In the next section we consider axioms which help us to characterize choice functions.

[2] k^{th} branch at a decision node in a tree is the next branch to the $(k-1)^{th}$ branch from the left emanated from that decision node

2.3 Axioms

In this section we introduce and discuss the axioms which we impose on choice functions from trees. These axioms are similar in spirit to the axioms used in Rubinstein and Salant (2006). First we define an operation on trees.

Definition 2 Let $t_1, t_2, t_3, \ldots, t_p \in \Gamma$, where $t_i \equiv (X_i^1, X_i^2, \ldots, X_i^{n_i})$. The *horizontal concatenation* of $t_1, t_2, \ldots, t_p$, denoted by $(t_1 \text{ o } t_2 \text{ o } \ldots \text{o } t_p)$, is the tree $t \equiv (X_1^1, X_1^2, \ldots, X_1^{n_1}, X_2^1, X_2^2, \ldots, X_2^{n_2}, ., X_p^1, X_p^2, \ldots, X_p^{n_p}) \equiv (X^1, X^2, \ldots, X^N)$; where $X^1 = X_1^1$, $X^2 = X_1^2$, $\ldots$, $X^N = X_p^{n_p}$. Alternatively t can also be represented as $t \equiv (\{x_1^1, x_2^1, \ldots, x_{k_1}^1\}, \{x_1^2, x_2^2, \ldots, x_{k_2}^2\}, \ldots \{x_1^l, x_2^l, \ldots, x_{k_l}^l\})$.

REMARK 1 We have defined horizontal concatenation for any set of trees in Γ. This operation can also be defined for trees in Γ^k, with $p = k$. Thus for k-ary trees, this operation has been defined such that resulting tree is also a k-ary tree. This operation can be used recursively. For instance, suppose t', t'', t''' are three ternary trees, each formed by horizontal concatenation of 3 trees from Γ_3. A new ternary tree t^* can be formed by horizontal concatenation of t', t'' and t'''. Thus a k-ary tree can be formed by horizontal concatenation of k trees from Γ_k where each of these k-ary trees are formed by horizontal concatenation of k trees from Γ_k and so on. We will use this recursive property in the axiom to be discussed next.

Backward Consistency (BC): For each tree $t \equiv \{t_1 \text{ o } t_2 \text{ o } \ldots \text{o } t_p\} \in \Gamma$, the following holds:

$$C(t) = C(C(t_1), C(t_2), \ldots, C(t_p))$$

Here t is a concatenation of $t_1, t_2, \ldots, t_p$ and t_i is concatenation of $t_i^1, t_i^2, \ldots, t_i^{k_i}$ $(i = 1, 2, \ldots, p)$ and so on. Observe that BC is defined recursively. If we partition a tree t into a set of sub-trees so that t is a concatenation of these sub-trees, then BC works on these partitions inductively. It implies that the DM chooses the same alternative from an arbitrary tree whether he chooses from the tree as a whole, or partitions the tree into sub-trees, chooses from each sub-tree and then again chooses from the reduced form tree and so on.

This axiom is called "Backward Consistency" because it is similar in spirit to the backward induction. Backward induction is widely used in economics, for instance in sub-game perfect equilibrium in sequential games and in various dynamic optimization problems. The BC axiom requires that one can start from the "end" of a tree choosing subsequently from the sub-trees which are concatenated backward to form the entire tree. This axiom is clearly based on the general principle of backward induction.

The BC axiom is also related to the *Partition Independence (PI)* axiom in Rubinstein and Salant (2006). A choice function satisfying PI chooses the same alternative from a list whether the DM chooses from the whole list or he partitions the list and chooses from the list comprising chosen alternatives from the sub-partitions. Here BC implies the same choice from a tree as a whole or from a tree comprising choices from sub-trees which are concatenated to form the whole tree. Similarly we can relate BC to several *Path Independence* axioms, (e.g. Plott (1973)) used in social choice theory.

Example 4 Let $t \equiv (\{a, b, c\}, \{d, e, f\}, \{g, h, i\})$ and let C satisfy BC. We have $C(t) = C(\{a, b, c\}, \{d, e, f\}, \{g, h, i\}) = C(C(a, b, c), C(d, e, f), C(g, h, i))$.

Reconsider the example of the online purchase of a television. Suppose the DM chooses a 42″ LCD television in this process. Consider the situation when the DM has the only option to choose from LCD televisions and she can choose any one from the following tree: $t_1 \equiv$ (25″ LCD, 42″ LCD). BC implies that the DM still chooses a 42″ LCD television. Also consider two separate cases. (i) The DM has to choose from Flat televisions and the options appear in the form of the following tree: $t_2 \equiv$ (14″ Flat, 22″ Flat, 29″ Flat). Suppose $C(t_2) = 22''$ Flat. (ii) The DM has to choose from Ordinary televisions and options appear in the form of the following tree $t_3 \equiv$ (21″ Ord., 29″ Ord., 17″ Ord.). Suppose $C(t_3) = 29''$ Ord. Now consider the same DM who has to choose from the following tree $t' \equiv$ (22″ Flat, 42″ LCD, 29″ Ord). BC implies the DM chooses the 42″ LCD television. Thus BC ensures that the choices are independent of the path in which the choices appear.

In order to introduce the next axiom, an additional definition is needed.

Definition 3 Let $t \equiv (X^1, X^2, \ldots, X^M)$, $t \in \Gamma$. For any alternative x_i^j (we note that x_i^j is the i-th alternative from the left in X^j) we define *the Partner Set* of x_i^j as the set $\{x_l^j; l \neq i$ and $x_l^j \in X^j\}$ and denote it by $\{\beta(x_i^j)\}$. A representative alternative from the Partner Set is $\beta(x_i^j)$, which is called a Partner alternative.

For an elementary binary tree, the Partner Set of an alternative is singleton. For instance, if $t \equiv \{a, b\}$, then $\{\beta(a)\} = \{b\}$ and $\{\beta(b)\} = \{a\}$.

Repetition Indifference (RI): For any tree $t \equiv (X^1, X^2, \ldots, X^M)$, if $C(t) = x_i^j$ then $C(t') = x_i^j$ where t' has been obtained from t by replacing any x_k^l ($x_k^l \neq x_i^j$) by $\beta(x_k^l)$ ($1 \leq l \leq M$), where x_m^n is the alternative that occupies m-the position from left in the set X^n.

This axiom requires that choice from a tree remains the same when some non-chosen alternative has been replaced by an alternative from its partner set. We provide examples:

Example 5 Let $t \equiv (\{a, b\}, \{c, d)\})$. Suppose that $C(t) = b$. Then RI implies that $C(t') = b$ where $t' \equiv (\{a, b\}, \{c, c)\})$.

Example 6 Let $t = (\{a, b, c\}, \{d, e, f\}, \{g, h, i)\})$. Suppose that $C(t) = b$. Then $C(t') = b$, where $t' \equiv (\{a, b, a\}, \{d, e, f\}, \{g, h, i)\})$.

RI is in the same spirit as the various Independence of Irrelevant Alternatives (IIA) axioms used in social choice theory. For instance, Sen's Property (Sen (1993)) α (or Contraction Consistency): for each pair of sets S and T, and for each $x \in S$, if $x \in C(T)$ and $S \subset T$, then $x \in C(S)$. RI requires choice from a tree to remain the same when some other alternative (i.e. an alternative which is not chosen from the tree) is replaced by an alternative from its partner set. This is justified because here choice is made sequentially- at each non-terminal node an offspring is chosen from the available ones, and ultimately an alternative at a terminal node is chosen. We observe that the partner set of an alternative (say, x) contains all other alternatives available for choice at the non-terminal node (call it n^*), from which the offspring containing x has emanated. Thus after DM has reached n^*, she has x and its partner set available to choose from. Consider choosing from a tree t in which n^* is a non-terminal node and $C(t) \neq x$. If we consider contraction consistency by replacing x, then it must be replaced by one of its partner alternatives because only partner alternatives are available as options from n^*. Otherwise the contraction may not be compatible with the choice problem. We explain with a simple example: consider purchasing a television online. Suppose initially DM needs to choose between black and white category (BW) and colored (C). Within BW, there are two types: flat screen (F-BW) and normal screen (N-BW) and similarly within C, there are two categories: (F-C) and (N-C). Suppose DM chooses a flat screen television in the colored category, i.e (F-C) is chosen. Now we consider checking contraction consistency of DM's choice function. To do so, if we replace (F) category under (BW) by a colored flat screen television (F-C), then we have fundamentally altered the structure of the initial choice problem by removing the colored vs BW distinction. But we can replace a (F-BW) by a (N-BW) under black and white category. This explains why we choose to replace an alternative by only one of its partner alternative.

We characterize all choice functions satisfying BC and RI. We first show that these conditions are independent.

Proposition 1 *Backward Consistency and Repetition Indifference are independent.*

Proof: (a) BC does not imply RI. Let $X = \{a, b, c\}$. Let C^* be defined on elementary binary trees as follows:

(i) $C^*(a, b) = b$; $C^*(b, a) = a$.

(ii) $C^*(b, c) = c$; $C^*(c, b) = b$.

(iii) $C^*(a, c) = a$; $C^*(c, a) = c$.

Choices from any arbitrary binary trees t are obtained by applying the choice function C^* recursively to elementary trees which are concatenated sequentially to form the tree t.

Note that choice function C^* satisfies BC by construction. But we show that this rule does not satisfy RI. Let $t \equiv (\{a, b\}, \{c, c\})$. Let $t_1 \equiv (\{a, b\})$; $t_2 \equiv (\{c, c\})$. Clearly $C^*(t) = C^*(t_1$ o $t_2)$. Applying the definition of C^* sequentially we get that $C^*(t) = c$. We now replace b by a in above tree and get $t' \equiv (\{a, a\}, \{c, c\})$. And $C^*(t') = a$, again using C^* for each elementary tree. But this contradicts RI.

(b) RI does not imply BC. Let $\hat{C}$: for any arbitrary binary tree $t \equiv (t_1 \text{ o } t_2 \text{ o} \ldots \text{o } t_k)$ or $t \equiv (X^1, \ldots, X^M)$, $C(t) = x^1_2$, where x^i_j: j-th alternative from the left in X^i. We claim that this rule satisfies RI. Let $t : \{(a, b), (c, d)\}$. Clearly $C(t) = b$. Therefore if any alternative (other than b) is replaced by its partner alternative, b is still the outcome in the choice from new tree. This argument holds for any arbitrary binary tree. Also observe $\hat{C}(\hat{C}(a, b), \hat{C}(c, d))=$ $\hat{C}(b, d) = d$. But $\hat{C}(\{a, b\}, \{c, d\})=b$. Clearly this rule violates BC.

■

It is important to compare our formal setting and axioms with those of Rubinstein and Salant (2006). In the latter, lists can be of arbitrary length. A tree can be thought of as an ordered n-tuples as we have seen in Section 2; however the Rubinstein and Salant (2006) axioms do not extend naturally to this setting. Let $t = \{(a, b), (c, d)\}$. Suppose $C(t) = a$. The Rubinstein and Salant (2006) axiom LIIA (List Independence of Irrelevant Alternatives) would require the choice to be a if any other alternative, say c, is removed. However c cannot be removed without destroying the binary structure of the set of alternatives. RI requires the choice to be a when c is *replaced* by d in t, i.e. when the binary structure is retained. PI (Partition Independence) axiom of RS is also distinct from BC. In particular, arbitrary partitions of the n-tuple describing a tree,s are precluded. For instance consider a tree $t = \{(a, b, c, d), (e, f)\}$. We cannot partition t into the sub-trees $\{(a, b), (c, d), (e, f)\}$. It follows that our problem (characterizing choice from trees satisfying BC and RI) is not equivalent to the Rubinstein and Salant (2006) problem of characterizing choice from lists satisfying PI or LIIA. Interestingly, our results are very similar to theirs for the case of binary trees but differ for higher order and mixed trees.

We now proceed to characterization results.

2.4 Choice from Binary trees

Our goal in this section is to show that choice functions from binary trees that satisfy the RI and BC axioms have a simple structure. We begin with some examples of choice functions that satisfy these axioms.

Example 7 Let $C : \Gamma^2 \to X$ be the PREF. For any $t \in \Gamma^2$, it chooses the maximal alternative from $X(t)$ according to the strict ordering $\succ$ over X. We note that if any alternative in $X(t)$ (apart from $C(t)$) is replaced by its Partner alternative and generate a new tree t', then $C(t)$ remains the maximal alternative according to $\succ$ in $X(t')$. Thus $C(t') = C(t)$ establishing that C satisfies RI. Suppose t is a concatenation of t_1 and t_2. Assume without loss of generality that $C(t) \in X(t_1)$. Then $C(t_1) = C(t)$ because $C(t)$ is the maximal alternative according to $\succ$ in $X(t_1)$ and $X(t_1) \subset X(t)$. Also $C(C(t_1), C(t_2) = C(C(t), C(t_2)) = C(t)$ because $C(t)$ is the maximal alternative in $\{C(t), C(t_2)\}$ according to $\succ$.

Example 8 Let $C : \Gamma^2 \to X$ be the SAT. Suppose $C(t) = x$ for some $t \in \Gamma^2$, i.e. x is the left-most alternative in t that is at least as good as the threshold alternative a^*. If an alternative (which is not x) is replaced by any of its partner alternatives, x remains the left-most alternative in the new tree (say t') that is at least as good as a^*. Therefore $C(t') = x$. It can be verified that C satisfies BC as well.

Given X and an ordering[3] $\succeq$ over X an *Indifference Set*, denoted by $I \subset X$ consists of alternatives which are indifferent to each other, i.e if $x, y \in I$, then $x \succeq y$ and $y \succeq x$. Given X and $\succeq$, we denote the set of all indifference sets by $I\{X, \succeq\}$. We note that $M(X, \succeq) \in I\{X, \succeq\}$.

Given X and an ordering $\succeq$ over X, an *admissible binary indicator function* δ is a map $\delta : X \to \{left, right\}$ such that $\delta(x) = \delta(y)$ whenever $x, y \in I_i$. For instance, if $x_1, x_2, \ldots, x_m \in M(X', \succeq)$, $X' \subset X$, then $\delta(x_1) = \delta(x_2) = \ldots = \delta(x_m)$ since $M(X', \succeq)$ is an indifference set.

Example 9 Let $X = \{a, b, c, d, e, f, g\}$ and let $\succeq$ be the following ordering: $b \succ c \sim a \sim d \succ e \sim f \succ g$. If δ is such that $\delta(c) = \delta(a) = \delta(d) = left$ and $\delta(e) = \delta(f) = right$, then it

[3]A binary relation is a *Ordering* if it is complete, reflexive and transitive, i.e. for any two alternatives $x, y \in X$, either $x \succeq y$, or $y \succeq x$. Also if $x \succeq y, y \succeq z$, then $x \succeq z$, i.e. $\succeq$ is transitive. Also if for $x \neq y$, $x \succeq y, y \succeq x$, then we say that x and y are indifferent.

is an admissible binary indicator function.

Definition 4 Let $\Gamma' \subset \Gamma$. A choice function $C : \Gamma' \to X$ is MIXED or MC, if there exists an ordering $\succeq$ over X and a binary indicator function δ (which may depend on $\succeq$) such that for any $t \in \Gamma'$:

(i) $C(t) = M(X(t), \succeq)$ whenever $M(X(t), \succeq)$ is unique.

(ii) If $M(X(t), \succeq)$ is not unique, then we have the following:

(a) if $\delta(x) = left$ for all $x \in M(X(t), \succeq)$, then $C(t)$ is the left-most alternative[4] in $M(X(t), \succeq)$.

(b) if $\delta(x) = right$ for any $x \in M(X(t), \succeq)$, then $C(t)$ is the right-most alternative in $M(X(t), \succeq)$.

Consider an arbitrary tree t. How does an MC choose an alternative for t? Note that an MC is associated with an ordering over X. Let this be called "binary-admissible" ordering. If a unique maximal alternative according to this ordering exists for $X(t)$, then the MC chooses this alternative. If the the maximal set (say M) contains more than one alternative, then the MC chooses the left-most or right-most alternative in M according as the admissible indicator function for M prescribes "left" or "right".

This choice function is described as being MIXED because it includes alternatives of both pure-preference based rules and procedural rules. The former occurs if $\succeq$ is anti-symmetric. The later occurs in the case where $\succeq$ is the universal indifference ordering, i.e. $x \sim y$ for all $x, y \in X$. Let $\mathcal{M}$ denote the set that contains all MCs.

[4]We have already demonstrated that a tree can be represented as an ordered sequence. Thus the left-most alternative in $M(X(t), \succeq)$ is the alternative in $M(X(t), \succeq)$ which lies to the left of the other alternatives in $M(X(t), \succeq)$ in t, when t is represented as an ordered sequence

Our main result is the following:

Theorem 1 *A choice function C, $C : \Gamma^2 \to X$ satisfies Backward Consistency and Repetition Indifference if and only if it is MIXED.*

Proof: The proof is provided in the Appendix I.

■

REMARK 2 We have already shown that a SAT from binary trees satisfies BC and RI. Following Theorem 1, a SAT is also an MC and thus a SAT belongs to $\mathcal{M}$. We can also directly show that a SAT is an MC. Let C be a SAT associated with strict ordering $\succ$ and threshold level alternative a^*. One can form a weak ordering $\succeq$ that induces two indifference sets of satisfactory and unsatisfactory alternatives. Also there exists an admissible binary indicator function δ, such that for all alternatives in satisfactory class δ is l, and for all unsatisfactory alternatives δ is r. This is the same argument as are given by Rubinstein and Salant (2006) to show that a satisficing choice function from lists can be presented with a weak ordering and an indicator function.

Similarly it is easy to check that if C_1 is a PREF from binary trees, then $C_1 \in \mathcal{M}$. Also if C_2 is a PROC from binary trees, then $C_2 \in \mathcal{M}$. In case C is a PREF, the weak ordering is the ordering (which is strict, according to the definition of PREF) according to which the choice is made from a tree by C. For a PROC, one can form a weak ordering such that all alternatives are indifferent according to the weak ordering and the δ for all the alternatives is l (if the PROC chooses $x(1;t)$ for any $t \in \Gamma^2$) or r (if the PROC chooses $x(2;t)$ for any $t \in \Gamma^2$). A PROC from binary trees is also a G-PROC from binary trees.

REMARK 3 Let C be G-PROC, but not PROC. We claim that $C \notin \mathcal{M}$, i.e. C is not MC.

To see this assume to the contrary that C is an MC. Suppose $\succeq$ is the binary-admissible ordering associated with C. Without loss of generality we assume that given any tree, C chooses 1^{st} branch at the initial node and 2^{nd} branch at all consecutive decision nodes. Consider $t \equiv \{(a,b),(c,d)\}$, $t' \equiv \{(b,a),(c,d)\}$, $t'' \equiv \{(c,d),(a,b)\}$ and $t''' \equiv \{(d,c),(a,b)\}$. Clearly $C(t) = b$, $C(t') = a$, $C(t'') = d$ and $C(t''') = c$. We observe that $X(t) = X(t') = X(t'') = X(t''') = \{a,b,c,d\}$. Thus $a \sim b \sim c \sim d$ and $a,b,c,d \in I_i$, where I_i is an indifference set. Let $\delta(a) = \delta(b) = \delta(c) = \delta(d) = l$. This implies that for $t \equiv \{(a,b),(c,d)\}$, $C(t) = a$, because a is the left-most alternative in $X(t)$. But this contradicts $C(t) = b$. Similarly we arrive at a contradiction if $\delta(a) = \delta(b) = \delta(c) = \delta(d) = r$. Thus a G-PROC that is not a PROC cannot be an MC.

2.5 Choice from Ternary and Higher order trees

In this section we characterize choice in ternary and higher order trees. We have noted that the results we have for binary trees are similar to the results in Rubinstein and Salant (2006). We show in this section that the results for choice in ternary or higher order trees are more subtle. In particular we show that there exists an ordering with the property that indifference is permitted for bottom-ranked alternatives. For any tree t, if $X(t)$ contains a non-bottom-ranked alternative, $C(t)$ is the (unique) maximal alternative according to this ordering. If $X(t)$ contains only bottom-ranked alternatives, then there are two possibilities (i) if there are more than two bottom-ranked alternatives, then the alternative chosen is $x(r;t)$ $(1 \leq r \leq k)$ (ii) if there are exactly two bottom-ranked alternatives and $X(t)$ consists of these two alternatives, then a relatively complicated tie-breaking rule can be used.

Below, we give an example of a choice function in ternary trees satisfying BC and RI which is not covered by the binary tree case.

Example 10 Let $X = \{a, b\}$. The choice function $\hat{C} : \Gamma^3(\{a, b\}) \to \{a, b\}$ is constructed as follows. Let

$\hat{C}(a, b, b) = a$, $\hat{C}(b, a, a) = a$.

$\hat{C}(a, a, b) = a$, $\hat{C}(b, b, a) = b$.

$\hat{C}(a, b, a) = a$, $\hat{C}(b, a, b) = b$.

For an arbitrary ternary tree t, $\hat{C}(t)$ is obtained by the concatenation operation and the C function defined above. It can be verified that this choice satisfies RI (BC is satisfied by construction).

We claim that $\hat{C}$ is not MC (as defined earlier). If $\hat{C}$ were MC, there would either be (i) a strict ordering over $\{a, b\}$, such that the maximal alternative from $X(t)$ according to this ordering would be chosen for any t; or (ii) a weak ordering R over $\{a, b\}$ with an indicator function δ, such that if aRb and bRa and for any t then $\hat{C}(t)$ would be the left-most (or the right-most) alternative in t if $\delta(a) = \delta(b) = l$ (or r). However $\hat{C}$ cannot satisfy (i) because $\hat{C}(a, a, b) = a$ and $\hat{C}(b, b, a) = b$. Nor can (ii) hold because $\hat{C}(a, b, b) = a$ and $\hat{C}(b, a, a) = a$, i.e. neither the left-most nor the right-most alternatives are chosen. It is also not the case that $\hat{C}$ chooses the "middle branch" at all trees. Therefore, this example demonstrates that choice functions satisfying BC and RI in ternary and higher order trees are qualitatively different from those in binary trees.

We now provide a characterization.

Definition 5 A weak-ordering $\succeq$ over X is *admissible* if $[x \succ y \implies \nexists z$, such that $x \sim z]$.

Indifference is permitted only for alternatives which are bottom-ranked in an admissible ordering. If $\succeq$ is an admissible preference ordering, let $BT(\succeq)$ denote the set of bottom-ranked alternatives in $\succeq$. We note that a strict preference ordering is an admissible ordering. The weak ordering $\succeq$ over X where $x \sim y$, for all $x, y \in X$ is also an admissible ordering. In this case $BT(\succeq) = X$.

Admissible preferences also appear in Ehlers (2002) in the context of a housing allocation model and preferences over houses. Ehlers (2002) justifies these preferences on the grounds that the DM does not possess (or does not have the incentives to acquire) information about the bottom-ranked alternatives and hence ranks them as indifferent to each other.

We now define choice functions for the special case of trees t where $X(t)$ has exactly two alternatives.

Definition 6 A choice function $C : \Gamma^k(\{a, b\}) \rightarrow \{a, b\}$ with $a, b \in X$, is *Pair-Consistent* (over $\{a, b\}$), if

(i) for any $t, t' \in \Gamma^k$, such that $t, t' \in \Gamma^k_e(\{a, b\})$ $(k \geq 2)$,$[C(t) \neq C(t')] \implies [\exists r^k(t, t') : 1 \leq r^k(t, t') \leq k$, such that $C(t) = x(r^k(t, t'); t)$ and $C(t') = x(r^k(t, t'); t')]$.

(ii) for any $t \in \Gamma^k$, such that $t \equiv t_1 \text{ o } t_2 \text{ o} \ldots \text{o } t_m$, $C(t) = C(C(t_1), C(t_2), \ldots, C(t_m))$.

Suppose C is a Pair-Consistent choice function and let $t, t' \in \Gamma^k$ be elementary k-ary trees such that $t, t' \in \Gamma^k_e(\{a, b\})$ and $C(t) \neq C(t')$. Then there exists a position in an elementary k-ary tree, say r^k $(r^k : 1 \leq r^k \leq k)$, such that $x(r^k; t) = C(t)$ and $x(r^k; t') = C(t')$. For an arbitrary $t \in \Gamma^k$, $C(t)$ is obtained by concatenation and the C function defined over elementary k-ary trees above.

Consider PROC(r) over k-ary trees with $X(t) = \{a, b\}$, i.e the r^{th} branch from the left is chosen at every node. This choice function is obviously Pair-Consistent.

We now define HYBRID choice functions which we will show, are the only choice functions which satisfy BC and RI for k-ary trees, $k \geq 2$.

Definition 7 A choice function $C : \Gamma^k \to X$, is HYBRID, if there exists an admissible preference ordering $(\succeq)$ over X, such that

(i) $C(t) = M(X(t), \succeq)$, whenever $X(t) \cap (X - BT(\succeq)) \neq \phi$.

(ii) $C(t) = x(r;t)$, $r : 1 \leq r \leq k$, if $X(t) \subset BT(\succeq)$ and $|BT(\succeq)| \geq 3$.

(iii) C is Pair-Consistent over $BT(\succeq)$, if $X(t) = BT(\succeq)$ and $|BT(\succeq)| = 2$.

For any $t \in \Gamma^k$, a HYBRID chooses the maximal alternative from $X(t)$, according to an admissible preference ordering, if the maximal alternative is unique. For all trees in Γ^k such that $X(t) \subset BT(\succeq)$ and $|BT(\succeq)| = 2$, C is Pair-Consistent over $BT(\succeq)$. If $|BT(\succeq)| \geq 3$, then for all trees in Γ^k, such that $X(t) \subset BT(\succeq)$, C is PROC(r). Let $\mathcal{H}$ denote the set that contains all HYBRIDs.

