Stability and (Obviously) Strategy-proofness in Matching Theory

a dissertation presented
by
Pinaki Mandal
to
Economic Research Unit

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Quantitative Economics

Indian Statistical Institute
Kolkata, West Bengal, India
August 2021

Acknowledgments

I thank my supervisor Prof. Souvik Roy.

I express my deepest gratitude to Prof. Szilvia Pápai for her rock-solid support. She stood by me during the darkest times of my PhD, and went above and beyond any mentor/letter writer can ever do for a student, let alone thousands of miles away. She is an example to follow - both as an academic of excellence, as well as an admirable human being.

My warm thanks to Prof. Lars Ehlers for his constructive comments on my thesis. I also thank Prof. Manipushpak Mitra, Prof. Samarjit Das, Prof. Indraneel Dasgupta, and my former and current lab-mates.

Shout-out to Sanhita Sengupta, Satyaki Mukherjee, and Oishee Banerjee for their unconditional support (and for biting my head off). And most importantly, for my parents and my sister - no amount of thanks is enough, they rock!

Contents

Author List

All the chapters are joint work with my supervisor, Prof. Souvik Roy.

Listing of figures

List of Tables

1

Introduction

This thesis comprises of four chapters related to stability and (obviously) strategy-proofness in matching theory. A brief introduction of the chapters is provided below.

1.1 Obviously Strategy-proof Implementation of Assignment Rules: A New Characterization

In this chapter, we consider assignment problems where individuals are to be assigned at most one indivisible object and monetary transfers are not allowed. We provide a characterization of assignment rules that are Pareto efficient, non-bossy, and implementable in obviously strategy-proof (OSP) mechanisms. As corollaries of our result, we obtain a characterization of OSP-implementable fixed priority top trading cycles (FPTTC) rules, hierarchical exchange rules, and trading cycles rules. Troyan (2019) provides a characterization of OSP-implementable FPTTC rules when there are equal number of individuals and objects. Our result generalizes this for arbitrary values of those.

1.2 On Obviously Strategy-proof Implementation of Fixed Priority Top Trading Cycles with Outside Options

In this chapter, we study the implementation of a fixed priority top trading cycles (FPTTC) rule via an obviously strategy-proof (OSP) mechanism (Li, 2017) in the context of assignment problems with outside options, where agents are to be assigned at most one indivisible object and monetary transfers are not allowed. In a model *without* outside options, Troyan (2019) gives a sufficient (but not necessary) and Mandal & Roy (2020) give a necessary and sufficient condition for an FPTTC rule to be OSP-implementable. This paper shows that in a model *with* outside options, the two conditions (in Troyan (2019) and Mandal & Roy (2020)) are equivalent for an FPTTC rule, and each of them is necessary and sufficient for an FPTTC rule to be OSP-implementable.

1.3 Strategy-proof Allocation of Indivisible Goods when Preferences are Single-peaked

In this chapter, we consider assignment problems where heterogeneous indivisible goods are to be assigned to individuals so that each individual receives at most one good. Individuals have single-peaked preferences over the goods. In this setting, first we show that there is no strategy-proof, non-bossy, Pareto efficient, and strongly pairwise reallocation-proof assignment rule on a minimally rich single-peaked domain when there are at least three individuals and at least three objects in the market. Next, we characterize all strategy-proof, Pareto efficient, top-envy-proof, non-bossy, and pairwise reallocation-proof assignment rules on a minimally rich single-peaked domain as hierarchical exchange rules. We additionally show that strategy-proofness and non-bossiness together are equivalent to group strategy-proofness on a minimally rich single-peaked domain, and every hierarchical exchange rule satisfies group-wise reallocation-proofness on a minimally rich single-peaked domain.

1.4 Matchings under Stability, Minimum Regret, and Forced and Forbidden Pairs in Marriage Problem

In this chapter, we provide a class of algorithms, called men-women proposing deferred acceptance (MWPDA) algorithms, that can produce all stable matchings at every preference profile for the marriage problem. Next, we provide an algorithm that produces a minimum regret stable matching at every preference profile. We also show that its outcome is always women-optimal in the set of all minimum regret stable matchings. Finally, we provide an algorithm that produces a stable matching with given sets of forced and forbidden pairs at every preference profile, whenever such a matching exists. As before, here too we show that the outcome of the said algorithm is women-optimal in the set of all stable matchings with given sets of forced and forbidden pairs.

2

Obviously Strategy-proof Implementation of Assignment Rules: A New Characterization

2.1 Introduction

We consider the problem where a set of objects are to be allocated over a set of individuals based on the individuals' preferences over the objects. Each individual can receive at most one object. An assignment rule selects an allocation (of the objects over the individuals) at every collection of preferences of the individuals.

Pareto efficiency, *non-bossiness*, and *(group) strategy-proofness* are standard requirements of an assignment rule.[1] Pareto efficiency ensures that there is no other way to allocate the objects so that each individual is weakly better-off (and hence some individual is strictly better-off). Non-bossiness says that an individual cannot change the assignment of another one without changing her own assignment. Strategy-proofness ensures that no individual can be strictly better-off by misreporting her

[1] The concept of non-bossiness is due to Satterthwaite & Sonnenschein (1981).

(true) preference. Group strategy-proofness ensures the same for every group of individuals, that is, no group of individuals can be better-off by misreporting their preferences. Here, we say a group of individuals is better-off if each member in it is weakly better-off and some member is strictly better-off.

Pápai (2000) showed that an assignment rule is strategy-proof, non-bossy, Pareto efficient, and reallocation-proof if and only if it is a *hierarchical exchange rule*. A hierarchical exchange rule works in stages. In each stage, the objects (available in that stage) are owned by certain individuals who then trade their objects by forming top trading cycles.[2] Ownership of the objects at the start of each stage is determined by a collection of trees, called *inheritance trees* in Pápai (2000). As observed in Troyan (2019), the use of hierarchical exchange rules in practice is rare as participating individuals find it difficult to understand them, particularly the fact that these rules are strategy-proof.[3]

Obvious strategy-proofness (Li, 2017) came to the literature as a remedy by strengthening strategy-proofness in a way so that it becomes transparent to the participating individuals that a rule is not manipulable. The concept of obvious strategy-proofness is based on the notion of *obvious dominance* in an *extensive-form game*. A strategy s_i of an individual i in an extensive-form game is obviously dominant if, for any deviating strategy s_i', starting from any earliest information set where s_i and s_i' diverge, the best possible outcome from s_i' is no better than the worst possible outcome from s_i. An assignment rule is *obviously strategy-proof (OSP)* if one can construct an extensive-form game that has an equilibrium in obviously dominant strategies. By construction, OSP depends on the extensive-form game, so two games with the same normal form may differ on this criterion.[4]

This chapter characterizes the structure of OSP-implementable assignment rules subject to Pareto efficiency and non-bossiness. We introduce the notion of *dual ownership* for this purpose. A hierarchical exchange rule satisfies dual ownership if for each preference profile and each stage of the hierarchical exchange rule at that preference profile, there are at most two individuals who own all the objects available in that stage.[5] Thus, the dual ownership property makes it very simple for the (at most two) owners in any stage to trade: they only interchange their favorite objects. In contrast, for an arbitrary hierarchical exchange rule, there might be arbitrary number of individuals trading their favorite objects in a stage, which makes it harder to asses what would happen if they do not do this truthfully.

We show that an assignment rule is OSP-implementable, Pareto efficient, and non-bossy if and only if it is a hierarchical exchange rule satisfying dual ownership (Theorem 2.4.1). Since strategy-proofness and non-bossiness together are equiva-

[2] Top trading cycle (TTC) is due to David Gale and discussed in Shapley & Scarf (1974).

[3] Similar phenomena is also observed in other settings, see Chen & Sönmez (2006), Hassidim et al. (2016), Hassidim et al. (2017), Rees-Jones (2018), and Shorrer & Sóvágó (2018) for details.

[4] This verbal description of obvious strategy-proofness is adapted from Li (2017).

[5] Ehlers (2002) characterizes a class of assignment rules called *mixed dictator-pairwise-exchange rules* as the unique class of assignment rules that satisfy *efficiency* and *coalitional strategy-proofness* on the unique maximal domain (for which the mentioned axioms are compatible). These rules resemble the hierarchical exchange rules satisfying dual ownership.

lent to group strategy-proofness (see Pápai, 2000 for details), Theorem 2.4.1 can be reformulated in terms of group strategy-proofness (Corollary 2.4.1). We also show that a hierarchical exchange rule is OSP-implementable if and only if it satisfies dual ownership, and a *trading cycles rule* is OSP-implementable if and only if it is a hierarchical exchange rule satisfying dual ownership.[6]

Troyan (2019) introduces the notion of *dual dictatorship* in the context of fixed priority top trading cycles (FPTTC) rules.[7] It follows from Theorem 1 and Theorem 2 of his paper that dual dictatorship is both necessary and sufficient condition for an FPTTC rule to be OSP-implementable. However, there is a mistake in his characterization–although dual dictatorship is a sufficient condition for OSP-implementability of an FPTTC rule, it is *not* necessary.[8] Since FPTTC rules are special cases of hierarchical exchange rules (see Pápai, 2000 for details), we obtain as a corollary (Corollary 2.5.2) of our result that dual ownership is a necessary and sufficient condition for OSP-implementability of an FPTTC rule. It is worth mentioning that Troyan (2019) assumes that the number of individuals is the same as the number of objects, whereas we derive our results for arbitrary values of those.

As we have mentioned earlier, Pápai (2000) characterized hierarchical exchange rules as the only assignment rules satisfying strategy-proofness, non-bossiness, Pareto efficiency and reallocation-proofness. Our results complement hers in two ways. Firstly, whereas strategy-proofness, non-bossiness, and Pareto efficiency are desirable, reallocation-proofness is not that desirable. So, replacing strategy-proofness and reallocation-proofness by OSP-implementability, and characterizing the relevant class of hierarchical exchange rules is a significant contribution in our opinion. Secondly, hierarchical exchange rules are somewhat complicated for participants to understand. So, finding the class of such rules that can be implemented by obviously strategy-proof mechanisms is important for their application. Nevertheless, OSP-implementability is a desirable criteria on its own.

2.1.1 Related literature

Obvious strategy-proofness is introduced by Li (2017), who studies this property extensively for both the scenarios where monetary transfers are allowed and not allowed. When monetary transfers are not allowed, he analyses the implementability of serial dictatorship and top trading cycles rules under obvious strategy-proofness. Bade & Gonczarowski (2017) *constructively* characterize Pareto-efficient social choice rules that admit obviously strategy-proof implementations in popular domains (object assignment, single-peaked preferences, and combinatorial auctions). Pycia & Troyan (2019) characterize

[6]Trading cycles rules are introduced in Pycia & Ünver (2017) as generalization of hierarchical exchange rules. They show that an assignment rule is strategy-proof, non-bossy, and Pareto efficient if and only if it is a trading cycles rule.

[7]Troyan (2019) uses the term "TTC rule" to refer to an FPTTC rule in his paper.

[8]Theorem 2 in Troyan (2019) states that "weak acyclicity" and dual dictatorship are equivalent properties of an FPTTC rule. This result is correct on its own, however, because of the mistake in Theorem 1, it is not correct that an FPTTC rule is OSP-implementable if and only if it satisfies dual dictatorship.

the full class of obviously strategy-proof mechanisms in environments without transfers. They also introduce a natural strengthening of obvious strategy-proofness called *strong obvious strategy-proofness* to characterize the well-known *random priority mechanism* as the unique mechanism that is efficient and fair. Ashlagi & Gonczarowski (2018) consider two-sided matching with one strategic side and show that for general preferences, no mechanism that implements the men-optimal stable matching (or any other stable matching) is obviously strategy-proof for men. They also provide a sufficient condition for a deferred acceptance rule to be OSP-implementable. Later, Thomas (2020) provides a necessary and sufficient condition for the same.

2.1.2 Organization of the chapter

The organization of this chapter is as follows. In Section 2.2, we introduce basic notions and notations that we use throughout the chapter, define assignment rules and discuss their standard properties, and introduce the notion of obvious strategy-proofness. Section 2.3 introduces the notion of hierarchical exchange rules. In Section 2.4, we introduce the dual ownership property of a hierarchical exchange rule and present our main result (characterization of all OSP-implementable, Pareto efficient, and non-bossy assignment rules). In Section 2.5, we present a characterization of OSP-implementable hierarchical exchange rules, a characterization of OSP-implementable trading cycles rules, and a characterization of OSP-implementable FPTTC rules. We further discuss the relation between our result regarding FPTTC rules and that of Troyan (2019).

2.2 Preliminaries

2.2.1 Basic notions and notations

Let $N = \{1, \ldots, n\}$ be a (finite) set of individuals and A be a (non-empty and finite) set of objects. An ***allocation*** is a function $\mu : N \to A \cup \{\emptyset\}$ such that $|\mu^{-1}(x)| \leq 1$ for all $x \in A$. Here, $\mu(i) = x$ means individual i is assigned object x under μ, and $\mu(i) = \emptyset$ means individual i is not assigned any object under μ. We denote by $\mathcal{M}$ the set of all allocations. For $N' \subseteq N, A' \subseteq A$ such that $|N'| = |A'| \neq 0$, let $\mathcal{M}(N', A')$ denote the set of all bijections from N' to A'.

Let $\mathbb{L}(A)$ denote the set of all strict linear orders over A.[9] An element of $\mathbb{L}(A)$ is called a ***preference*** over A. For a preference P, let R denote the weak part of P, that is, for all $x, y \in A$, xRy if and only if $[xPy$ or $x = y]$. We assume that the set of admissible preferences of each individual is $\mathbb{L}(A)$. An element $P_N = (P_1, \ldots, P_n)$ of $\mathbb{L}^n(A)$ is called a ***preference profile***. Given a preference profile P_N, we denote by (P'_i, P_{-i}) the preference profile obtained from P_N by changing the preference of individual i from P_i to P'_i and keeping all other preferences unchanged. For $P \in \mathbb{L}(A)$ and non-empty

[9] A ***strict linear order*** is a semiconnex, asymmetric, and transitive binary relation.

$A' \subseteq A$, let $\tau(P, A')$ denote the most-preferred object in A' according to P, that is, $\tau(P, A') = x$ if and only if $\big[x \in A'$ and xPy for all $y \in A' \setminus \{x\}\big]$. For ease of presentation, we denote $\tau(P, A)$ by $\tau(P)$.

For ease of presentation we use the following convention throughout the chapter: for a set $\{1, \ldots, g\}$ of integers, whenever we refer to the number $g + 1$, we mean 1. For instance, if we write $s_t \geq r_{t+1}$ for all $t = 1, \ldots, g$, we mean $s_1 \geq r_2, \ldots, s_{g-1} \geq r_g$, and $s_g \geq r_1$.

2.2.2 Assignment rules and their standard properties

An ***assignment rule*** is a function $f : \mathbb{L}^n(A) \to \mathcal{M}$. For an assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ and a preference profile $P_N \in \mathbb{L}^n(A)$, let $f_i(P_N)$ denote the assignment of individual i by f at P_N.

An allocation μ ***Pareto dominates*** another allocation ν at a preference profile P_N if $\mu(i) R_i \nu(i)$ for all $i \in N$ and $\mu(j) P_j \nu(j)$ for some $j \in N$. An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is called ***Pareto efficient*** at a preference profile $P_N \in \mathbb{L}^n(A)$ if there is no allocation that Pareto dominates $f(P_N)$ at P_N, and it is called ***Pareto efficient*** if it is Pareto efficient at every preference profile in $\mathbb{L}^n(A)$.

Non-bossiness is a standard notion in matching theory which says that if an individual misreports her preference and her assignment does not change by the same, then the assignment of any other individual cannot change. Formally, an assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is ***non-bossy*** if for all $P_N \in \mathbb{L}^n(A)$, all $i \in N$, and all $\tilde{P}_i \in \mathbb{L}(A)$, $f_i(P_N) = f_i(\tilde{P}_i, P_{-i})$ implies $f(P_N) = f(\tilde{P}_i, P_{-i})$.

An individual i *manipulates* an assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ at a preference profile $P_N \in \mathbb{L}^n(A)$ via a preference $\tilde{P}_i \in \mathbb{L}(A)$ if $f_i(\tilde{P}_i, P_{-i}) P_i f_i(P_N)$. An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is ***strategy-proof*** if no individual can manipulate it at any preference profile.

Group strategy-proofness says that no group of individuals will have an incentive to misreport their preferences. More formally, a group of individuals $N' \subseteq N$ *manipulates* an assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ at a preference profile $P_N \in \mathbb{L}^n(A)$ via a collection of preferences $\tilde{P}_{N'} \in \mathbb{L}^{|N'|}(A)$ if $f_i(\tilde{P}_{N'}, P_{-N'}) R_i f_i(P_N)$ for all $i \in N'$ and $f_j(\tilde{P}_{N'}, P_{-N'}) P_j f_j(P_N)$ for some $j \in N'$. An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is ***group strategy-proof*** if no group of individuals can manipulate it at any preference profile.