We now state the main result of this section:

Theorem 2 *A choice function $C : \Gamma^k \to X$ satisfies Backward Consistency and Repetition Indifference if and only if C is HYBRID.*

Proof: Provided in the Appendix.

■

REMARK 4 Let C^* be a PREF and let C^{**} be a PROC. Clearly $C^*, C^{**} \in \mathcal{H}$, i.e. PREF and PROC are HYBRID. In the case of PREF, the admissible ordering is a strict ordering while in the case of PROC, the admissible ordering $\succeq$ is such that $BT(\succeq) = X$.

REMARK 5 SAT is not HYBRID. To see this, assume to the contrary that SAT is HYBRID. Pick an arbitrary SAT (C) and let $\succ$ and d be the ordering and threshold respectively, associated with it. Let $t \equiv \{(e, f, g), (f, d, c), (a, b, b)\}$ and $t' \equiv \{(e, f, g), (c, d, c), (a, b, b)\}$. Suppose $a \succ b \succ c \succ d \succ x$, for any $x \in X - \{a, b, c, d\}$. Clearly, $C(t) = d$ and $C(t') = c$. Since C is a HYBRID by assumption, let R be the associated admissible ordering. Since $C(t) = d$, d must either be the maximal alternative in $X(t)$ according to R, or $X(t) \subseteq BT(R)$. Since $X(t) = X(t')$, if $M(X(t), R) = d$, then we must have $M(X(t'), R) = d$ implying $C(t') = d$. But this contradicts $C(t') = c$. Thus $X(t) \subseteq BT(R)$. Here $|X(t)| > 2$; according to the definition of HYBRID, we must have $C(t) = x(r; t)$, $C(t') = x(r; t')$, where $r : 1 \leq r \leq 3$. But this is false because $x(r; t) \neq d$, for any r with $1 \leq r \leq 3$.

We can also show directly that SAT does not satisfy RI. Let $t \equiv (\{e, e, g\}, \{f, d, c\}, \{a, b, c\})$. Here the first alternative from the left in the tree t that is not inferior to d, is d and thus $C(t) = d$. Suppose we replace f by its partner alternative c and we get $t' \equiv (\{e, e, g\}, \{c, d, c\}, \{a, b, c\})$. Applying RI, we get $C(t') = d$. But since C is SAT, it chooses the first alternative from the left in the tree t' that is not inferior to d, which is c. Thus we have a contradiction.

We end this section by emphasizing once again the difference in our results between the binary and ternary and higher order trees. In the former case, the admissible ordering permits indifference at all "levels" while in the latter case, indifference is permitted only at the bottom. Tie-breaking in the binary case is done by picking "left" or "right" in the maximal set in $X(t)$. In the latter case, if there are are at least three bottom-ranked alternatives, the chosen alternative is the alternative obtained by picking the r^{th} branch in the tree at every node. A more complicated tie-breaking rule can be used if there are exactly two bottom-ranked alternatives.

2.6 Choice in general trees

In this section we characterize choice in general trees, i.e. all trees in Γ. Observe that $\Gamma^k \subset \Gamma$, $\forall k \geq 2$. Since a choice function defined over Γ also defines a choice function over Γ^k, $k \geq 2$, we can use the results of the previous sections to characterize choices in the general case.

Definition 8 Let $W = \{\succeq\}_i$, $i = 1, 2, \ldots$ be a collection of orderings over X. The set W satisfies Mutual Consistency if $\nexists \succeq_i \in W$ and $x, y, z \in X$ such that following conditions hold: (i) $x \succ_i y \succ_i z$ or $x \succ_i y \succeq_i z$ or $x \succeq_i y \succ_i z$ and (ii) there is a sequence of orderings $\succeq_1, \succeq_2, \ldots, \succeq_k \in W$ and a sequence of alternatives $v_0, v_1, \ldots, v_k \in X$ such that $v_0 = z$, $v_k = x$ and $v_i \succeq_{i+1} v_{i+1}$, $i = 0, 1, \ldots, k-1$.

The mutual consistency property requires the orderings in W to satisfy a "non-reversality" property. Pick a pair of alternatives x, y and an ordering $\succeq_i$ such that $x \succ_i z$ with the property that there exists an alternative y which lies "between" x and y according to y. Then there should not exist an ordering $\succeq_j$ which "reverses" the ordering between x and z; more generally there should not exist sequences of orderings in W and alternatives in X through which the "reverse" ordering between x and z can be obtained. We illustrate this definition below.

Example 11 Let $X = \{a, b, c\}$. Let $W = \{\succeq_1, \succeq_2, \succeq_3\}$ where $a \succ_1 b \sim_1 c$, $b \succ_2 aP \succ_2 c$, $a \sim_3 b \sim_3 c$. W is not Mutually Consistent, since $a \succ_1 b \sim_1 c$ and $c \sim_3 a$.

Definition 9 A choice function $C : \Gamma \to X$ is General-Hybrid (G-HYBRID) if (i) there exists an MC choice function C^2 over Γ^2 (with an associated binary-admissible preference ordering R^2) (ii) there exist HYBRID choice functions $\{C^k\}_{k \geq 3}$ (with associated admissible orderings R^k, $k \geq 2$) (iii) the set $W = \{R^k, k \geq 2\}$ is mutually consistent. Moreover,

(i) for any $t \in \Gamma_e^k$, $C(t) = C^k(t)$, $k \geq 2$.

(ii) for any $t \in \Gamma$, such that $t \equiv t_1$ o t_2 o... o t_m, $C(t) = C(C(t_1), \ldots, C(t_m))$.

A G-HYBRID defines choice function C^k over k-ary trees ($k \geq 2$), where C^k is MC for $k = 2$ and HYBRID for $k \geq 3$. Thus for any $t \in \Gamma^k$, $C(t) = C^k(t)$, and for any mixed tree t, $C(t)$ is obtained by concatenation operation and C function defined over k-ary trees above. Also C^2 is associated with a binary-admissible ordering and C^k, $k \geq 3$ is associated with an admissible ordering, such that the set of these orderings is mutually consistent. Let $\mathcal{G}$ denote the set of all General-Hybrid choice functions.

We state the main result of this section:

Theorem 3 *A choice function $C : \Gamma \rightarrow X$ satisfies Backward Consistency and Repetition Indifference, if and only if C is G-HYBRID.*

Proof: Provided in Appendix.

■

REMARK 6 It is easy to show that PREF, PROC choice functions are G-HYBRID and thus belong to $\mathcal{G}$. But a G-PROC choice function which is not a PROC is not G-HYBRID. This follows from the same argument which establishes that a G-PROC, but not a PROC is not MC over Γ^2. Also a SAT choice function is not a G-HYBRID. To see this, assume to the contrary that SAT is G-HYBRID. Pick an arbitrary SAT (C) defined over Γ. Clearly C is SAT over k-ary, ($k \geq 2$) also. On the other hand since C is G-HYBRID by assumption, C is also HYBRID over k-ary trees ($k \geq 3$). This is because a G-HYBRID choice function defines a HYBRID choice function for $k \geq 3$. Thus C is a HYBRID as well as SAT over k

-ary trees ($k \geq 3$). But we have shown earlier that a SAT choice function from k-ary trees ($k \geq 3$) is not HYBRID. Thus we arrive at a contradiction.

2.7 Conclusion

This chapter characterizes choice functions when the DM encounters the set of alternatives in the form of a tree. The choice functions are assumed to satisfy two behavioral axioms- backward consistency and repetition indifference. These two axioms are shown to be independent of each other. We characterize the choice functions from binary trees followed by the characterizations of higher order and general trees.

We note that several questions remain unanswered. Other behavioral axioms can be imposed on the choice functions and characterization results can be obtained for the same. Apart from the axiomatic approach, non-cooperative game theoretic frameworks can also be interesting. In case the trees resulting from the decision problems are large or are unknown, the DM can have a belief over expected outcomes at each node and can update it along the process. An optimal search problem can be of interest in this respect. We hope to address some of these issues in future work.

Chapter 3

Implementation in undominated strategies

3.1 Introduction

Implementation theory aims to characterize the outcomes of group decision-making processes under various information structures. The group's collective objectives are specified by a social choice correspondence that selects a set of alternatives from the available set in every possible "state of the world". Implementation theory attempts to structure the interactions amongst the agents by designing a game-form such that in every state, equilibrium actions of agents according to some pre-specified equilibrium notion, lead to outcomes that are socially desirable, i.e. belong to the image of the social correspondence in that state.

The literature on implementation is vast and considers various equilibrium notions. In the complete information model, all agents are assumed to know the state while the mechanism designer does not. A natural notion of equilibrium in this context is Nash equilibrium (Maskin (1999)). Other notions that are consistent with this information setting and have been studied include the iterated eliminated of weakly undominated strategies (Moulin (1979)), sub-game perfect Nash equilibrium and various other Nash equilibrium refinements.[1]

In this chapter, we consider a private information setting. Each agent has private information about her type and a state of the world is a collection of types, one for each agent. In what follows, an agent's type will be her *preference ordering* over a finite set of alternatives. Thus each agent knows her own preferences but is ignorant of the preferences of others. The mechanism designer has no information regarding the state. In such a model, equilibrium notions such as the iterated elimination of dominated strategies or Nash implementation are inappropriate. For instance, an agent will be unable to predict the strategies that will be eliminated by other agents (since they depend on the types of the other agents which are unobservable).

Natural notions of equilibria under such an information structure are dominant strategies and Bayes-Nash equilibrium. If the dominant strategy notion is used, then there exists a message at each state of the world that dominates all other messages. On the other hand, a strategy is Bayes-Nash equilibrium if unilateral deviations are not profitable in terms of expected pay-offs where these expectations are computed based on prior beliefs of the other agents. The significant advantage of using dominant strategy concept is that the mechanism designed does not depend on the prior beliefs of the agents. However dominant strategy implementation is a very demanding requirement. For instance, according to the Gibbard-Satterthwaite (Gibbard (1973), Satterthwaite (1975)) Theorem, the only social choice functions that can be implemented in a complete domain of preferences are *the dictatorial* (provided that the range of the social choice function has a range of at least three alternatives).

[1]See Corchòn (2009), Jackson (2001) and Serrano (2004) for surveys of this literature.

In this chapter we will consider another natural prior-free notion of equilibrium that is consistent with the private information assumption. This is the *single round elimination of weakly dominated strategies*. We will refer to implementation with this solution concept as implementation in undominated strategies. From a decision theoretic perspective, implementation in undominated strategies has many of the strengths of dominant strategy implementation. However as we shall see, implementation in undominated strategies is not as restrictive as dominant strategy implementation provided we consider social choice correspondences rather social choice functions.

Implementation in undominated strategies was first introduced and studied in an important paper by Jackson (1992). The paper proved a surprising and powerful result: *all* social choice correspondences can be implemented in undominated strategies. The mechanism constructed involves infinite strings of strategies each of which dominates the earlier one. The paper interpreted this as a weakness in the solution concept and proposed the following restriction on admissible mechanisms: for each state each weakly dominated strategy must be dominated by an undominated strategy. The paper refers to such mechanisms as *bounded* mechanisms.

A natural question is the following: what are the social choice correspondences that can be implemented in undominated strategies by bounded mechanisms Unfortunately, this appears to be an exceedingly difficult question to answer. The only available general result available so far is in Jackson (1992), which shows that a social choice function is implementable over a complete domain of preferences if and only if it is dictatorial. In this chapter we shall investigate some special aspects of the implementation in undominated strategies. Throughout the chapter we will consider the mechanisms in which the message spaces are finite. Note that such a mechanism is bounded. However the converse is not true (see Jackson (1992) for such an example). Borgers (1991) and Yamashita (2010) also use finite message spaces and provide additional justification for their use.

We first provide a complete characterization of all implementable SCCs for the case where there is a single agent and three alternatives, where the agent has an anti-symmetric ordering over the alternatives. We show that it is important to consider the mechanisms where there is a dummy agent who sends dummy messages. We provide a necessary and sufficient condition called the "Neighborhood Flip (NF) condition" for implementation. This condition can be thought of as a generalization of the standard monotonicity condition that is necessary for implementation in dominant strategies. We also highlight the differences with implementation in dominant strategies by showing that a wider class of SCCs can be implemented.

Next, we investigate implementation (for an arbitrary number of players and alternatives) in a more restricted class of mechanisms which we call "covered mechanisms". These are mechanisms where every message of an agent is undominated for at least at one preference ordering of the agent. In general a mechanism that implements a SCC may contain a message for an agent that is always dominated. We show that if a SCC is implementable, then there exists a covered mechanism that implements the SCC weakly, i.e. implements a sub-correspondence of the SCC. We identify a condition which we call "General Monotonicity (GM)" that is necessary for implementation by covered mechanisms. The GM condition builds on the NF condition and also implies the strategy-resistance condition that Jackson shows is necessary for implementation. We show that the Pareto correspondence for two agents and three alternatives fails the GM condition and hence cannot be implemented by a covered mechanism. However, we show that there exists a complicated uncovered mechanism that implements this SCC. These results further underscore the earlier point that a full characterization of implementable SCCs is very difficult.

It is well-known that SCCs which pick unions of top-ranked alternatives of agents at every preference profile, are implementable.[2] A drawback of these SCCs is that they overlook "compromise" alternatives. Consider a case where there are two agents and a hundred alternatives. Alternative a is ranked first by agent 1 but hundredth by agent 2. Alternative b is ranked first by agent 2 but hundredth by agent 1. On the other hand, alternative c is ranked second by both agents. It seems very natural for a SCC to pick c at this profile while the union of top-ranked alternatives SCC will pick the set $\{a, b\}$. Borgers (1991) investigates the implementability of SCCs which pick compromise alternatives. In particular an alternative is a compromise at a profile if is not first-ranked by an agent but is (Pareto) efficient and a SCC satisfies the *Compromise Axiom* if there exists a preference profile where the SCC picks *only* compromises. The chapter proves that if there are either two agents or three alternatives, then there doest not exist an efficient, implementable SCC satisfying the compromise axiom.

We extend and refine the Borgers' result (Borgers (1991)) in a number of ways by proving a number of possibility as well as impossibility results. In particular we show the following:

- There does not exist an *efficient*, implementable SCC for any number of agents and alternatives satisfying the compromise axiom at preference profile that are "near unanimous", i.e. preference profiles where all but one agents agree on the top-most alternative.
- There does not exist a *neutral*, implementable SCC for two agents and an arbitrary number of alternatives satisfying the compromise axiom. A neutral SCC is one that treats alternatives symmetrically.
- There does not exist a *unanimous*, implementable SCC for two agents and an arbitrary number of alternatives satisfying the compromise axiom and the additional axiom of

[2] "Integer game" mechanisms of the type used in complete information implementation theory (Maskin (1999)) can be used for the purpose.

minimality. A unanimous SCC uniquely picks the alternative that is top-ranked by all agents at a preference profile. Each alternative in the image set of a minimal SCC at a profile is the maximal alternative in that set for some agent at that profile.

- There exists a *unanimous*, implementable SCC for two agents and three alternatives satisfying the compromise axiom.
- There exists an efficient, implementable SCC satisfying the compromise axiom for an arbitrary number of agents and alternatives in the special case where *a single agent* has only *one* preference ordering (i.e. a single "type").

Finally, we investigate the structure of SCCs satisfying the property of *strategy resistance* that Jackson (1992) introduced as a necessary (but not sufficient) condition for the implementablity of a SCC. We provide a general characterization of minimal (as defined previously) strategy-resistant SCCs. We show that such SCCs have a particularly simple form in the case where there are two agents but an arbitrary number of alternatives.

The chapter is organized as follows: In next section we set up the model, introduce definitions and state some preliminary results. Section 3 considers the single agent and three alternatives environment and the NF condition while Section 4 concerns covered mechanisms. Sections 5 deals with implementation with the Compromise Axiom. The final section concludes.

3.2 The Model and Preliminaries

Let X denote a finite set of alternatives and $N = \{1, 2, ..., n\}$, the set of agents or players. Let $|X| = m$. We shall assume that each agent i has a linear (or anti-symmetric) preference ordering P_i over the alternatives of X. If xP_iy then "x is strictly better than y". In general, for any two alternatives $x, y \in X$, we will write xR_iy if either xP_iy or $x = y$.

Let $\mathcal{P}$ denote the set of all orderings over the alternatives of X. An admissible domain for every agent i denoted by $\mathcal{D}_i$ is a subset of $\mathcal{P}$. Let $\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times .... \times \mathcal{D}_n$.

An *environment* is a collection $(N, X, \mathcal{D})$.

Let P denote a preference profile which is the n-tuple $(P_1, P_2, ..., P_n)$, where $P \in \mathcal{D}$

For all orderings P_i we shall denote the k-th ranked alternative by $t^k(P_i)$.

A *social choice correspondence* (SCC) S is a map $S : \mathcal{D} \to 2^X \setminus \emptyset$. Thus a SCC associates a non-empty subset of X with every profile $P \in \mathcal{D}$. Note that a SCC can be defined over an arbitrary domain.

A sub-correspondence S' of a SCC S is a map $S' : \mathcal{D} \to 2^X \setminus \emptyset$ such that $S'(P) \subseteq S(P)$ for a preference profile $P \in \mathcal{D}$.

A *mechanism* or *game-form* is an $N+1$ tuple $\Gamma = \langle M_1, M_2, ...M_n; g\rangle$ where M_i, $i = 1, 2, ...n$ is the message set of player i and $g : M_1 \times M_2 \times ... \times M_n \to A$ is an outcome function.

Let a mechanism $\Gamma = (M, g)$ be given. We say that $m_i \in M_i$ *weakly dominates* $m'_i \in M_i$ at P_i, if $g(m_i, m_{-i})R_ig(m'_i, m_{-i})$ for all $m_{-i} \in M_{-i}$ with $g(m_i, m_{-i})P_ig(m'_i, m_{-i})$ for at least one $m_{-i} \in M_{-i}$. If there is a strategy $m^*_i \in M_i$ that weakly dominates all other strategies in M_i at P_i, then we say that m^*_i a *dominant strategy* at P_i. Let $D_i(\Gamma, P_i)$ denote the set of all messages of agent i which are not weakly dominated at P_i. Also let $H_i(\Gamma, P_i)$ denote the set of dominant strategies of agent i at P_i. Since P_i is a strict ordering $H_i(\Gamma, P_i)$ can consist of at most one alternative. A strategy m_i is *undominated* at P_i, if there is no strategy in M_i that weakly dominates m_i at P_i.

For any preference profile P, let $D(\Gamma, P) \equiv D_1(\Gamma, P_1) \times ... \times D_n(\Gamma, P_n)$ denote the set of undominated strategy profiles. Also, for any profile P, let $H(\Gamma, P) \equiv H_1(\Gamma, P_1) \times ... \times H_n(\Gamma, P_n)$ denote the set of dominant strategy profiles and an alternative of $H(\Gamma, P)$ is denoted by $h(\Gamma, P)$. Note that $H(\Gamma, P)$ is a singleton for all P.

Definition 10 The SCC S is *implementable in dominant strategies* if there exists a mechanism $\Gamma = \langle M_1, M_2, ...M_N; g\rangle$ such that $S(P) = g(h(\Gamma, P))$ for all $P \in \mathcal{D}$. Note that if S is implementable in dominant strategies, then $S(P)$ is singleton-valued at all profiles P.

We assume that the agents know their own preference orderings or "types". When dominant strategies exist in the mechanism, it is natural to assume that agents will play them. This strategy is a best-response for an agent to *any* belief that the agent may have regarding the the choice of strategies of other agents.

Include the necessary and sufficient condition for implementation in dominant strategies.

Definition 11 The SCC S is *implementable in undominated strategies* if there exists a mechanism $\Gamma = \langle M_1, M_2, ...M_N; g\rangle$ such that $S(P) = g(D(\Gamma, P))$ for all $P \in \mathcal{D}$. Note that $S(P)$ may be multi-valued.

We say that a SCC S is *weakly* implementable in undominated strategies if there exists an implementable sub-correspondence S' of S.

The set of all undominated strategies at P_i for an agent contain only those strategies that survive elimination of weakly dominated strategies at P_i. The solution concept of implementation in undominated strategies is weaker than the implementation in dominant strategies. It requires that the outcome set in the game form at a state of the world coincides with the socially desired outcome set when all agents play the undominated strategies at that state of the world. We provide an example to highlight the difference with the implementation in dominant strategies.

Example 12 There is a single agent and the set of alternatives $X = \{a, b, c\}$. Let P_i, $i = 1, 2, .., 6$, denote the six possible preference orderings over X such that aP_1bP_1c, aP_2cP_2b, cP_3aP_3b, cP_4bP_4a, bP_5cP_5a and bP_6aP_6c. Let S be the following SCC: $S(P_1) = \{a, b\}$, $S(P_2) = \{a, b\}$, $S(P_3) = \{a, b, c\}$, $S(P_4) = \{a, b, c\}$, $S(P_5) = \{b\}$ and $S(P_6) = \{b\}$.

We claim that S cannot be implemented in dominant strategies. Suppose it can. Let (M, g) be a mechanism that implements it. We allow the mechanism designer (who has no preferences) to send an arbitrary number of *dummy* messages. (We are allowing for the case where the designer sends a single or no message.) Let m_i be the message which is dominant at P_4. Since $a \in S(P_4)$, it must be that m_i produces a against some dummy message. But the message (say, m_i') that is dominant at P_5 produces only b for all dummy messages since $S(P_5) = \{b\}$. However since bP_4a, m_i is not a dominant at P_4.

However, it can be verified that the mechanism (Γ) below implements S in undominated strategies. In this mechanism, the mechanism designer sends three dummy messages, α_1, α_2, α_3, and while the single agent can send one of two messages m_1 and m_2. The outcome function is described below.

	m_1	m_2
α_1	a	b
α_2	b	b
α_3	c	b

Table 3.1: Mechanism Γ implementing S

We note that $m_1, m_2 \in D_i(\Gamma, P_1^1)$, $m_1 \in D_i(\Gamma, P_2^1)$, $m_1 \in D_i(\Gamma, P_3^1)$, $m_1, m_2 \in D_i(\Gamma, P_4^1)$,

	P_1		P_2	P_3	P_4		P_5	P_6
	m_1	m_2	m_1	m_1	m_1	m_2	m_1	m_1
α_1	a	b	a	a	a	b	b	b
α_2	b	b	b	b	b	b	b	b
α_3	c	b	c	c	c	b	b	b

Table 3.2: Description of undominated strategies in Γ implementing S

$m_1 \in D_i(\Gamma, P_5^1)$, $m_1 \in D_i(\Gamma, P_6^1)$. The example shows that a wider class of SCCs can be implemented in undominated strategies.

In our study of implementation in undominated strategies, it is important to introduce a qualification. Jackson (1992) has shown that it is possible to implement *any* SCC in undominated strategies if no restrictions are imposed on the mechanism. However the canonical mechanism used for this result involves an infinite string of strategies each of which dominates the earlier strategy without having any undominated strategy. This is unsatisfactory because an agents' best-response according to the criteria specified, does not exist. Jackson (1992) therefore introduces the restriction of "bounded-ness" on admissible mechanisms.

Definition 12 A mechanism Γ is *bounded* if, for all $m_i \in M_i$, if $m_i \notin D_i(\Gamma, P_i)$ then there exists $\hat{m}_i \in D_i(\Gamma, P_i)$ and $\hat{m}_i$ weakly dominates m_i at P_i.

Thus the bounded-ness property puts the restriction that there cannot be an infinite sequence of dominated strategies without any undominated strategy at the "end". We restrict attention only to bounded mechanisms. In fact we make the stronger assumption throughout this chapter that the message spaces for all agents are *finite*. This ensures that every agent i will have at least one undominated strategy at any P_i. Thus, a finite mechanism is bounded although the converse is false. Throughout the chapter we shall say that S is implementable if S is implementable in undominated strategies by a finite mechanism.

Jackson (1992) provides a necessary condition that an implementable SCC must satisfy. Jackson (1992) calls this condition "strategy-resistance".

Definition 13 Strategy-resistant SCC:A SCC S is *strategy-resistant* if for all i, P, and P_i^* and for each $b \in S(P_i^*, P_{-i})$ there exists $a \in S(P)$ such that aR_ib.

Equivalently, S is strategy-resistant if for all i, P, P_i^*, we have $\tau(P_i, S(P_i, P_{-i}))R_i\tau(P_i, S(P_i^*, P_{-i}))$ where $\tau(P_i, B)$ is the maximal alternative in B according to P_i. Thus a strategy-resistant SCC is not manipulable by any agent by misrepresenting her ordering.

We now introduce some general properties of SCCs that we will use later.

Definition 14 A SCC $S : \mathcal{D} \to 2^X \setminus \emptyset$ satisfies *neutrality* if following holds: let λ be a bijection $\lambda : X \to X$ and let $P, P' \in \mathcal{D}$ such that xP_iy implies that $\lambda(x)P_i'\lambda(y)$, for all $x, y \in X$, for all $i \in N$. Then $S(P) = \lambda(S(P'))$.

If a SCC satisfies neutrality, then it does not discriminate between alternatives.

Definition 15 Pareto Efficiency: A SCC $S : \mathcal{D} \to 2^X \setminus \emptyset$ satisfies *Pareto efficiency* if for no $P \in \mathcal{D}$, there are $x, y \in X$ such that xP_iy for all $i \in N$ and $y \in S(P)$.

An alternative is Pareto dominated at a profile if there exists another alternative which is strictly preferred to this alternative by all agents. A SCC is Pareto-efficient if it never picks a Pareto dominated alternative at any profile.

Definition 16 A SCC $S : \mathcal{D} \to 2^X \setminus \emptyset$ satisfies *unanimity* if for all $P \in \mathcal{D}$ and $a \in X$ such that $t^1(P_i) = a$ for all $i \in N$, we have $S(P) = a$.

A unanimous SCC always respects consensus if it exists. Thus it always picks an alternative ranked first by all agents in a profile. Clearly a Pareo efficient SCC satisfies unanimity. Note that the properties of neutrality, unanimity and Pareto efficiency are familiar in the literature and these are discussed in Moulin (1983).

3.3 The single agent and three alternatives environment

In this section, we consider an environment E^0 with a single agent and three alternatives, i.e. $E^0 = (\{i\}, X, \mathcal{P})$ where $|X| = 3$. We will characterize the class of SCCs that can be implemented in this environment.

As we have seen earlier, an important issue is the set of dummy messages that the designer can send. A natural case to consider is the one where the designer is completely passive. It is best to think of this case as the one where the agent sends a message which leads to an outcome. Equivalently one can think of the designer sending a single dummy message i.e. $|M_{-i}| = 1$. We shall call these mechanisms *simple mechanisms*.

The characterization of an implementable SCC with simple mechanisms is trivial and is given below. We first introduce a piece of notation that will be used extensively in this section.

Let $x, y \in X$ and $P_i, P_j \in \mathcal{P}$. We say (x, y) is a *neighborhood flip* for P_i, P_j, if

1. xP_iy, yP_jx and there is no $z \in X$ such that xP_izP_iy and yP_jzP_jx;
2. for any alternative $w \in X$, such that wP_ix, we have wP_jy and for any alternative $q \in X$ such that yP_iq, we have xP_jq.

Thus x and y are contiguous in P_i and P_j. Their rankings are reversed in these two orderings while those of all other alternatives remain unchanged.

Proposition 2 *A SCC S is implementable by a simple mechanism if and only if it satisfies the following condition. If (x, y) is a neighborhood flip between P_i and P_i' and $S(P_i) \neq S(P_i')$, then $S(P_i) = x$ and $S(P_i') = y$.*

The proof of Proposition 2 is straightforward and therefore omitted. An implementable SCC must be singleton-valued at all orderings. In addition, if P_i and P_i' have a neighborhood flip, then $S(P_i)$ and $S(P_i')$ are the same or $(S(P_i), S(P_i'))$ is the neighborhood flip between P_i and P_i'. Example 13 provides an example of a SCC satisfying the condition.

We now consider the general case where the mechanism designer can send multiple dummy messages which together with the message sent by the agent determines the outcome. We show that the *Neighborhood Flip* condition defined below is necessary and sufficient for implementation.

Definition 17 Let (x, y) be a neighborhood flip between P_i and P_i'. A SCC S satisfies the *Neighborhood Flip (NF)* condition if for all $a \in S(P_i) \setminus S(P_i')$,

1. either $a = x$ and $y \in S(P_i')$ or
2. yP_ia and $x, y \in S(P_i)$, $y \in S(P_i')$.

The NF condition says the following. Suppose that $a \in S(P_i)$ and $a \notin S(P_i')$, where P_i and P_i' have a neighborhood flip. Then there exists an alternative $b \in S(P_i')$ such that b is strictly preferred to a at P_i'. If a is not strictly preferred to b at P_i, then both x, y must belong to $S(P_i)$ and y must belong to $S(P_i')$.

We give two examples of SCCs satisfying the NF condition and one which does not. Let $X = \{a, b, c\}$. Let P_i, $i = 1, 2, .., 6$, denote the possible preference orderings aP_1bP_1c, aP_2cP_2b, cP_3aP_3b, cP_4bP_4a, bP_5cP_5a and bP_6aP_6c.

Example 13 Let S be the following SCC: $S(P_1) = \{a\}$, $S(P_2) = \{a\}$, $S(P_3) = \{a\}$, $S(P_4) = \{b\}$, $S(P_5) = \{b\}$ and $S(P_6) = \{b\}$. S satisfies the NF condition and is implementable by a simple mechanism.

Example 14 Let S be the following SCC: $S(P_1) = \{a, b\}$, $S(P_2) = \{a, b\}$, $S(P_3) = \{a, b, c\}$, $S(P_4) = \{a, b, c\}$, $S(P_5) = \{b\}$ and $S(P_6) = \{b\}$. Clearly S satisfies the NF condition. In Example 12 we have illustrated the mechanism that implements S.

Example 15 Let S be the following SCC: $S(P_1) = \{a, b\}$, $S(P_2) = \{a, b\}$, $S(P_3) = \{a, b, c\}$, $S(P_4) = \{a, b, c\}$, $S(P_5) = \{b\}$ and $S(P_6) = \{b, c\}$. We note that (b, a) is a neighborhood flip between P_6, P_1 and $c \in S(P_6) \setminus S(P_1)$. But $b \notin S(P_6)$. Thus S does not satisfy the NF condition.

We have the following characterization result.

Theorem 4 *A SCC S is implementable in environment E^0 if and only if it satisfies the NF Condition.*

Proof: See Appendix II for the proof. ■

We illustrate the algorithm and the argument in the proof for the SCC described in Example 16

Example 16 We provide an example to illustrate the algorithm for designing the implementing mechanism in case (B). Let P_i, $i = 1, 2, .., 6$, denote the six possible preference orderings over X such that aP_1bP_1c, aP_2cP_2b, cP_3aP_3b, cP_4bP_4a, bP_5cP_5a and bP_6aP_6c. Let S be the following SCC: $S(P_1) = \{a, b\}$, $S(P_2) = \{a, b, c\}$, $S(P_3) = \{a, c\}$, $S(P_4) = \{b, c\}$, $S(P_5) = \{b, c\}$ and $S(P_6) = \{a, b, c\}$. We note that $b \in S(P_2) \setminus S(P_3)$, $c \in S(P_6) \setminus S(P_1)$ and $a \in S(P_5) \setminus S(P_4)$. Following the algorithm described above we construct the mechanism implementing S in Step 1, 2 and 3.

The first, second and third blocks from Steps 1, 2 and 3 are given in Tables 3.3, 3.4, 3.5 and the augmented mechanism that implements S is given in Table 3.6 .

	P_1	P_2		P_3	P_4	P_5	P_6
	m_6	m_1	m_1'	m_2	m_3	m_4	m_5
α_1	b	b	c	c	c	b	b
α_2	a	a	c	c	c	c	a
α_3	a	a	a	a	b	b	b

Table 3.3: The first block from Step 1

	P_1	P_2	P_3	P_4	P_5	P_6	
	m_2	m_3	m_4	m_5	m_6	m_1	m_1'
α_4	a	a	c	c	c	c	a
α_5	a	a	a	b	b	b	a
α_6	b	c	c	c	b	b	b

Table 3.4: The second block from Step 2

	P_1	P_2	P_3	P_4	P_5		P_6
	m_5	m_4	m_3	m_2	m_1	m_1'	m_6
α_7	a	a	c	c	a	c	a
α_8	b	c	c	c	b	c	b
α_9	a	a	a	b	b	b	b

Table 3.5: The third block from Step 3

	P_1^1	P_1^2		P_3^1	P_4^1	P_5^1		P_6^1	
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
α_1	b	b	c	c	c	b	b	b	b
α_2	a	a	c	c	c	c	c	a	a
α_3	a	a	a	a	b	b	b	b	b
α_4	a	a	a	c	c	c	c	c	a
α_5	a	a	a	a	b	b	b	b	a
α_6	b	c	c	c	c	b	b	b	b
α_7	a	a	a	c	c	a	c	a	a
α_8	b	c	c	c	c	b	c	b	b
α_9	a	a	a	a	b	b	b	b	b

Table 3.6: The mechanism implementing S

3.4 Covered and uncovered mechanisms

In this section we investigate implementation of SCCs by a more restricted class of mechanisms. An implementing mechanism may include a message that it is dominated at *all* $P_i \in \mathcal{D}_i$, i.e. this message is *"never"* undominated. Yamashita (2010) has termed such messages *nuisance* messages. In this section we investigate the role played by such messages in implementation.

We define the class of mechanisms that do not contain nuisance messages.

Definition 18 A mechanism $\gamma = (M, g)$ is covered if for each agent $i \in N$ and for each message $m_i \in M_i$, there exists $P_i \in \mathcal{D}_i$ such that m_i is undominated at P_i.

A mechanism $\Gamma = (M, g)$ is covered if all messages (for every agent) are undominated in at least one preference ordering. According to the next Proposition attention can be restricted to the covered mechanisms provided the solution concept is weakened from implementation to weak implementation.

Proposition 3 *If a SCC is implementable then there exists a covered mechanism that weakly implements it.*

Proof: We provide the proof for $n = 2$, but it can be generalized to an arbitrary number of agents. Let $\Gamma = (M, g)$ be a finite mechanism that implements S. For an arbitrary agent i, let $M_i^* = \{m_i^* \in M_i |$ there is no $P_i \in \mathcal{D}_i$, such that $m_i^* \in D_i(\Gamma, P_i)\}$, $i = 1, 2$. Thus, $M_i^* \subset M_i$ denotes the set of messages for the agent, so that all messages in this set are *never* undominated at an ordering for the agent. Let $\tilde{M}_i = M_i \setminus M_i^*$. For an arbitrary message in $\tilde{M}_i$, there exists $P_i \in \mathcal{D}_i$ such that the message is undominated at P_i. We can say that $\tilde{M}_i$ contains all *no nuisance* messages for i in Γ.

Let $m_1^* \in M_1^*$. We construct a mechanism $\Gamma' = (M', g)$ such that $M_2 = M_2'$ and $M_1' = M_1 \setminus \{m_1^*\}$. Thus the mechanism Γ' is obtained from Γ by eliminating the message m_1^* from the message space for the agent 1 without affecting the message space for the other agent and the outcome function g. We analyze the impacts of this elimination on the other messages in the mechanism, i.e. whether an undominated message at an arbitrary $P_2 \in \mathcal{P}$ in the mechanism Γ remains undominated at P_2 in the mechanism Γ' and whether an undominated message at an arbitrary $P_1 \in \mathcal{P}$ in the mechanism Γ remains undominated at P_1 in the mechanism Γ'.

Let $P_1 \in \mathcal{D}_1$ be an arbitrary preference ordering of the agent 1. There must exist a message, say $m_1' \in M_1'$ so that $m_1' \in D_1(\Gamma, P_1)$ and m_1' weakly dominates m_1^* at P_1 in Γ. This is because $m_1^* \in M_1^*$ and thus m_1^* is never undominated. Clearly the elimination of m_1^* from Γ will leave m_1' undominated at P_1. All other messages in $D_1(\Gamma, P_1)$ which do not weakly dominate m_1^* at P_1 in Γ also remain undominated at P_1. Also the elimination of m_1^* from Γ does not convert any message m_1'' into an undominated message at $P_1^* \in \mathcal{D}_1$ in Γ', where $m_1'' \notin D_1(\Gamma, P_1^*)$. Thus after the elimination of m_1^* from Γ also, $\tilde{M}_1$ remains the set of all "no nuisance" messages for the agent 1 in Γ'.

Thus if we eliminate a message of the agent 1 from M_1 in Γ, such that the message is weakly dominated at all $P_1 \in \mathcal{D}_1$, it does not affect the set of messages for the agent 1 which are undominated at a preference ordering of the agent 1.

Next we consider the impact of the elimination of m_1^* on the set of messages for the agent 2, which are undominated at a preference ordering of agent 2, i.e. $\tilde{M}_2$. Let $m_2'' \in \tilde{M}_2$ where $m_2'' \in D_2(\Gamma, P_2)$ for $P_2 \in \mathcal{D}_2$. There can be two possible effects of the elimination of $m^*{}_1$:

(a) There exists $m_2''' \in D_2(\Gamma, P_2) \subset \tilde{M}_2$ such that $g(m_1, m_2''') P_2 g(m_1, m_2'')$ for all $m_1 \in M_1 \setminus \{m_1^*\}$ and $g(m_1^*, m_2'') P_2 g(m_1^*, m_2''')$. Clearly the elimination of m_1^* from the mechanism allows m_2''' to dominate m_2'' at P_2. But m_2''' remains undominated at P_2 in the mechanism Γ', i.e. after the elimination of m_1^* from Γ. Thus although $\tilde{M}_2$ contains m_2'', but $\tilde{M}_2{}'$ does not contain m_2''. Also a message m_2^* that belongs to M_2^* cannot become undominated at P_2 after the elimination of m_1^*. This is because if a message is dominated by another undominated message at P_2 in Γ, even after eliminating the message m_1^* it remains dominated by the later message at P_2. It can be easily verified.

(b) m_2'' remains undominated at P_2 in Γ'. Thus if we delete a message which is not undominated at any possible preference ordering of an agent from a mechanism, the set of messages which are undominated at some preference ordering of that agent remains the same. It may contract the set of messages which are undominated at some possible preference ordering of the other agent. But the deletion of the message can never expand the set of messages which are undominated at some possible preference ordering of the other agent.

The implication of eliminating a message which is never undominated in a mechanism is that no message which was not an undominated message, can become undominated at a preference ordering for an agent. Thus the mechanism obtained after deleting the message can implement a sub-correspondence of the SCC implemented by the former mechanism. Since Γ implements S, Γ' implements a sub-correspondence of S and hence it weakly implements S. If we keep on deleting all messages from M_1^* and M_2^* then we arrive at a mechanism Γ'' so that $\tilde{M_i}'' \subseteq \tilde{M}_i$, $i = 1, 2$. Thus Γ'' weakly implements S.

■

In the last section we have showed that the NF condition is necessary as well as sufficient for implementation when there is a single agent and three alternatives. Next we identify a condition which we call the *General Monotonicity (GM)* condition that is necessary for implementation by a covered mechanism for an arbitrary number of agents and alternatives. The GM condition generalizes the NF condition when we consider implementation by covered mechanisms. We also show that the GM condition implies the strategy-resistance condition which is necessary for implementation (see Jackson (1992)). We state the condition after we introduce a definition.

Definition 19 A tuple $z = (a, P, i, P_i')$ where $a \in X$, $P \in \mathcal{D}$, $i \in N$, $P_i' \in \mathcal{D}_i$, is an "admissible tuple" if $a \in S(P) \setminus S(P_i', P_{-i})$. Let z_a denote the first alternative in z, i.e.

$z_a = a$.

Definition 20 A SCC $S : \mathcal{D} \to X$ satisfies the *General Monotonicity (GM)* condition if there exists an association $\Psi : X \times \mathcal{D} \times N \times \cup\mathcal{D}_i \to X \times X \times X \times X \times X \times \cup\mathcal{D}_{-i}$, such that

1. (a) if $a_1 \in S(P_i, P_{-i}) \cap S(P_i', P_{-i})$, then $\Psi(a_1, P, i, P_i') = (a_1, a_1, a_1, a_1, a_1, P_{-i})$;

   (b) if $a_1 \notin S(P_i, P_{-i}) \setminus S(P_i', P_{-i})$, i.e. (a_1, P, i, P_i') is an admissible tuple, then $\Psi(a_1, P, i, P_i') = (a_2, a_3, a_4, a_5, a_6, P_{-i}')$, where $a_2 \in S(P_i', P_{-i})$, $a_3 \in S(P_i, P_{-i})$, such that $a_2 P_i' a_1$, $a_2 R_i' a_3$, $a_3 R_i a_2$. If $a_3 = a_1$, then $a_4 = a_5 = a_6 = a_1, P_{-i}' = P_{-i}$. If $a_3 \neq a_1$, then $a_4 P_i a_5$, $a_5 R_i a_6$, $a_6 R_i' a_5$, $a_6 P_i' a_4$, where $a_4, a_5 \in S(P)$, $a_6 \in S(P_i', P_{-i})$.

   (c) if $z_1 = (a_1, P, i, P_i')$ and $z_2 = (a_1, P, j, P_j')$ be two admissible tuples, where $i \neq j$; $i, j \in N$, then there exist $b_1 \in S(P)$, $b_2 \in S(P_i, P_j', P_{-ij})$, $b_3 \in S(P_i', P_j, P_{-ij})$ and $b_4 \in S(P_i', P_j', P_{-ij})$, such that (i) $b_1 R_i b_3$, $b_2 R_i b_4$, $b_3 R_i' b_1$, $b_3 R_i' \Psi_2(z_2)$, $b_4 R_i' b_2$ and $b_4 R_i' \Psi_1(z_2)$; (ii) $b_1 R_j b_2$, $b_3 R_j b_4$, $b_2 R_j' \Psi_2(z_1)$, $b_2 R_j' b_1$, $b_4 R_j' \Psi_1(z_1)$ and $b_4 R_j' b_3$.

2. there does not exist a sequence of admissible tuples $z^0, z^1, ..., z^T, z^{T+1}$, where

   (a) $z^{T+1} = z^0$;

   (b) $z_a^r = \Psi_k(z^{r-1})$ for some $k = 1, ..., 5$, $1 \leq r \leq T+1$, where $\Psi_k(a, P, i, P_{-i})$ denotes an alternative in $\Psi(a, P, i, P_{-i})$, $k = 1, 2, 3, 4, 5$.

Proposition 4 *If a SCC can be implemented by a covered mechanism, then it satisfies the GM condition.*

Proof: Let S be a SCC that is implemented by a covered mechanism $\Gamma = (M, g)$. Let $P_i, P_i' \in \mathcal{D}_i$, $P_{-i} \in \mathcal{D}_{-i}$. Let $a_1 \in S(P_i, P_{-i})$. There is a message m_i which is undominated at

P_i and a message m_{-i} which is undominated at P_{-i}, such that $g(m_i, m_{-i}) = a_1$. There are two possibilities:

1. $a_1 \in S(P'_i, P_{-i})$, or

2. $a_1 \in S(P_i, P_{-i}) \setminus S(P'_i, P_{-i})$.

In the later case, there is a message m'_i which is undominated at P'_i and a message m''_i which is undominated at P_i, such that m'_i dominates m_i and m''_i at P'_i and m''_i dominates m'_i at P_i.

Let $g(m'_i, m_{-i}) = a_2$ and $g(m''_i, m_{-i}) = a_3$, so that $a_2 P'_i a_1$, $a_3 R_i a_2$ and $a_2 R'_i a_3$. We note that a_2 is strictly preferred to a_1 according to P'_i because $m_i \notin D(\Gamma, P'_i)$. This is because $a_1 \notin S(P'_i, P_{-i})$. But m'_i and m''_i dominate each other at P'_i and P_i respectively and it is also possible that $a_2 = a_3$. Since $m'_i \in D(\Gamma, P'_i)$ and $m''_i \in D(\Gamma, P_i)$ and $m_{-i} \in D(\Gamma, P_{-i})$, we must have $a_2 \in S(P'_i, P_{-i})$ and $a_3 \in S(P_i, P_{-i})$. If $m''_i = m_i$ then clearly $a_3 = a_1$. But if $m''_i \neq m_i$, since both m_i and m''_i are undominated at P_i there must exist a message profile, say $m'_{-i} \in M_{-i}$ and there exists P'_{-i} such that $m'_{-i} \in D(\Gamma, P'_{-i})$ and $g(m_i, m'_{-i}) P_i g(m''_i, m'_{-i})$. This is because since $m''_i \neq m_i$ and thus $a_3 \neq a_1$, it must be that $a_3 P_i a_1$. But since both m_i and $m''_i \in D(\Gamma, P_i)$, there must exist a message profile by other players, say, $m'_{-i} \in D(\Gamma, P'_{-i})$ such that $g(m_i, m'_{-i}) P_i g(m''_i, m'_{-i})$. Let $g(m''_i, m'_{-i}) = a_5$ and $g(m_i, m'_{-i}) = a_6$. Since m''_i dominates m'_i at P_i and m'_i dominates m''_i at P'_i, we must also have $a_5 R_i a_6$ and $a_6 R'_i a_5$. Since m'_i dominates m_i at P'_i, we also have $a_6 P'_i a_4$.

It easy to construct an association $\Psi : X \times \mathcal{P}^N \times N \times \cup \mathcal{P}_i \to X \times X \times X \times X \times X \times \cup \mathcal{P}_{-i}$, such that if $a_1 \in S(P_i, P_{-i}) \cap S(P'_i, P_{-i})$, then $\Psi(a_1, P, i, P'_i) = (a_1, a_1, a_1, a_1, a_1, P_{-i})$. If $a_1 \notin S(P_i, P_{-i}) \setminus S(P'_i, P_{-i})$, then the following holds.

We have shown that if $a_1 \notin S(P_i, P_{-i}) \setminus S(P'_i, P_{-i})$, there are $a_2 \in S(P'_i, P_{-i})$ and

$a_3 \in S(P_i, P_{-i})$ such that $a_2 P_i' a_1$, $a_3 R_i a_2$ and $a_2 R_i' a_3$. Thus we have $\Psi(a_1, P, i, P_i') = (a_2, a_3, a_4, a_5, a_6, P_{-i}')$, such that if $a_3 = a_1$, then $\Psi(a_1, P, i, P_i') = (a_2, a_1, a_1, a_1, a_1, P_{-i})$. If $a_3 \neq a_1$, then we have $P_{-i}' \in \mathcal{P}_{-i}$, such that $a_4 P_i a_5$, $a_5 R_i a_6$, $a_6 R_i' a_5$ and $a_6 P_i' a_4$. Thus, in this case $\Psi(a_1, P, i, P_i') = (a_2, a_3, a_4, a_5, a_6, P_{-i}')$. Combining both the cases we have that if $a_1 \notin S(P_i, P_{-i}) \setminus S(P_i', P_{-i})$, then $\Psi(a_1, P, i, P_i') = (a_2, a_3, a_4, a_5, a_6, P_{-i}')$, where $a_2 \in S(P_i', P_{-i})$, $a_3 \in S(P_i, P_{-i})$, such that $a_2 P_i' a_1$, $a_2 R_i' a_3$, $a_3 R_i a_2$. If $a_3 = a_1$, then $a_4 = a_5 = a_6 = a_1, P_{-i}' = P_{-i}$. If $a_3 \neq a_1$, then $a_4 P_i a_5, a_5 R_i a_6, a_6 R_i' a_5, a_6 P_i' a_4$, where $a_4, a_5 \in S(P), a_6 \in S(P_i', P_{-i})$. Hence 1.$(a)$ and 1.(b) are proved.

Let $z_1 = (a_1, P, i, P_i')$ and $z_2 = (a_1, P, j, P_j')$ be two admissible tuples, where $i \neq j$; $i, j \in N$. It follows from the discussion above that there are three messages, m_i, $m_i'' \in D(\Gamma, P_i)$ and $m_i' \in D(\Gamma, P_i')$for the agent i and three messages, m_j, $m_j'' \in D(\Gamma, P_j)$ and $m_j' \in D(\Gamma, P_j')$ for the agent j, such that (i) m_i is weakly dominated by m_i' at P_i', m_i'' is weakly dominated by m_i' at P_i' and m_i' is weakly dominated by m_i'' at P_i; (ii) m_j is weakly dominated by m_j' at P_j', m_j'' is weakly dominated by m_j' at P_j' and m_j' is weakly dominated by m_j'' at P_j.

Clearly, $g(m_i, m_j, m_{-ij}) = a_1$, $g(m_i, m_j'', m_{-ij}) = \Psi_2(z_2)$, $g(m_i, m_j', m_{-ij}) = \Psi_1(z_2)$, $g(m_i'', m_j, m_{-ij}) = \Psi_2(z_1)$ and $g(m_i', m_j, m_{-ij}) = \Psi_1(z_1)$. This follows from the design of the implementing mechanism Γ, where m_{-ij} represents the message profile by all other agents except agents i and j.

Let $g(m_i'', m_j'', m_{-ij}) = b_1$, $g(m_i'', m_j', m_{-ij}) = b_2$, $g(m_i', m_j'', m_{-ij}) = b_3$ and $g(m_i', m_j', m_{-ij}) = b_4$. Since m_i, $m_i'' \in D(\Gamma, P_i)$, $m_i' \in D(\Gamma, P_i')$, m_j, $m_j'' \in D(\Gamma, P_j)$ and $m_j' \in D(\Gamma, P_j')$, we have $b_1 \in S(P)$, $b_2 \in S(P_i, P_j', P_{-ij})$, $b_3 \in S(P_i', P_j, P_{-ij})$ and $b_4 \in S(P_i', P_j', P_{-ij})$. It is easy to check that (i) $b_1 R_i b_3$, $b_2 R_i b_4$, $b_3 R_i' b_1$, $b_3 R_i' \Psi_2(z_2)$, $b_4 R_i' b_2$ and $b_4 R_i' \Psi_1(z_2)$; (ii) $b_1 R_j b_2$, $b_3 R_j b_4$, $b_2 R_j' \Psi_2(z_1)$, $b_2 R_j' b_1$, $b_4 R_j' \Psi_1(z_1)$ and $b_4 R_j' b_3$. If $\Psi_2(z_2) = a_1$ and $\Psi_2(z_1) = a_1$, then $b_1 = a_1$, $b_2 = \Psi_1(z_2)$ and $b_3 = \Psi_1(z_1)$. If $\Psi_2(z_1) = a_1$, $\Psi_2(z_2) \neq a_1$, then $b_1 = \Psi_2(z_2)$ and $b_2 = \Psi_1(z_2)$. Similarly, if $\Psi_2(z_2) = a_1$, $\Psi_2(z_1) \neq a_1$, then $b_1 = \Psi_2(z_1)$ and $b_3 = \Psi_1(z_1)$. Hence we prove

1.(c).

We prove the last part of the condition by the method of contradiction. Let S be an implementable SCC and let $\Gamma = (M, g)$ be the covered mechanism that implements S. Let $z^0, z^1, ..., z^T, z^{T+1}$ be a sequence of admissible tuples such that

1. $z^{T+1} = z^0$;
2. $z^r_a = \Psi_k(z^{r-1})$ for some $k = 1, ..., 5$, $1 \leq r \leq T+1$.