2.2.3 Obviously strategy-proof assignment rules

Li (2017) introduces the notion of *obviously strategy-proof implementation*. We use the following notions and notations to present it.

We denote a rooted (directed) tree by T. For a tree T, we denote its set of nodes by $V(T)$, set of all edges by $E(T)$, root by $r(T)$, and set of leaves (terminal nodes) by $L(T)$. For a node $v \in V(T)$, we denote the set of all outgoing edges from v by $E^{out}(v)$. For an edge $e \in E(T)$, we denote its source node by $s(e)$. A *path* in a tree is a sequence of nodes such that every two consecutive nodes form an edge.

A *leaves-to-allocations* function $\eta^{LA} : L(T) \to \mathcal{M}$ assigns an allocation to each leaf of T, and a *nodes-to-individuals* function $\eta^{NI} : V(T) \setminus L(T) \to N$ assigns an individual to each internal node of T. An *edges-to-preferences* function $\eta^{EP} : E(T) \to 2^{\mathbb{L}(A)} \setminus \{\emptyset\}$ assigns each edge a subset of preferences satisfying the following criteria:

(i) for all distinct $e, e' \in E(T)$ such that $s(e) = s(e')$, we have $\eta^{EP}(e) \cap \eta^{EP}(e') = \emptyset$, and

(ii) for any $v \in V(T) \setminus L(T)$,

(a) if there exists a path $(v^1, \ldots, v^t)$ from $r(T)$ to v and some $1 \leq r < t$ such that $\eta^{NI}(v^r) = \eta^{NI}(v)$ and $\eta^{NI}(v^s) \neq \eta^{NI}(v)$ for all $s = r+1, \ldots, t-1$, then $\bigcup_{e \in E^{out}(v)} \eta^{EP}(e) = \eta^{EP}(v^r, v^{r+1})$, and

(b) if there is no such path, then $\bigcup_{e \in E^{out}(v)} \eta^{EP}(e) = \mathbb{L}(A)$.

An ***extensive-form assignment mechanism*** is defined as a tuple $G = \langle T, \eta^{LA}, \eta^{NI}, \eta^{EP} \rangle$, where T is a rooted tree, η^{LA} is a leaves-to-allocations function, η^{NI} is a nodes-to-individuals function, and η^{EP} is an edges-to-preferences function.

Note that for a given extensive-form assignment mechanism G, every preference profile P_N identifies a unique path from the root to some leaf in T in the following manner: for each node v, follow the outgoing edge e from v such that $\eta^{EP}(e)$ contains the preference $P_{\eta^{NI}(v)}$. If a node v lies in such a path, then we say that the preference profile P_N *passes through the node* v. Furthermore, we say two preferences P_i and P'_i of some individual i *diverge at a node* $v \in V(T) \setminus L(T)$ if $\eta^{NI}(v) = i$ and there are two distinct outgoing edges e and e' in $E^{out}(v)$ such that $P_i \in \eta^{EP}(e)$ and $P'_i \in \eta^{EP}(e')$.

For a given extensive-form assignment mechanism G, the ***extensive-form assignment rule*** f^G implemented by G is defined as follows: for all preference profiles P_N, $f^G(P_N) = \eta^{LA}(l)$, where l is the leaf that appears at the end of the unique path characterized by P_N.

In what follows, we define the notion of obvious strategy-proofness.

Definition 2.2.1. An extensive-form assignment mechanism G is ***Obviously Strategy-Proof (OSP)*** if for all $i \in N$, all nodes v such that $\eta^{NI}(v) = i$, and all $P_N, \tilde{P}_N \in \mathbb{L}^n(A)$ passing through v such that P_i and $\tilde{P}_i$ diverge at v, we have $f^G_i(P_N) R_i f^G_i(\tilde{P}_N)$.

An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is ***OSP-implementable*** if there exists an OSP mechanism G such that $f = f^G$.[10,11]

Remark 2.2.1. Every OSP-implementable assignment rule is strategy-proof (see Li, 2017 for details).

[10]Definition 2.2.1 is taken from Troyan (2019). However, his definition has a typo as it does not mention that P_N and $\tilde{P}_N$ must pass through v. We have corrected it here.
[11]An extensive-form assignment mechanism is called an ***OSP mechanism*** if it is OSP.

2.3 Hierarchical exchange rules

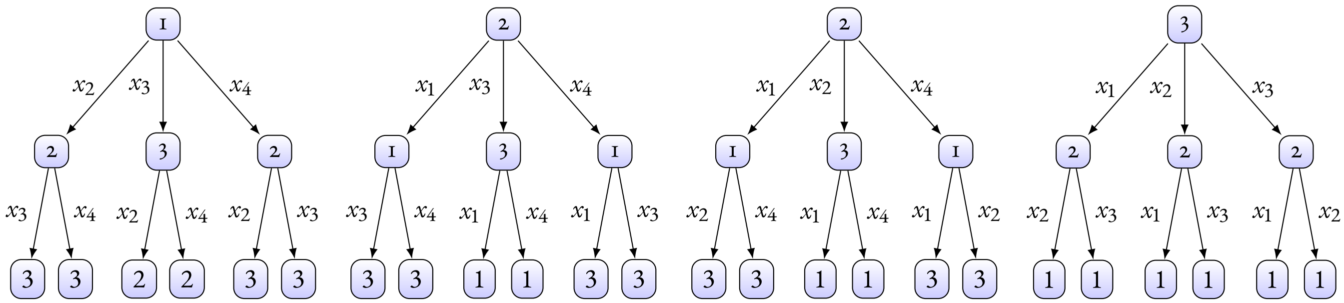
The notion of *hierarchical exchange rules* is introduced in Pápai (2000). We explain how such a rule works by means of an example.[12]

We begin with the notion of a *TTC procedure* with respect to a given endowments of the objects over the individuals. Suppose that each object is owned by exactly one individual (an individual may own more than one objects). A directed graph is constructed in the following manner. The set of nodes is the same as the set of individuals. There is a directed edge from individual i to individual j if and only if individual j owns individual i's most-preferred object. Note that such a graph will have exactly one outgoing edge from every node (though possibly many incoming edges to a node). Further, there may be an edge from a node to itself. It is clear that such a graph will always have a cycle. This cycle is called a *top trading cycle (TTC)*. After forming a TTC, the individuals in the TTC are assigned their most-preferred objects.

Example 2.3.1. Suppose $N = \{1, 2, 3\}$ and $A = \{x_1, x_2, x_3, x_4\}$. A hierarchical exchange rule is based on a collection of *inheritance trees*, one tree for each object.[13] Figure 2.1 presents a collection of inheritance trees $\Gamma_{x_1}, \ldots, \Gamma_{x_4}$. Consider Γ_{x_1} to have an understanding of their structure. Each maximal path of this tree has $\min\{|N|, |A|\} - 1 = 2$ edges. In any maximal path, each individual appears *at most* once at the nodes. For instance, individuals $1, 2$ and 3 appear at the nodes (in that order) in the left most path of Γ_{x_1}. Each object other than x_1 appears *exactly* once at the outgoing edges from the root (thus there are three edges from the root). For every subsequent node which is not the end node of a maximal path, each object other than x_1, that has *not* already appeared in the path from the root to that node, appears *exactly* once at the outgoing edges from that node. For instance, consider the node marked with 2 in the left most path of Γ_{x_1}. Since this node is not the end node of the left most maximal path and object x_2 has already appeared at the edge from the root to this node, objects x_3 and x_4 appear exactly once at the outgoing edges from this node. Thus, each object other than x_1 appears *at most* once at the edges in any maximal path of Γ_{x_1}. For instance, objects x_2 and x_3 appear at the edges (in that order) in the left most path of Γ_{x_1}. It can be verified that other inheritance trees have the same structure.

[12] See Pápai (2000) for an intuitive explanation of these rules.

[13] We define this notion formally in Section 2.3.1.



(a) Γ_{x_1} (b) Γ_{x_2} (c) Γ_{x_3} (d) Γ_{x_4}

Figure 2.1: Inheritance trees for Example 2.3.1

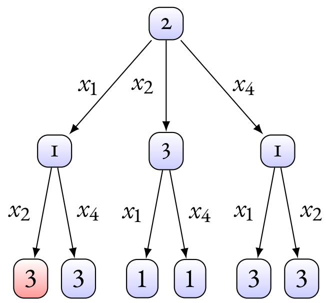
Consider the hierarchical exchange rule based on the collection of inheritance trees given in Figure 2.1 and consider the preference profile P_N such that $x_2P_1x_1P_1x_3P_1x_4$, $x_1P_2x_2P_2x_3P_2x_4$, and $x_1P_3x_2P_3x_3P_3x_4$. The outcome is computed through a number of stages. In each stage, endowments of the individuals are determined by means of the inheritance trees, and TTC procedure is performed with respect to the endowments.

Stage 1. In Stage 1, the "owner" of an object x is the individual who is assigned to the root-node of the inheritance tree Γ_x. Thus, object x_1 is owned by individual 1, objects x_2 and x_3 are owned by individual 2, and object x_4 is owned by individual 3. TTC procedure is performed with respect to these endowments to decide the outcome of Stage 1. Individuals who are assigned some object in Stage 1 leave the market with the corresponding objects. It can be verified that for the given preference profile P_N, individual 1 gets object x_2 and individual 2 gets object x_1. So, individuals 1 and 2 leave the market with objects x_2 and x_1, respectively.

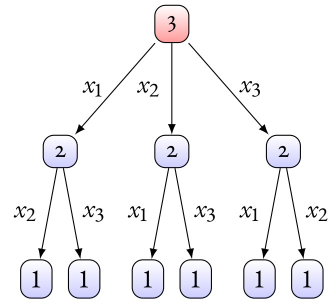
Stage 2. As in Stage 1, the endowments of the individuals are decided first and then TTC procedure is performed with respect to the endowments. To decide the owner of a (remaining) object x, look at the root of the inheritance tree Γ_x. If the individual who appears there, say individual i, is remained in the market, then i becomes the owner of x. Otherwise, that is, if i is assigned an object in Stage 1, say y, then follow the edge from the root that is marked with y. If the individual appearing at the node following this edge, say j, is remained in the market, then j becomes the owner of x. Otherwise, that is, if j is assigned an object in Stage 1, say z, then follow the edge that is marked with z from the current node. As before, check whether the individual appearing at the end of this edge is remained in the market or not. Continue in this manner until an individual is found in the particular path who is not already assigned an object and decide that individual as the owner of x.

For the example at hand, the remaining market in Stage 2 consists of objects x_3 and x_4, and individual 3. Consider object x_3. Individual 2 appears at the root of Γ_{x_3}. Since individual 2 is assigned object x_1 in Stage 1, we follow the edge from the root that is marked with x_1 and come to individual 1. Since individual 1 is assigned object x_2, we follow the edge marked with x_2 from this node and come to individual 3. Since individual 3 is remained in the market, she becomes the owner of

x_3. For object x_4, individual 3 appears at the root of Γ_{x_4} and she is remained in the market. So, individual 3 becomes the owner of x_4 in Stage 2. To emphasize the process of deciding the owner of an object, we have highlighted the node in red in the corresponding inheritance tree in Figure 2.2.



(a) Γ_{x_3}



(b) Γ_{x_4}

Figure 2.2: Stage 2

Once the endowments are decided for Stage 2, TTC procedure is performed with respect to the endowments to decide the outcome of this stage. As in Stage 1, individuals who are assigned some object in Stage 2 leave the market with the corresponding objects. It can be verified that for the current example, individual 3 gets object x_3 in this stage. So, individual 3 leave the market with objects x_3.

Stage 3 is followed on the remaining market in a similar way as Stage 2. For the current example, everybody is assigned some object by the end of Stage 2 and hence the algorithm stops in this stage. Thus, individuals 1, 2, and 3 get objects x_2, x_1, and x_3, respectively, at the outcome of the hierarchical exchange rule.

In what follows, we present a formal description of hierarchical exchange rules.

2.3.1 Inheritance trees

For a rooted tree T, the *level* of a node $v \in V(T)$ is defined as the number of edges appearing in the (unique) path from $r(T)$ to v.

Definition 2.3.1. For an object $x \in A$, an ***inheritance tree for*** $x \in A$ is defined as a tuple $\Gamma_x = \langle T_x, \zeta_x^{NI}, \zeta_x^{EO} \rangle$, where

(i) T_x is a rooted tree with

  (a) $\max_{v \in V(T_x)} \mathit{level}(v) = \min\{|N|, |A|\} - 1$, and

  (b) $|E^{out}(v)| = |A| - \mathit{level}(v) - 1$ for all $v \in V(T_x)$ with $\mathit{level}(v) < \min\{|N|, |A|\} - 1$,

(ii) $\zeta_x^{NI} : V(T_x) \to N$ is a nodes-to-individuals function with $\zeta_x^{NI}(v) \neq \zeta_x^{NI}(\tilde{v})$ for all distinct $v, \tilde{v} \in V(T_x)$ that appear in same path, and

(iii) $\zeta_x^{EO} : E(T_x) \to A \setminus \{x\}$ is an edges-to-objects function with $\zeta_x^{EO}(e) \neq \zeta_x^{EO}(\tilde{e})$ for all distinct $e, \tilde{e} \in E(T_x)$ that appear in same path or have same source node (that is, $s(e) = s(\tilde{e})$).

2.3.2 Endowments

A hierarchical exchange rule works in several stages and in each stage, endowments of individuals are determined by using a (fixed) collection of inheritance trees.

Given a collection of inheritance trees $\Gamma = (\Gamma_x)_{x \in A}$, one for each object $x \in A$, we define a class of endowments $\mathcal{E}^{\Gamma}$ as follows:

(i) The ***initial endowment*** $\mathcal{E}_i^{\Gamma}(\emptyset)$ of individual i is given by

$$\mathcal{E}_i^{\Gamma}(\emptyset) = \{x \in A \mid \zeta_x^{NI}(r(T_x)) = i\}.$$

(ii) For all $N' \subseteq N \setminus \{i\}$ and $A' \subseteq A$ with $|N'| = |A'| \neq 0$, and all $\mu' \in \mathcal{M}(N', A')$, the ***endowment*** $\mathcal{E}_i^{\Gamma}(\mu')$ of individual i is given by

$$\begin{aligned} \mathcal{E}_i^{\Gamma}(\mu') = \{ & x \in A \setminus A' \mid \zeta_x^{NI}(r(T_x)) = i, \text{ or} \\ & \text{there exists a path } (v_x^1, \ldots, v_x^{r_x}) \text{ from } r(T_x) \text{ to } v_x^{r_x} \text{ in } \Gamma_x \text{ such that } \zeta_x^{NI}(v_x^{r_x}) = i \\ & \text{and for all } s = 1, \ldots, r_x - 1, \text{ we have } \zeta_x^{NI}(v_x^s) \in N' \text{ and } \mu'(\zeta_x^{NI}(v_x^s)) = \zeta_x^{EO}(v_x^s, v_x^{s+1}) \}. \end{aligned}$$

2.3.3 Iterative procedure to compute the outcome of a hierarchical exchange rule

For a given collection of inheritance trees $\Gamma = (\Gamma_x)_{x \in A}$, the ***hierarchical exchange rule*** f^{Γ} ***associated with*** Γ is defined by an iterative procedure with at most $\min\{|N|, |A|\}$ number of stages. Consider a preference profile $P_N \in \mathbb{L}^n(A)$.

Stage 1.

Hierarchical Endowments (Initial Endowments): For all $i \in N$, $E_1(i, P_N) = \mathcal{E}_i^{\Gamma}(\emptyset)$.

Top Choices: For all $i \in N$, $T_1(i, P_N) = \tau(P_i)$.

Trading Cycles: For all $i \in N$,

$$C_1(i, P_N) = \begin{cases} \{j_1, \ldots, j_g\} & \text{if there exist } j_1, \ldots, j_g \in N \text{ such that} \\ & \text{for all } s = 1, \ldots, g, T_1(j_s, P_N) \in E_1(j_{s+1}, P_N), \text{ and} \\ & \text{for some } \hat{s} = 1, \ldots, g, j_{\hat{s}} = i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Since each individual can be in at most one trading cycle, $C_1(i, P_N)$ is well-defined for all $i \in N$. Furthermore, since both the number of individuals and the number of objects are finite, there is always at least one trading cycle. Note that $C_1(i, P_N) = \{i\}$ if $T_1(i, P_N) \in E_1(i, P_N)$.

Assigned Individuals: $N_1(P_N) = \{i \mid C_1(i, P_N) \neq \emptyset\}$.

Assignments: For all $i \in N_1(P_N)$, $f_i^\Gamma(P_N) = T_1(i, P_N)$.

Assigned Objects: $A_1(P_N) = \{T_1(i, P_N) \mid i \in N_1(P_N)\}$.