Thus, $\Psi_k(z^T) = z^{T+1}_a$. Since $z^{T+1} = z^0$, we have $z^{T+1}_a = z^0_a$. Let $z^0 = (z^0_a, P, i, P_{-i})$ for $P \in \mathcal{D}, i \in N, P_{-i} \in \mathcal{D}_{-i}$. Since $z^0_a \in S(P_i, P_{-i}) \setminus S(P'_i, P_{-i})$, there must exist $m_i \in D(\Gamma, P_i)$, $m'_i \in D(\Gamma, P'_i)$, $m''_i \in D(\Gamma, P_i)$, $m_{-i} \in D(\Gamma, P_i)$ such that $g(m_i, m_{-i}) = z^0_a$ and m'_i dominates m_i and m''_i at P'_i. Also m''_i dominates m'_i at P_i. There also exists $P'_{-i} \in \mathcal{D}_{-i}$, $m'_{-i} \in D(\Gamma, P'_{-i})$ such that $g(m_i, m'_{-i}) P_i g(m''_i, m'_{-i})$, $g(m''_i, m'_{-i}) R_i g(m'_i, m'_{-i})$, $g(m'_i, m'_{-i}) R'_i g(m''_i, m'_{-i})$. Clearly, $g(m'_i, m_{-i}) = \Psi_1(z^0)$, $g(m''_i, m_{-i}) = \Psi_2(z^0)$, $g(m_i, m'_{-i}) = \Psi_3(z^0)$, $g(m''_i, m'_{-i}) = \Psi_4(z^0)$ and $g(m'_i, m'_{-i}) = \Psi_5(z^0)$. Thus from an admissible tuple z^0 we have a set of messages, as described above.

We note that $z^1_0 = \Psi_k(z^0)$, $k = 1, .., 5$. Since z^1 is an admissible tuple, we have a set of messages in a similar way as described above and this forms a chain of messages as z^2, z^3,...., z^T are admissible tuples. Since $z^{T+1} = z^0$, we complete a round of having sets of messages in the mechanism and come back to the initial admissible tuple, which again requires to have a set of messages and so on. This leads to cycles after cycles of messages and thus we have infinite messages in the mechanism. Thus the implementing mechanism cannot be finite. This proves that there does not exist a sequence of admissible tuples z^0, z^1,....,z^T, z^{T+1}, with $z^{T+1} = z^0$ and $z^r_a = \Psi(z^{r-1})$ (for some $k = 1, ..., 5$ and $1 \leq r \leq T+1$). Hence the theorem is proved.

■

In the following proposition we show that the GM condition implies the strategy-resistance of SCC.

Proposition 5 *A SCC satisfying the GM condition satisfies strategy-resistance.*

Proof: We prove it by the method of contradiction. Let S be a SCC satisfying GM condition. Suppose that it does not satisfy strategy-resistance. This implies that there are $a, b \in X$, $P_i, P_i' \in \mathcal{D}_i$ such that $a = Max_{P_i}\{S(P_i, P_{-i})\}$, $b \in S(P_i', P_{-i}) \setminus S(P_i, P_{-i})$ and bP_ia.

Since $(b, (P_i', P_{-i}), i, P_i)$ is an admissible tuple and S satisfies the GM condition, we must have an association $\Psi : X \times \mathcal{D} \times N \times \cup\mathcal{D}_i \to X \times X \times X \times X \times X \times \cup\mathcal{D}_{-i}$, such that $\Psi(b, (P_i', P_{-i}), i, P_i) = (b_2, b_3, b_4, b_5, b_6, P_{-i}')$, where $b_2 \in S(P_i, P_{-i})$, $b_3 \in S(P_i', P_{-i})$, such that b_2P_ib, $b_2R_i'b_3$, $b_3R_ib_2$.

Since b_2P_ib and bP_ia, we have b_2P_ia. But this contradicts $a = Max_{P_i}\{S(P_i, P_{-i})\}$, since $b_2 \in S(P_i, P_{-i})$. ■

3.4.1 An Application

In this subsection, we explore the covered versus uncovered mechanism question and provide an illustration of Proposition 4. In what follows, the *Pareto correspondence* (denoted by $S : \mathcal{P} \to 2^X \setminus \emptyset$) is the correspondence that selects the set of efficient alternatives at all profiles.

Proposition 6 *The Pareto correspondence is not implementable by a covered mechanism in the case $n = 2$ and $m = 3$.*

Proof: We will show that the Pareto Correspondence violates the GM condition and then invoke Proposition 4.

Let $N = \{1, 2\}$, $X = \{a, b, c\}$ and $\mathcal{P}_i = \{P_1^i, P_2^i, P_3^i, P_4^i, P_5^i, P_6^i\}$ for $i \in N$ such that $aP_1^i bP_1^i c$, $aP_2^i cP_2^i b$, $cP_3^i aP_3^i b$, $cP_4^i bP_4^i a$, $bP_5^i cP_5^i a$, $bP_6^i aP_6^i c$. The following table presents the Pareto correspondence S in this environment.

	P_1^2	P_2^2	P_3^2	P_4^2	P_5^2	P_6^2
P_1^1	$\{a\}$	$\{a\}$	$\{a, c\}$	$\{a, b, c\}$	$\{a, b\}$	$\{a, b\}$
P_2^1	$\{a\}$	$\{a\}$	$\{a, c\}$	$\{a, c\}$	$\{a, b, c\}$	$\{a, b\}$
P_3^1	$\{a, c\}$	$\{a, c\}$	$\{c\}$	$\{c\}$	$\{b, c\}$	$\{b, c, a\}$
P_4^1	$\{a, b, c\}$	$\{a, c\}$	$\{c\}$	$\{c\}$	$\{b, c\}$	$\{b, c\}$
P_5^1	$\{a, b\}$	$\{a, b\}$	$\{b, c\}$	$\{b, c\}$	$\{b\}$	$\{b\}$
P_6^1	$\{a, b\}$	$\{a, b\}$	$\{b, c, a\}$	$\{b, c\}$	$\{b\}$	$\{b\}$

Table 3.7: Pareto Correspondence for the $n = 2$, $m = 3$ case

We note that $S(P_1^1, P_4^2) = \{a, b, c\}$, $S(P_1^1, P_3^2) = \{a, c\}$, $S(P_2^1, P_3^2) = \{a, c\}$, $S(P_2^1, P_4^2) = \{a, c\}$. To see that S cannot be implemented by a covered mechanism we prove it by the method of contradiction. Let $\Gamma = (M, g)$ be a covered mechanism that implements S. Since $b \in S(P_1^1, P_4^2) \backslash S(P_2^1, P_4^2)$, $(b, (P_1^1, P_4^2), 1, P_2^1)$ is an admissible tuple. Since S is implementable it must satisfy the GM condition and thus there must exist an association $\Psi : X \times \mathcal{P} \times N \times \cup \mathcal{P}_i \to X \times X \times X \times X \times X \times \mathcal{P}_2$, such that $\Psi(b, (P_1^1, P_4^2), \{1\}, P_2^1) = (x_1, x_2, x_3, x_4, x_5, P_2')$, where $x_1 P_2^1 b$, $x_2 R_1^1 x_1$ and $x_1 R_2^1 x_2$.

There are the following subcases.

$(A.1)$ $x_1 = c$, $x_2 = b$. Following the GM condition we have $x_3 = x_4 = x_5 = b$ and $P^{2'} = P^2_4$.

$(A.2)$ $x_1 = a$, $x_2 = a$, Following the GM condition we have $x_3 = b$, $x_4 = c$, $x_5 = c$ and $P^{2'} = P^2_4$.

Similarly, since $b \in S(P^1_1, P^2_4) \setminus S(P^1_1, P^2_3)$, $(b, (P^1_1, P^2_4), \{2\}, P^2_3)$ is an admissible tuple. Following the GM condition, there is an association $\Psi : X \times \mathcal{P} \times N \times \cup\mathcal{P}_i \rightarrow X \times X \times X \times X \times X \times \mathcal{P}_1$, such that $\Psi(b, (P^1_1, P^2_4), \{2\}, P^2_3) = (y_1, y_2, y_3, y_4, y_5, P'_1)$, where $y_1 P^2_3 b$, $y_2 R^2_4 y_1$ and $y_1 R^2_3 y_2$.

We have two subcases.

$(B.1)$ $y_1 = a$, $y_2 = b$, $y_3 = y_4 = y_5 = b$ and $P^{1'} = P^1_1$.

$(B.2)$ $y_1 = c$, $y_2 = c$, $y_3 = b$, $y_4 = a$, $y_5 = a$ and $P^{1'} = P^1_1$.

We consider various possibilities discussed above.

(1) $(A.1)$ and $(B.1)$: Thus $x_1 = c$, $x_2 = b$, $x_3 = x_4 = x_5 = b$ and $y_1 = a$, $y_2 = b$, $y_3 = y_4 = y_5 = b$. Following 1.(c) of the GM condition, we must have $b_1 \in S(P^1_1, P^2_4)$, $b_2 \in S(P^1_1, P^2_3)$, $b_3 \in S(P^1_2, P^2_4)$ and $b_4 \in S(P^1_2, P^2_3)$, such that (i) $b_1 R^1_1 b_3$, $b_2 R^1_1 b_4$, $b_3 R^1_2 b_1$, $b_3 R^1_2 b$, $b_4 R^1_2 b_2$ and $b_4 R^1_2 a$; (ii) $b_1 R^2_4 b_2$, $b_3 R^2_4 b_4$, $b_2 R^2_3 b$, $b_2 R^2_3 b_1$, $b_4 R^2_3 c$ and $b_4 R^2_3 b_3$.

Clearly, $b_1 = b$, $b_2 = a$, $b_3 = c$. But there is no alternative b_4 satisfying the condition above. For instance, if $b_4 = a$ then $b_4 R^2_3 c$ does not hold good. Similarly, if $b_4 = c$, then $b_4 R^1_2 a$ does not hold good. Thus 1.(c) of the GM condition is violated.

(2) $(A.1)$ and $(B.2)$: In this case, according to 1.(c) of the GM condition, we must have $b_1 \in S(P^1_1, P^2_4)$, $b_2 \in S(P^1_1, P^2_3)$, $b_3 \in S(P^1_2, P^2_4)$ and $b_4 \in S(P^1_2, P^2_3)$, such that (i) $b_1 R^1_1 b_3$, $b_2 R^1_1 b_4$, $b_3 R^1_2 b_1$, $b_3 R^1_2 c$, $b_4 R^1_2 b_2$ and $b_4 R^1_2 c$; (ii) $b_1 R^2_4 b_2$, $b_3 R^2_4 b_4$, $b_2 R^2_3 b$, $b_2 R^2_3 b_1$, $b_4 R^2_3 c$ and $b_4 R^2_3 b_3$.

Clearly, $b_1 = c$, $b_2 = c$, $b_3 = c$ and $b_4 = c$. We also note that $y_3 = b$ and $P^{1'} = P^1_1$. Let $z^0 = (b, (P^1_1, P^2_4), \{2\}, P^2_3)$. We note that $\Psi_3(z^0) = b$ and $(b, (P^1_1, P^2_4), \{2\}, P^2_3)$ is again admissible tuple. This violates the part (2) in the GM condition.

Similarly we can show that if $(A.2)$ and $(B.1)$ hold good or if $(A.2)$ and $(B.2)$ hold good, then the GM condition is violated.

■

However, note that the Pareto correspondence is implementable for $n = 2$ and $m = 2$. In particular the following mechanism $\hat{\Gamma}$ in Table 3.8 can be used.

	m^2_1	m^2_2	m^2_3	m^2_4
m^1_1	a	a	a	b
m^1_2	a	a	b	a
m^1_3	a	b	b	b
m^1_4	b	a	b	b

Table 3.8: Mechanism $\hat{\Gamma}$

Proposition 7 *The Pareto correspondence is implementable in the case where $n = 2$ and $m = 3$.*

Proof: We consider the environment described in the proof of Proposition 6. We provide a mechanism $\bar{\Gamma} = (M, g)$ in Table 3.9 that implements the Pareto correspondence (S) when there are two agents and three alternatives.

The message space for the agent 1 in the mechanism in Table 3.9 has 20 messages and the message space for the agent 2 has 18 messages: from m^2_1 to m^2_{18}. The messages for the

	m_1^2	m_2^2	m_3^2	m_4^2	m_5^2	m_6^2	m_7^2	m_8^2	m_9^2	m_{10}^2	m_{11}^2	m_{12}^2	m_{13}^2	m_{14}^2	m_{15}^2	m_{16}^2	m_{17}^2	m_{18}^2
m_1^1	a	a	a	a	a	b	b	a	a	c	a	a	b	a	a	a	a	a
m_2^1	a	a	a	a	b	a	a	b	a	a	a	c	a	a	a	b	b	a
m_3^1	a	a	a	a	a	c	b	a	a	c	a	a	c	a	a	a	a	a
m_4^1	a	a	a	a	b	a	a	b	a	a	a	c	a	a	a	c	c	a
m_5^1	a	b	b	c	b	b	b	b	b	c	c	b	b	b	b	b	b	b
m_6^1	b	b	a	b	b	b	b	b	b	b	b	c	b	b	c	b	b	c
m_7^1	a	b	b	a	b	b	b	b	b	c	a	b	b	b	b	b	b	b
m_8^1	b	b	a	b	b	b	b	b	b	b	b	c	b	b	a	b	b	a
m_9^1	a	a	c	c	c	c	b	c	a	c	c	c	c	c	c	c	c	c
m_{10}^1	c	c	a	c	b	c	c	b	c	c	c	c	c	c	c	c	c	c
m_{11}^1	a	b	c	c	c	c	b	c	b	c	c	c	c	c	c	c	c	c
m_{12}^1	c	c	a	c	b	c	c	b	c	c	c	c	c	c	c	c	c	c
m_{13}^1	a	a	a	c	c	b	b	a	a	c	c	c	c	c	c	c	c	c
m_{14}^1	a	a	a	a	b	b	b	b	b	a	a	c	b	c	c	c	c	a
m_{15}^1	a	a	a	a	b	b	b	b	b	b	a	c	b	b	c	c	b	a
m_{16}^1	a	a	a	a	b	c	a	b	a	c	c	c	c	c	c	c	c	a
m_{17}^1	a	c	a	a	b	c	c	b	a	c	c	c	c	c	c	c	c	a
m_{18}^1	b	b	a	c	b	b	b	b	b	c	c	c	c	c	c	c	c	a
m_{19}^1	c	b	a	c	b	c	b	b	c	c	c	c	c	c	c	c	c	a
m_{20}^1	a	a	a	a	b	b	b	b	b	c	a	c	c	c	a	a	c	a

Table 3.9: Mechanism $\bar{\Gamma}$

agent 1 are presented as row messages and the messages for the agent 2 are presented as column messages. For agent 1, m_1^1 and m_2^1 are undominated at P_1^1, i.e. $m_1^1, m_2^1 \in D(\bar{\Gamma}, P_1^1)$. Similarly, $m_3^1, m_4^1 \in D(\bar{\Gamma}, P_2^1)$, $m_5^1, m_6^1 \in D(\bar{\Gamma}, P_5^1)$, $m_7^1, m_8^1 \in D(\bar{\Gamma}, P_6^1)$, $m_9^1, m_{10}^1 \in D(\bar{\Gamma}, P_3^1)$, $m_{11}^1, m_{12}^1 \in D(\bar{\Gamma}, P_4^1)$. For the agent 2, we have $m_1^2, m_2^2 \in D(\bar{\Gamma}, P_1^2)$, $m_3^2, m_4^2 \in D(\bar{\Gamma}, P_2^2)$, $m_5^2, m_6^2, m_7^2 \in D(\bar{\Gamma}, P_5^2)$, $m_8^2, m_9^2 \in D(\bar{\Gamma}, P_6^2)$, $m_{10}^2, m_{11}^2, m_{12}^2 \in D(\bar{\Gamma}, P_3^2)$ and $m_{13}^2 \in D(\bar{\Gamma}, P_4^2)$.

■

REMARK 7 In view of Proposition 6, it must be the case the mechanism used in the proof of this Proposition is uncovered. This is indeed the case and can be verified directly as follows. The message space for the agent 1 in mechanism $\bar{\Gamma}$ can be been partitioned into two sets, $\bar{M}^1$ consisting of messages m_1^1 through m_{12}^1 and $\hat{M}^1$ from m_{13}^1 through m_{20}^1. Every message in $\bar{M}^1$ is undominated for some ordering for agent 1 while messages in $\hat{M}^1$ are dominated all orderings. Thus $\hat{M}^1$ is a set of nuisance messages for 1. Similarly, messages from m_{14}^2

through m_{18}^2 are nuisance messages for 2. Clearly $\bar{\Gamma}$ is uncovered. A critical observation is that implementation fails if nuisance messages are deleted. For instance consider a truncated mechanism where the nuisance messages for agent 2 are deleted. Then m_5^1 weakly dominates m_1^1 in the truncated mechanism; thus m_1^1 is no longer undominated at P_1^1. Consequently, there does not exist undominated messages for agents 1 and 2 which yield b at the profile P_1^1, P_4^2 even though it is Pareto-efficient at the profile. Hence the truncated mechanism does not implement S (although it weakly implements it according to Proposition 3).

The discussion above makes it clear that the GM condition is not necessary for implementation (if we allow for uncovered mechanisms). It also illustrates complexities involved in characterizing implementable SCCs.

3.5 Enforcing Compromises

In this section, we investigate some issues raised in Borgers (1991).

Definition 21 The alternative x is a compromise at profile P if

(i) There does not exist an agent $i \in N$ such that $t^1(P_i) = x$

(ii) x is Pareto efficient.

A SCC S satisfies the Compromise Axiom (CA) if there exists a profile $P \in \mathcal{D}$ such that

$$[x \in S(P)] \Rightarrow [x \text{ is a compromise at } P]$$

Thus S satisfies CA if there exists a profile where the outcome set contains *only* compromises. Borgers (1991) investigates the implementability of SCCs which satisfies CA. He shows the following.

Theorem 5 *(Borgers (1991)) There does not exist an efficient, implementable SCC $S : \mathcal{P} \to 2^X \setminus \emptyset$ satisfying CA in the case $n = 2$ or $m = 3$.*

We extend and refine Theorem 5 in several ways in this section. The next three propositions extend the impossibility result while the next two are possibility results.

The next result shows that the impossibility result can be extended to an arbitrary number of agents and alternatives provided that the CA axiom is strengthened. In particular compromises cannot be the only outcomes at *near-unanimous* profiles.

Definition 22 A preference profile $P \in \mathcal{D}$ is near-unanimous if there exists an agent $i \in N$ such that $t^1(P_j) = t^1(P_k) \neq t^1(P_i)$, for all $j, k \in N \setminus \{i\}$.

All except one agent agrees on the top-ranked alternative at a near-unanimous profile.

Definition 23 A SCC S satisfies the Strong Compromise Axiom (SCA) if there exists a near-unanimous profile $\hat{P} \in \mathcal{D}$ such that

$$[x \in S(\hat{P})] \Rightarrow [x \text{ is a compromise at } \hat{P}]$$

We note that SCA is a stronger axiom than CA. If $n = 2$, then SCA is equivalent to CA and Proposition 8 below generalizes one part of Theorem 5. For large n, SCA may be regarded as restrictive because it is "close" to violating unanimity. Thus SCA would be incompatible with the assumption of *No Veto Power* which is widely used in the theory of Nash implementation. However, as Proposition 8 below demonstrates, compromises at near-unanimous profiles are problematic while implementing efficient SCCs.

Proposition 8 *Let $n \geq 2$ and $m \geq 3$ be arbitrary. There does not exist an efficient, implementable SCC $S : \mathcal{P} \to 2^X \setminus \emptyset$ satisfying SCA.*

Proof: We prove it by the method of contradiction. Let $S : \mathcal{P} \to 2^X \setminus \emptyset$ be an efficient SCC that is implementable by $\Gamma = (M, g)$ and satisfies SCA. Let $P \in \mathcal{P}$ be a near-unanimous profile such that $t^1(P^1) = t^1(P^2) = ... = t^1(P^{n-1}) = a$ and $t^1(P^n) = b$. We can present P in the following matrix.

$$
\begin{array}{c} \\ = \end{array}
\begin{array}{cccccc} P_1 & P_2 & . & . & P_{n-1} & P_n \end{array}
$$

$$
= \begin{pmatrix} a & a & . & . & a & b \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{pmatrix}
$$

Let $P' \in \mathcal{P}$, such that $t^1(P'_1) = t^1(P'_2) = ... = t^1(P'_{n-1}) = a$, $t^1(P'_n) = b$, $t^2(P'_1) = t^2(P'_2) = ... = t^2(P'_{n-1}) = b$ and $t^2(P'_n) = a$.

$$
= \begin{pmatrix} \overset{P'_1}{a} & \overset{P'_2}{a} & \overset{.}{.} & \overset{.}{.} & \overset{P'_{n-1}}{a} & \overset{P'_n}{b} \\ b & b & . & . & b & a \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{pmatrix}
$$

Let $S(P) = c$, where c is a compromise. The analysis remains the same if we have more than one compromise alternative in $S(P)$. Clearly, if $P_n = P'_n$ then following the efficiency axiom, we have $S(P) = a$ or b. Thus S cannot satisfy SCA. Interesting case arises when $P_n \neq P'_n$. We prove the following Claims.

Claim 1: $S(P_1, P_2, ..., P'_n) = a$.

Proof: Since S is efficient, $S(P_1, P_2, ..., P'_n) \in \{a, b\}$. But if $S(P_1, P_2, ..., P'_n) = b$, the agent n can manipulate at P via P'_n. This is because we have assumed that $S(P) = c$, where c is a compromise and thus bP_nc. Thus strategy-resistance, which is a necessary condition for implementability (Jackson (1992)), is violated.

Claim 2: $a \notin S(P'_1, P_2, ..., P_n)$.

Proof: This claim also follows directly from the strategy-resistance of S.

Claim 3: $S(P'_1, P'_2, ..., P'_{n-1}, P_n) = b$.

Proof: Since S is Pareto efficient, $S(P_1', P_2', ..., P_{n-1}', P_n)$ is a subset of $\{a, b\}$. Let $a \in S(P_1', P_2', ..., P_{n-1}', P_n)$. Therefore there are $m_1' \in D(\Gamma, P_1')$,...., $m_{n-1}' \in D(\Gamma, P_{n-1}')$, $m_n \in D(\Gamma, P_n)$ such that $g(m_1', ..., m_{n-1}', m_n) = a$. Either $m_{n-1}' \in D(\Gamma, P_{n-1})$, or there is message $\tilde{m}_{n-1}' \in D(\Gamma, P_{n-1})$, such that $g(m_1, m_2, ..., \tilde{m}_{n-1}', m_n)\ R_{n-1}\ g(m_1, m_2, ..., m_{n-1}', m_n)$ for all $m_1 \in M_1$, $m_2 \in M_2$,...,$m_{n-2} \in M_{n-2}$, $m_n \in M_n$. In particular, we have the following: $g(m_1', m_2', ..., m_{n-2}', \tilde{m}_{n-1}', m_n)\ R_{n-1}\ g(m_1', m_2', ..., m_{n-2}', m_{n-1}', m_n)$.

This follows from the finiteness of the mechanism. Note that we have assumed that $g(m_1', m_2', ..., m_{n-2}', m_{n-1}', m_n) = a$. We observe that $g(m_1', m_2', ..., m_{n-2}', \tilde{m}_{n-1}', m_n)\ R_{n-1}$ $g(m_1', m_2', ..., m_{n-2}', m_{n-1}', m_n)$ and $t^1(R^{n-1}) = a$. Thus $g(m_1', m_2', ..., m_{n-2}', \tilde{m}_{n-1}', m_n) = a$. This follows from the strategy-resistance. Therefore $a \in S(P_1{}', P_2', ..., P_{n-2}', P_{n-1}, P_n)$.

Therefore there are $m_1' \in D(\Gamma, P_1')$,...., $m_{n-2}' \in D(\Gamma, P_{n-2}')$, $m_{n-1} \in D(\Gamma, P_{n-1})$, $m_n \in D(\Gamma, P_n)$ such that $g(m_1', ..., m_{n-2}', m_{n-1}, m_n) = a$. There are two possibilities, one of which holds good: (i) $m_{n-2}' \in D(\Gamma, P_{n-2})$; (ii) there is message $\tilde{m}_{n-2}' \in D(\Gamma, P_{n-2})$, such that $g(m_1, m_2, ..., m_{n-3}, \tilde{m}_{n-2}', m_{n-1}, m_n)\ R_{n-2}\ g(m_1, m_2, ..., m_{n-3}, m_{n-2}', m_{n-1}, m_n)$ for all $m_1 \in M_1$, $m_2 \in M_2$,...,$m_{n-3} \in M_{n-3}$, $m_{n-1} \in M_{n-1}$, $m_n \in M_n$.

In particular, $g(m_1', m_2', ..., m_{n-3}', \tilde{m}_{n-2}', m_{n-1}, m_n)\ R_{n-2}\ g(m_1', m_2', ..., m_{n-3}', m_{n-2}', m_{n-1}, m_n)$. But $g(m_1', ..., m_{n-2}', m_{n-1}, m_n) = a$ and $t^1(R^{n-2}) = a$. Thus $g(m_1', m_2', ..., m_{n-3}', \tilde{m}_{n-2}', m_{n-1}, m_n) = a$. Therefore we have $a \in S(P_1{}', P_2', ..., P_{n-3}', P_{n-2}, P_{n-1}, P_n)$.

We can thus show that $a \in S(P_1{}', P_2, ..., P_{n-3}, P_{n-2}, P_{n-1}, P_n)$. But this contradicts Claim (2). Therefore $a \notin S(P_1', P_2', ..., P_{n-1}', P_n)$ and $S(P_1', P_2', ..., P_{n-1}', P_n) = b$. This completes the proof of Claim (3).