This procedure is repeated iteratively in the remaining reduced market. For each stage t, define $N^t(P_N) = \bigcup_{u=1}^{t} N_u(P_N)$ and $A^t(P_N) = \bigcup_{u=1}^{t} A_u(P_N)$. In what follows, we present Stage $t+1$ of f^Γ.

***Stage** $t+1$.*

Hierarchical Endowments (Non-initial Endowments): Let $\mu^t \in \mathcal{M}(N^t(P_N), A^t(P_N))$ such that for all $i \in N^t(P_N)$,

$$\mu^t(i) = f_i^\Gamma(P_N).$$

For all $i \in N \setminus N^t(P_N)$, $E_{t+1}(i, P_N) = \mathcal{E}_i^\Gamma(\mu^t)$.

Top Choices: For all $i \in N \setminus N^t(P_N)$, $T_{t+1}(i, P_N) = \tau(P_i, A \setminus A^t(P_N))$.

Trading Cycles: For all $i \in N \setminus N^t(P_N)$,

$$C_{t+1}(i, P_N) = \begin{cases} \{j_1, \ldots, j_g\} & \text{if there exist } j_1, \ldots, j_g \in N \setminus N^t(P_N) \text{ such that} \\ & \text{for all } s = 1, \ldots, g, T_{t+1}(j_s, P_N) \in E_{t+1}(j_{s+1}, P_N), \text{ and} \\ & \text{for some } \hat{s} = 1, \ldots, g, j_{\hat{s}} = i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Assigned Individuals: $N_{t+1}(P_N) = \{i \mid C_{t+1}(i, P_N) \neq \emptyset\}$.

Assignments: For all $i \in N_{t+1}(P_N), f_i^{\Gamma}(P_N) = T_{t+1}(i, P_N)$.

Assigned Objects: $A_{t+1}(P_N) = \{T_{t+1}(i, P_N) \mid i \in N_{t+1}(P_N)\}$.

This procedure is repeated iteratively until either all individuals are assigned or all objects are assigned. The hierarchical exchange rule f^{Γ} associated with Γ is defined as follows. For all $i \in N$,

$$f_i^{\Gamma}(P_N) = \begin{cases} T_t(i, P_N) & \text{if } i \in N_t(P_N) \text{ for some stage } t; \\ \emptyset & \text{otherwise.} \end{cases}$$

Since for every preference profile P_N and every individual i, there exists at most one stage t such that $i \in N_t(P_N)$, f^{Γ} is well-defined.

Remark 2.3.1. Note that a collection of inheritance trees do not uniquely identify a hierarchical exchange rule. More formally, two different collections of inheritance trees Γ and $\overline{\Gamma}$ may give rise to the same hierarchical exchange rule, that is, $f^{\Gamma} \equiv f^{\overline{\Gamma}}$.

2.4 A characterization of OSP-implementable assignment rules

In this section, we introduce a property called *dual ownership* of a hierarchical exchange rule and provide a characterization of OSP-implementable, Pareto efficient, and non-bossy assignment rules by means of this property. We also explain the practical usefulness of the dual ownership property.

2.4.1 Dual ownership

Troyan (2019) introduces the notion of *dual dictatorship* in the context of fixed priority top trading cycles (FPTTC) rules.[14] We introduce a closely related notion for hierarchical exchange rules which we call *dual ownership*. A hierarchical exchange rule satisfies ***dual ownership*** if for any preference profile and any stage of the hierarchical exchange rule at that preference profile, there are at most two individuals who own all the objects that remain in the reduced market in that stage.

[14] Troyan (2019) uses the term "TTC rule" to refer to an FPTTC rule. In Section 2.5.2, we provide a formal description of FPTTC rules.

2.4.2 The characterization result

In this section, we provide a characterization of OSP-implementable assignment rules under two mild and desirable properties, namely Pareto efficiency and non-bossiness.[15]

Theorem 2.4.1. *An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is OSP-implementable, Pareto efficient and non-bossy if and only if f is a hierarchical exchange rule satisfying dual ownership.*

The proof of this theorem is relegated to Section 2.7.

Since OSP-implementability implies strategy-proofness (see Remark 2.2.1) and group strategy-proofness is equivalent to strategy-proofness and non-bossiness (see Pápai, 2000 for details), we obtain the following corollary from Theorem 2.4.1.

Corollary 2.4.1. *A group strategy-proof and Pareto efficient assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is OSP-implementable if and only if f is a hierarchical exchange rule satisfying dual ownership.*

It is worth mentioning that OSP-implementability and non-bossiness together do not imply Pareto efficiency. For instance, any constant assignment rule satisfies the former two properties, but does not satisfy the latter. Furthermore, it follows from Pápai (2000) that non-bossiness and Pareto efficiency together do not imply strategy-proofness. Since OSP-implementability is stronger than strategy-proofness (by Remark 2.2.1), non-bossiness and Pareto efficiency cannot imply it either. Example 2.4.1 shows that OSP-implementability and Pareto efficiency together do not imply non-bossiness.

Example 2.4.1. Consider an allocation problem with three individuals $N = \{1, 2, 3\}$ and three objects $A = \{x_1, x_2, x_3\}$. Consider the assignment rule f such that

$$f = \begin{cases} \text{Serial dictatorship with priority } (1 \succ 2 \succ 3) & \text{if } x_2 P_1 x_3 \\ \text{Serial dictatorship with priority } (1 \succ 3 \succ 2) & \text{if } x_3 P_1 x_2 \end{cases}$$

Consider the preference profiles $P_N = (x_1x_2x_3, x_1x_2x_3, x_1x_2x_3)$ and $\tilde{P}_N = (x_1x_3x_2, x_1x_2x_3, x_1x_2x_3)$.[16] Note that only individual 1 changes her preference from P_N to $\tilde{P}_N$. This, together with the facts $f(P_N) = [(1, x_1), (2, x_2), (3, x_3)]$ and $f(\tilde{P}_N) = [(1, x_1), (2, x_3), (3, x_2)]$, implies f violates non-bossiness. However, the OSP mechanism in Figure 2.3 implements f.[17]

[15] Bade & Gonczarowski (2017) characterize OSP-implementable and Pareto efficient assignment rules as the ones that can be implemented via a mechanism they call *sequential barter with lurkers*. Sequential barter with lurkers violates non-bossiness in general, and we do not see any obvious way to relate their result to ours.

[16] Here, we denote by $(x_1x_2x_3, x_2x_3x_1, x_3x_2x_1)$ a preference profile where individuals 1, 2 and 3 have preferences $x_1x_2x_3$, $x_2x_3x_1$, and $x_3x_2x_1$, respectively.

[17] We use the following notation in Figure 2.3: by x_1x_2 we denote the set of preferences where x_1 is preferred to x_2 and we denote an

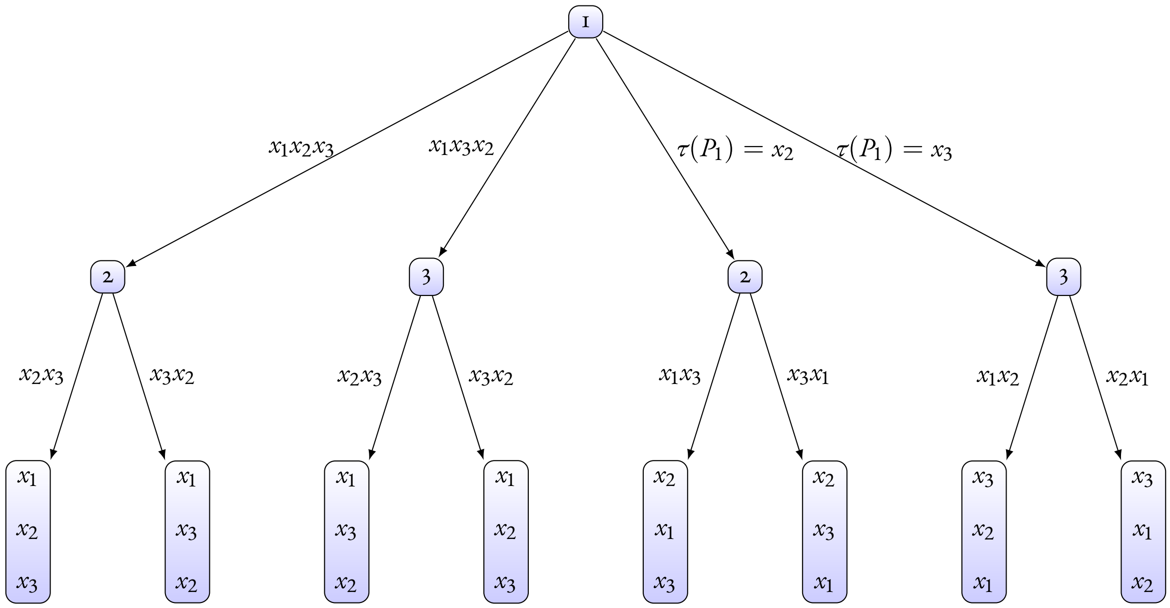


Figure 2.3: Tree Representation for Example 2.4.1

2.4.3 Advantage of using hierarchical exchange rules satisfying dual ownership property

In this section, we show how a hierarchical exchange rule satisfying the dual ownership property can be explained to the participating individuals and how the explanation helps in convincing individuals that such rules are indeed strategy-proof.[18]

In Stage 1:

(1) We call at most two individuals who will be the owners in this stage.

(2) We tell them their endowed sets.

(3) We tell them that each of them can "take" something from her endowed set (and leave the market), or "wait" to see if she gets something better. We additionally mention that if someone chooses to "wait", she can leave the market anytime in the future with an object from her current endowment set.

To see that the owners will act truthfully in (3), first note that the owners are asked to choose between "take" or "wait", in particular, they are not asked to reveal their top choices. Therefore,

(a) if any of the owners has her favorite object in her endowment, then she will "take" that object and leave the market, and

allocation $[(1, x_1), (2, x_2), (3, x_3)]$ by

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

[18]This explanation does not highlight many of the key features of hierarchical exchange rules satisfying the dual ownership property.

(b) if any of the owners does not have her favorite object in her endowment, then she will "wait" as she can leave the market anytime in the future with an object from her current endowment set.

(4) (i) If any of the owners chooses to "take" in (3). We get a submarket.

(ii) On the other hand, if both of them choose to "wait", we tell each of them to "take" something from other's endowment and leave the market, and again we get a submarket. Clearly, there is no question of manipulation for an individual at this step as she will simply take her favorite object from other's endowment.

In Stage 2:

(1) We call at most two individuals who will be the owners in this stage. If one of the owners in Stage 1 remains in the reduced market in Stage 2, we make her one of the owners in Stage 2.[19]

(2) We tell them their endowed sets. If one of the owners in Stage 2 was also an owner in Stage 1, all the objects in her endowment in Stage 1 must be included in her endowment in Stage 2.

(3) Same as Stage 1. For the same reason as we have discussed in (3) of Stage 1, individuals will act truthfully at this step of Stage 2.

(4) Same as Stage 1.

We continue this procedure until everyone is assigned or all objects are assigned.

The main reason why a hierarchical exchange rule satisfying dual ownership is simpler than an arbitrary hierarchical exchange rule is as follows. The dual ownership property ensures that at most two individuals will get to act in each stage. Therefore, the only way they can trade is to interchange their favorite objects. This makes it easy to see that they cannot strictly benefit by misreporting. For an arbitrary hierarchical exchange rule, there might be a lot more individuals acting in a stage, and consequently it may become harder for an individual to see the consequences of all possible misreports.

2.5 Discussion

2.5.1 OSP-implementability of hierarchical exchange rules and trading cycles rules

In this section, we provide a necessary and sufficient condition for a hierarchical exchange rule and a trading cycles rule to be OSP-implementable.

[19]Note that both owners in Stage 1 can not remain in the reduced market in Stage 2.

Proposition 2.5.1. *A hierarchical exchange rule is OSP-implementable if and only if it satisfies dual ownership.*

The proof of this proposition is relegated to Section 2.6.[20]

Pycia & Ünver (2017) introduce a general version of hierarchical exchange rules which they call *trading cycles rules*. They show that an assignment rule is group strategy-proof and Pareto efficient if and only if it is a trading cycles rule. Combining this result with Corollary 2.4.1, we obtain the following corollary.

Corollary 2.5.1. *A trading cycles rule is OSP-implementable if and only if it is a hierarchical exchange rule satisfying dual ownership.*

2.5.2 OSP-implementability of FPTTC rules

In this section, we discuss OSP-implementability of FPTTC rules. FPTTC rules are well-known in the literature; we present a brief description for the sake of completeness.

For each object $x \in A$, we define the ***priority*** of x as a "preference" $\succ_x$ over N.[21] We call a collection $\succ_A := (\succ_x)_{x \in A}$ a ***priority structure***. For a given priority structure $\succ_A$, the ***FPTTC rule*** $T^{\succ_A}$ ***associated with*** $\succ_A$ is defined by an iterative procedure as follows. Consider an arbitrary preference profile $P_N \in \mathbb{L}^n(A)$.

Step 1. Each object x is owned by the individual who has the highest priority according to $\succ_x$, that is, the most-preferred individual of $\succ_x$. TTC procedure is performed with respect to these endowments. Individuals who are assigned some object leave the market with their assigned objects.

This procedure is repeated iteratively in the remaining reduced market. We present a general step of $T^{\succ_A}$.

Step t. Consider the reduced market with the remaining individuals and objects. Each remaining object x is owned by the individual who has the highest priority among the remaining individuals according to $\succ_x$, that is, the individual who is remained in the reduced market at this step and is preferred to every other remaining individual according to $\succ_x$. TTC procedure is performed on the reduced market with respect to these endowments, and individuals who are assigned some object at this step leave the market.[22]

This procedure is repeated iteratively until either all individuals are assigned or all objects are assigned. The final outcome is obtained by combining all the assignments at all steps. This completes the description of an FPTTC rule.

[20]Proposition 2.5.1 follows as a corollary of Theorem 2.4.1. However, we do not present it as a corollary as we use this proposition in the proof of Theorem 2.4.1.

[21]That is, $\succ_x \in \mathbb{L}(N)$.

[22]In this TTC procedure, an individual i point to an individual j if j owns i's most-preferred object among the remaining objects.

Since FPTTC rules are special cases of hierarchical exchange rules (see Pápai, 2000 for details), the ***dual ownership property of FPTTC rules*** implies the following: for any preference profile and any step of the FPTTC rule at that preference profile, there are at most two individuals who own all the objects that remain in the reduced market at that step. This yields the following corollary from Proposition 2.5.1.

Corollary 2.5.2. *An FPTTC rule is OSP-implementable if and only if it satisfies dual ownership.*

Now, we discuss the relation between *dual dictatorship* (Troyan, 2019) and dual ownership of FPTTC rules. It follows from Theorem 1 and Theorem 2 in Troyan (2019) that an FPTTC rule is OSP-implementable if and only if it satisfies dual dictatorship, whereas Corollary 2.5.2 of our chapter says that an FPTTC rule is OSP-implementable if and only if it satisfies dual ownership. In what follows, we clarify the difference between these two (conflicting) results and conclude that while dual dictatorship is a sufficient condition for an FPTTC rule to be OSP-implementable, it is *not* necessary.[23]

Dual dictatorship property of an FPTTC rule requires that in any submarket, at most two individuals will own all the objects in the submarket. In contrast, dual ownership property of an FPTTC rule requires that for every preference profile and every step of that FPTTC rule at that preference profile, at most two individuals will own all the objects that will remain in the reduced market at that step. The difference between these two properties arises from the fact that *not every* submarket arises at some step at some preference profile of an FPTTC rule. In other words, dual dictatorship is stronger than dual ownership. In Section 2.8, we clarify this fact by means of an example.

2.6 Proof of Proposition 2.5.1

Before we formally start proving Proposition 2.5.1, to facilitate the proof we introduce the notion of a reduced tree structure and make two observations.

2.6.1 Reduced tree structure

For an inheritance tree $\Gamma_a = \langle T_a, \zeta_a^{NI}, \zeta_a^{EO} \rangle$ and an edge $(v, v') \in E(T_a)$, we say that an inheritance tree $\tilde{\Gamma}_a = \langle \tilde{T}_a, \tilde{\zeta}_a^{NI}, \tilde{\zeta}_a^{EO} \rangle$ ***is obtained by collapsing the edge*** (v, v') if

(i) $V(\tilde{T}_a) = V(T_a) \setminus \Big(\{v\} \cup \{v'' \mid \text{ there exists a path in } T_a \text{ from } v \text{ to } v'' \text{ which does not contain } v'\}\Big)$,

(ii) $E(\tilde{T}_a) = \Big(E(T_a) \cap \big(V(\tilde{T}_a) \times V(\tilde{T}_a)\big)\Big) \cup \{(\hat{v}, v')\}$, where $\hat{v}$ is the parent node of v in T_a. If $v = r(T_a)$, then $\hat{v}$ does not exist, and consequently, we take $\{(\hat{v}, v')\} = \emptyset$,

[23] In order to prove the "only-if" part of Theorem 1, Troyan (2019) reduces the whole problem to a restricted domain and uses a result from Li (2017). However, for the purpose of Troyan (2019), this reduction step is *not* correct.

(iii) $\tilde{\zeta}_a^{NI}(v) = \zeta_a^{NI}(v)$ for all $v \in V(\tilde{T}_a)$, and

(iv) $\tilde{\zeta}_a^{EO}(e) = \zeta_a^{EO}(e)$ for all $e \in \Big(E(T_a) \cap \big(V(\tilde{T}_a) \times V(\tilde{T}_a)\big)\Big)$ and $\tilde{\zeta}_a^{EO}(\hat{v}, v') = \zeta_a^{EO}(\hat{v}, v)$.

For an inheritance tree $\Gamma_a = \langle T_a, \zeta_a^{NI}, \zeta_a^{EO} \rangle$ and an edge $(v, v') \in E(T_a)$, we say that an inheritance tree $\tilde{\Gamma}_a = \langle \tilde{T}_a, \tilde{\zeta}_a^{NI}, \tilde{\zeta}_a^{EO} \rangle$ ***is obtained by dropping the edge*** (v, v') if

(i) $V(\tilde{T}_a) = V(T_a) \setminus \{v'' \mid$ there exists a path in T_a from v to v'' which contains $v'\}$,

(ii) $E(\tilde{T}_a) = E(T_a) \cap \big(V(\tilde{T}_a) \times V(\tilde{T}_a)\big)$,

(iii) $\tilde{\zeta}_a^{NI}(v) = \zeta_a^{NI}(v)$ for all $v \in V(\tilde{T}_a)$, and

(iv) $\tilde{\zeta}_a^{EO}(e) = \zeta_a^{EO}(e)$ for all $e \in E(\tilde{T}_a)$.

For an inheritance tree $\Gamma_a = \langle T_a, \zeta_a^{NI}, \zeta_a^{EO} \rangle$, we denote an edge $(v, v') \in E(T_a)$ by (i, x) if $\zeta_a^{NI}(v) = i$ and $\zeta_a^{EO}(v, v') = x$ in Γ_a. By the construction of Γ_a, $\zeta_a^{EO}(v, v') = x$ implies $a \neq x$.

For a pair $(i, x) \in N \times A$ and a collection of inheritance trees $\Gamma = (\Gamma_x)_{x \in A}$, we define the ***reduced collection*** $\Gamma \setminus (i, x)$ as follows:

(i) If $a = x$, then drop the inheritance tree Γ_a.

(ii) If $a \neq x$ and $\zeta_a^{NI}(r(T_a)) = i$, then $\Gamma_a \setminus (i, x)$ is obtained by collapsing the edge (i, x) in Γ_a.[24]

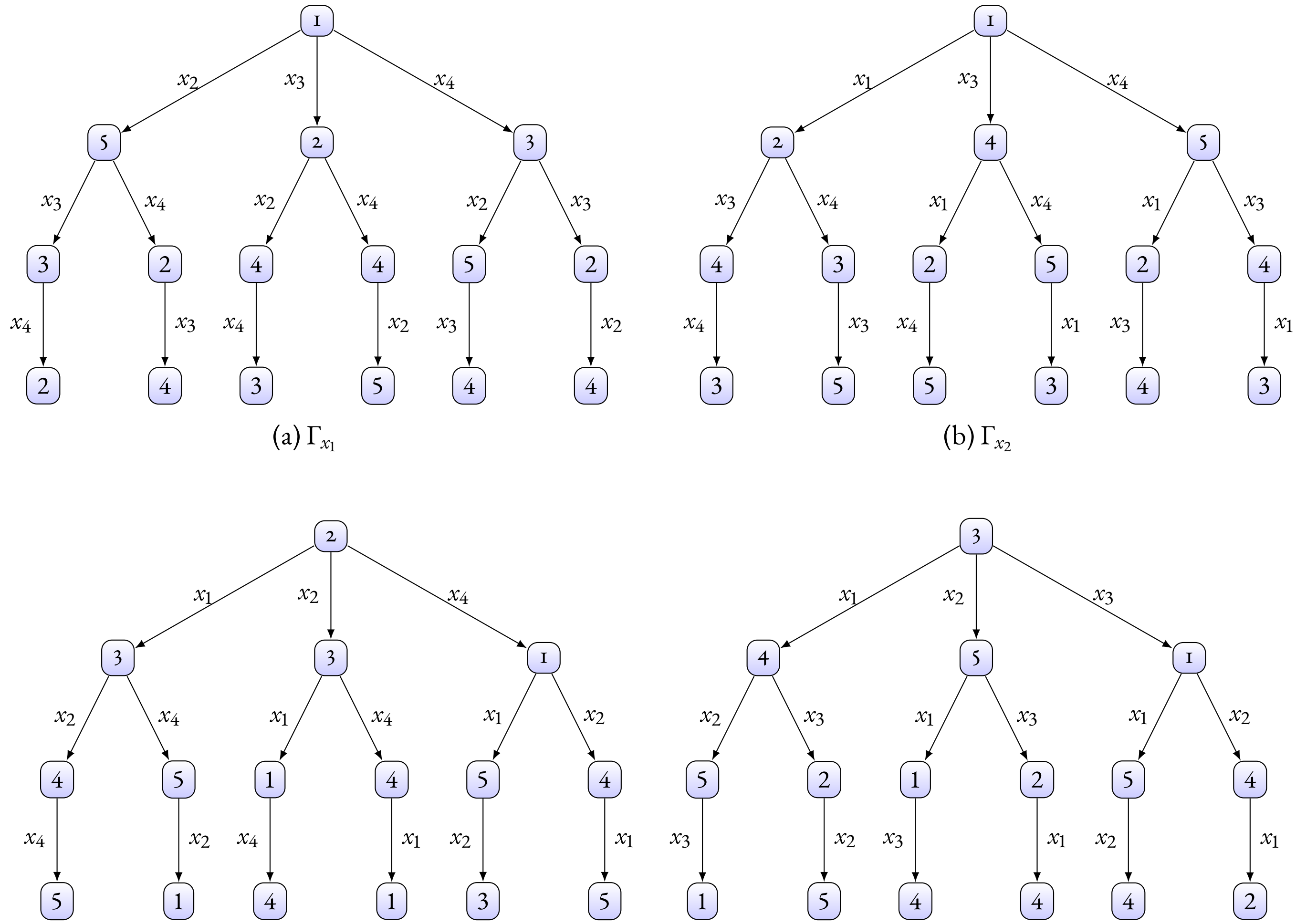
(iii) If $a \neq x$ and $\zeta_a^{NI}(r(T_a)) \neq i$, then $\Gamma_a \setminus (i, x)$ is obtained by collapsing all edges (i, x) and dropping all edges (j, x) with $j \neq i$ in Γ_a.

For $(i, x), (j, y) \in N \times A$ and a collection of inheritance trees $\Gamma = (\Gamma_x)_{x \in A}$, we denote the reduced collection $\Big(\Gamma \setminus (i, x)\Big) \setminus (j, y)$ by $\Gamma \setminus \Big((i, x), (j, y)\Big)$.

Remark 2.6.1. For $(i, x), (j, y) \in N \times A$ and a collection of inheritance trees $\Gamma = (\Gamma_x)_{x \in A}$, we have $\Gamma \setminus \Big((i, x), (j, y)\Big) = \Gamma \setminus \Big((j, y), (i, x)\Big)$.

Example 2.6.1. Suppose $N = \{1, 2, 3, 4, 5\}$ and $A = \{x_1, x_2, x_3, x_4\}$. Consider the collection of inheritance trees Γ given in Figure 2.4.

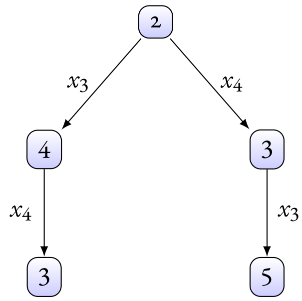
[24] Note that in this case, there is only one such edge (i, x).



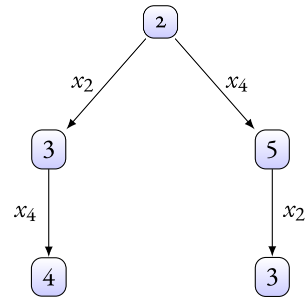
(a) Γ_{x_1} (b) Γ_{x_2} (c) Γ_{x_3} (d) Γ_{x_4}

Figure 2.4: Collection of inheritance trees Γ for Example 2.6.1

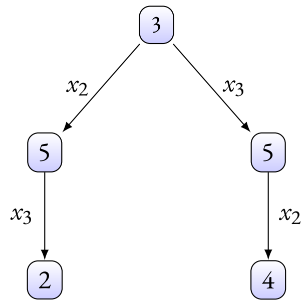
Consider the pair $(1, x_1) \in N \times A$. The reduced collection $\Gamma \setminus (1, x_1)$ is given in Figure 2.5.



(a) $\Gamma_{x_2} \setminus (1, x_1)$



(b) $\Gamma_{x_3} \setminus (1, x_1)$



(c) $\Gamma_{x_4} \setminus (1, x_1)$

Figure 2.5: Reduced collection $\Gamma \setminus (1, x_1)$

2.6.2 Two observations

Let $\mathcal{T}(\Gamma) = \{i \mid \zeta_x^{NI}(r(T_x)) = i \text{ for some } x \in A\}$ be the set of individuals who appear at the root-node of some inheritance tree in the collection of inheritance trees Γ. We now make two observations. The first observation is straightforward, and see Step 2.a in the "Necessity Proof" of Pápai (2000) for the second observation.

Observation 2.6.1. *Suppose f^Γ satisfies dual ownership. Then, $|\mathcal{T}(\Gamma)| \leq 2$.*

Observation 2.6.2. *Suppose $\zeta_x^{NI}(r(T_x)) = i$ for some $x \in A$ and some $i \in N$. Then, for all $P_N \in \mathbb{L}^n(A)$, $f_i^\Gamma(P_N) R_i x$.*

2.6.3 The proof

(If part) Suppose f^Γ satisfies dual ownership. We show that f^Γ is OSP-implementable by using induction on the number of individuals, which we refer to as the *size of the market*.

Base Case: Suppose $|N| = 1$.[25] The following extensive-form assignment mechanism, labeled as G^1, implements f^Γ.

Step 1. Ask the (only) individual which object is her top choice and assign her that object.

It is simple to check that the extensive-form assignment mechanism G^1 is OSP. Since the OSP mechanism G^1 implements f^Γ, it follows that f^Γ is OSP-implementable. Now, we proceed to prove the induction step.

[25] With only one individual, f^Γ trivially satisfies dual ownership.

Induction Hypothesis: Assume that f^Γ is OSP-implementable for $|N| \leq m$. We show f^Γ is OSP-implementable for $|N| = m+1$. Since f^Γ satisfies dual ownership, by Observation 2.6.1, we have $|\mathcal{T}(\Gamma)| \leq 2$. We distinguish the following two cases.

Case A: Suppose $|\mathcal{T}(\Gamma)| = 1$.

Let $\mathcal{T}(\Gamma) = \{i\}$. Define the extensive-form assignment mechanism G^{m+1} as follows:

Step 1. Ask individual i which object is her top choice and assign her that object, say x.

Step 2. Consider the reduced market $(N \setminus \{i\}, A \setminus \{x\})$ where individual i is removed from the market together with the object x she is assigned. This reduced market $(N \setminus \{i\}, A \setminus \{x\})$ is of size m.

Claim 2.6.1. *$f^{\Gamma \setminus (i,x)}$ satisfies dual ownership on the reduced market $(N \setminus \{i\}, A \setminus \{x\})$.*[26]

By the induction hypothesis and Claim 2.6.1, it follows that there exists an OSP mechanism G^m that implements $f^{\Gamma \setminus (i,x)}$ on the reduced market $(N \setminus \{i\}, A \setminus \{x\})$. Run the extensive-form assignment mechanism G^m on the reduced market $(N \setminus \{i\}, A \setminus \{x\})$.

By definition, the extensive-form assignment mechanism G^{m+1} implements f^Γ. This extensive-form assignment mechanism is OSP for individual i since she receives her top choice. For every other individual, her first decision node comes after i has been assigned, and hence, her strategic decision is equivalent to that under the OSP mechanism that implements f^Γ restricted to the corresponding reduced market. Thus, the above extensive-form assignment mechanism is OSP for all individuals, and hence, f^Γ is OSP-implementable.

Case B: Suppose $|\mathcal{T}(\Gamma)| = 2$.

Let $\mathcal{T}(\Gamma) = \{i, j\}$. Let $A_i = \{x \in A \mid \zeta_x^{NI}(r(T_x)) = i\}$ and $A_j = \{y \in A \mid \zeta_y^{NI}(r(T_y)) = j\}$. Define the extensive-form assignment mechanism G^{m+1} as follows:

Step 1. For each $x \in A_i$, ask i if her top choice is x. If i answers "Yes" for some x, assign her this x, and go to Step $1(a)$. Otherwise, jump to Step 2.

Step 1(a). We now have a reduced market $(N \setminus \{i\}, A \setminus \{x\})$ of size m.

Claim 2.6.2. *$f^{\Gamma \setminus (i,x)}$ satisfies dual ownership on the reduced market $(N \setminus \{i\}, A \setminus \{x\})$.*[27]

[26] The proof of Claim 2.6.1 is relegated to Section 2.6.4.

[27] The proof of Claim 2.6.2 follows by using similar logic as for the proof of Claim 2.6.1. The only adjustment needed for the proof of Claim 2.6.2 over the proof of Claim 2.6.1 is that instead of $\mathcal{T}(\Gamma) = \{i\}$ (which is an assumption of Case A) meaning that individual i is assigned to the root-node of every inheritance tree, we need to consider $x \in A_i$ (which is an assumption of Step 1 in Case B) meaning that individual i is assigned to the root-node of the inheritance tree for x.

By the induction hypothesis and Claim 2.6.2, it follows that there exists an OSP mechanism G^m that implements $f^{\Gamma\setminus(i,x)}$ on the reduced market $(N \setminus \{i\}, A \setminus \{x\})$. Run the extensive-form assignment mechanism G^m on the reduced market $(N \setminus \{i\}, A \setminus \{x\})$.

Step 2. For each $y \in A_j$, ask j if her top choice is y. If j answers "Yes" for some y, assign her this y, and go to Step $2(a)$. Otherwise, jump to Step 3.

Step 2(a). We now have a reduced market $(N \setminus \{j\}, A \setminus \{y\})$ of size m. Similar to Claim 2.6.2, we have the following claim.

Claim 2.6.3. *$f^{\Gamma\setminus(j,y)}$ satisfies dual ownership on the reduced market $(N \setminus \{j\}, A \setminus \{y\})$.*

By the induction hypothesis and Claim 2.6.3, it follows that there exists an OSP mechanism G^m that implements $f^{\Gamma\setminus(j,y)}$ on the reduced market $(N \setminus \{j\}, A \setminus \{y\})$. Run the extensive-form assignment mechanism G^m on the reduced market $(N \setminus \{j\}, A \setminus \{y\})$.

Step 3. If the answers to both Step 1 and Step 2 are "No", then i's top choice belongs to A_j, and j's top choice belongs to A_i. Ask i for her top choice x, and j for her top choice y. Assign x to i and y to j, and go to Step $3(a)$.

Step 3(a). We now have a reduced market $(N \setminus \{i,j\}, A \setminus \{x,y\})$ of size $m-1$.

Claim 2.6.4. *$f^{\Gamma\setminus((i,x),(j,y))}$ satisfies dual ownership on the reduced market $(N \setminus \{i,j\}, A \setminus \{x,y\})$.*[28]

By the induction hypothesis and Claim 2.6.4, it follows that there exists an OSP mechanism G^{m-1} that implements $f^{\Gamma\setminus((i,x),(j,y))}$ on the reduced market $(N \setminus \{i,j\}, A \setminus \{x,y\})$. Run the extensive-form assignment mechanism G^{m-1} on the reduced market $(N \setminus \{i,j\}, A \setminus \{x,y\})$.

By definition, the extensive-form assignment mechanism G^{m+1} implements f^{Γ}. We show that G^{m+1} is OSP for all individuals by showing it for the case where $|N| = 4$. The proof for other cases is similar.

Consider an allocation problem with four individuals $N = \{i_1, i_2, i_3, i_4\}$ and five objects $A = \{x_1, x_2, x_3, x_4, x_5\}$. Let Γ be a collection of inheritance trees such that $\mathcal{T}(\Gamma) = \{i_1, i_2\}$, $A_{i_1} = \{x_1, x_2\}$, and $A_{i_2} = \{x_3, x_4, x_5\}$. In Figure 2.6, we provide the structure of the extensive-form assignment mechanism G^4 which implements the hierarchical exchange rule f^{Γ}.