From Claim (3) we know that $S(P_1', P_2', ..., P_{n-1}', P_n) = b$. Thus for all $m_1' \in D(\Gamma, P_1')$, $m_2' \in D(\Gamma, P_2')$,....,$m_{n-1}' \in D(\Gamma, P_{n-1}')$, $m_n \in D(\Gamma, P_n)$. Let $m_1' \in D(\Gamma, P_1')$, $m_2' \in D(\Gamma, P_2')$,....,$m_{n-1}' \in D(\Gamma, P_{n-1}')$, $m_n \in D(\Gamma, P_n)$. We have two possibilities. Either (i) $m_n \in D(\Gamma, P_n')$ or (ii) there exists $\hat{m}_n' \in D(\Gamma, P_n')$, such that $g(m_1, m_2, ..., m_{n-1}, \hat{m}_n')\ R_n'\ g(m_1, m_2, ..., m_{n-1}, m_n)$, $\forall m_1 \in M_1, m_2 \in M_2, ..., m_{n-1} \in M_{n-1}$. Therefore $g(m_1', m_2', ..., m_{n-1}', \hat{m}_n')\ R_n'\ g(m_1', m_2', ..., m_{n-1}', m_n)$. Thus $g(m_1', m_2', ..., \hat{m}_n') = b$. This is because $t^1(P_n') = b$. We Note this holds for *all* messages in $D(\Gamma, P_1')$, $D(\Gamma, P_2')$,....,$D(\Gamma, P_{n-1}')$.

From Claim 1 we know that for all $m_1 \in D(\Gamma, P_1)$, $m_2 \in D(\Gamma, P_2)$,...., $m_{n-1} \in D(\Gamma, P_{n-1})$ and for the message $\hat{m}_n'$, we have $g(m_1, m_2, ..., m_{n-1}, \hat{m}_n') = a$. Let $m_1^* \in D(\Gamma, P_1)$ be an arbitrary message. Either $m_1^* \in D(\Gamma, {P_1}')$ or there is $m_1^{**} \in D(\Gamma, P_1')$ such that $g(m_1^{**}, m_2, ..., m_n)$ R_1' $g(m_1^*, m_2, ..., m_n)$ for all $m_2, ..., m_n$. Thus $g(m_1^{**}, m_2, ..., \hat{m}_n')$ R_1' $g(m_1^*, m_2, ..., \hat{m}_n')$, for all $m_2 \in D(\Gamma, P_2)$,...., $m_{n-1} \in D(\Gamma, P_{n-1})$.

We know that for all $m_1 \in D(\Gamma, P_1)$, $m_2 \in D(\Gamma, P_2)$, ..., $m_{n-1} \in D(\Gamma, P_{n-1})$ and for $\hat{m}_n{}' \in D(\Gamma, P_n{}')$ it must be that $g(m_1, m_2, ..., m_{n-1}, m_n{}') = a$. Thus $g(m_1^*, m_2, ..., m_{n-1}, \hat{m}_n{}') = a$, for all $m_2 \in D(\Gamma, P_2)$, ...,$m_{n-1} \in D(\Gamma, P_{n-1})$. Since $t^1({P_1}') = a$, we must have that $g(m_1^{**}, m_2, ..., m_{n-1}, \hat{m}_n') = a$, for all $m_2 \in D(\Gamma, P_2)$, ..., $m_{n-1} \in D(\Gamma, P_{n-1})$.

Therefore $g(m_1^{**}, \tilde{m}_2, m_3, ..., m_{n-1}, \hat{m}_n') = a$ for any arbitrary $\tilde{m}_2 \in D(\Gamma, P_2)$ and for all $m_3 \in D(\Gamma, P_3)$, ..., $m_{n-1} \in D(\Gamma, P_{n-1})$. Either $\tilde{m}_2 \in D(\Gamma, P_2')$ or there exists $\hat{m}_2 \in D(\Gamma, P_2')$ such that $g(m_1, \hat{m}_2, m_3, ..., m_n)$ R_2' $g(m_1, \tilde{m}_2, m_3, ..., m_n)$ for all $m_1, m_3, ..., m_n$. Therefore $g(m_1^{**}, \hat{m}_2, m_3, ..., m_{n-1}, \hat{m}_n')$ R_2' $g(m_1^{**}, \tilde{m}_2, m_3, ..., m_{n-1}, \hat{m}_n')$, where $m_3 \in D(\Gamma, P_3)$,...., $m_{n-1} \in D(\Gamma, P_{n-1})$. Thus $g(m_1^{**}, \hat{m}_2, m_3, ..., m_{n-1}, m_n') = a$. This is because the following holds good:

$g(m_1^{**}, \tilde{m}_2, m_3, ..., m_{n-1}, \hat{m}_n') = a$, where $m_3 \in D(\Gamma, P_3$,...., $m_{n-1} \in D(\Gamma, P_{n-1})$ and $t^1(R_2') = a$.

Continuing like this we can show that $g(m_1^{**}, \hat{m}_2, \hat{m}_3, ..., \hat{m}_{n-1}, m_n') = a$, where $\hat{m}_3 \in D(\Gamma, P_3)$, ..., $\hat{m}_{n-1} \in D(\Gamma, P_{n-1})$. But we arrive at a contradiction to the finding above that $g(m_1, m_2, m_3, ..., m_{n-1}, m_n') = b$ for all $m_1 \in D(\Gamma, P_1')$, ..., $m_{n-1} \in D(\Gamma, P_{n-1}')$.

We have proved that $S(P)$ must have either a or b. Therefore it is not possible that the outcome set at a near-unanimous profile consists of only compromises. Hence an implementable, efficient SCC does not satisfy SCA.

■

The next proposition shows that the impossibility result is retained if the efficiency axiom is replaced by *neutrality*. However this result holds only when there are two agents and three alternatives.

Proposition 9 *: There exists no implementable SCC* $S : \mathcal{P} \to 2^X \setminus \emptyset$ *satisfying neutrality and CA in the case* $n = 2$ *and* $m = 3$.

Proof: Let $X = \{a, b, c\}$ and $N = \{1, 2\}$. Let $\mathcal{P}_i = \{P^i_1, P^i_2, P^i_3, P^i_4, P^i_5, P^i_6\}$, where $aP^i_1bP^i_1c$, $aP^i_2cP^i_2b$, $cP^i_3aP^i_3b$, $cP^i_4bP^i_4a$, $bP^i_5cP^i_5a$ and $bP^i_6aP^i_6c$, $i = 1, 2$. Let S be an implementable SCC that satisfies neutrality and CA. Let $\Gamma = (M, g)$ be a mechanism that implements S. Since S satisfies CA, there is a profile such that the image set of S at that profile consists of only compromise alternatives. Without loss of generality, we assume that $S(P^1_1, P^2_4) = b$. We note that only compromise alternative at this profile is b. Thus, for all $m^1_1 \in D(\Gamma, P^1_1)$ and for all $m^2_4 \in D(\Gamma, P^2_4)$, we have $g(m^1_1, m^2_4) = b$. Since $t^1(P^1_5) = b$ there exists $m^{1'}_5 \in D(\Gamma, P^1_5)$ such that $g(m^{1'}_5, m^2_4) = b$, for all $m^2_4 \in D(\Gamma, P^2_4)$. This is because, all $m^1_1 \in D(\Gamma, P^1_1)$ must either be undominated or be weakly dominated by an undominated message at P^1_5.

Let $h : X \to X$ be a bijection such that $h(a) = b$, $h(b) = c$, $h(c) = a$. Since $S(P^1_1, P^2_4) = b$ and S is neutral, it follows that at the profile $P' = (P^1_5, P^2_2)$ we have $S(P') = c$. Thus for $m^1_5 \in D(\Gamma, P^1_5)$ and for $m^2_2 \in D(\Gamma, P^2_2)$, we have $g(m^1_5, m^2_2) = c$. Since c is the top-most alternative in P^4_2, there is an undominated strategy $m^{2'}_4$ at P^4_2, such that together with all the undominated strategies at P^1_5 it produces c. This implies that $g(m^1_5, m^{2'}_4) = c$, for all

$m_5^1 \in D(\Gamma, P_5^1)$. But we have already observed that there is $m_5^{1'} \in D(\Gamma, P_5^1)$ such that $g(m_5^{1'}, m_4^2) = b$, for all $m_4^2 \in D(\Gamma, P_4^2)$. This leads to a contradiction.

■

Our next (impossibility) result relates to the implementation of "small" correspondences. Roughly speaking there is a bias in favor of large SCCs in the matter of implementability. For instance, there is a super-correspondence of every SCC that is implementable. Although this may be trivial SCC that selects the entire set of alternatives at every profile. Consider the following sort of question: given a SCC, is there a sub-correspondence that is also implementable? An immediate consequence of an implementable sub-correspondence must be that it is *minimal*, which we define below.

Definition 24 A SCC S is minimal if for all P, $S(P) = \cup_i \{\tau(P_i, S(P))\}$.

Thus each alternative in the image set of a minimal SCC at a profile is the maximal alternative for some agent at that profile. Minimality is of interest because of the proposition below.

Proposition 10 *If a SCC is implementable, then it must be the super-correspondence of a minimal SCC.*

The proof of Proposition 10 is an immediate consequence of the necessity of strategy-resistance for implementation which is demonstrated in Jackson (1992). Two examples of minimal SCCs are the dictatorial SCC [3] and the SCC that picks the top-ranked alternatives of all agents. Observe that neither of these two SCCs satisfies the CA axiom. Our next

[3] A SCC S is dictatorial if there exists an agent $i \in N$ such that for every profile P, $S(P) = t^1(P_i)$.

result states that this holds generally; minimality, unanimity, implementability and CA are irreconcilable.

Proposition 11 *Assume $n = 2$. There does not exist a unanimous, minimal and implementable SCC $S : \mathcal{P} \to 2^X \setminus \emptyset$ satisfying CA.*

In order to prove the result we will use the lemma below.

Lemma 1 *Let $S : \mathcal{P} \to 2^X \setminus \emptyset$ be a strategy-resistant SCC for an arbitrary number of agents and alternatives. There exists mappings $\psi_i : \mathcal{P}_{-i} \to 2^X$, $\forall i \in N$ such that $[\tau(P_i, \psi_i(P_{-i}))] \in S(P)$ for all P, i.e. S is a super-correspondence of $[\tau(P_i, \psi_i(P_{-i}))]$. If S is minimal, then we have $S(P) = [\tau(P_i, \psi_i(P_{-i}))]$.*

Proof: We first show that $\tau(P_i, \psi_i(P_{-i})) \in S(P)$. We define $\psi_i(P_{-i}) = \cup_{P_i}\{S(P_i, P_{-i})\}$. Therefore, $\tau(P_i, \psi_i(P_{-i})) = \tau(P_i, \cup_{P_i} S(P_i, P_{-i}))$ and let $\tau(P_i, \psi_i(P_{-i})) = x$. To see that $x \in S(P)$, we use the method of contradiction. Suppose $x \notin S(P)$. Since $\tau(P_i, \psi_i(P_{-i})) = x$, there must be P_i', $P_i' \neq P_i$, such that $x \in S(P_i', P_{-i})$. Since $x = \tau(P_i, \cup_{P_i}\{S(P_i, P_{-i})\})$, we must have $xR_iy, \forall y \in S(P)$. Therefore player i can manipulate via P_i'. This is a violation of the strategy-resistance. Therefore $\tau(P_i, \psi_i(P_{-i})) \in S(P)$ for all $i \in N$.

Since S is minimal, it is easy to show that $S(P) = [\tau(P_i, \psi_i(P_{-i}))]$. This follows from the following argument. We have to show that $S(P)$ does not consist of an alternative other than $[\tau(P_i, \psi_i(P_{-i}))]$. We prove this by the method of contradiction. Let $z \in S(P)$ where $z \neq \tau(P_i, \psi_i(P_{-i}))$, for $i \in N$. Since S is minimal and $z \in S(P)$ then it must be that $z = \tau(P_j, S(P_j, P_{-j}))$ for $j \in N$. Let $\tau(P_j, \psi_j(P_{-j}) = z'$, $z \neq z'$. Since $\psi_j(P_{-j}) = \cup_{P_j} S(P_j, P_{-j})$, we must have $z'R_jx$, for $x \in S(P)$. Thus $z'R_jz$, since $z \in S(P)$. But this contradicts $z = \tau(P_j, S(P_j, P_{-j}))$.

■

We now return to the main proof. *Proof*: Let S be an implementable SCC satisfying unanimity and minimality. From Lemma 1 we have $\psi_i : \mathcal{P}_{-i} \rightarrow 2^X$, $\forall i \in N$ such that $S(P) = [\tau(P_i, \psi_i(P_j))]$ for all P. Since S is unanimous $t^1(P_j) \in \psi_i(P_j) \forall i, j; i \neq j$. This can be shown by the method of contradiction. Without loss of generality, we assume that $t^1(P_2) \notin \psi_1(P_2)$. Let $t^1(P_2)$ be denoted by y. Let $P_1^* \in \mathcal{P}_1$ such that $t^1(P_1^*) = t^1(P_2) = y$. We know that $\tau(P_1^*, \psi_1(P_2)) \in S(P_1^*, P_2)$. But $y \notin \psi_1(P_2)$ and hence $\tau(P_1^*, \psi_1(P_2)) \neq y$. Let $\tau(P_1^*, \psi_1(P_2)) = x$, $x \neq y$. Thus $x \in S(P_1^*, P_2)$. But since S is unanimous and $t^1(P_1^*) = t^1(P_2) = y$, it must be that $S(P_1^*, P_2) = y$. Thus we arrive at a contradiction.

To see that S cannot satisfy CA we assume that the converse is true. Let P be a profile such that $S(P) = b$, where b is a compromise. Let $t^1(P_1) = x, t^2(P_2) = y$. Thus $x \in \psi_2(P_1)$ and $y \in \psi_1(P_2)$. It must be that bP_1y, bP_2x, since b is an efficient alternative at P. Let $P_1' \in \mathcal{P}_1$ where rankings of all alternatives are same as in P_1, except that y and b have interchanged their ranks in P_1'. Let $P_2' \in \mathcal{P}_2$, where rankings of all alternatives are same as in P_2, except that x and b have interchanged their ranks in P_2'. We know that $S(P_1, P_2') = \{\tau(P_1, \psi_1(P_2')), \tau(P_2', \psi_2(P_1))$. Since $x \in (\psi_2(P_1)$, the best alternative in $(\psi_2(P_1)$ according to P_2' is x and thus $x \in S(P_1, P_2')$. This is because, player 2 cannot manipulate to get an alternative which is ranked above b in P_2 or above x in P_2'. This follows from the strategy-resistance of S.

Since S is minimal, $S(P_1, P_2')$ contains only x. This is because, if $S(P_1, P_2')$ would contain an alternative t, $t \neq x$, then t would not be the best alternative in $S(P_1, P_2')$ for an agent according to the orderings P_1 or P_2'. Thus minimality is violated. Similarly $S(P_1', P_2) = y$. We consider the profile $P_1', P_2\prime$. and the mechanism which implements S. Since top-ranked alternative in P_1' is x, there exists a message that is undominated at P_1' and results x together with all messages that are undominated at P_2'. [We call this "Observation 1"]

But the top alternative of P_2' is y. Thus at least an undominated message at P_2' results in y together with all undominated messages at P_1'. [We call this "Observation 2"]

We complete the proof by noting that Observations 1 and 2 contradict each other.

■

The next example shows that minimality is critical for Proposition 11. In particular we show that there exists a unanimous and implementable SCC $S : \mathcal{P} \to 2^X \setminus \emptyset$ satisfying CA in the case $n = 2$ and $m = 3$.

Example 17 We use the same environment and notation as in Example 12. In Table 3.10 we present an unanimous SCC S^* for two agents and three alternatives. We note that S^* satisfies unanimity and $S^*(P_1^1, P_4^2) = b$. Thus S^* also satisfies CA. We prove that S^* is implementable. In particular the mechanism in Table 3.11 can be used to implement S^*.

In this mechanism m_1^1 are m_2^1 undominated at P_1^1 and P_2^1 respectively. There are two strategies, m_3^1 and m_4^1, that are undominated at P_3^1. m_3^1 is undominated at P_4^1 and m_5^1 is undominated at both P_5^1 and P_6^1. We note that this is a symmetric mechanism i.e. the agent 2 has the same undominated strategies as the agent 1 at different orderings.

Whether Borgers' result (Borgers (1991)) can be extended to an arbitrary number of agents and alternatives, is an open question. We consider an environment with an arbitrary number agents and alternatives, where an agent, i, has an arbitrarily fixed ordering $\bar{P}^i$. Thus $\mathcal{D}_i = \{\bar{P}_i\}$. We say that this agent is a "passive" agent. Other agents have all preference orderings over X. Let $E^d = (N, X, \tilde{\mathcal{D}})$ denote this environment. We have the following result.

	P_1^2	P_2^2	P_3^2	P_4^2	P_5^2	P_6^2
P_1^1	$\{a\}$	$\{a\}$	$\{a,b\}$	$\{b\}$	$\{b\}$	$\{b\}$
P_2^1	$\{a\}$	$\{a\}$	$\{a,c\}$	$\{c\}$	$\{b\}$	$\{b\}$
P_3^1	$\{a,b\}$	$\{a,c\}$	$\{c\}$	$\{c\}$	$\{b\}$	$\{b\}$
P_4^1	$\{b\}$	$\{c\}$	$\{c\}$	$\{c\}$	$\{b\}$	$\{b\}$
P_5^1	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$
P_6^1	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$

Table 3.10: The SCC S^*

	m_1^2	m_2^2	m_3^2	m_4^2	m_5^2
m_1^1	a	a	b	a	b
m_2^1	a	a	c	a	b
m_3^1	b	c	c	c	b
m_4^1	a	a	c	c	b
m_5^1	b	b	b	b	b

Table 3.11: The mechanism implementing S^*

Proposition 12 *There is an implementable SCC in environment E^d, efficient SCC satisfying CA when there is an arbitrary number of agents and alternatives.*

Proof: We prove it for an arbitrary number of agents and alternatives. Without loss of generality, we assume that agent 1 is the passive agent. Thus the ordering of agent 1 is fixed, say at $\bar{P}_1$. Thus the domain of orderings for the agent 1 is restricted to $\mathcal{D}_1 = \{\bar{P}_1\}$. The domains of orderings for the other agents are assumed to be complete, i.e. for an agent j, $j \neq i$, we have $\mathcal{D}_j = \mathcal{P}_j$.

Let S be a SCC such that the following holds:

for a profile, $\bar{P} = (\bar{P}_1, P_2, ..., P_n)$, where $P_2 \in \mathcal{P}_2$,....,$P_n \in \mathcal{P}_n$, we have

(i) $S(\bar{P}) = t^2(\bar{P}_1)$, if $t^2(\bar{P}_1)$ is an efficient alternative at $\bar{P}$;

(ii) $S(\bar{P}) = t^1(\bar{P}_1)$, if $t^2(\bar{P}_1)$ is inefficient at $\bar{P}$.

It is easy to check that S is a Pareto efficient SCC.

Let $\Gamma = (M, g)$ be a direct mechanism, i.e. $M_i = \mathcal{P}_i$, $\forall i \in N$. Also let $g(\bar{P}_1, P_2, ..., P_n) = S(\bar{P}_1, P_2, ..., P_n)$ for all $P_2 \in \mathcal{P}_2$,....,$P_n \in \mathcal{P}_n$.

Let $\bar{P}_2 \in \mathcal{P}_2$,....., $\bar{P}_n \in \mathcal{P}_n$ such that $t^1(\bar{P}_2) = t^3(\bar{P}_1)$ and $t^1(\bar{P}_j) = t^4(\bar{P}_1)$, for all $j \in N \setminus \{1, 2\}$. We note that $S(\bar{P}_1, \bar{P}_2, ..., \bar{P}_n) = t^2(\bar{P}_1)$. This is because b is efficient at $(\bar{P}_1, \bar{P}_2, ..., \bar{P}_n)$. We also note that $t^2(\bar{P}_1)$ is a compromise alternative at $(\bar{P}_1, \bar{P}_2, ..., \bar{P}_n)$. Thus S satisfies the CA. It is easy to check that telling the truth is the only undominated strategy at an ordering for an agent. Thus Γ implements an efficient SCC that satisfies the CA.

■

3.6 Conclusion

This chapter investigates some aspects of implementation in undominated strategies by finite mechanisms. We provide a condition called the NF condition that characterizes implementable SCCs in an environment with a single agent and three alternatives. We also provide another condition called the GM condition and show that this is necessary for implementable SCCs in a general environment. We also prove various possibility and impossibility results regarding SCCs satisfying the Compromise axiom and efficiency.

A full characterization of implementable SCCs remains very difficult. We are yet to prove whether the GM condition is also sufficient for implementation. Whether Borgers' (Borgers (1991)) impossibility result generalizes also remains open. In future we hope to address some of these questions.

Chapter 4

Appendices

4.1 Appendix I

Appendix 1 We give the proofs of Theorems 1 and 2 here.

Proof of Theorem 1 Necessity: Let C be a choice function from binary trees that satisfies BC and RI. Define the following binary relations: for all $a, b \in X$ (i) $a \sim_1 b$ if and only if $C(a, b) = a, C(b, a) = b$; (ii) $a \sim_2 b$ if and only if $C(a, b) = b, C(b, a) = a$; and (iii) $a \succ b$ if and only if $C(a, b) = C(b, a) = a$. We claim that

CLAIM 1. $\sim_i$ is symmetric for $i = 1, 2$;

CLAIM 2. $\succ$ is asymmetric;

CLAIM 3. $\sim_i$ is transitive for $i = 1, 2$.

CLAIM 4. $\succ$ is transitive.

Claims 1 and 2 follow obviously from the definitions of $\sim_i$ and $\succ$. Let us prove Claim 3.

Let $a \sim_1 b, b \sim_1 c$. Let $t \equiv (\{a,a\},\{b,c\})$. By BC, $C(t) = C(C(a,a), C(b,c))$. Since $a \sim_1 b$ and $b \sim_1 c$ we have $C(t) = C(a,b) = a$. By RI, $C(\{a,a\},\{c,c\}) = a$. By BC, $C(a,c) = a$ [Observation (1)]

Let $t' \equiv (\{c,c\},\{b,a\})$. Using the definitions of $\sim_i, i = 1,2$ and by BC, $C(t') = C(C(c,c), C(b,a))$=$C(c,b) = c$. By RI and replacing b by a, we get $C(\{c,c\},\{a,a\}) = c$. By BC, $C(c,a) = c$ [Observation (2)]

From observations (1) and (2), we conclude that $\sim_1$ is transitive. We can show similarly that $\sim_2$ is also transitive.

Next we show that $\succ$ is also transitive. Let $a \succ b$, $b \succ c$. Let $t \equiv (\{a,a\},\{b,c\})$. By BC and the definition of $\succ$, we get $C(t)$=$C(C(a,a), C(b,c)) = C(a,b) = a$. By RI, $C(\{a,a\},\{c,c\}) = a$. By BC, $C(a,c) = a$. Let $t'' \equiv (\{c,b\},\{a,a\})$. Applying BC and the definition of $\succ$ we get $C(t'') = C(C(c,b), C(a,a)) = C(b,a) = a$. RI implies that $C(\{c,c\},\{a,a\}) = a$. BC then implies $C(c,a) = a$. Thus $a \succ c$. Thus $\succ$ is transitive.

We prove following claims for any $\{a,b,c\} \in X$:

CLAIM 1. $a \succ b,\, b \sim_1 c \implies a \succ c$,

Let $a \succ b$ and $b \sim_1 c$. Let $t \equiv (\{a,a\},\{b,c\})$. By BC and using the binary relations, $C(C(a,a), C(b,c)) = C(a,b) = a$. By RI, $C(\{a,a\},\{c,c\}) = a$. Thus $C(a,c) = a$. Let $t' \equiv (\{b,c\},\{a,a\})$ and by BC, $C(t') = C(b,a) = a$. By RI again, $C(\{b,c\},\{a,a\}) = C(\{c,c\},\{a,a\}) = a$. Thus $C(c,a) = a$. Therefore $a \succ c$.

CLAIM 2. $a \succ b,\, b \sim_2 c \implies a \succ c$,

We can prove CLAIM 2 similarly.

CLAIM 3. $a \sim_1 b,\, b \succ c \implies a \succ c$, Let $a \sim_1 b$, and $b \succ c$. Let $t \equiv (\{a, b\}, \{b, c\})$. Applying BC and using the definitions of binary relations $\sim_1$ and $\succ$ we get $C(C(a, b), C(b, c)) = C(a, b) = a$. Applying RI we get $C(C(a, a), C(c, c)) = a$. But applying BC yields $C(C(a, a), C(c, c)) = C(a, c)$ and hence $C(a, c) = a$. Therefore only possible relations are $a \sim_1 c$ and $a \succ c$. But if $a \sim_1 c$ holds then using symmetry of $\sim_1$, we get $a \sim_1 c$ implies $c \sim_1 a$. Also using transitivity of $\sim_1$ and given $a \sim_1 b$ we get $c \sim_1 b$. But this is in contradiction with $b \succ c$. Thus $a \succ c$.

CLAIM 4. $a \sim_2 b,\ b \succ c \implies a \succ c$.

We can show similarly that CLAIM 4 holds.

CLAIM 5. It is not possible to have $a \sim_1 b$ and $b \sim_2 c$.

Suppose the claim is not true. Let $t \equiv (\{b, a\}, \{c, c\})$. Applying BC we get $C(t) = C(C(b, a), C(c, c))$. Given $a \sim_1 b$ and $b \sim_2 c$, we get $C(t) = C(b, c) = c$. Now applying RI, $C(\{a, a\}, \{c, c\}) = c$. Again BC implies that $C(a, c) = c$. We can show $a \sim_2 c$ is not possible. This is because had that been true, then $a \sim_2 c$, $b \sim_2 c$ or $c \sim_2 b$ (by symmetry of $\sim_2$) would imply $a \sim_2 b$ by transitivity of $\sim_2$. But this contradicts $a \sim_1 b$.