[28] The proof of Claim 2.6.4 is relegated to Section 2.6.5.

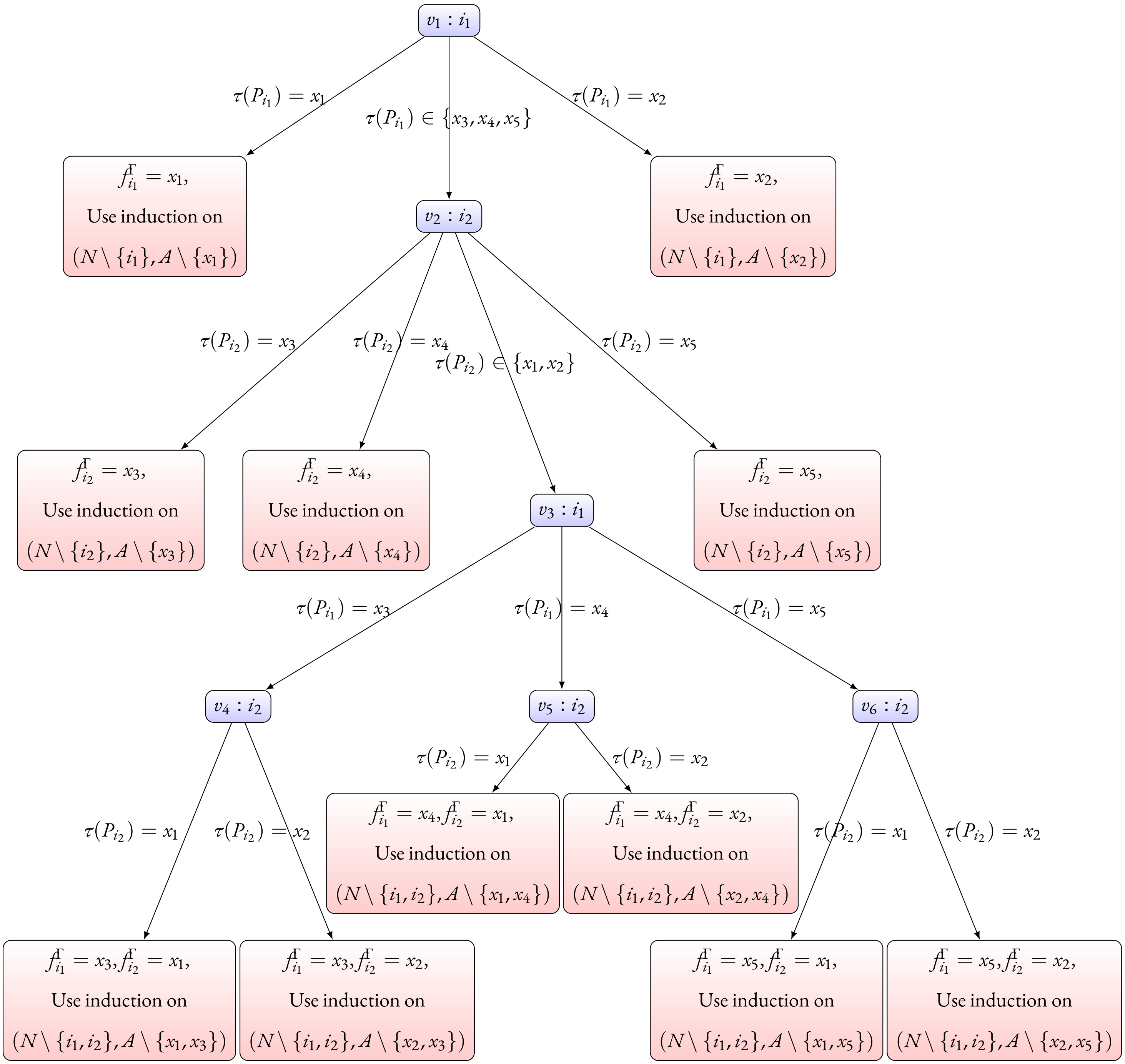


Figure 2.6: Structure of G^4

In Figure 2.6, node v_1 (which is the root-node of G^4) is assigned to individual i_1 and there are $|A_{i_1}| + 1$ outgoing edges from this node, node v_2 is assigned to individual i_2 and there are $|A_{i_2}| + 1$ outgoing edges from this node, and node v_3 is assigned to individual i_1 and there are $|A_{i_2}|$ outgoing edges from this node. Nodes v_4, v_5, and v_6 are assigned to individual i_2 and there are $|A_{i_1}|$ outgoing edges from each of these nodes.

It follows from the definition of G^4 and Observation 2.6.2 that G^4 satisfies the OSP property at node v_1 (for individual

i_1). We distinguish two cases.

(i) Suppose $\tau(P_{i_1}) \in \{x_1, x_2\}$.

Individual i_1 receives her top choice. The first decision node of every other individual comes after i_1 has been assigned, and hence, their strategic decisions are equivalent to that under the OSP mechanism that implements f^{Γ} restricted to the reduced market.

(ii) Suppose $\tau(P_{i_1}) \in \{x_3, x_4, x_5\}$.

It follows from the definition of G^4 and Observation 2.6.2 that G^4 satisfies the OSP property at node v_2 (for individual i_2).

(a) Suppose $\tau(P_{i_2}) \in \{x_3, x_4, x_5\}$. Individual i_2 receives her top choice. For every other individual, her strategic decision is equivalent to that under the OSP mechanism that implements f^{Γ} restricted to the reduced market.

(b) Suppose $\tau(P_{i_2}) \in \{x_1, x_2\}$. Both i_1 and i_2 receive their top choices. The first decision node of every other individual comes after i_1 and i_2 have been assigned, and hence, their strategic decisions are equivalent to that under the OSP mechanism that implements f^{Γ} restricted to the reduced market.

Since Cases (i) and (ii) are exhaustive, it follows that the extensive-form assignment mechanism G^4 is OSP for all individuals, and hence, f^{Γ} is OSP-implementable for this particular instance.

Since Case A and Case B are exhaustive, it follows that f^{Γ} is OSP-implementable for $|N| = m + 1$. This completes the proof of the induction step, and thereby completes the proof of the "if" part of Proposition 2.5.1.

(Only-if part) Suppose f^{Γ} does not satisfy dual ownership. We show that f^{Γ} is not OSP-implementable. Since f^{Γ} does not satisfy dual ownership, there exist a preference profile P'_N and a stage s^* of f^{Γ} at P'_N such that there are three individuals i_1, i_2, i_3 and three objects x_1, x_2, x_3 in the reduced market in Stage s^* with the property that for all $h = 1, 2, 3$, individual i_h owns the object x_h in Stage s^*.

Note that if an assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is not OSP-implementable on some restricted domain $\tilde{\mathcal{P}}_N \subseteq \mathbb{L}^n(A)$, then f is not OSP-implementable on the whole domain $\mathbb{L}^n(A)$ (see Li, 2017 for details). We distinguish the following two cases.

Case A: Suppose $s^* = 1$.

Consider the restricted domain $\tilde{\mathcal{P}}_N$ defined as follows. Each $l \in N \setminus \{i_1, i_2, i_3\}$ has only one (admissible) preference P'_l, and each individual in $\{i_1, i_2, i_3\}$ has two preferences, defined as follows (the dots indicate that all preferences for the corresponding parts are irrelevant and can be chosen arbitrarily).[29]

Individual i_1	Individual i_2	Individual i_3
$x_2x_3x_1\ldots$	$x_3x_1x_2\ldots$	$x_1x_2x_3\ldots$
$x_3x_2x_1\ldots$	$x_1x_3x_2\ldots$	$x_2x_1x_3\ldots$

Table 2.1: Admissible preferences of individuals i_1, i_2, and i_3

In Table 2.2, we present some facts regarding the outcome of f^Γ on the restricted domain $\tilde{\mathcal{P}}_N$. These facts are deduced by the construction of $\tilde{\mathcal{P}}_N$ along with the assumptions for Case A.

Preference profile	Individual i_1	Individual i_2	Individual i_3	$f^\Gamma_{i_1}$	$f^\Gamma_{i_2}$	$f^\Gamma_{i_3}$
$\tilde{P}^1_N$	$x_2x_3x_1\ldots$	$x_3x_1x_2\ldots$	$x_1x_2x_3\ldots$	x_2	x_3	x_1
$\tilde{P}^2_N$	$x_2x_3x_1\ldots$	$x_1x_3x_2\ldots$	$x_1x_2x_3\ldots$	x_2	x_1	x_3
$\tilde{P}^3_N$	$x_2x_3x_1\ldots$	$x_3x_1x_2\ldots$	$x_2x_1x_3\ldots$	x_1	x_3	x_2
$\tilde{P}^4_N$	$x_2x_3x_1\ldots$	$x_1x_3x_2\ldots$	$x_2x_1x_3\ldots$	x_2	x_1	x_3
$\tilde{P}^5_N$	$x_3x_2x_1\ldots$	$x_3x_1x_2\ldots$	$x_1x_2x_3\ldots$	x_3	x_2	x_1
$\tilde{P}^6_N$	$x_3x_2x_1\ldots$	$x_1x_3x_2\ldots$	$x_1x_2x_3\ldots$	x_3	x_2	x_1
$\tilde{P}^7_N$	$x_3x_2x_1\ldots$	$x_3x_1x_2\ldots$	$x_2x_1x_3\ldots$	x_1	x_3	x_2
$\tilde{P}^8_N$	$x_3x_2x_1\ldots$	$x_1x_3x_2\ldots$	$x_2x_1x_3\ldots$	x_3	x_1	x_2

Table 2.2: Partial outcome of f^Γ on $\tilde{\mathcal{P}}_N$

Assume for contradiction that f^Γ is OSP-implementable on $\tilde{\mathcal{P}}_N$. So, there exists an OSP mechanism $\tilde{G}$ that implements f^Γ on $\tilde{\mathcal{P}}_N$. Note that since $f^\Gamma(\tilde{P}^1_N) \neq f^\Gamma(\tilde{P}^8_N)$, there exists a node in the OSP mechanism $\tilde{G}$ that has at least two edges. Also, note that since each individual $l \in N \setminus \{i_1, i_2, i_3\}$ has exactly one preference in $\tilde{\mathcal{P}}_l$, whenever there are more than one outgoing edges from a node, the node must be assigned to some individual in $\{i_1, i_2, i_3\}$. Consider the first node (from the root) v that has two edges and, without loss of generality, assume $\eta^{NI}(v) = i_1$. Consider the preference profiles $\tilde{P}^3_N$ and $\tilde{P}^5_N$. Note that both of them pass through the node v at which $\tilde{P}^3_{i_1}$ and $\tilde{P}^5_{i_1}$ diverge. Further note that $x_3\tilde{P}^3_{i_1}x_1, f^\Gamma_{i_1}(\tilde{P}^3_N) = x_1$, and $f^\Gamma_{i_1}(\tilde{P}^5_N) = x_3$. However, the facts that $x_3\tilde{P}^3_{i_1}x_1, f^\Gamma_{i_1}(\tilde{P}^3_N) = x_1$, and $f^\Gamma_{i_1}(\tilde{P}^5_N) = x_3$ together contradict OSP-implementability of f^Γ on $\tilde{\mathcal{P}}_N$.

Case B: Suppose $s^* > 1$.

[29] For instance, $x_1x_2x_3\ldots$ indicates (any) preference that ranks x_1 first, x_2 second, and x_3 third.

Recall that for the preference profile P'_N, $A^{s^*-1}(P'_N)$ is the set of assigned objects up to Stage $s^* - 1$ (including Stage $s^* - 1$) of f^Γ at P'_N. Fix a preference $\hat{P} \in \mathbb{L}(A^{s^*-1}(P'_N))$ over these objects.

Consider the restricted domain $\tilde{\mathcal{P}}_N$ defined as follows. Each $l \in N \setminus \{i_1, i_2, i_3\}$ has only one (admissible) preference P'_l, and each individual in $\{i_1, i_2, i_3\}$ has two preferences, defined as follows.[30]

Individual i_1	Individual i_2	Individual i_3
$\hat{P}x_2x_3x_1\ldots$	$\hat{P}x_3x_1x_2\ldots$	$\hat{P}x_1x_2x_3\ldots$
$\hat{P}x_3x_2x_1\ldots$	$\hat{P}x_1x_3x_2\ldots$	$\hat{P}x_2x_1x_3\ldots$

Table 2.3: Admissible preferences of individuals i_1, i_2, and i_3

In Table 2.4, we present some facts regarding the outcome of f^Γ on the restricted domain $\tilde{\mathcal{P}}_N$ that can be deduced by the construction of the restricted domain $\tilde{\mathcal{P}}_N$ along with the assumptions for Case B. The verification of these facts is left to the reader.

Preference profile	Individual i_1	Individual i_2	Individual i_3	$f^\Gamma_{i_1}$	$f^\Gamma_{i_2}$	$f^\Gamma_{i_3}$
$\tilde{P}^1_N$	$\hat{P}x_2x_3x_1\ldots$	$\hat{P}x_3x_1x_2\ldots$	$\hat{P}x_1x_2x_3\ldots$	x_2	x_3	x_1
$\tilde{P}^2_N$	$\hat{P}x_2x_3x_1\ldots$	$\hat{P}x_1x_3x_2\ldots$	$\hat{P}x_1x_2x_3\ldots$	x_2	x_1	x_3
$\tilde{P}^3_N$	$\hat{P}x_2x_3x_1\ldots$	$\hat{P}x_3x_1x_2\ldots$	$\hat{P}x_2x_1x_3\ldots$	x_1	x_3	x_2
$\tilde{P}^4_N$	$\hat{P}x_2x_3x_1\ldots$	$\hat{P}x_1x_3x_2\ldots$	$\hat{P}x_2x_1x_3\ldots$	x_2	x_1	x_3
$\tilde{P}^5_N$	$\hat{P}x_3x_2x_1\ldots$	$\hat{P}x_3x_1x_2\ldots$	$\hat{P}x_1x_2x_3\ldots$	x_3	x_2	x_1
$\tilde{P}^6_N$	$\hat{P}x_3x_2x_1\ldots$	$\hat{P}x_1x_3x_2\ldots$	$\hat{P}x_1x_2x_3\ldots$	x_3	x_2	x_1
$\tilde{P}^7_N$	$\hat{P}x_3x_2x_1\ldots$	$\hat{P}x_3x_1x_2\ldots$	$\hat{P}x_2x_1x_3\ldots$	x_1	x_3	x_2
$\tilde{P}^8_N$	$\hat{P}x_3x_2x_1\ldots$	$\hat{P}x_1x_3x_2\ldots$	$\hat{P}x_2x_1x_3\ldots$	x_3	x_1	x_2

Table 2.4: Partial outcome of f^Γ on $\tilde{\mathcal{P}}_N$

Using a similar argument as for Case A, it follows from Table 2.4 that f^Γ is not OSP-implementable on $\tilde{\mathcal{P}}_N$. This completes the proof of the "only-if" part of Proposition 2.5.1. ■

[30]For instance, $\hat{P}x_1x_2x_3\ldots$ denotes a preference where objects in $A^{s^*-1}(P'_N)$ are ranked at the top according to the preference $\hat{P}$, objects x_1, x_2, and x_3 are ranked consecutively after that (in that order), and the ranking of the rest of the objects is arbitrarily.

2.6.4 Proof of Claim 2.6.1

Assume for contradiction that $f^{\Gamma \setminus (i,x)}$ does not satisfy dual ownership on the submarket $(N \setminus \{i\}, A \setminus \{x\})$. Then, there exist $\tilde{P}_{N \setminus \{i\}} \in \mathbb{L}^{|N \setminus \{i\}|}(A \setminus \{x\})$ and a stage s^* of $f^{\Gamma \setminus (i,x)}$ at $\tilde{P}_{N \setminus \{i\}}$ such that there are three individuals i_1, i_2, i_3 and three objects x_1, x_2, x_3 in the reduced market in Stage s^* of $f^{\Gamma \setminus (i,x)}$ at $\tilde{P}_{N \setminus \{i\}}$ with the property that for all $h = 1, 2, 3$, individual i_h owns the object x_h in Stage s^* of $f^{\Gamma \setminus (i,x)}$ at $\tilde{P}_{N \setminus \{i\}}$.

Consider the preference profile $P_N \in \mathbb{L}^n(A)$ such that $\tau(P_i) = x$ and $P_k = x\tilde{P}_k$ for all $k \in N \setminus \{i\}$.[31] By the assumption of Case A, $\mathcal{T}(\Gamma) = \{i\}$, which implies that individual i is assigned to the root-node of Γ_x. This, together with the construction of P_N and the definition of f^{Γ}, implies that individuals i_1, i_2, and i_3 own the objects x_1, x_2, and x_3, respectively, in Stage $s^* + 1$ of f^{Γ} at P_N, a contradiction to the fact that f^{Γ} satisfies dual ownership. This completes the proof of Claim 2.6.1. □

2.6.5 Proof of Claim 2.6.4

Assume for contradiction that $f^{\Gamma \setminus ((i,x),(j,y))}$ does not satisfy dual ownership on the submarket $(N \setminus \{i,j\}, A \setminus \{x,y\})$. Then, there exist $\tilde{P}_{N \setminus \{i,j\}} \in \mathbb{L}^{|N \setminus \{i,j\}|}(A \setminus \{x,y\})$ and a stage s^* of $f^{\Gamma \setminus ((i,x),(j,y))}$ at $\tilde{P}_{N \setminus \{i,j\}}$ such that there are three individuals i_1, i_2, i_3 and three objects x_1, x_2, x_3 in the reduced market in Stage s^* of $f^{\Gamma \setminus ((i,x),(j,y))}$ at $\tilde{P}_{N \setminus \{i,j\}}$ with the property that for all $h = 1, 2, 3$, individual i_h owns the object x_h in Stage s^* of $f^{\Gamma \setminus ((i,x),(j,y))}$ at $\tilde{P}_{N \setminus \{i,j\}}$.

Consider the preference profile $P_N \in \mathbb{L}^n(A)$ such that $\tau(P_i) = x$, $\tau(P_j) = y$ and $P_k = xy\tilde{P}_k$ for all $k \in N \setminus \{i,j\}$.[32] By the assumption of Step 3 in Case B, $x \in A_j$ and $y \in A_i$, which imply that individuals i and j are assigned to the root-nodes of Γ_y and Γ_x, respectively. This, together with the construction of P_N and the definition of f^{Γ}, implies that individuals i_1, i_2, and i_3 own the objects x_1, x_2, and x_3, respectively, in Stage $s^* + 1$ of f^{Γ} at P_N, a contradiction to the fact that f^{Γ} satisfies dual ownership. This completes the proof of Claim 2.6.4. □

2.7 Proof of Theorem 2.4.1

We use Proposition 2.5.1 (which is presented after Theorem 2.4.1) in the proof of Theorem 2.4.1. Therefore, we have already presented the proof of Proposition 2.5.1 in Section 2.6.

We first prove a lemma which says that every OSP-implementable, non-bossy, and Pareto efficient assignment rule is reallocation-proof. Next, we combine this lemma with Proposition 2.5.1 and two results of Pápai (2000) to complete the proof of Theorem 2.4.1.

[31] $x\tilde{P}_k$ denotes the preference that ranks x first, and follows $\tilde{P}_k$ for the ranking of the rest of the objects.
[32] $xy\tilde{P}_k$ denotes the preference that ranks x first, y second, and follows $\tilde{P}_k$ for the ranking of the rest of the objects.

2.7.1 Lemma 2.7.1 and its proof

Lemma 2.7.1 involves the notion of reallocation-proof assignment rules, which we present first.

Definition 2.7.1 (Pápai, 2000)**.** An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is *manipulable through reallocation* if there exist $P_N \in \mathbb{L}^n(A)$, distinct individuals $i, j \in N$, and $\tilde{P}_i \in \mathbb{L}(A)$, $\tilde{P}_j \in \mathbb{L}(A)$ such that

(i) $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i(P_N)$,

(ii) $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$, and

(iii) $f_i(P_N) = f_i(\tilde{P}_i, P_{-i}) \neq f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ and $f_j(P_N) = f_j(\tilde{P}_j, P_{-j}) \neq f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$.

An assignment rule is ***reallocation-proof*** if it is not manipulable through reallocation.

Lemma 2.7.1. *Suppose an assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is OSP-implementable, non-bossy, and Pareto efficient. Then, f is reallocation-proof.*

Proof of Lemma 2.7.1. Since f is OSP-implementable, by Remark 2.2.1, f is strategy-proof. Assume for contradiction that f is not reallocation-proof. Then, there exist $P_N \in \mathbb{L}^n(A)$, distinct individuals $i, j \in N$, and $\tilde{P}_i \in \mathbb{L}(A)$, $\tilde{P}_j \in \mathbb{L}(A)$ such that

(i) $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i(P_N)$,

(ii) $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$, and

(iii) $f_i(P_N) = f_i(\tilde{P}_i, P_{-i}) \neq f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ and $f_j(P_N) = f_j(\tilde{P}_j, P_{-j}) \neq f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$.

Using non-bossiness, $f_i(P_N) = f_i(\tilde{P}_i, P_{-i})$ implies $f(P_N) = f(\tilde{P}_i, P_{-i})$, and $f_j(P_N) = f_j(\tilde{P}_j, P_{-j})$ implies $f(P_N) = f(\tilde{P}_j, P_{-j})$. Combining the facts that $f(P_N) = f(\tilde{P}_i, P_{-i})$ and $f(P_N) = f(\tilde{P}_j, P_{-j})$, we have

$$f(P_N) = f(\tilde{P}_i, P_{-i}) = f(\tilde{P}_j, P_{-j}). \tag{2.1}$$

Claim 2.7.1. $\left\{f_i(P_N), f_j(P_N), f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}), f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})\right\} \subseteq A$.

Proof of Claim 2.7.1. Assume for contradiction that $f_i(P_N) = \emptyset$. By (2.1), we have $f_i(P_N) = f_i(\tilde{P}_j, P_{-j})$. Because $f_i(P_N) = \emptyset$ and $f_i(P_N) = f_i(\tilde{P}_j, P_{-j})$, we have $f_i(\tilde{P}_j, P_{-j}) = \emptyset$. Since f is strategy-proof, $f_i(\tilde{P}_j, P_{-j}) = \emptyset$ implies $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = \emptyset$. However, as $f_i(P_N) = \emptyset$ and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = \emptyset$, we have a contradiction to $f_i(P_N) \neq f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. So, it must be that

$$f_i(P_N) \neq \emptyset. \tag{2.2}$$

Using a similar argument, we have

$$f_j(P_N) \neq \emptyset. \tag{2.3}$$

Since $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$, (2.3) implies $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) \neq \emptyset$. Also, the fact $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i(P_N)$, together with (2.2), implies $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) \neq \emptyset$. This completes the proof of Claim 2.7.1. □

Claim 2.7.2. $f_i(P_N) = f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$.

Proof of Claim 2.7.2. Assume for contradiction that $f_i(P_N) \neq f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. Let $f_i(P_N) = w, f_j(P_N) = x, f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = y$, and $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = z$. By Claim 2.7.1, we have $w, x, y, z \neq \emptyset$. Since $f_i(P_N) = w$ and $f_j(P_N) = x$, we have $w \neq x$. Similarly, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = y$ and $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = z$ together imply $y \neq z$. Since $f_i(P_N) \neq f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$, we have $w \neq y$. Similarly $f_j(P_N) \neq f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ implies $x \neq z$, and $f_i(P_N) \neq f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ implies $w \neq z$. Moreover, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ implies $x \neq y$. However, the facts $w, x, y, z \neq \emptyset$, $w \neq x$, $y \neq z$, $w \neq y$, $x \neq z$, $w \neq z$, and $x \neq y$ together imply w, x, y, and z are all distinct objects.

Since $f_i(P_N) \neq f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$, $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i(P_N)$ implies $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_i f_i(P_N)$. The facts $f_i(P_N) = w, f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = z$, and $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_i f_i(P_N)$ together imply zP_iw. Since zP_iw and $f_i(P_N) = w$, by strategy-proofness, we have

$$f_i(P_i', P_{-i}) \neq z \text{ for all } P_i' \in \mathbb{L}(A). \tag{2.4}$$

By (2.1) we have $f_i(P_N) = f_i(\tilde{P}_j, P_{-j})$. This, along with the fact that $f_i(P_N) = w$, yields $f_i(\tilde{P}_j, P_{-j}) = w$. Since f is strategy-proof, the facts $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = y$ and $f_i(\tilde{P}_j, P_{-j}) = w$ together imply $y\tilde{R}_iw$, which, along with the fact that $w \neq y$, yields $y\tilde{P}_iw$. Also, combining the facts that $f_i(P_N) = w$ and $f_i(P_N) = f_i(\tilde{P}_i, P_{-i})$, we have $f_i(\tilde{P}_i, P_{-i}) = w$. Since $y\tilde{P}_iw$ and $f_i(\tilde{P}_i, P_{-i}) = w$, by strategy-proofness, we have

$$f_i(P_i', P_{-i}) \neq y \text{ for all } P_i' \in \mathbb{L}(A). \tag{2.5}$$

Moreover, since zP_iw and $f_i(\tilde{P}_j, P_{-j}) = w$, by strategy-proofness, we have

$$f_i(P_i', \tilde{P}_j, P_{-i,j}) \neq z \text{ for all } P_i' \in \mathbb{L}(A). \tag{2.6}$$

Let $\hat{P}_i$ rank z first, y second, and w third. Since f is strategy-proof and non-bossy, the fact $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = y$ and (2.6) imply

$$f(\hat{P}_i, \tilde{P}_j, P_{-i,j}) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.7}$$

Similarly, by strategy-proofness and non-bossiness, the fact that $f_i(P_N) = w$ along with (2.4) and (2.5), yields

$$f(\hat{P}_i, P_{-i}) = f(P_N). \tag{2.8}$$

By (2.8) we have $f_j(\hat{P}_i, P_{-i}) = f_j(P_N)$. This, along with the fact $f_j(P_N) = x$, yields $f_j(\hat{P}_i, P_{-i}) = x$. Also, the facts $f_j(P_N) = x$, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = y$, and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ together imply yP_jx. Since yP_jx and $f_j(\hat{P}_i, P_{-i}) = x$, by strategy-proofness, we have

$$f_j(\hat{P}_i, P'_j, P_{-i,j}) \neq y \text{ for all } P'_j \in \mathbb{L}(A). \tag{2.9}$$

Let $\hat{P}_j$ rank y first and z second. By (2.7) we have $f_j(\hat{P}_i, \tilde{P}_j, P_{-i,j}) = f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. This, along with the fact $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = z$, yields $f_j(\hat{P}_i, \tilde{P}_j, P_{-i,j}) = z$. Since f is strategy-proof and non-bossy, the fact $f_j(\hat{P}_i, \tilde{P}_j, P_{-i,j}) = z$ and (2.9) imply $f(\hat{P}_i, \hat{P}_j, P_{-i,j}) = f(\hat{P}_i, \tilde{P}_j, P_{-i,j})$. This, along with (2.7), yields

$$f(\hat{P}_i, \hat{P}_j, P_{-i,j}) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.10}$$

Because $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = y$ and $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = z$, (2.10) implies $f_i(\hat{P}_i, \hat{P}_j, P_{-i,j}) = y$ and $f_j(\hat{P}_i, \hat{P}_j, P_{-i,j}) = z$. However, since $z\hat{P}_iy$ and $y\hat{P}_jz$, the facts $f_i(\hat{P}_i, \hat{P}_j, P_{-i,j}) = y$ and $f_j(\hat{P}_i, \hat{P}_j, P_{-i,j}) = z$ together contradict Pareto efficiency.

So, it must be that $f_i(P_N) = f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. This completes the proof of Claim 2.7.2. $\square$

Since f is Pareto efficient, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ implies that there exists $k \in N \setminus \{j\}$ such that $f_k(P_N) = f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. Also, the facts $f_k(P_N) = f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ and $f_i(P_N) \neq f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ together imply $k \neq i$. Let $f_i(P_N) = a$, $f_j(P_N) = b$, and $f_k(P_N) = c$. Combining the facts that $f_k(P_N) = f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ and $f_k(P_N) = c$, we have $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$. Also the fact $f_i(P_N) = a$ along with Claim 2.7.2, implies $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$. Let $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = d$.

Claim 2.7.3. *a, b, and c are distinct objects, $d \in A$, and a, c, and d are distinct objects.*

Proof of Claim 2.7.3. Since $f_i(P_N) = a$, $f_j(P_N) = b$, and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$, by Claim 2.7.1, we have $a \neq \emptyset$, $b \neq \emptyset$, and $c \neq \emptyset$. Moreover, since $f_i(P_N) = a$, $f_j(P_N) = b$, and $f_k(P_N) = c$, it follows that a, b, and c are all distinct objects.

Now, we show $d \in A$. Assume for contradiction that $d = \emptyset$. Consider the preference profiles presented in Table 2.5. In addition to the structure provided in the table, suppose that $P^1_j = P^3_j$, $P^2_j = P^4_j$, and $P^1_k = P^2_k$. Here, l denotes an individual (might be empty) other than i, j, k. Note that such an individual does not change her preference across the mentioned preference profiles.

Preference profiles	Individual i	Individual j	Individual k	$\dots$	Individual l
P_N^1	$\tilde{P}_i$	$ca\dots$	$bc\dots$	$\dots$	P_l
P_N^2	$\tilde{P}_i$	$cba\dots$	$bc\dots$	$\dots$	P_l
P_N^3	$\tilde{P}_i$	$ca\dots$	P_k	$\dots$	P_l
P_N^4	$\tilde{P}_i$	$cba\dots$	P_k	$\dots$	P_l

Table 2.5: Preference profiles for Claim 2.7.3

The facts $f_j(P_N) = b$, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$, and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ together imply cP_jb. Moreover, $f_j(P_N) = b$ and (2.1) yield $f_j(\tilde{P}_i, P_{-i}) = b$. Since cP_jb and $f_j(\tilde{P}_i, P_{-i}) = b$, by strategy-proofness, we have

$$f_j(\tilde{P}_i, P'_j, P_{-i,j}) \neq c \text{ for all } P'_j \in \mathbb{L}(A). \tag{2.11}$$

By strategy-proofness and non-bossiness, the fact $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$ and (2.11) imply

$$f(P_N^3) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.12}$$

The facts $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = d$ and $d = \emptyset$ together imply $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = \emptyset$. Moreover, $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = \emptyset$ and (2.12) imply $f_k(P_N^3) = \emptyset$. Since f is strategy-proof and non-bossy, $f_k(P_N^3) = \emptyset$ yields $f(P_N^1) = f(P_N^3)$. This, together with (2.12), implies

$$f(P_N^1) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.13}$$

Similarly, by strategy-proofness and non-bossiness, the fact $f_j(\tilde{P}_i, P_{-i}) = b$ and (2.11) imply $f(P_N^4) = f(\tilde{P}_i, P_{-i})$. This, along with (2.1), yields

$$f(P_N^4) = f(P_N). \tag{2.14}$$

Since $f_j(P_N) = b$ and $f_k(P_N) = c$, by (2.14) we have $f_j(P_N^4) = b$ and $f_k(P_N^4) = c$. By strategy-proofness, $f_k(P_N^4) = c$ implies $f_k(P_N^2) \in \{b, c\}$. Suppose $f_k(P_N^2) = c$. Since $f_k(P_N^2) = c$ and $f_k(P_N^4) = c$, by non-bossiness and the fact that $f_j(P_N^4) = b$, we have $f_j(P_N^2) = b$. However, $f_j(P_N^2) = b$ and $f_k(P_N^2) = c$ together contradict Pareto efficiency. So, it must be that

$$f_k(P_N^2) = b. \tag{2.15}$$

Since $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$, by (2.13) we have $f_j(P_N^1) = a$. Also, by (2.15) we have $f_j(P_N^2) \neq b$. By strategy-proofness, the facts $f_j(P^1_N) = a$ and $f_j(P^2_N) \neq b$ imply $f_j(P^2_N) = a$. Since $f_j(P^1_N) = a$ and $f_j(P^2_N) = a$, by non-bossiness and (2.13), we have

$$f(P^2_N) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.16}$$

However, since $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = \emptyset$, by (2.16) we have $f_k(P^2_N) = \emptyset$, a contradiction to (2.15). So, it must be that

$$d \in A. \tag{2.17}$$

Since $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$, $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$, and $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = d$, it follows that a, c, and d are all distinct objects. This completes the proof of Claim 2.7.3. □

Claim 2.7.4. *cP_kd.*

Proof of Claim 2.7.4. Assume for contradiction that dR_kc. By Claim 2.7.3, this means dP_kc. Suppose $b = d$. Because dP_kc, this implies bP_kc. Also, the facts $f_j(P_N) = b$, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$, and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ together imply cP_jb. However, since cP_jb and bP_kc, the facts $f_j(P_N) = b$ and $f_k(P_N) = c$ together contradict Pareto efficiency. So, it must be that $b \neq d$. This, along with Claim 2.7.3, yields that a, b, c, and d are all distinct objects.

Consider the preference profiles presented in Table 2.6. In addition to the structure provided in the table, suppose $P^1_j = P^3_j$, $P^2_j = P^4_j$, and $P^1_k = P^2_k$.