Note that $C(a, c) = c$ rules out the case $a \succ c$ and $a \sim_1 c$ directly. The only remaining possibility is $c \succ a$. Suppose this is true. Let $t' \equiv (\{c, c\}, \{a, b\})$. By BC, $C(t') = C(c, a) = c$. By RI and BC, we thus get $C(\{c, c\}, \{a, b\}) = c = C((c, c), (b, b)) = C(c, b)$. This contradicts $b \sim_2 c$. Thus $a \sim_1 b$ and $b \sim_2 c$ is not possible.

Define an order $\succeq^*$ as follows: for all $a, b \in X$, $a \succ^* b$ if $a \succ b$ and $a \sim^* b$ if either $a \sim_1 b$ or $a \sim_2 b$. It follows from earlier claims that $\succeq^*$ is complete, reflexive and transitive. Thus $\succeq^*$ is a weak ordering over the set of alternatives X. We define an indifference set I as a subset of X such that for any $x, y \in I$ we have $x \succeq^* y$ and $y \succeq^* x$. By CLAIM 5, all members in any indifference set are related to each other by binary relation $\sim_1$ or by $\sim_2$, but both cannot hold good. We define an indicator function $\delta(.)$ over indifference sets in following manner: for an indifference set I_i, $\delta(I_i) = 1$ if for any $x, y \in I_i, x \sim_1 y$ and $\delta(I_i) = 2$ if for any $x, y \in I_i, x \sim_2 y$.

Let $t \in \Gamma^2$ and suppose $C(t) = x_i^j$. We claim that $x_i^j \in M(X(t), \succeq^*)$. Let us prove it by contradiction. Suppose $x_i^j \notin M(X(t), \succeq^*)$. Therefore $\exists x_m^n \in X(t)$ such that $x_m^n \succ^* x_i^j$. We now define an algorithm: Apply BC sequentially on trees which are recursively concatenated to form t. Since t ia binary tree, X^n has two alternatives, one of which is x_m^n. Suppose the other alternative of X^n is x_l^n $(l, m = 1, 2; l \neq m)$. Consider the choice from the elementary binary tree represented by X^n. If x_m^n is chosen then $x_m^n \succeq^* x_l^n$, or if x_l^n is chosen then $x_l^n \succeq^* x_m^n$. We denote the chosen alternative from this first round by $x(1)$. In the next round of applying BC, we observe the choice problem from an elementary tree comprising of two alternatives one of which must be $x(1)$ ($= x_m^n$ or x_l^n, depending on the outcome in the last round of choice problem). The outcome from the present round of the choice problem, say $x(2)$, will certainly satisfy following: $x(2) \succeq^* x(1)$, which implies $x(2) \succeq^* x_m^n$ or $x' \succeq^* x_l^n$, depending on the outcome in the last round of sequential choice problem. We observe anyways $x(2) \succeq^* x_m^n$ by the transitivity property of $\succeq^*$. In fact this holds for outcome for any p-th round $x(p)$, i.e. $x(p) \succeq^* x_m^n$. We carry on applying BC sequentially to reach the last round, say the r-th round, where we face the problem $C(C(t_1), C(t_2))$ such that $t \equiv (t_1 o t_2)$ and $x(r-1)$ is the alternative that entered this last round of exercise we are carrying recursively and we have $x(r)$ that is the outcome of $C(C(t_1), C(t_2))$. Thus we have $x(r) \succeq^* x_m^n$. But since $C(t) = x_i^j$ we have $x(r) = x_i^j$. Thus $x_i^j \succeq^* x_m^n$. This is in contradiction to what we assumed, i.e. $x_m^n \succeq^* x_i^j$.

Let us characterize $C(t)$ when $M(X(t), \succeq^*)$ is not a singleton. We have $C(t) = x_i^j$ and we have proved that $x_i^j \in M(X(t), \succeq^*)$. Consider any other alternative from the set $M(X(t), \succeq^*)$. Suppose $x_{i'}^{j'} \in M(X(t), \succeq^*)$. We get from the definition of a maximal set

$x_i^j \succeq^* x_{i'}^{j'}$ and $x_{i'}^{j'} \succeq^* x_i^j$. Thus the maximal set is an indifference set characterized by either $\sim_1$ or $\sim_2$. W.l.o.g. assume that $x_i^j \sim_1 x_{i'}^{j'}$. Thus $\delta(M(X(t), \succeq^*)) = 1$. We claim that the left-most alternative in $M(X(t), \succeq^*)$ is chosen. We prove this below.

Suppose the left-most alternative in $M(X(t), \succeq^*)$ is x_i^j, i.e. i-th alternative from the left in the set X^j. We note that for any alternative x_m^n which is to the left of x_i^j, we have $x_i^j \succ^* x_m^n$ since x_i^j is the left-most alternative of $M(X(t), \succeq^*)$ in the tree t. We now consider the algorithm described earlier starting from x_i^j. We note that for each round p, $x(p)$ (the alternative that is chosen from the choice problem in p-th round) is x_i^j. This is because in each round p, staring from $p = 1$, if the elementary binary tree is of the form $\{x_i^j, y\}$ then choice is x_i^j. This is because if $y \in M(X(t), \succeq^*)$, then it must be that $x_i^j \sim_1 y$ and x_i^j is chosen. If $y \notin M(X(t), \succeq^*)$ then $x_i^j \succ^* y$ and thus x_i^j is chosen. If the elementary binary tree in round p is of the form $\{y, x_i^j\}$ then choice is again x_i^j. This is because if $y \notin M(X(t), \succeq^*)$ then $x_i^j \succ^* y$ and thus x_i^j is chosen. Now we claim that the tree cannot be of the form $\{y, x_i^j\}$ if $x_i^j \sim_1 y$. This is ensured from the fact that in t, x_i^j is the left-most alternative in the set $M(X(t), \succeq^*)$ and from the definition of the algorithm. We note that following this algorithm in the last round we get $x(r)$ which is $C(t)$. Thus $C(t) = x_i^j$.

We have shown that if a choice function C from binary trees satisfies BC and RI then there exists a weak order ($\succeq^*$) such that for any tree $t \in \Gamma^2$, we have $C(t) \in M(X(t), \succeq^*$). If $M(X(t), \succeq^*)$ contains more than one alternative, then the left-most alternative in $M(X(t), \succeq^*)$ is chosen if $\delta(x) = 1$ or the right-most alternative in $M(X(t), \succeq^*)$ is chosen if $\delta(x) = 2$. Thus C is MC.

*S*ufficiency: It is easy to verify that a mixed choice function satisfies BC and RI. Let C be a mixed choice function. If C is characterized by such an weak ordering ($\succeq^*$) over X, using an algorithm similar to the algorithm described earlier one can show that $C(t) = C(t')$, where $t \equiv \{t_1 o t_2\}$ and $t' \equiv (C(t_1), C(t_2))$. This is because $M(X(t), \succeq^*)$ is same as $M(X(t'), \succeq^*)$.

If maximal set is not singleton, the tie-breaking rule through indicator function is also same. It is easy to check C satisfies RI: suppose for $t \in \Gamma^2$, $C(t) = x$. Thus $x \in M(X(t), \succeq^*)$ or if $M(X(t), \succeq^*)$ is not singleton, then tie is broken according to the indicator function $\delta(M(X(t), \succeq^*))$. W.l.o.g. assume that the left-most alternative is chosen from $M(X(t), \succeq^*)$. Suppose any alternative $x'(\neq x)$ is replaced by its partner and the new tree be denoted by t'. We note that $M(X(t'), \succeq^*)$ is still x if $M(X(t), \succeq^*) = x$. If $M(X(t), \succeq^*)$ is not singleton (and x is the left-most or the right-most alternative in $M(X(t), \succeq^*)$ according to the indicator function), then we observe $M(X(t'), \succeq^*) \subset M(X(t), \succeq^*)$. Thus the indicator function chooses the same alterative, i.e. x, because even after above replacement x remains the left-most alternative in $M(X(t), \succeq^*)$. Thus we show C satisfies RI. ■

Next we prove Theorem 2. Before the proof we need to discuss few definitions and we also prove some lemmas.

Definition 25 Let $C : \Gamma^k \to X$. We define $a \succ b$, for any $a, b \in X$, if and only if $C(t) = a$ for any $t \in \Gamma^k_e(\{a, b\})$.

Definition 26 Let $C : \Gamma^k \to X$. We define $a \sim_r b$, for any $a, b \in X$, if and only if $C(t) = x(t, r)$, $1 \leq r \leq k$ for any $t \in \Gamma^k_e(\{a, b\})$.

Definition 27 Let $C : \Gamma^k \to X$. We define $a \leftrightarrow b$, for any $a, b \in X$, if and only if following holds: (i) for any $l : 1 \leq l \leq k$, $\exists t_1 \in \Gamma^k_e(\{a, b\})$ such that $C(t_1) \neq x(t_1, l)$; and (ii) for any $t, t' \in \Gamma^k_e(\{a, b\})$, $C(t) \neq C(t') \implies \exists r : 1 \leq r \leq k$, such that $C(t) = x(r; t)$ and $C(t') = x(r; t')$.

Definition 28 A choice function $C : \Gamma^k(\{a, b\}) \to \{a, b\}$, for any $a, b \in X$, is $\leftrightarrow$-based if following holds:

(i) $a \leftrightarrow b$.

(ii) for any $t \in \Gamma^k(\{a,b\})$, such that $t \equiv t_1$ o t_2 o ....t_k, $C(t) = C(C(t_1), C(t_2), ..., C(t_k))$.

We give an example to explain:

Example 18 Consider a $\leftrightarrow$-based choice function from ternary trees $C : \Gamma^3(\{a,b\}) \rightarrow \{a,b\}$. Suppose:

$C(a,b,b) = a$, $C(b,a,a) = a$.

$C(a,a,b) = a$, $C(b,b,a) = b$.

$C(a,b,a) = a$, $C(b,a,b) = b$.

Consider a tree $t \equiv (\{a,b,b\}$,$\{b,a,b\}$, $\{a,a,b)\})$.

Then we have $C(t) = C(C(a,b,b), C(b,a,b), C(a,a,b)) = C(a,b,a) = a$.

Lemma 2 *Let $C : \Gamma^k \rightarrow X$ and let C satisfy RI. Then for any $\{a,b\} \in X$ either of following holds: (i) $a > b$, (ii) $a \sim_r b$; $r : 1 \leq r \leq k$ or (iii) $a \leftrightarrow b$.*

Proof: Let $C : \Gamma^k_e(\{a,b\}) \rightarrow \{a,b\}$ and let C satisfy RI. There are two possibilities: (i) $\forall t \in \Gamma^k_e(\{a,b\})$, $C(t) = a$ (or b). (ii) $\exists A, A \subset \Gamma^k_e(\{a,b\})$ such that $C(t) = a$, $\forall t \in A$ and $C(t) = b$, $\forall t \in \Gamma^k_e(\{a,b\}) - A$. We observe that if (i) occurs then $a \succ b$ (or $b \succ a$). If (ii) occurs then $\forall t_1, t_2 \in \Gamma^k_e(\{a,b\})$, $C(t_1) \neq C(t_2) \implies \exists r : 1 \leq r \leq k$, such that $C(t_1) = x(r; t_1)$ and $C(t_2) = x(r; t_2)$. This can be shown as follows: suppose $C(t_1) = a$ and $C(t_2) = b$. Suppose $\nexists r : 1 \leq r \leq k$ such that $x(r; t_1) = a$, $x(r; t_2) = b$. Consider the set $I = \{i : 1 \leq i \leq k; x(i, t_1) = a\}$ and the set $J = \{j : 1 \leq j \leq k; x(j, t_2) = a\}$. It follows that $I \subset J$. Thus a occupies at least those positions in the tree t_2, which are occupied by a in the tree t_1. Thus following RI we get that $C(t_2) = C(t_1) = a$. This contradicts $C(t_2) = b$. We note that if $\exists r : 1 \leq r \leq k$ such that $C(t) = x(r;t)$, $\forall t \in \Gamma_e^k(\{a,b\})$, then $a \sim_r b$. Otherwise, if for any $l : 1 \leq l \leq k$, $\exists t_1 \in \Gamma_e^k(\{a,b\})$ such that $C(t_1) \neq x(t_1, l)$, then we have $a \leftrightarrow b$.

■

Lemma 3 *Let $C : \Gamma^k \to X$ and suppose C satisfies RI. Then for any $t_1, t_2 \in \Gamma_e^k(X'), X' \subseteq X$, $[C(t_1) \neq C(t_2)] \implies [\exists r : 1 \leq r \leq k$, such that $C(t_1) = x(r;t_1)$ and $C(t_2) = x(r;t_2)]$.*

Proof: Let $C : \Gamma^k \to X$. Suppose C satisfies RI. Consider choice from trees in $\Gamma_e^k(\{a,b\})$. There are three possibilities: (i) $\forall t \in \Gamma_e^k(\{a,b\})$, $C(t) = a$; (ii) $\forall t \in \Gamma_e^k(\{a,b\})$, $C(t) = b$; and (iii) $\exists A : A \subset \Gamma_e^k(\{a,b\})$, such that $C(t) = a$, $\forall t \in A$ and $C(t) = b$, $\forall t \in \Gamma_e^k(\{a,b\}) - A$. If (i) occurs, then clearly $a \succ b$ and similarly if (ii) occurs, then $b \succ a$. If (iii) occurs, we can sow that $\forall t_1, t_2 \in \Gamma_e^k(\{a,b\})$, $C(t_1) \neq C(t_2) \implies \exists r : 1 \leq r \leq k$, such that $C(t_1) = x(r;t_1)$ and $C(t_2) = x(r;t_2)$. This can be proved by contradiction: suppose $C(t_1) = a$ and $C(t_2) = b$ and $\nexists r : 1 \leq r \leq k$, such that $x(r;t_1) = a$, $x(r;t_2) = b$. Consider the set $I = \{i : 1 \leq i \leq k; x(i;t_1) = a\}$ and the set $J = \{j : 1 \leq j \leq k; x(j;t_2) = a\}$. It follows that $I \subseteq J$. Thus a appears at least in those positions in the tree t_2, which are occupied by a in the tree t_1. Following RI we must have $C(t_2) = C(t_1) = a$. But this contradicts $C(t_2) = b$.

■

Let $C : \Gamma^k \to X$, such that C satisfies BC and RI. We know that C is associated with either $\succ$, $\sim_r$ $(1 \leq r \leq k)$ or $\leftrightarrow$. Additionally we have following lemmas:

Lemma 4 *$\succ$ is transitive i.e. for any $a, b, c \in X$, $a \succ b$ and $b \succ c$ implies $a \succ c$.*

Proof: Suppose $a \succ b$ and $b \succ c$. Let $t \in \Gamma^k(\{a,b,c\})$. Let $t \equiv (X^1, X^2, ..., X^k)$ such that t can be represented as concatenation of k elementary k-ary trees where X^i (for all $i = 1, 2, 3, ..., k$) represents elementary k-ary tree. Also suppose for any $i : 1 \leq i \leq k$, $x_r^i = a$ for all $r : 1 \leq r \leq k$. Also suppose X^j (for any j: $1 \leq j \leq k$, $j \neq i$) is any elementary k-ary tree with only b and c labeled at the terminal nodes. Given $a \succ b$, $b \succ c$ and since C satisfies BC it must be that $C(t) = a$.

We replace b by c in X^j (for all j: $1 \leq j \leq k$, $j \neq i$) in t and get t'. Applying RI we have $C(t') = a$. Applying BC on t' we further obtain $C(t') = C(t'') = a$, where t'' is an elementary k-ary tree with $x(i; t'') = a$ and $x(l; t'') = c$, $l : 1 \leq l \leq k$, $l \neq i$. We observe in t'' only one position (i.e. the i-th position from the left) is occupied by a. It follows from RI that for any $t^* \in \Gamma_e^k(\{a, c\}$ such that $x(i; t^*) = a$, we must have $C(t^*) = a$. Since i is any arbitrary position in the k-ary tree it follows that $a \succ c$.

■

Lemma 5 *Suppose for any $a, b, c \in X$, $a \succ b$ and $b \sim_r c$. Then $a \succ c$.*

Proof: Let $a, b, c \in X$ such that $a \succ b$ and $b \sim_r c$. Let $t' \in \Gamma^k(\{a, b, c\})$ such that $t \equiv (X^1, X^2, ..., X^k)$, where t can be represented as a concatenation of k elementary trees and X^i ($i = 1, 2, ..., k$) represents an elementary k-ary tree. Also let for arbitrary $i : 1 \leq i \leq k$, $x_l^i = a$ for all $l : 1 \leq l \leq k$. Also for X^j (any $j : 1 \leq j \leq k$; $j \neq i$), let $x_r^j = b$ and $x_s^j = c$, for all $s : 1 \leq s \leq k$; $s \neq r$. Since $a \succ b$ and $b \sim_r c$ and applying BC we get $C(t) = a$.

In t we replace b by c in X^j for all $j : 1 \leq j \leq k$; $j \neq i$ to get tree t'. Applying RI we get $C(t') = a$. Applying BC we get $C(t'') = a$, where t'' is an elementary k-ary tree with $x(i, t'') = a$ and $x(t'', l) = c$ for all $l : 1 \leq l \leq k$; $l \neq i$. We observe that in t'' only one position (i-th position) is occupied by a and all other positions are occupied by c. Since initial choice of position i is arbitrary we can similarly show that for any $t \in \Gamma_e^k(\{a, c\})$ such that $x(j;t) = a$ (for any $j : 1 \leq j \leq k$) and $x(i;t) = c$ (for all $i : 1 \leq i \leq k$, $i \neq j$), we have $C(t) = a$. Applying RI we get for any $t \in \Gamma_e^k(\{a,c\})$, $C(t) = a$. This proves that $a \succ c$.

■

Lemma 6 *Consider any $a, b, c \in X$ such that $a \succ b$, $b \leftrightarrow c$. Then $a \succ c$.*

Proof: Similar to the proof of Lemma 5.

■

Lemma 7 *$\sim_r$ is transitive, i.e. for any $a, b, c \in X$, $a \sim_r b$ and $b \sim_r c$ implies $a \sim_r c$.*

Proof: Suppose $a \sim_r b$ and $b \sim_r c$. Let $t' \in \Gamma^k(\{a,b,c\})$ such that $t \equiv (X^1, X^2, ..., X^k)$, where t can be represented as a concatenation of k elementary trees and X^i ($i = 1, 2, ..., k$) represents an elementary k-ary tree. Let $x_l^r = a$ for all $l : 1 \leq l \leq k$. Also suppose for X^j (for all $j : 1 \leq j \leq k$; $j \neq r$) we have $x_r^j = b$ and $x_l^j = c$ for any $l : 1 \leq l \leq k$; $l \neq r$. Given that C satisfies BC and since $b \sim_r c$, we have $C(t) = a$.

We now replace b by c in each X^j (for any $j : 1 \leq j \leq k$, $r \neq j$) in the tree t to get t'. Applying RI we get $C(t') = a$. Let $t'' \in \Gamma_e^k(\{a,c\})$ such that $x(r;t'') = a$ and $x(l;t'') = c$ for all $l : 1 \leq l \leq k; l \neq r$. Again applying BC on t', and using $a \sim_r b$, we get $C(t'') = C(t') = a$. Similarly we can show $C(t''') = c$, where $x(r;t''') = c$ and $x(l;t''') = a$ for all $l : 1 \leq l \leq k$; $l \neq r$. Therefore $a \sim_r c$. ■

Lemma 8 *For any $a, b, c \in X$ if $a \sim_i b$ ($i : 1 \leq i \leq k$) then it is not possible to have $b \sim_j c$ ($j : 1 \leq j \leq k$, $j \neq i$).*

Proof: Let $a \sim_i b$ and $b \sim_j c$, $i, j : 1 \leq i, j \leq k$; $i \neq j$. Let $t \in \Gamma_e^k(\{a, b, c\})$ such that $x(i;t) = c$, $x(j;t) = a$ and $x(l;t) = b$, for all $l : 1 \leq l \leq k$; $l \neq i, j$. We can show that $C(t) = a$. Let $t' \in \Gamma_e^k(\{a, b\})$ such that $x(i;t') = b$, $x(j;t') = a$ and $x(l;t') = b$, for all $l : 1 \leq l \leq k$, $l \neq i, j$. Since $C(t) = a$, then applying RI we get $C(t') = a$ by replacing c by b in t. But given $a \sim_i b$ we have $C(t') = b$ because $x(i;t') = b$.

Following similar argument we can show that $C(t) \neq c$. If $C(t) = c$ then we use RI and replace a by b in t to get t'' such that $C(t'') = c$. But $x(j;t'') = b$ and $b \sim_j c$ implying $C(t'') = b$. Thus we have a contradiction. Similar argument shows $C(t) \neq b$ also. Thus it is impossible to have $a \sim_i b$ and $b \sim_j c; i \neq j$).

■

Lemma 9 *For any $a, b, c \in X$ it is not possible to have $a \sim_r b$ ($r : 1 \leq r \leq k$) and $a \succ c$.*

Proof: We prove it by contradiction. Let $a, b, c \in X$ such that $a \sim_r b$ and $a \succ c$. Let $t \in \Gamma_e^k$ such that $x(r;t) = c$, $x(m;t) = a$ $(r, m : 1 \leq r, m \leq k; m \neq r)$ and $x(n;t) = b$ for all $n : 1 \leq n \leq k; n \neq r, m$.

Clearly $C(t) \neq c$. This is because since C satisfies RI, $C(t) = c$ contradicts $a \succ c$. Similarly if we have $C(t) = a$, then we apply RI and replace c by b in the tree t to get tree t', such that $C(t') = a$. But $x(r;t') = b$ and $a \sim_r b$. This is a contradiction. Also if $C(t) = b$, then we apply RI again and replace c by a in t to get tree t'', such that $C(t'') = b$. But this contradicts $a \sim_r b$ because $x(r;t) = a$.

Thus we can show that for any $a, b, c \in X$ if $a \sim_r b$ $(r : 1 \leq r \leq k)$ then it is not possible to have $a \succ c$.

■

Lemma 10 *For any $a, b, c \in X$ it is not possible to have $a \leftrightarrow b$ and $b \succ c$.*

Proof: We prove it by contradiction. Let $a, b, c \in X$ such that $a \leftrightarrow b$ and $b \succ c$. Since $a \leftrightarrow b$, from the definition of $\leftrightarrow$ we have $\exists t_1 \in \Gamma_e^k\{a, b\}$ such that $C(t_1) = a$. Suppose $x(r; t_1) = a$ where $r : 1 \leq r \leq k$. We have either

(i) $\exists$ a tree $t_2 \in \Gamma_e^k\{a, b\}$ such that $x(r; t_2) = a$ and $C(t_2) = b$; Or

(ii) $\exists t_3, t_4 \in \Gamma_e^k\{a, b\}$, such that $x(r; t_3) = b$, $x(r; t_4) = b$, $C(t_3) = b$ and $C(t_4) = a$. This follows from $a \leftrightarrow b$.

Thus given $a \leftrightarrow b$ we must either have (i) $\exists t_1, t_2 \in \Gamma_e^k\{a, b\}$ such that $x(r; t_1) = x(r; t_2) = a$ and $C(t_1) = a$, $C(t_2) = b$; or (ii)$\exists t_3, t_4 \in \Gamma_e^k\{a, b\}$ such that $x(r, t_3) = x(r; t_4) = b$ and $C(t_3) = b$, $C(t_4) = a$.

W.l.o.g. we assume $\exists t_1, t_2 \in \Gamma_e^k\{a, b\}$, such that $x(r, t_1) = x(r, t_2) = a$, and $C(t_1) = a$, $C(t_2) = b$. Since $a \leftrightarrow b$, $\exists l; l : 1 \leq l \leq k; l \neq r$ such that $x(l; t_1) = a$ and $x(l; t_2) = b$. Also suppose $I = \{i | x(i; t_1) = b$, i.e. I is the set of all positions (from the left) in elementary k-ary tree such that for each such position i, we have $x(i; t_1) = b$. Clearly $r, l \notin I$.

Let $t^* \in \Gamma_e^k\{a, b, c\}$ such that $x(r; t^*) = a$, $x(l; t^*) = c$, $x(i; t^*) = b$, for all $i \in I$ and $x(j; t^*) = c$, $\forall j : 1 \leq j \leq k$; $j \neq r, l$; $j \notin I$.

Our claim is $C(t^*) \neq c$. This follows from $b \succ c$ and applying RI. We can also show $C(t^*) \neq b$. This is because if $C(t^*) = b$, then we can replace all c in t^* by a to get t^{**} and applying RI repetitively we get $C(t^{**}) = b$. But we note that $t^{**} = t_1$ and $C(t_1) = a$. This leads to a contradiction. Now let us show $C(t^*) \neq a$ by contradiction. Suppose $C(t^*) = a$. We can replace c by b in t^* in all positions occupied by c in t^* to get t^{***} and apply RI to get $C(t^{***}) = a$. We also know $C(t_2) = b$. We replace a by b in t_2 in all positions in t_2 occupied by a, except the r-th position and get t^{****}. Applying RI repetitively we get $C(t^{****}) = b$.