Preference profiles	Individual i	Individual j	Individual k	$\dots$	Individual l
P^1_N	$\tilde{P}_i$	$ca\dots$	$dbc\dots$	$\dots$	P_l
P^2_N	$\tilde{P}_i$	$cba\dots$	$dbc\dots$	$\dots$	P_l
P^3_N	$\tilde{P}_i$	$ca\dots$	P_k	$\dots$	P_l
P^4_N	$\tilde{P}_i$	$cba\dots$	P_k	$\dots$	P_l

Table 2.6: Preference profiles for Claim 2.7.4

The fact $f_j(P_N) = b$ and (2.1) yield $f_j(\tilde{P}_i, P_{-i}) = b$. Moreover, the facts $f_j(P_N) = b$, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$, and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ together imply cP_jb. Since cP_jb and $f_j(\tilde{P}_i, P_{-i}) = b$, by strategy-proofness, we have

$$f_j(\tilde{P}_i, P'_j, P_{-i,j}) \neq c \text{ for all } P'_j \in \mathbb{L}(A). \tag{2.18}$$

By strategy-proofness and non-bossiness, the fact $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$ and (2.18) imply

$$f(P^3_N) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.19}$$

The fact $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = d$ and (2.19) imply $f_k(P^3_N) = d$. Since f is strategy-proof and non-bossy, $f_k(P^3_N) = d$ yields $f(P^1_N) = f(P^3_N)$. This, together with (2.19), implies

$$f(P^1_N) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.20}$$

Similarly, by strategy-proofness and non-bossiness, the fact $f_j(\tilde{P}_i, P_{-i}) = b$ and (2.18) imply $f(P^4_N) = f(\tilde{P}_i, P_{-i})$. This, along with (2.1), yields

$$f(P^4_N) = f(P_N). \tag{2.21}$$

Since $f_j(P_N) = b$ and $f_k(P_N) = c$, by (2.21) we have $f_j(P^4_N) = b$ and $f_k(P^4_N) = c$. By strategy-proofness, dP_kc and $f_k(P^4_N) = c$ together imply $f_k(P^2_N) \in \{b, c\}$. Suppose $f_k(P^2_N) = c$. Since $f_k(P^2_N) = c$ and $f_k(P^4_N) = c$, by non-bossiness and the fact that $f_j(P^4_N) = b$, we have $f_j(P^2_N) = b$. However, $f_j(P^2_N) = b$ and $f_k(P^2_N) = c$ together contradict Pareto efficiency. So, it must be that

$$f_k(P^2_N) = b. \tag{2.22}$$

Since $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$, by (2.20) we have $f_j(P^1_N) = a$. Also, by (2.22) we have $f_j(P^2_N) \neq b$. By strategy-proofness, the facts $f_j(P^1_N) = a$ and $f_j(P^2_N) \neq b$ together imply $f_j(P^2_N) = a$. Since $f_j(P^1_N) = a$ and $f_j(P^2_N) = a$, by non-bossiness and (2.20), we have

$$f(P^2_N) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j}). \tag{2.23}$$

However, since $f_k(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = d$, by (2.23) we have $f_k(P^2_N) = d$, a contradiction to (2.22). This completes the proof of Claim 2.7.4. □

Fix a preference $\hat{P} \in \mathbb{L}(A \setminus \{a, b, c\})$ over the objects in $A \setminus \{a, b, c\}$. Consider the preference profiles presented in Table 2.7. Assume that $P^5_k = P^{10}_k = P^{11}_k$.

Preference profiles	Individual i	Individual j	Individual k	$\dots$	Individual l
P_N^1	$abc\hat{P}$	$cab\hat{P}$	$acb\hat{P}$	$\dots$	P_l
P_N^2	$abc\hat{P}$	$cba\hat{P}$	$acb\hat{P}$	$\dots$	P_l
P_N^3	$acb\hat{P}$	$cab\hat{P}$	$acb\hat{P}$	$\dots$	P_l
P_N^4	$acb\hat{P}$	$cab\hat{P}$	$cab\hat{P}$	$\dots$	P_l
P_N^5	$acb\hat{P}$	$cab\hat{P}$	$cd\dots$	$\dots$	P_l
P_N^6	$bca\hat{P}$	$cba\hat{P}$	$acb\hat{P}$	$\dots$	P_l
P_N^7	$bca\hat{P}$	$cba\hat{P}$	$cab\hat{P}$	$\dots$	P_l
P_N^8	$cab\hat{P}$	$cab\hat{P}$	$cab\hat{P}$	$\dots$	P_l
P_N^9	$cab\hat{P}$	$cba\hat{P}$	$cab\hat{P}$	$\dots$	P_l
P_N^{10}	$cab\hat{P}$	$cab\hat{P}$	$cd\dots$	$\dots$	P_l
P_N^{11}	$cab\hat{P}$	$cba\hat{P}$	$cd\dots$	$\dots$	P_l
P_N^{12}	$cba\hat{P}$	$cab\hat{P}$	$acb\hat{P}$	$\dots$	P_l
P_N^{13}	$cba\hat{P}$	$cba\hat{P}$	$acb\hat{P}$	$\dots$	P_l
P_N^{14}	$cba\hat{P}$	$cab\hat{P}$	$cab\hat{P}$	$\dots$	P_l
P_N^{15}	$cba\hat{P}$	$cba\hat{P}$	$cab\hat{P}$	$\dots$	P_l

Table 2.7: Preference profiles for Lemma 2.7.1

The facts $f_j(P_N) = b$, $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$, and $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$ together imply cP_jb. Since cP_jb and $f_j(P_N) = b$, by strategy-proofness, we have

$$f_j(P'_j, P_{-j}) \neq c \text{ for all } P'_j \in \mathbb{L}(A). \tag{2.24}$$

Combining the fact $f_j(P_N) = b$ with (2.1), we have $f_j(\tilde{P}_i, P_{-i}) = f_j(\tilde{P}_j, P_{-j}) = b$. Since f is strategy-proof, the facts $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$ and $f_j(\tilde{P}_i, P_{-i}) = b$ together imply $a\tilde{R}_jb$, which along with Claim 2.7.3, yields $a\tilde{P}_jb$. Since $a\tilde{P}_jb$ and $f_j(\tilde{P}_j, P_{-j}) = b$, by strategy-proofness, we have

$$f_j(P'_j, P_{-j}) \neq a \text{ for all } P'_j \in \mathbb{L}(A). \tag{2.25}$$

However, since $f_j(\tilde{P}_j, P_{-j}) = b$, by strategy-proofness and non-bossiness along with (2.24) and (2.25), we have $f(P_j^5,$

$P_{-j}) = f(\tilde{P}_j, P_{-j})$. By (2.1), this, in particular, means

$$f_i(P^5_j, P_{-j}) = a, f_j(P^5_j, P_{-j}) = b, \text{ and } f_k(P^5_j, P_{-j}) = c. \tag{2.26}$$

By moving the preferences of the individuals $l \in \{i, k\}$ from P_l to P^5_l one by one, and by applying strategy-proofness and non-bossiness on (2.26) each time, we conclude

$$f_i(P^5_N) = a, f_j(P^5_N) = b, \text{ and } f_k(P^5_N) = c. \tag{2.27}$$

Using strategy-proofness and non-bossiness, we obtain from (2.27) that

$$f_i(P^4_N) = a, f_j(P^4_N) = b, \text{ and } f_k(P^4_N) = c. \tag{2.28}$$

By strategy-proofness, the facts cP_jb and $f_j(\tilde{P}_i, P_{-i}) = b$ together imply

$$f_j(\tilde{P}_i, P'_j, P_{-i,j}) \neq c \text{ for all } P'_j \in \mathbb{L}(A). \tag{2.29}$$

Since f is strategy-proof, the fact $f_j(\tilde{P}_i, P_{-i}) = b$ and (2.29) imply $f_j(\tilde{P}_i, P^{11}_j, P_{-i,j}) = b$. Moreover, since $f_j(\tilde{P}_i, P_{-i}) = b$ and $f_j(\tilde{P}_i, P^{11}_j, P_{-i,j}) = b$, by non-bossiness, we have $f(\tilde{P}_i, P^{11}_j, P_{-i,j}) = f(\tilde{P}_i, P_{-i})$. This, together with (2.1), yields

$$f(\tilde{P}_i, P^{11}_j, P_{-i,j}) = f(P_N). \tag{2.30}$$

By (2.1) we have $f_i(P_N) = f_i(\tilde{P}_j, P_{-j})$. This, along with the fact that $f_i(P_N) = a$, yields $f_i(\tilde{P}_j, P_{-j}) = a$. Since f is strategy-proof, the facts $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = c$ and $f_i(\tilde{P}_j, P_{-j}) = a$ together imply $c\tilde{R}_ia$, which along with Claim 2.7.3, yields $c\tilde{P}_ia$. Also, the fact $f_i(P_N) = a$, together with (2.30), implies $f_i(\tilde{P}_i, P^{11}_j, P_{-i,j}) = a$. Since $c\tilde{P}_ia$ and $f_i(\tilde{P}_i, P^{11}_j, P_{-i,j}) = a$, by strategy-proofness, we have $f_i(P^{11}_i, P^{11}_j, P_{-i,j}) = a$. Moreover, since $f_i(\tilde{P}_i, P^{11}_j, P_{-i,j}) = a$ and $f_i(P^{11}_i, P^{11}_j, P_{-i,j}) = a$, by non-bossiness, we have $f(P^{11}_i, P^{11}_j, P_{-i,j}) = f_i(\tilde{P}_i, P^{11}_j, P_{-i,j})$. This, together with (2.30), implies

$$f_i(P^{11}_i, P^{11}_j, P_{-i,j}) = a, f_j(P^{11}_i, P^{11}_j, P_{-i,j}) = b, \text{ and } f_k(P^{11}_i, P^{11}_j, P_{-i,j}) = c. \tag{2.31}$$

Using strategy-proofness and non-bossiness, we obtain from (2.31) that

$$f_i(P^{11}_N) = a, f_j(P^{11}_N) = b, \text{ and } f_k(P^{11}_N) = c. \tag{2.32}$$

Again, using strategy-proofness and non-bossiness, we obtain from (2.32) that

$$f_i(P_N^9) = a, f_j(P_N^9) = b, \text{ and } f_k(P_N^9) = c. \tag{2.33}$$

Since f is strategy-proof, the fact $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$ and (2.29) imply $f_j(\tilde{P}_i, P_j^{10}, P_{-i,j}) = a$. Since $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = a$ and $f_j(\tilde{P}_i, P_j^{10}, P_{-i,j}) = a$, by non-bossiness, we have $f(\tilde{P}_i, P_j^{10}, P_{-i,j}) = f(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. This, in particular, means

$$f_i(\tilde{P}_i, P_j^{10}, P_{-i,j}) = c, f_j(\tilde{P}_i, P_j^{10}, P_{-i,j}) = a, \text{ and } f_k(\tilde{P}_i, P_j^{10}, P_{-i,j}) = d. \tag{2.34}$$

From Claim 2.7.4, we have cP_kd. Since f is strategy-proof and cP_kd, (2.34) implies $f_k(\tilde{P}_i, P_j^{10}, P_k^{10}, P_{-i,j,k}) = d$. Moreover, since $f_k(\tilde{P}_i, P_j^{10}, P_{-i,j}) = d$ and $f_k(\tilde{P}_i, P_j^{10}, P_k^{10}, P_{-i,j,k}) = d$, by non-bossiness, (2.34) implies

$$f_i(\tilde{P}_i, P_j^{10}, P_k^{10}, P_{-i,j,k}) = c, f_j(\tilde{P}_i, P_j^{10}, P_k^{10}, P_{-i,j,k}) = a, \text{ and } f_k(\tilde{P}_i, P_j^{10}, P_k^{10}, P_{-i,j,k}) = d. \tag{2.35}$$

Using strategy-proofness and non-bossiness, we obtain from (2.35) that

$$f_i(P_N^{10}) = c, f_j(P_N^{10}) = a, \text{ and } f_k(P_N^{10}) = d. \tag{2.36}$$

By strategy-proofness, (2.33) implies $f_j(P_N^8) \in \{a, b\}$. Suppose $f_j(P_N^8) = b$. Since $f_j(P_N^8) = b$ and $f_j(P_N^9) = b$, by non-bossiness, (2.33) implies $f_k(P_N^8) = c$. However, since $f_k(P_N^8) = c$, (2.36) contradicts strategy-proofness. So, it must be that $f_j(P_N^8) = a$. By strategy-proofness, (2.28) implies $f_i(P_N^8) \in \{a, c\}$. This, along with the fact that $f_j(P_N^8) = a$, yields

$$f_i(P_N^8) = c \text{ and } f_j(P_N^8) = a. \tag{2.37}$$

Using strategy-proofness and non-bossiness, we obtain from (2.37) that

$$f_i(P_N^{14}) = c \text{ and } f_j(P_N^{14}) = a. \tag{2.38}$$

By strategy-proofness, (2.38) implies $f_j(P_N^{15}) \in \{a, b\}$. Suppose $f_j(P_N^{15}) = a$. Since $f_j(P_N^{14}) = a$ and $f_j(P_N^{15}) = a$, by non-bossiness and (2.38), we have $f_i(P_N^{15}) = c$. However, since $f_i(P_N^{15}) = c$, (2.33) contradicts strategy-proofness. So, it must be that $f_j(P_N^{15}) = b$. By strategy-proofness, (2.33) implies $f_i(P_N^{15}) \in \{a, b\}$. This, along with the fact that $f_j(P_N^{15}) = b$, yields $f_i(P_N^{15}) = a$. By non-bossiness, this and (2.33) imply

$$f_i(P_N^{15}) = a, f_j(P_N^{15}) = b, \text{ and } f_k(P_N^{15}) = c. \tag{2.39}$$

Using strategy-proofness and non-bossiness, we obtain from (2.39) that

$$f_i(P_N^{7}) = a, f_j(P_N^{7}) = b, \text{ and } f_k(P_N^{7}) = c. \tag{2.40}$$

By (2.38) we have $f_k(P_N^{14}) \notin \{a, c\}$. By strategy-proofness, the fact $f_k(P_N^{14}) \notin \{a, c\}$ implies $f_k(P_N^{12}) = f_k(P_N^{14})$. This, by non-bossiness and (2.38), implies

$$f_i(P_N^{12}) = c \text{ and } f_j(P_N^{12}) = a. \tag{2.41}$$

By strategy-proofness, (2.41) implies $f_i(P_N^{3}) \in \{a, c\}$. Suppose $f_i(P_N^{3}) = c$. Since $f_i(P_N^{12}) = c$ and $f_i(P_N^{3}) = c$, by non-bossiness and (2.41), we have $f_j(P_N^{3}) = a$. However, $f_i(P_N^{3}) = c$ and $f_j(P_N^{3}) = a$ together contradict Pareto efficiency. So, it must be that $f_i(P_N^{3}) = a$. By strategy-proofness, (2.27) implies $f_k(P_N^{3}) \in \{a, c\}$. This, along with the fact that $f_i(P_N^{3}) = a$, yields

$$f_i(P_N^{3}) = a \text{ and } f_k(P_N^{3}) = c. \tag{2.42}$$

Using strategy-proofness and non-bossiness, we obtain from (2.42) that

$$f_i(P_N^{1}) = a \text{ and } f_k(P_N^{1}) = c. \tag{2.43}$$

By (2.43) we have $f_j(P_N^{1}) \notin \{a, c\}$. By strategy-proofness, $f_j(P_N^{1}) \notin \{a, c\}$ implies $f_j(P_N^{2}) = f_j(P_N^{1})$. This, by non-bossiness and (2.43), implies

$$f_i(P_N^{2}) = a \text{ and } f_k(P_N^{2}) = c. \tag{2.44}$$

By (2.39) we have $f_i(P_N^{15}) = a$ and $f_k(P_N^{15}) = c$. By strategy-proofness, $f_k(P_N^{15}) = c$ implies $f_k(P_N^{13}) \in \{a, c\}$. Suppose $f_k(P_N^{13}) = c$. Since $f_k(P_N^{15}) = c$ and $f_k(P_N^{13}) = c$, by non-bossiness and the fact that $f_i(P_N^{15}) = a$, we have $f_i(P_N^{13}) = a$. However, $f_i(P_N^{13}) = a$ and $f_k(P_N^{13}) = c$ together contradict Pareto efficiency. So, it must be that $f_k(P_N^{13}) = a$. By strategy-proofness, (2.41) implies $f_j(P_N^{13}) \in \{a, b\}$. This, along with the fact that $f_k(P_N^{13}) = a$, yields $f_j(P_N^{13}) = b$. By strategy-proofness, (2.44) implies $f_i(P_N^{13}) \in \{a, b, c\}$. This, together with the facts that $f_j(P_N^{13}) = b$ and $f_k(P_N^{13}) = a$, implies

$$f_i(P_N^{13}) = c, f_j(P_N^{13}) = b, \text{ and } f_k(P_N^{13}) = a. \tag{2.45}$$

By strategy-proofness, (2.40) implies $f_k(P_N^6) \in \{a, c\}$. Suppose $f_k(P_N^6) = c$. Since $f_k(P_N^7) = c$ and $f_k(P_N^6) = c$, by non-bossiness and (2.40), we have $f_i(P_N^6) = a$. However, $f_i(P_N^6) = a$ and $f_k(P_N^6) = c$ together contradict Pareto efficiency. So, it must be that $f_k(P_N^6) = a$. Also, by (2.45) we have $f_i(P_N^{13}) = c$ and $f_j(P_N^{13}) = b$. By strategy-proofness, $f_i(P_N^{13}) = c$ implies $f_i(P_N^6) \in \{b, c\}$. Suppose $f_i(P_N^6) = c$. Since $f_i(P_N^{13}) = c$ and $f_i(P_N^6) = c$, by non-bossiness and the fact that $f_j(P_N^{13}) = b$, we have $f_j(P_N^6) = b$. However, $f_i(P_N^6) = c$ and $f_j(P_N^6) = b$ together contradict Pareto efficiency. So, it must be that $f_i(P_N^6) = b$. Combining the facts that $f_i(P_N^6) = b$ and $f_k(P_N^6) = a$, we have

$$f_i(P_N^6) = b \text{ and } f_k(P_N^6) = a. \tag{2.46}$$

Now we complete the proof of Lemma 2.7.1. Consider the restricted domain $\tilde{\mathcal{P}}_N \subseteq \mathbb{L}^n(A)$ with only three preference profiles as follows.