But we observe $t^{***} = t^{****}$ and we have $C(t^{***}) = a$. Hence we get a contradiction and prove the Lemma. ■

Lemma 11 *For any $a, b, c \in X$, it is not possible to have $a \sim_r b$ and $b \leftrightarrow c$.*

Proof: Suppose $a \sim_r b$, $b \leftrightarrow c$, where $a, b, c \in X$. Let $t \in \Gamma_e^k\{a, b, c\}$ such that $x(r; t) = c$ and $x(i; t) \in \{a, b\}$, for all $i : 1 \leq i \leq k; i \neq r$. Clearly $C(t) \neq a$ or b. To see this, assume to the contrary, $C(t) = a$. We replace c by b in t and applying RI we observe that the choice from the new tree should remain unaltered. But this violates $a \sim_r b$. Similarly we can show that $C(t) \neq b$. Thus $C(t) = c$. Let $t' \in \Gamma_e^k\{a, b, c\}$ such that $x(r; t') = c$, $x(j; t') = b$ for any arbitrary $j : 1 \leq j \leq k; j \neq r$, and $x(i; t') = a$, for all $i : 1 \leq i \leq k; i \neq r, j$. Since $C(t) = c$, applying RI we get $C(t') = c$. Let $t'' \in \Gamma_e^k\{a, b, c\}$ such that $x(r; t'') = b$, $x(j; t'') = c$, and $x(i; t'') = a$ for all $i : 1 \leq i \leq k; i \neq r, j$. Following Lemma 3 we have $C(t'') \neq a$. Also $C(t'') \neq b$. This is because if $C(t'') = b$, then using $C(t') = c$ and applying RI we can show that $b \sim_r c$. But this contradicts $b \leftrightarrow c$. Thus $C(t'') = c$. We observe that in t'' only j-th position is occupied by c and choice of this j-th position is arbitrary. We already have $C(t') = c$, where only r-th position is occupied by c. Thus, applying RI for any $t \in \Gamma^k\{b, c\}$ such that $x(i; t) = c$ (where $i : 1 \leq i \leq k$) and $x(j; t) = b$, for all $j : 1 \leq j \leq k; j \neq i$, we get $C(t) = c$. This implies $c \succ b$, which contradicts $b \leftrightarrow c$. Thus we can conclude that it is not possible to have $a \sim_r b$ and $b \leftrightarrow c$ for any $a, b, c \in X$. ■

Lemma 12 *For any $a, b, c \in X$, it is not possible to have $a \leftrightarrow b$ and $b \leftrightarrow c$.*

Proof: Suppose we have $a \leftrightarrow b$ and $b \leftrightarrow c$. Let $t_1 \in \Gamma_e^k\{a, b, c\}$ such that $x(i; t_1) = a$ for any arbitrary $i : 1 \leq i \leq k$ and $x(l; t_1) \in \{b, c\}$ for all $l : 1 \leq l \leq k; l \neq i$. We claim $C(t_1) \neq a$. We prove this by contradiction. Assume $C(t_1) = a$. Consider t_2 such that $x(i, t_2) = a$ and

$x(j, t_2) = b$ where $j : 1 \leq j \leq k$; $j \neq i$ and $x(l, t_2) = c$ for all $l : 1 \leq l \leq k$; $l \neq i, j$. Since $C(t_1) = a$, applying RI we get that $C(t_2) = a$. Consider t_3 such that $x(i, t_3) = b$, $x(j, t_3) = a$ and $x(l, t_3) = c$ for all $l : 1 \leq l \leq k$; $l \neq i, j$.

Following Lemma 3 we get $C(t_3) \neq c$. We check if $C(t_3) = b$. Since $C(t_2) = a$, $x(i; t_2) = a$ and $x(i; t_3) = b$, then if $C(t_3) = b$, applying RI we get $a \sim_i b$. But this contradicts $a \leftrightarrow b$. Thus $C(t_3) = a$.

We note that $C(t_1) = a$ and a occupies only i-th position in t_1. Consider t such that $x(i; t) = a$ and $x(l; t) = b$ for all $l : 1 \leq l \leq k$; $l \neq i$. Applying RI on $C(t_1) = a$, we get $C(t) = a$. Also we note that $C(t_3) = a$ and a occupies only j-th position in t_3. Similarly for any t' such that $x(j; t') = a$, $x(l; t') = b$ for all $l : 1 \leq l \leq k$; $l \neq j$, we have $C(t) = a$. But choosing the position j was arbitrary. Thus we observe that choice from any tree t such that $x(j; t) = a$ (any $j : 1 \leq j \leq k$) and $x(l; t) = b$ for all $l : 1 \leq l \leq k$; $l \neq j$, we have $C(t) = a$. Applying RI one can show that for any $t \in \Gamma_e^k(\{a, b\})$, we have $C(t) = a$. But this implies $a \succ b$ which contradicts $a \leftrightarrow b$. Similarly we can show that for any $t_1 \in \Gamma_e^k\{a, b, c\}$ such that $x(i; t_1) = b$ (or c) for any arbitrary $i : 1 \leq i \leq k$ and $x(l; t_1) \in \{a, c\}(or\{a, b\})$ for all $l : 1 \leq l \leq k; l \neq i$, we cannot have $C(t_1) = b$ (or c).

Consider choosing from elementary k-ary trees formed with $\{a, b, c\}$, i.e. trees in $\Gamma_e^k\{a, b, c\}$. Consider the case when $k = 3$. Since $a \leftrightarrow b$ and $b \leftrightarrow c$, we cannot have either $C(a, b, c) = a$, $C(a, b, c) = b$ or $C(a, b, c) = c$. This is because in $t \equiv (a, b, c)$ each alternative occupies only single position. Thus it is clear that if $k = 3$ we cannot have $a \leftrightarrow b$ and $b \leftrightarrow c$.

For $k = 4$ consider t_1 and t_2 such that $t_1 \equiv (a, b, c, a)$ and $t_1 \equiv (a, b, b, c)$. Let $a \leftrightarrow b$ and $b \leftrightarrow c$. Again following the same argument as above $C(t_1) = a$ and $C(t_2) = b$. This is because in t_1 each of b and c occupies single position. Similarly in t_2 each of a and c occupies single position. Applying RI to $t_1 \equiv (a, b, c, a)$ yields $C(a, b, b, a) = a$, because

$C(a, b, c, a) = a$. But again applying RI to $t_2 \equiv (a, b, b, c)$ yields $C(a, b, b, a) = b$, because $C(a, b, b, c) = b$. Thus we arrive at a contradiction. Thus if $k = 4$ we cannot have $a \leftrightarrow b$ and $b \leftrightarrow c$.

For $k = 5$ we prove that we cannot have $a \leftrightarrow b$ and $b \leftrightarrow c$. Let $t_1 \equiv (c, b, a, a, b)$ and $t_2 \equiv (c, b, a, b, a)$. We first show that we cannot have $C(t_1) = C(t_2) = a$. To see this, assume to the contrary that $C(t_1) = C(t_2) = a$. Since $C(c, b, a, a, b) = a$, we have $C(c, b, a, c, b) \neq b$ following Lemma 3. $C(c, b, a, c, b) \neq a$ since a occupies single position in this tree. Thus we have $C(c, b, a, c, b) = c$. But $C(c, b, a, b, a) = a$ implies $C(c, b, a, c, a) = a$ following RI. Thus we arrive at contradiction.

We note that $C(t_1) \neq c$ and $C(t_2) \neq c$ since c occupies single position in both the trees t_1 and t_2. $C(t_1) = C(t_2) = a$ is already ruled out. Thus either $C(t_1) = b$ or $C(t_2) = b$ or both. Without loss of generality we assume that $C(t_1) = b$. We have $C(c, a, a, a, b) = a$ because both c and b occupy single position in this tree. Applying RI we get $C(c, a, a, a, c) = a$. Since $C(c, b, a, a, b) = b$ we have $C(c, b, a, a, c) = c$ applying Lemma 3 and also because b occupies single position. Applying RI we get $C(c, a, a, a, c) = c$. Thus we arrive at a contradiction. Hence we cannot have $a \leftrightarrow b$ and $b \leftrightarrow c$.

Next we prove that we cannot have $a \leftrightarrow b$ and $b \leftrightarrow c$ for $k \geq 6$. Consider those trees in $\Gamma_e^k\{a, b, c\}$ such that for each such tree t, $x(1; t) = c$, $x(2; t) = b$ and $x(3; t) = a$, i.e. first three positions in these trees are occupied by c, b and a respectively. Also in each t out of the last $k-3$ positions all the positions except one position are occupied by a. In particular we consider following trees:

$t_1 \equiv (c, b, a, \ldots, a, b)$

$t_2 \equiv (c, b, a, \ldots, a, b, a)$

.

.

$t_{k-3} \equiv (c, b, a, b, a, \ldots, a)$.

We first show it is not possible to have $C(t_i) = a$ for all $i = 1, 2, \ldots, (k-3)$. We prove this by contradiction. Suppose $C(t_i) = a$, for all $i = 1, 2, \ldots, (k-3)$. Thus $C(c, b, a, b, a, \ldots, a) = a$. Consider $C(c, b, a, b, c, a, \ldots, a)$. This cannot be b. This is because if $C(c, b, a, b, c, a, \ldots, a) = b$, then applying RI we get that $C(c, b, a, b, a, \ldots, a) = b$, which contradicts above.

Also $C(c, b, a, b, c, a, \ldots, a) \neq c$. To see this, we assume that $C(c, b, a, b, c, a, \ldots, a) = c$. Applying RI again we get that $C(c, b, a, a, c, a, \ldots, a) = c$. But we have assumed that $C(c, b, a, a, b, a, \ldots, a) = a$. Applying RI we get that $C(c, b, a, a, c, a, \ldots, a) = a$. Thus we arrive at a contradiction. Thus we must have $C(c, b, a, b, c, a, \ldots, a) = a$. Applying RI we get that $C(c, b, a, b, b, a, \ldots, a) = a$.

Continuing like this we can show $C(c, b, a, b, \ldots, b, a) = a$. But we started assuming $C(c, b, a, \ldots, a, b) = a$. This implies $C(c, b, a, c, \ldots, c, b) = c$. We can explain this as follows: $C(c, b, a, c, \ldots, c, b) \neq a$, because a occupies only one position in this tree. Also $C(c, b, a, c, \ldots, c, b) \neq b$, following Lemma 3. Using RI we get that $C(c, b, a, c, \ldots, c, a) = c$. But applying RI we can show that $C(c, b, a, b, \ldots, b, a) = a$ implies $C(c, b, a, c, \ldots, c, a) = a$. Thus we arrive at a contradiction.

Therefore it is not possible to have $C(t_i) = a$, for all $i = 1, 2, \ldots, (k-3)$. Thus $\exists t_j, j \in \{1, 2, \ldots, (k-3)\}$, such that $C(t_j) = b$ ($C(t_j) \neq c$, because c occupies only single position in $t_j, \forall j = 1, 2, ..(k-3)$). W.l.o.g. we assume $t_j \equiv (c, b, a, a, a, a, b, a, \ldots, a)$. Thus $C(t_j) = b$. But we must have $C(c, a, a, a, a, a, b, a, \ldots, a) = a$. This is because both b, c occupy single position in this tree. Therefore by applying RI we have $C(c, a, a, a, a, a, c, a, \ldots, a) = a$. [We call this observation (*)]

But we have $C(c, b, a, a, a, a, b, a, \ldots, a) = b$. Hence $C(c, b, a, a, a, a, c, a, \ldots, a) = c$. This follows from RI and Lemma 3. This implies, by RI, $C(c, a, a, a, a, a, c, a, \ldots, a) = c$. This contradicts observation (*). Hence we prove that we cannot have $a \leftrightarrow b$ and $b \leftrightarrow c$.

■

Proof of Theorem 2: Sufficiency is straightforward. We prove the necessity part in detail. Let $C : \Gamma^k \to X$ and let C satisfy BC and RI. It follows from Lemma 2 that C induces either $\succ$, or $\sim_r$ $(r : 1 \leq r \leq k)$ or $\leftrightarrow$ over any $\{a, b\} \in X$. Lemmas 2 - 9 further characterize these binary relations over X. These imply that there exists a weak ordering $\succeq^*$ over X defined as follows: $a \succ^* b$, if $a \succ b$ and $a \sim^* b$, if $a \sim_r b$ or $a \leftrightarrow b$. We observe that if $a \succ^* b$ then $\nexists c \in X$ such that $a \sim^* c$. This follows from Lemmas 5 and 6, which says that following cannot happen: $a \succ b$ and $a \sim_r b$ or $a \leftrightarrow b$. Thus $\succeq^*$ is an admissible ordering.

Let $t \in \Gamma^k$. Since C satisfies BC we apply BC sequentially on trees which are recursively concatenated to form the tree t. It follows that $C(t) = M(X(t), \succeq^*)$ if $X(t) \cap (X - BT(\succeq^*)) \neq \phi$ (where $BT(\succeq^*)$ is the set of bottom-ranked alternatives in $\succeq^*$). This is because in this case $M(X(t), \succeq^*)$ is unique and also is the outcome while applying BC sequentially on the tree t.

If $X(t) \subset BT(\succeq^*)$, then we can have two possibilities: (a) $|BT(\succeq^*)| > 2$: it must be the case that for any $x, y \in BT(\succeq^*)$, $x \sim_r y$ $(1 \leq r \leq k)$. This follows from Lemma 8, Lemma 11 and Lemma 12: Lemma 8 says that it is not possible to have $a \sim_i b$ and $a \sim_j c$, $i \neq j$ for any $a, b, c \in X$; Lemma 11 says that it is not possible to have $a \sim_r b$ and $a \leftrightarrow c$ for any $a, b, c \in X$; and Lemma 12 says that it is not possible to have $a \leftrightarrow b$ and $a \leftrightarrow c$ for any $a, b, c \in X$. In this case since $X(t) \subset BT(\succeq^*)$, applying BC sequentially on the tree t as mentioned above yields $C(t) = x(r;t)$;

(b) $|BT(\succeq^*)| = 2$: it must be the case that for $x, y \in BT(\succeq^*)$, $x \sim_r y$ or $x \leftrightarrow y$. If $x \sim_r y$ then $C(t) = x(r;t)$ whenever $X(t) = \{x, y\}$ and if $x \leftrightarrow y$ then C is $\leftrightarrow$-based whenever $X(t) = \{x, y\}$. Following definition 15 this implies that if $X(t) \subset BT(\succeq^*)$ and $|BT(\succeq^*)| = 2$ then C must be a mutually consistent choice function. Thus we prove that C must be a HYBRID. ■

Next we provide the proof of Theorem 3.

Proof of Theorem 3: Sufficiency is straightforward. We prove the necessity in detail. Let $C : \Gamma \to X$ and let C satisfy BC and RI. Since $\Gamma^k \subset \Gamma$, $k \geq 2$, C also defines choice functions from k-ary trees. Since C satisfies BC and RI, using the results in sections 4 and 5 we know that C must define an MC (say, C^2) over Γ^2 and a HYBRID choice function (say, C^k) over Γ^k, $k \geq 3$. We know that an MC is associated with a binary-admissible preference ordering and a HYBRID is associated with an admissible preference ordering over X. Let R_2 be the binary-admissible ordering and let $\{R_k\}_{k \geq 3}$ be the admissible preference orderings associated with C.

It follows that for any $t \in \Gamma^k$; $k \geq 2$, we have $C(t) = C^k(t)$. We have noted that any mixed tree is formed after concatenating the trees recursively starting from elementary trees. Since C satisfies BC, for any mixed tree t, $C(t)$ can be obtained by concatenation operation sequentially and C function defined for trees in Γ^k above. We also show that the set (say, W) of orderings $\{R_k\}_{k \geq 2}$ must satisfy mutual consistency. To see this, assume to the contrary that the set of orderings $\{R_k\}_{k \geq 2}$ do not satisfy mutual consistency. Thus there must exist $R_i \in W$ and $x, y, z \in X$ such that (i) xP_iyP_iz or xP_iyR_iz or xR_iyP_iz, and (ii) there is a sequence of orderings $R^1, R^2, \ldots R^k \in W$ and a sequence of alternatives $v_0, v_1, \ldots v_k \in X$ such that $v_0 = z$, $v_k = x$ and $v_i R^{i+1} v_{i+1}$, $i = 0, 1, \ldots k-1$. We make following assumptions without loss of generality: $R_i = R_2$, xI_2yP_2z, $k = 1$, $R^1 = R_3 = I_3$. Further we assume that $I_2 = \sim_1$ and $I_3 = \sim_2$. These assumptions, however, can be generalized to other possibilities. Consider $t \equiv t_1$ o t_2 where $t_1 \equiv (\{x, z, z\}, \{x, z, z\}, \{x, z, z\})$ and $t_2 \equiv (\{y, y\}, \{y, z\})$. Here t_1 is a ternary tree and t_2 is a binary tree. Using BC and the definitions of the relations we have $C(t) = y$. We replace z by x in t and apply RI to get $C(t') = y$, where $t' \equiv t_1^*$ o t_2 and $t_1^* \equiv (\{x, x, x\}, \{x, x, x\}, \{x, x, x\})$ is the tree obtained by replacing z by x in t_1.

But we observe that $C(t_1^*) = x$ and using BC we get that $C(t') = C(C(t_1^*), C(t_2)) = C(x, y) =$ x. But this contradicts $C(t') = y$. Although we have shown that the set W must satisfy mutual consistency with specific assumptions regarding R_i, k etc., this can be shown for other cases as well in similar way. Hence the set of orderings W must satisfy mutual consistency. Thus C must be a G-HYBRID choice function. ■

4.2 Appendix II

We provide the proof of Theorem 4.

Proof of Theorem 4*:* We prove the necessity part first. Let S be an implementable SCC in environment E^0. Let $\Gamma = (M, g)$ be a mechanism that implements S where M_i is the message space for the agent. Let (x, y) be a Neighborhood Flip between P_i and P_i', where $P_i, P_i' \in \mathcal{P}$. Let $a \in S(P_i) - S(P_i')$. Since S is implementable, we have $m_i \in M_i$, such that $m_i \in D(\Gamma, P_i)$ and $g(m_i, m_{-i}) = a$, for $m_{-i} \in M_{-i}$. Also, since $a \notin S(P_i')$, we have m_i' which is undominated at P_i' and dominates m_i at P_i'. This also implies there exists $b \in X$, such that $g(m_i', m_{-i}) = b$ and $bP_i'a$. Thus $b \in S(P_i')$. It must be that either m_i' is undominated at P_i or m_i' is dominated by m_i^* which is undominated at P_i. We have two possibilities.

(1) If aP_ib then we do not require further conditions. This is because it is possible to construct

m_i such that it produces a together with dummy messages by the designer and m_i' such that it produces b together with dummy messages by the designer. Thus m_i dominates m_i' at P_i and m_i' dominates m_i at P_i'. In this case $m_i^* = m_i$. Since aP_ib and $bP_i'a$, we have $a = x$, $b = y$.

(2) If bP_ia then there are two possibilities: (i) m_i' is undominated at P_i; or (ii) m_i' is dominated by m_i^* which is undominated at P_i ($m_i^* \neq m_i$) and m_i' dominates m_i^* at P_i'. If (i) holds then we have a dummy message $m_{-i'}$ such that $g(m_i, m_{-i'})P_ig(m_i', m_{-i'})$ but $g(m_i', m_{-i'})P_i'g(m_i, m_{-i'})$. This is because, both m_i and m_i' are undominated at P_i and bP_ia. It must then be that $g(m_i, m_{-i'}) = x$ and $g(m_i', m_{-i'}) = y$. This implies that we must have $x, y \in S(P_i)$ and $y \in S(P_i')$. Since m_i' is undominated at P_i, we also have $b \in S(P_i)$.

If (ii) holds, then we have $g(m_i^*, m_{-i}) = c$ such that cR_ib and $bR_i'c$. Since m_i' is dominated by $m_i^* \in D(\Gamma, P_i)$ and $m_i^* \neq m_i$, we must have $x, y \in S(P_i)$ and $y \in S(P_i')$. This is because cP_ia and both m_i and m_i^* are undominated at P_i.

In either case $a \neq x$, because if $a = x$ and since $y \in S(P_i')$, we go back to (1). Also $a \neq y$, because $|X| = 3$ and x and y are contiguous in P_i and P_i'. Clearly, it must be that a is the bottom-ranked alternative in both P_i and P_i'. Thus yP_ia.

It thus follows that if $a \in S(P_i) - S(P_i')$, we have either (i) $a = x$ and $y \in S(P_i')$; or (ii) $x, y \in S(P_i)$, $y \in S(P_i')$ and yP_ia.

Next we prove sufficiency. Let $X = \{a, b, c\}$. Let $\mathcal{B} = \{(P, P')|P, P' \in \mathcal{P},\ t^3(P) = t^3(P')$ and P, P' has a neighborhood flip$\}$. We call (P, P') an admissible pair if $(P, P') \in \mathcal{B}$. Let S be an arbitrary SCC satisfying the NF condition. We describe an algorithm to construct a mechanism implementing S. Several cases need to be considered.

Case (A): For all $(P, P') \in \mathcal{B}$, we have $t^3(P) = t^3(P') \in S(P) \cap S(P')$.

In this case we have three dummy messages $\alpha_1, \alpha_2, \alpha_3$ for the mechanism designer and a single message m_1 for i. We have $g(m_1, \alpha_1) = a$, $g(m_1, \alpha_2) = b$ and $g(m_1, \alpha_3) = c$.

Case (B): There exists $(P^*, P^{**}) \in \mathcal{B}$, so that $t^3(P^*) \in S(P^*) \cup S(P^{**})$ but $t^3(P^*) \notin S(P^*) \cap S(P^{**})$. Without loss of generality let $t^3(P^*) \in S(P^*) - S(P^{**})$. We design the mechanism in the following steps.

Step $(B.1)$

Let $P : \{1, 2, 3, 4, 5, 6\} \to \mathcal{P}$ be a bijection such that $P(1) = P^*$, $P(2) = P^{**}$. Also let $P(j), P(j+1)$, $j = 2, 3, 4, 5$ have a neighborhood flip. For instance, if $aP(1)cP(1)b$ then we have $cP(2)aP(2)b$, $cP(3)bP(3)a$, $bP(4)cP(4)a$, $bP(5)aP(5)c$, $aP(6)bP(6)c$. There are dummy messages $\alpha_1, .., \alpha_l$, $l \leq 6$, depending on various possibilities,four dummy messages for the designer and 7 messages for the agent $m_1, m_2, m_3, m_4, m_5, m_6, m_7$. We have $g(m_1, \alpha_1) = t^3(P(1))$, $g(m_1, \alpha_2) = t^1(P(1))$ and $g(m_1, \alpha_3) = t^1(P(1))$. We also have $g(m_7, \alpha_1) = t^2(P(1))$, $g(m_7, \alpha_2) = t^2(P(1))$ and $g(m_7, \alpha_3) = t^1(P(1))$. Next we have (i) $g(m_2, \alpha_1) = t^2(P(1))$ and $g(m_2, \alpha_2) = t^2(P(1))$; (ii) $g(m_3, \alpha_1) = t^2(P(1))$ and $g(m_3, \alpha_2) = t^2(P(1))$; (iii) $g(m_4, \alpha_1) = t^3(P(1))$; $g(m_5, \alpha_1) = t^3(P(1))$; $g(m_6, \alpha_1) = t^3(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$ and $g(m_6, \alpha_3) = t^1(P(1))$. There are several subcases depending on whether $t^1(P(1)) \in S(P(2))$ or $t^1(P(1)) \notin S(P(2))$.

Subcase $B.1.1$: $t^1(P(1)) \in S(P(2))$.

There are four further subcases.

$B.1.1.1$: $t^1(P(1)) \in S(P(3)) \cap S(P(4))$.

There are six subcases that remain.

$B.1.1.1.(i)$: $t^3(P(1)) \notin S(P(3))$ and $t^2(P(1)) \notin S(P(4))$.

We let $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^1(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^1(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_5, \alpha_3) = t^1(P(1))$.

$B.1.1.1.(ii)$: $t^3(P(1)) \in S(P(3))$ and $t^2(P(1)) \notin S(P(4))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^1(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^1(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_5, \alpha_3) = t^1(P(1))$. We have another dummy message α_4 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_2, \alpha_4) = g(m_6, \alpha_4) = t^1(P(1))$, $g(m_3, \alpha_4) = g(m_4, \alpha_4) = g(m_5, \alpha_4) = t^3(P(1))$.

$B.1.1.1.(iii)$: $t^3(P(1)) \notin S(P(3))$, $t^2(P(1)) \in S(P(4))$, $t^2(P(1)) \notin S(P_5) \cup S(P_6)$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^1(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^1(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_5, \alpha_3) = t^1(P(1))$. We also have dummy messages α_4 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_5, \alpha_4) = g(m_6, \alpha_4) = t^1(P(1))$, $g(m_2, \alpha_4) = t^2(P(1))$, $g(m_3, \alpha_4) = g(m_4, \alpha_4) = t^2(P(1))$.

$B.1.1.1.(iv)$: $t^3(P(1)) \notin S(P(3))$, $t^2(P(1)) \in S(P(4))$, $t^2(P(1)) \in S(P_5) \cap S(P_6)$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^1(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^1(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_5, \alpha_3) = t^1(P(1))$. Next we have dummy messages α_4 and α_5 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_5, \alpha_4) = g(m_6, \alpha_4) = t^1(P(1))$, $g(m_2, \alpha_4) = t^2(P(1))$, $g(m_3, \alpha_4) = g(m_4, \alpha_4) = t^2(P(1))$. We have $g(m_1, \alpha_5) = g(m_7, \alpha_5) = g(m_2, \alpha_5) = g(m_3, \alpha_5) = g(m_4, \alpha_5) = g(m_5, \alpha_5) = g(m_6, \alpha_5) = t^2(P(1))$.

$B.1.1.1.(v)$: $t^3(P(1)) \in S(P(3))$, $t^2(P(1)) \notin S(P(4))$ and $t^2(P(1)) \notin S(P_5) \cup S(P_6)$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^1(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^1(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_5, \alpha_3) = t^1(P(1))$. We have dummy messages α_4 and α_5 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_2, \alpha_4) = g(m_6, \alpha_4) = t^1(P(1))$, $g(m_3, \alpha_4) = g(m_4, \alpha_4) = g(m_5, \alpha_4) = t^3(P(1))$ and $g(m_1, \alpha_5) = g(m_7, \alpha_5) = t^1(P(1))$, $g(m_2, \alpha_5) = g(m_3, \alpha_5) = g(m_4, \alpha_5) = t^2(P(1))$, $g(m_5, \alpha_5) = g(m_6, \alpha_5) = t^1(P(1))$.

$B.1.1.1.(vi)$: $t^3(P(1)) \in S(P(3))$, $t^2(P(1)) \notin S(P(4))$ and $t^2(P(1)) \in S(P_5) \cap S(P_6)$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^1(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^1(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_5, \alpha_3) = t^1(P(1))$. We have dummy messages α_4, α_5 and α_6 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_2, \alpha_4) = g(m_6, \alpha_4) = t^1(P(1))$, $g(m_3, \alpha_4) = g(m_4, \alpha_4) = g(m_5, \alpha_4) = t^3(P(1))$ and $g(m_1, \alpha_5) = g(m_7, \alpha_5) = t^1(P(1))$, $g(m_2, \alpha_5) = g(m_3, \alpha_5) = g(m_4, \alpha_5) = t^2(P(1))$, $g(m_5, \alpha_5) = g(m_6, \alpha_5) = t^1(P(1))$. Also we let $g(m_1, \alpha_6) = g(m_7, \alpha_6) = g(m_2, \alpha_6) = g(m_3, \alpha_6) = g(m_4, \alpha_6) = g(m_5, \alpha_6) = g(m_6, \alpha_6) = t^2(P(1))$.

$B.1.1.2$: $t^1(P(1)) \in S(P(3)) - S(P(4))$. There are three subcases that remain.

$B.1.1.2.(i)$: $t^2(P(1)) \notin S(P(4))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^3(P(1))$.

$B.1.1.2.(ii)$: $t^2(P(1)) \in S(P(4))$ and $t^2(P(1)) \notin S(P(5)) \cup S(P(6))$.

We let $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$, $g(m_5, \alpha_3) = t^3(P(1))$.

$B.1.1.2.(iii)$: $t^2(P(1)) \in S(P(4))$ and $t^2(P(1)) \in S(P(5)) \cap S(P(6))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^3(P(1))$. We have dummy message α_4 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_2, \alpha_4) = g(m_3, \alpha_4) = g(m_4, \alpha_4) = g(m_5, \alpha_4) = g(m_6, \alpha_4) = t^2(P(1))$.

$B.1.1.3$: $t^1(P(1)) \notin S(P(3)) \cup S(P(4))$. There are three sub-cases that remain.

$B.1.1.3.(i)$: $t^3(P(1)) \in S(P(3))$, $t^2(P(2)) \notin S(P(4))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = g(m_4, \alpha_3) = t^3(P(1))$,

$g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^3(P(1))$.

B.1.1.3.(ii): $t^3(P(1)) \in S(P(3))$, $t^2(P(2)) \in S(P(4))$ and $t^2(P(2)) \notin S(P(5)) \cup S(P(6))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$, $g(m_5, \alpha_3) = t^3(P(1))$.

B.1.1.3.(iii): $t^3(P(1)) \in S(P(3))$, $t^2(P(2)) \in S(P(4))$ and $t^2(P(2)) \in S(P(5)) \cap S(P(6))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$, $g(m_5, \alpha_3) = t^3(P(1))$.

B.1.1.4: $t^1(P(1)) \in S(P(4)) - S(P(3))$.

There are two subcases that remain.

B.1.1.4.(i): $t^2(P(1)) \notin S(P(5)) \cup S(P(6))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^3(P(1))$. We also have dummy message α_4 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_5, \alpha_4) = g(m_6, \alpha_4) = t^2(P(1))$ and $g(m_2, \alpha_4) = g(m_3, \alpha_4) = g(m_4, \alpha_4) = t^2(P(1))$.

B.1.1.4.(ii): $t^2(P(1)) \in S(P(5)) \cap S(P(6))$.

We have $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_2) = g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^3(P(1))$. Next we have dummy messages α_4 and α_5 for the designer, so that (i) $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_5, \alpha_4) = g(m_6, \alpha_4) = t^2(P(1))$, $g(m_2, \alpha_4) = g(m_3, \alpha_4) = g(m_4, \alpha_4) = t^2(P(1))$; and (ii) $g(m_1, \alpha_5) = g(m_7, \alpha_5) = g(m_2, \alpha_5) = g(m_3, \alpha_5) = g(m_4, \alpha_5) = g(m_5, \alpha_5) = g(m_6, \alpha_5) = t^2(P(1))$.

Subcase $B.1.2$: $t^1(P(1)) \notin S(P(2))$.

There are two possibilities which we discuss as subcases.

$B.1.2.1$: $t^2(P(1)) \notin S(P(4))$.

We have $g(m_2, \alpha_3) = t^2(P(1))$, $g(m_3, \alpha_3) = t^2(P(1))$, $g(m_4, \alpha_2) = g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^3(P(1))$.

$B.1.2.2$: $t^2(P(1)) \in S(P(4))$.

There are two possible subcases that remain.

$B.1.2.2.(i)$: $t^1(P(1)) \in S(P(4))$, $t^2(P(1)) \in S(P(5)) \cap S(P(6))$.

We let $g(m_2, \alpha_3) = t^2(P(1))$, $g(m_3, \alpha_3) = t^2(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^2(P(1))$, $g(m_5, \alpha_2) = g(m_5, \alpha_3) = t^1(P(1))$. We also have dummy message α_4 for the designer, so that $g(m_1, \alpha_4) = g(m_7, \alpha_4) = g(m_2, \alpha_4) = g(m_3, \alpha_4) = g(m_4, \alpha_4) = g(m_5, \alpha_4) = g(m_6, \alpha_4) = t^2(P(1))$.

$B.1.2.2.(ii)$: $t^1(P(1)) \in S(P(4))$, $t^2(P(1)) \notin S(P(5)) \cup S(P(6))$.

We have $g(m_2, \alpha_3) = t^2(P(1))$, $g(m_3, \alpha_3) = t^2(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_3) = t^2(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$, $g(m_5, \alpha_3) = t^1(P(1))$.

At the end of the Step $(B.1)$ we get a set of seven messages, $(m_1, m_2, m_3, m_4, m_5, m_6, m_7)$, for the agent, a set of dummy messages, $\alpha_1, .., \alpha_l$, $l \leq 6$ depending on various possibilities, for the designer and the outcomes. If there is no admissible pair $(P'', P''') \in \mathcal{B}$ such that $t^3(P'') \in S(P'') \cup S(P''')$ but $t^3(P'') \notin S(P'') \cap S(P''')$, then the algorithm for designing the mechanism ends after Step $(B.1)$. If there exists (P'', P''') such that $t^3(P'') \in S(P'') \cup S(P''')$ but $t^3(P'') \notin S(P'') \cap S(P''')$, then we move to Step $(B.2)$.

Step $B.2$

Without loss of generality, let $t^3(P'') = t^3(P''') \in S(P'') - S(P''')$. We again follow the algorithm developed in step (1) with $P'' = \tilde{P}(1)$ and $P''' = \tilde{P}(2)$, where $\tilde{P} : \{1, 2, 3, 4, 5, 6\} \rightarrow \mathcal{P}$ is a bijection, similar to P in step 1. We get a set of messages, $(\tilde{m_1}, \tilde{m_2}, \tilde{m_3}, \tilde{m_4}, \tilde{m_5}, \tilde{m_6}, \tilde{m_7})$, for the agent and a set of dummy messages, $\tilde{\alpha}_1, .., \tilde{\alpha}_k$, $k \leq 6$, for the designer and an outcome function $\tilde{g}$.

Step $B.3$

If there is another admissible pair $(P, P') \in \mathcal{B}$, such that $t^3(P) \in S(P) \cup S(P')$ and $t^3(P) \notin S(P) \cap S(P')$, then we similarly build another block of messages for the agent and the designer in Step $B.3$. Otherwise we move to Step $B.4$.

Step $B.4$

We note that we have got blocks of messages by the agent and dummy messages from Steps $B.1$, $B.2$ and $B.3$. In this step we concatenate these blocks to form a single block of messages by the agent and dummy messages. For sake of simplicity we show it for two blocks, but it can be extended for three blocks in similar way. Let $A : \{1, 2, 3, 4, 5, 6\} \to \{1, 2, 3, 4, 5, 6\}$ be a bijection so that $A(i) = j$ if $P(i) = \tilde{P}(j)$. We augment the mechanism obtained in Step 1 in the following way. We already have dummy messages, $\alpha_1, .., \alpha_k$, $k \leq 6$, for the designer from Step 1 and we now augment the message m_i, such that $g(m_i, \tilde{\alpha}_j) = \tilde{g}(\tilde{m}_{A(i)}, \tilde{\alpha}_j)$, for all $i \in \{1, 2, 3, 4, 5, 6\}$, $j = 1, .., k$. We augment the message m_7, such that $g(m_7, \tilde{\alpha}_j) = \tilde{g}(\tilde{m}_{A(1)}, \tilde{\alpha}_j)$, for all $j = 1, .., k$. We also have m_8 for the agent such that $g(m_8, \alpha_j) = g(m_{A^{-1}(1)}, \alpha_j)$, for all $j \leq l$ and $g(m_8, \tilde{\alpha}_j) = \tilde{g}(\tilde{m}_7, \tilde{\alpha}_j)$, for all $j \leq k$. Here we complete the description of Step $(B.4)$.

Case (C): For all $(P, P') \in \mathcal{B}$, we have $t^3(P) = t^3(P') \notin S(P) \cup S(P')$.

Let $(P^*, P^{**}) \in \mathcal{B}$ be an arbitrary admissible pair. Let $P : \{1, 2, 3, 4, 5, 6\} \to \mathcal{P}$ be a bijection such that $P(1) = P^*$, $P(2) = P^{**}$. Also let $P(j), P(j+1)$, $j = 2, 3, 4, 5$ have a neighborhood flip. For instance, if $aP(1)cP(1)b$ then we have $cP(2)aP(2)b$, $cP(3)bP(3)a$, $bP(4)cP(4)a$, $bP(5)aP(5)c$, $aP(6)bP(6)c$. We provide an algorithm for designing the mechanism $\Gamma = (M, g)$. We have three possibilities that we discuss as subcases.

Subcase $C.1$: $S(P(1)) = t^1(P(1))$.

we have messages $m_1, m_2, m_3, m_4, m_5, m_6$, for the agent, a dummy message α_1 for the designer and we let $g(m_1, \alpha_1) = t^1(P(1))$, $g(m_2, \alpha_1) = t^2(P(1))$, $g(m_3, \alpha_1) = t^2(P(1))$, $g(m_4, \alpha_1) = t^2(P(1))$, $g(m_5, \alpha_1) = t^1(P(1))$, $g(m_6, \alpha_1) = t^1(P(1))$. There are two possibilities which we describe as subcases.

C.1.1: $S(P(2)) = t^2(P(1))$.

There are two further subcases that need to be discussed.

C.1.1.1: $t^3(P(4)) \notin S(P(4))$.

No further step is required and we finish the algorithm for designing the mechanism.

C.1.1.2: $t^3(P(4)) \in S(P(4))$.

We have a dummy message α_2, for the designer such that $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^2(P(1))$, $g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$.

C.1.2: $t^1(P(1)) \in S(P(2))$.

We have a dummy message α_2, for the designer such that $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^1(P(1))$, $g(m_3, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$.

Subcase $C.2$: $S(P(1)) = t^2(P(1))$.

We have messages $m_1, m_2, m_3, m_4, m_5, m_6$, for the agent, a dummy message α_1, for the designer and we let $g(m_1, \alpha_1) = t^2(P(1))$, $g(m_2, \alpha_1) = t^2(P(1))$, $g(m_3, \alpha_1) = t^2(P(1))$, $g(m_4, \alpha_1) = t^3(P(1))$, $g(m_5, \alpha_1) = t^3(P(1))$, $g(m_6, \alpha_1) = t^3(P(1))$.

Subcase $C.3$: $S(P(1)) = \{t^1(P(1)), t^2(P(1))\}$.

We have messages $m_1, m_2, m_3, m_4, m_5, m_6$, for the agent, dummy messages α_1, α_2, for the designer and we have $g(m_1, \alpha_1) = t^2(P(1))$, $g(m_2, \alpha_1) = t^2(P(1))$, $g(m_3, \alpha_1) = t^2(P(1))$, $g(m_4, \alpha_1) = t^3(P(1))$, $g(m_5, \alpha_1) = t^3(P(1))$, $g(m_6, \alpha_1) = t^3(P(1))$. Next we have two possibilities which we discuss as subcases.

$C.3.1$: $S(P(2)) = t^2(P(1))$.

Let $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^2(P(1))$, $g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$.

$C.3.2$: $S(P(2)) = \{t^1(P(1)), t^2(P(1))\}$.

There are more subcases that need to be considered.

$C.3.2.1$: $t^2(P(1)) \in S(P(4))$.

We have messages $m'_1, m'_2, m'_3, m'_4, m'_5, m'_6$, for the agent and we have $g(m'_1, \alpha_1) = t^1(P(1))$, $g(m'_2, \alpha_1) = t^1(P(1))$, $g(m'_3, \alpha_1) = t^2(P(1))$, $g(m'_4, \alpha_1) = t^2(P(1))$, $g(m'_5, \alpha_1) = t^1(P(1))$, $g(m'_6, \alpha_1) = t^1(P(1))$, $g(m'_1, \alpha_2) = t^2(P(1))$, $g(m'_2, \alpha_2) = t^2(P(1))$, $g(m'_3, \alpha_2) = t^3(P(1))$, $g(m'_4, \alpha_2) = t^3(P(1))$, $g(m'_5, \alpha_2) = t^3(P(1))$, $g(m'_6, \alpha_2) = t^3(P(1))$, $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^1(P(1))$, $g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$.

C.3.2.2: $t^2(P(1)) \notin S(P(4))$.

We have $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^1(P(1))$, $g(m_3, \alpha_2) = t^3(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$.

Case (D): There exist $(P, P'), (P'', P''') \in \mathcal{B}$, such that $t^3(P) = t^3(P') \in S(P) \cap S(P')$ and $t^3(P'') = t^3(P''') \notin S(P'') \cup S(P''')$.

Let $P : \{1, 2, 3, 4, 5, 6\} \to \mathcal{P}$ be a bijection such that $P(1) = P$, $P(2) = P'$. Also let $P(j), P(j+1)$, $j = 2, 3, 4, 5$ have a neighborhood flip. For instance, if $aP(1)cP(1)b$ then we have $cP(2)aP(2)b$, $cP(3)bP(3)a$, $bP(4)cP(4)a$, $bP(5)aP(5)c$, $aP(6)bP(6)c$. We note that $(P(3), P(4))$ and $(P(5), P(6))$ are admissible pairs. Without loss of generality, we assume that $P'' = P(5)$ and $P''' = P(6)$. We provide an algorithm for designing the mechanism $\Gamma = (M, g)$.

We have messages $m_1, m_2, m_3, m_4, m_5, m_6$, for the agent, a dummy message α_1 for the designer and we let $g(m_1, \alpha_1) = g(m_2, \alpha_1) = g(m_3, \alpha_1) = g(m_4, \alpha_1) = g(m_5, \alpha_1) = g(m_6, \alpha_1) = t^3(P(1))$. There are four possibilities which we describe as subcases.

Subcase $D.1$: $S(P(1)) = S(P(2)) = S(P(3)) = S(P(4)) = S(P(5)) = S(P(6)) = t^3(P(1))$. In this case, we do not need to proceed further.

Subcase $D.2$: $S(P(1)) = \{t^1(P(1)), t^3(P(1))\}$. There are two possibilities which we describe as subcases.

$D.2.1$: $S(P(2)) = \{t^2(P(1)), t^3(P(1))\}$. We have another dummy message α_2 and we let $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^2(P(1))$, $g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$ and $g(m_6, \alpha_2) = t^1(P(1))$.

$D.2.2$: $S(P(2)) = \{t^1(P(1)), t^2(P(1)), t^3(P(1))\}$. There are three possibilities which we describe as subcases.

$D.2.2.1$: $S(P(4)) = \{t^2(P(1)), t^3(P(1))\}$. We have dummy messages α_2, α_3 and we let $g(m_1, \alpha_2) = t^1(P(1))$, $g(m_2, \alpha_2) = t^2(P(1))$, $g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = t^2(P(1))$, $g(m_5, \alpha_2) = t^1(P(1))$, $g(m_6, \alpha_2) = t^1(P(1))$, $g(m_1, \alpha_3) = t^1(P(1))$, $g(m_2, \alpha_3) = t^1(P(1))$, $g(m_3, \alpha_3) = t^3(P(1))$, $g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_3) = t^3(P(1))$, and $g(m_6, \alpha_3) = t^1(P(1))$.

$D.2.2.2$: $S(P(4)) = \{t^1(P(1)), t^3(P(1))\}$. We have dummy messages α_2, α_3 and we let $g(m_1, \alpha_2) = g(m_2, \alpha_2) = g(m_3, \alpha_2) = g(m_4, \alpha_2) = g(m_5, \alpha_2) = g(m_6, \alpha_2) = t^1(P(1))$, $g(m_1, \alpha_3) = t^1(P(1))$, $g(m_2, \alpha_3) = t^2(P(1))$, $g(m_3, \alpha_3) = t^2(P(1))$, $g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_3) = t^3(P(1))$, and $g(m_6, \alpha_3) = t^1(P(1))$.

$D.2.2.3$: $S(P(4)) = \{t^1(P(1)), t^2(P(1)), t^3(P(1))\}$. We have dummy messages α_2, α_3, α_4 and we let $g(m_1, \alpha_2) = g(m_2, \alpha_2) = g(m_3, \alpha_2) = g(m_4, \alpha_2) = g(m_5, \alpha_2) = g(m_6, \alpha_2) = t^1(P(1))$, $g(m_1, \alpha_3) = t^1(P(1))$, $g(m_2, \alpha_3) = t^2(P(1))$, $g(m_3, \alpha_3) = t^2(P(1))$, $g(m_4, \alpha_3) = t^3(P(1))$, $g(m_5, \alpha_3) = t^3(P(1))$, $g(m_6, \alpha_3) = t^1(P(1))$, $g(m_1, \alpha_4) = t^1(P(1))$, $g(m_2, \alpha_4) = t^2(P(1))$, $g(m_3, \alpha_4) = t^2(P(1))$, $g(m_4, \alpha_4) = t^2(P(1))$, $g(m_5, \alpha_4) = t^1(P(1))$, $g(m_6, \alpha_4) = t^1(P(1))$.

Subcase $D.3$: $S(P(1)) = \{t^2(P(1)), t^3(P(1))\}$. We have another dummy message α_2 and we let $g(m_1, \alpha_2) = t^2(P(1))$, $g(m_2, \alpha_2) = t^2(P(1))$, $g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = t^3(P(1))$, $g(m_5, \alpha_2) = t^3(P(1))$ and $g(m_6, \alpha_2) = t^3(P(1))$.

Subcase $D.4$: $S(P(1)) = \{t^1(P(1)), t^2(P(1)), t^3(P(1))\}$. There are two possibilities which we describe as subcases.

$D.4.1$: $S(P(2)) = \{t^2(P(1)), t^3(P(1))\}$. We have dummy messages α_2, α_3 and we let $g(m_1, \alpha_2) = g(m_2, \alpha_2) = g(m_3, \alpha_2) = t^2(P(1))$, $g(m_4, \alpha_2) = g(m_5, \alpha_2) = g(m_6, \alpha_2) = t^3(P(1))$, $g(m_1, \alpha_3) = t^1(P(1))$, $g(m_2, \alpha_3) = t^2(P(1))$, $g(m_3, \alpha_3) = t^2(P(1))$, $g(m_4, \alpha_3) = t^2(P(1))$, $g(m_5, \alpha_3) = g(m_6, \alpha_3) = t^1(P(1))$.

$D.4.2$: $S(P(2)) = \{t^1(P(1)), t^2(P(1)), t^3(P(1))\}$. We have dummy messages α_2, α_3, α_4 and we let $g(m_1, \alpha_2) = g(m_2, \alpha_2) = g(m_3, \alpha_2) = g(m_4, \alpha_2) = g(m_5, \alpha_2) = g(m_6, \alpha_2) =$

$t^1(P(1))$, $g(m_1,\alpha_3) = t^1(P(1))$, $g(m_2,\alpha_3) = t^2(P(1))$, $g(m_3,\alpha_3) = t^2(P(1))$, $g(m_4,\alpha_3) = t^3(P(1))$, $g(m_5,\alpha_3) = t^3(P(1))$, $g(m_6,\alpha_3) = t^1(P(1))$, $g(m_1,\alpha_4) = t^2(P(1))$, $g(m_2,\alpha_4) = t^2(P(1))$, $g(m_3,\alpha_4) = t^2(P(1))$, $g(m_4,\alpha_4) = t^3(P(1))$, $g(m_5,\alpha_4) = t^3(P(1))$, $g(m_6,\alpha_4) = t^3(P(1))$.

This completes the description of the algorithm for designing the implementing mechanism. We prove that the mechanism obtained from the algorithm described above implements S. We show it for all four possible cases, (A) - (D).

Case (A). In this case, since S satisfies the NF condition and the bottom-ranked alternative belongs to the image sets at all orderings, it must be that $S(P) = \{a, b, c\}$, for all $P \in \mathcal{P}$. The message space M_i for the agent consists of only one message m_1, which together with three dummy messages sent by the designer produces a, b and c. Clearly m_1 is undominated at all orderings in $\mathcal{P}$.

Case (B). There is at least one admissible pair of orderings with the same bottom-ranked alternative for both the orderings, such that, the bottom-ranked alternative is included only in one of the image sets of S at these two orderings. Clearly the bijection P described in the algorithm implies that $P(1)$ contains its bottom-ranked alternative while $P(2)$ does not, where $P(1)$ and $P(2)$ have the same bottom-ranked alternative. The mechanism constructed has at least seven messages for the agent and a set of dummy messages for the designer. The message m_1 produces the bottom-ranked alternative and the top-ranked alternative of $P(1)$, while m_1' produces the middle-ranked alternative of $P(1)$ together with two dummy messages sent by the designer. m_2 also produces the middle-ranked alternative of $P(1)$ together with these two dummy messages sent by the designer. Since the middle-ranked alternative of $P(1)$ is the top-ranked alternative of $P(2)$ and the top-ranked alternative of $P(1)$ is the middle-ranked alternative of $P(2)$, m_2 weakly dominates m_1 and m_1' at $P(2)$. Also m_1 and m_1' are undominated at $P(1)$ because m_1 produces the bottom-ranked alternative and the top-ranked alternative, while m_1' produces the middle-ranked alternative at $P(1)$ together with the dummy messages sent by the designer.

In step (1) of the algorithm we build a block with seven messages for the agent together with dummy messages sent by the designer and in step (2) we add another block constructed in the similar fashion. We note that these two blocks are conjoint together such that the following holds good: any two messages m and m' in the first block, which are undominated at P and P' respectively and m weakly dominates m' at P, m' weakly dominates m at P', continue to hold these properties when augmented by the messages in the second block. Clearly for all subcases, m_j is undominated at $P(j)$, $j = 2, .., 6$. Also m_j weakly dominates all other messages at $P(j)$, $j = 2, .., 6$. At $P(1)$, m_1 or m_1' weakly dominates all other messages for the agent. Thus the mechanism implements S.

In Cases (C) and (D), it is easy to verify that the construction of the algorithm itself ensures the mechanism implements S. ■

Bibliography

ATTALI, Y. AND M. BAR-HILLEL (2003): "Guess where: the position of correct answers in multiple-choice test items as a psychometric variable," *Journal of Educational Measurement*, 40, 109–128.

BORGERS, T. (1991): "Undominated Strategies and Coordination in Normalform Games," *Social Choice and Welfare*, 8, 65–78.

CHRISTENFELD, N. (1995): "Choices from Identical Options," *Psychological Science*, 6, 50–55.

CORCHÒN, L. (2009): *Encyclopedia of Complexity and Systems Editor Meyers, Robert A. Springer*, Springer, chap. The Theory of Implementation: What Did We Learn?, editor: Robert A. Meyers.

EHLERS, L. (2002): "Coalitional Strategy-proof House Allocation," *Journal of Economic Theory*, 105, 298–317.

GIBBARD, A. (1973): "The Manipulation of Voting Schemes: A General Result," *Econometrica*, 41, 587–601.

HARARY, F. (1969): *Graph theory*, Addison-Wesley Publishing Company, Inc.

JACKSON, M. O. (1992): "Implementation in Undominated Strategies: A Look at Bounded Mechanisms," *The Review of Economic Studies*, 59, 757–775.

——— (2001): "A Crash Course in Implementation Theory," *Social Choice and Welfare*, 18, 655–708.

Maskin, E. (1999): "Nash Equilibrium and Welfare Optimality," *The Review of Economic Studies*, 66, 23–38.

Moulin, H. (1979): "Dominance Solvable Voting Schemes," *Econometrica*, 47, 1337– 1351.

——— (1983): *The Strategy of Social Choice*, North Holland, Advanced Textbooks in Economics, No. 18, 214 p.

Ordeshook, P. and T. Schwartz (1987): "Agendas and the Control of Political Outcomes," *American Political Science Review*, 81, 179–200.

Plott, C. (1973): " Path Independence, Rationality, and Social Choice," *Econometrica*, 41, 1075–1091.

Rubinstein, A. and Y. Salant (2006): " A Model of Choice from Lists," *Theoretical Economics*, 1, 3–17.

Rubinstein, A., A. Tversky, and D. Hellers (1996): *Understanding Strategic Interaction: Essays in Honor of Reinhard Selten*, Springer Verlag, Berlin, ediors: W. Albers, W. Guth, P. Hammerstein, B. Moldovanu and E. van Damme.

Satterthwaite, M. (1975): "Strategy-proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions," *Journal of Economic Theory*, 10, 187–217.

Sen, A. (1993): " Internal Consistency of Choice," *Econometrica*, 61, 495–521.

Serrano, R. (2004): "The Theory of Implementation of Social Choice Rules," *SIAM Review*, 46, 377–414.

Simon, H. A. (1955): “ A Behavioral Model of Rational Choice,” *Quarterly Journal of Economics*, 69, 99–118.

Yamashita, T. (2010): “Worst-case Mechanism Design in Undominated Strategies,” Working Paper, Stanford University.