Preference profiles	Individual i	Individual j	Individual k	...	Individual l
P_N^6	$bca\hat{P}$	$cba\hat{P}$	$acb\hat{P}$	...	P_l
P_N^7	$bca\hat{P}$	$cba\hat{P}$	$cab\hat{P}$	...	P_l
P_N^{14}	$cba\hat{P}$	$cab\hat{P}$	$cab\hat{P}$	...	P_l

Table 2.8: Preference profiles of $\tilde{\mathcal{P}}_N$

By (2.38), (2.40), and (2.46), we have

Preference profiles	$f_i(P_N)$	$f_j(P_N)$	$f_k(P_N)$
P_N^6	b		a
P_N^7	a	b	c
P_N^{14}	c	a	

Table 2.9: Partial outcome of f on $\tilde{\mathcal{P}}_N$

Since f is OSP-implementable on $\mathbb{L}^n(A)$, it must be OSP-implementable on the restricted domain $\tilde{\mathcal{P}}_N$. Let $\tilde{G}$ be an OSP mechanism that implements f on $\tilde{\mathcal{P}}_N$.

Note that since $f(P_N^6) \neq f(P_N^7)$, there exists a node in the OSP mechanism $\tilde{G}$ that has at least two edges. Also, note that since each individual $l \in N \setminus \{i, j, k\}$ has exactly one preference in $\tilde{\mathcal{P}}_l$, whenever there are at least two outgoing edges from a node, that node must be assigned to some individual in $\{i, j, k\}$. Consider the first node (from the root) v that has two edges.

Suppose $\eta^{NI}(v) = i$. Consider the preference profiles P^7_N and P^{14}_N. Note that both of them pass through the node v at which P^7_i and P^{14}_i diverge. Further note that $cP^7_i a$, $f_i(P^7_N) = a$, and $f_i(P^{14}_N) = c$. However, the facts that $cP^7_i a$, $f_i(P^7_N) = a$, and $f_i(P^{14}_N) = c$ together contradict OSP-implementability of f on $\tilde{\mathcal{P}}_N$. So, it must be that $\eta^{NI}(v) \neq i$.

Suppose $\eta^{NI}(v) = k$. Consider the preference profiles P^6_N and P^{14}_N. Note that both of them pass through the node v at which P^6_k and P^{14}_k diverge. Further note that $f_k(P^6_N) = a$, $f_k(P^{14}_N) \notin \{a, c\}$, and $aP^{14}_k x$ for all $x \in A \setminus \{a, c\}$. Since $aP^{14}_k x$ for all $x \in A \setminus \{a, c\}$, the facts that $f_k(P^6_N) = a$ and $f_k(P^{14}_N) \notin \{a, c\}$ together contradict OSP-implementability of f on $\tilde{\mathcal{P}}_N$. So, it must be that $\eta^{NI}(v) \neq k$.

Since $\eta^{NI}(v) \neq i$ and $\eta^{NI}(v) \neq k$, it must be that $\eta^{NI}(v) = j$. We distinguish the following two cases.

Case 1: $f_j(P^6_N) = c$.

Consider the preference profiles P^6_N and P^{14}_N. Note that both of them pass through the node v at which P^6_j and P^{14}_j diverge. Further note that $cP^{14}_j a$, $f_j(P^6_N) = c$, and $f_j(P^{14}_N) = a$. However, the facts that $cP^{14}_j a$, $f_j(P^6_N) = c$, and $f_j(P^{14}_N) = a$ together contradict OSP-implementability of f on $\tilde{\mathcal{P}}_N$.

Case 2: $f_j(P^6_N) \neq c$.

Consider the preference profiles P^6_N and P^{14}_N. Note that both of them pass through the node v at which P^6_j and P^{14}_j diverge. Further note that $f_j(P^6_N) \notin \{a, b, c\}$, $f_j(P^{14}_N) = a$, and $aP^6_j x$ for all $x \in A \setminus \{a, b, c\}$. Since $aP^6_j x$ for all $x \in A \setminus \{a, b, c\}$, the facts that $f_j(P^6_N) \notin \{a, b, c\}$ and $f_j(P^{14}_N) = a$ together contradict OSP-implementability of f on $\tilde{\mathcal{P}}_N$. This completes the proof of Lemma 2.7.1. ■

2.7.2 Completion of the proof of Theorem 2.4.1

We present two results from Pápai (2000), which we use to complete the proof of Theorem 2.4.1.

Theorem 2.7.1 (Main theorem in Pápai, 2000). *An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is group strategy-proof, Pareto efficient, and reallocation-proof if and only if f is a hierarchical exchange rule.*

Lemma 2.7.2 (Lemma 1 in Pápai, 2000). *An assignment rule $f : \mathbb{L}^n(A) \to \mathcal{M}$ is group strategy-proof if and only if it is strategy-proof and non-bossy.*

Proof of Theorem 2.4.1. ***(If part)*** Let f be a hierarchical exchange rule satisfying dual ownership. By Proposition 2.5.1, f is OSP-implementable. Moreover, since f is a hierarchical exchange rule, by Theorem 2.7.1, f is group strategy-proof and Pareto efficient. The fact that f is group strategy-proof along with Lemma 2.7.2, implies f is non-bossy. This completes the proof of the "if" part of Theorem 2.4.1.

(Only-if part) Let f be an OSP-implementable, non-bossy, and Pareto efficient assignment rule. By Lemma 2.7.1, f is reallocation-proof. Since f is OSP-implementable, by Remark 2.2.1, f is strategy-proof. This, together with Lemma 2.7.2 and the fact that f is non-bossy, implies f is group strategy-proof. Since f is group strategy-proof, Pareto efficient, and reallocation-proof, by Theorem 2.7.1, f is a hierarchical exchange rule. Moreover, since f is an OSP-implementable hierarchical exchange rule, by Proposition 2.5.1, f is a hierarchical exchange rule satisfying dual ownership. This completes the proof of the "only-if" part of Theorem 2.4.1. ■

2.8 Example to clarify the difference between dual dictatorship (Troyan, 2019) and dual ownership of FPTTC rules

Troyan (2019) deals with the case where $|N| = |A|$. Therefore, we explain the difference between dual dictatorship and dual ownership of FPTTC rules for this case only.

Example 2.8.1. Consider an allocation problem with four individuals $N = \{i, j, k, l\}$ and four objects $A = \{w, x, y, z\}$. Let $\succ_A$ be as follows:

$\succ_w$	$\succ_x$	$\succ_y$	$\succ_z$
i	i	l	l
j	j	j	k
k	k	k	j
l	l	i	i

Table 2.10: Priority structure for Example 2.8.1

Consider the FPTTC rule $T^{\succ_A}$ associated with the priority structure given in Table 2.10. First, we argue that it satisfies dual ownership. Since either individual i or individual l appears at the top position in each priority, it follows that for any preference profile, individuals i and l will own all the objects at Step 1 of $T^{\succ_A}$. Moreover, since there are only four individuals in the original market, for any preference profile, at any step from Step 3 onward of $T^{\succ_A}$, there will remain at most two individuals in the corresponding submarket and hence dual ownership will be vacuously satisfied. In what follows, we show that dual ownership will also be satisfied at Step 2 for any preference profile. We distinguish three cases based on the possible assignments at Step 1.

(i) Suppose only individual i is assigned some object at Step 1. No matter whether individual i is assigned object w or object x, individuals j and l will own all the objects at Step 2.

(ii) Suppose only individual l is assigned some object at Step 1.

  (a) If l is assigned object y, then individuals i and k will own all the objects at Step 2.

  (b) If l is assigned object z, then individuals i and j will own all the objects at Step 2.

(iii) Suppose both i and l are assigned some objects at Step 1. Since there are only four individuals in the original market, only two individuals will remain in the reduced market at Step 2.

Since Cases (i), (ii), and (iii) are exhaustive, it follows that $T^{\succ_A}$ satisfies dual ownership. We now proceed to show that it does not satisfy dual dictatorship. Consider the submarket consisting of individuals i, j, and k and objects x, y, and z. Here, individuals i, j, and k will own objects x, y, and z, respectively, and hence $T^{\succ_A}$ under consideration violates dual dictatorship.

3

On Obviously Strategy-proof Implementation of Fixed Priority Top Trading Cycles with Outside Options

3.1 Introduction

The objective of mechanism design is to implement desirable outcomes when participating agents are strategic. The standard notion of strategy-proofness requires truth-telling to be a dominant strategy. However, the structure of such incentive compatible mechanisms are at times quite involved, and consequently, agents are not convinced that they are indeed strategy-proof. The notion of *obvious strategy-proofness (OSP)* (Li, 2017) has emerged to resolve this issue.[33]

[33] There is a rapidly growing body of work on OSP-implementability in variety of settings; see Bade & Gonczarowski (2017), Ashlagi & Gonczarowski (2018), Bade (2019), Pycia & Troyan (2019), Arribillaga et al. (2020), Mackenzie (2020).

We consider the problem of OSP-implementability of fixed priority top trading cycles (FPTTC) rules when outside options are available, that is, each object need not be "acceptable" to an agent. Troyan (2019) and Mandal & Roy (2020) deal with OSP-implementability of FPTTC rules when outside options are *not* available.[34] Troyan (2019) introduces the notion of *dual dictatorship* and shows that it is both necessary and sufficient condition for an FPTTC rule to be OSP-implementable when there are equal number of agents and objects.[35,36] Later, Mandal & Roy (2020) point out that while dual dictatorship is a sufficient condition for the same, it is *not* necessary. They consequently introduce the notion of *dual ownership* (a weaker condition than dual dictatorship) and show that it is both necessary and sufficient condition for an FPTTC rule to be OSP-implementable.[37]

In a model with outside options, we show that dual dictatorship and dual ownership are equivalent properties of an FPTTC rule (Theorem 3.4.1), and dual ownership is a necessary and sufficient condition for an FPTTC rule to be OSP-implementable (Theorem 3.4.2). It is worth mentioning that we consider arbitrary (not necessarily equal) values of the number of agents and the number of objects.

3.2 Preliminaries

3.2.1 Model

Let $N = \{1, \ldots, n\}$ be a finite set of agents and A be a finite set of objects. Let a_0 denote the ***outside option***. An ***allocation*** is a function $\mu : N \to A \cup \{a_0\}$ such that $|\mu^{-1}(a)| \leq 1$ for all $a \in A$. Here, $\mu(i) = a$ means agent i is assigned object a under μ, and $\mu(i) = a_0$ means agent i is not assigned any object. We denote by $\mathcal{M}$ the set of all allocations.

Let $\mathbb{L}(A \cup \{a_0\})$ denote the set of all strict linear orders over $A \cup \{a_0\}$.[38] An element of $\mathbb{L}(A \cup \{a_0\})$ is called a ***preference*** over $A \cup \{a_0\}$. For a preference P, let R denote the weak part of P.[39] An element $P_N = (P_1, \ldots, P_n)$ of $\mathbb{L}^n(A \cup \{a_0\})$ is called a ***profile***.

An ***assignment rule*** is a function $f : \mathbb{L}^n(A \cup \{a_0\}) \to \mathcal{M}$. Let $f_i(P_N)$ denote the assignment of agent i by f at P_N.

[34] Troyan (2019) uses the term "TTC rule" to refer to an FPTTC rule in his paper.
[35] See Theorem 1 and Theorem 2 in Troyan (2019) for details.
[36] Kesten (2006) introduces the notion of *acyclicity*, and shows that an FPTTC rule is stable if and only if it is acyclic. It can be verified that acyclic FPTTC rules satisfy dual dictatorship.
[37] See Corollary 5.2 in Mandal & Roy (2020) for details.
[38] A ***strict linear order*** is a semiconnex, asymmetric, and transitive binary relation.
[39] For all $a, b \in A \cup \{a_0\}$, aRb if and only if $[aPb \text{ or } a = b]$.

3.2.2 Obviously strategy-proof implementation

The notion of obviously strategy-proof implementation is introduced by Li (2017). We use the following notions and notations to present it. For a rooted tree T, we denote its set of nodes by $V(T)$, set of edges by $E(T)$, root by $r(T)$, and set of leaves (terminal nodes) by $L(T)$. For a node $v \in V(T)$, let $E^{out}(v)$ denote the set of outgoing edges from v. For an edge $e \in E(T)$, let $s(e)$ denote its source node. A *path* in a tree is a sequence of nodes such that every two consecutive nodes form an edge.

Definition 3.2.1. A ***mechanism*** is defined as a tuple $G = \langle T, \eta^{LA}, \eta^{NI}, \eta^{EP} \rangle$, where

(i) T is a rooted tree,

(ii) $\eta^{LA} : L(T) \to \mathcal{M}$ is a leaves-to-allocations function,

(iii) $\eta^{NI} : V(T) \setminus L(T) \to N$ is a nodes-to-agents function,

(iv) $\eta^{EP} : E(T) \to 2^{\mathbb{L}(A \cup \{a_0\})} \setminus \{\emptyset\}$ is an edges-to-preferences function such that

  (a) for all distinct $e, e' \in E(T)$ with $s(e) = s(e')$, $\eta^{EP}(e) \cap \eta^{EP}(e') = \emptyset$,

  (b) for any $v \in V(T) \setminus L(T)$,

    (1) if there exists a path $(v^1, \ldots, v^t)$ from $r(T)$ to v and some $1 \le r < t$ such that $\eta^{NI}(v^r) = \eta^{NI}(v)$ and $\eta^{NI}(v^s) \neq \eta^{NI}(v)$ for all $s = r+1, \ldots, t-1$, then $\bigcup_{e \in E^{out}(v)} \eta^{EP}(e) = \eta^{EP}(v^r, v^{r+1})$,

    (2) if there is no such path, then $\bigcup_{e \in E^{out}(v)} \eta^{EP}(e) = \mathbb{L}(A \cup \{a_0\})$.

For a mechanism G, every profile P_N identifies a unique path from the root to some leaf in T in the following manner: from each node v, follow the outgoing edge e from v such that $\eta^{EP}(e)$ contains the preference $P_{\eta^{NI}(v)}$. If a node v lies in such a path, then we say that the profile P_N *passes through the node* v. Furthermore, we say two preferences P_i and P'_i of some agent i *diverge at a node* $v \in V(T) \setminus L(T)$ if $\eta^{NI}(v) = i$ and there are two distinct edges e and e' in $E^{out}(v)$ such that $P_i \in \eta^{EP}(e)$ and $P'_i \in \eta^{EP}(e')$.

For a mechanism G, the ***assignment rule*** f^G ***implemented by*** G is defined as follows: for all profiles P_N, $f^G(P_N) = \eta^{LA}(l)$, where l is the leaf that appears at the end of the unique path characterized by P_N.

Definition 3.2.2. A mechanism G is ***Obviously Strategy-Proof (OSP)*** if for all $i \in N$, all nodes v such that $\eta^{NI}(v) = i$, and all $P_N, \tilde{P}_N \in \mathbb{L}^n(A \cup \{a_0\})$ passing through v such that P_i and $\tilde{P}_i$ diverge at v, we have $f^G_i(P_N) R_i f^G_i(\tilde{P}_N)$.

An assignment rule $f : \mathbb{L}^n(A \cup \{a_0\}) \to \mathcal{M}$ is ***OSP-implementable*** if there exists an OSP mechanism G such that $f = f^G$.

3.3 Fixed priority top trading cycles (FPTTC) rules

For each object $a \in A$, we define the ***priority*** of a as a "preference" $\succ_a$ over N. We call a collection $\succ_A := (\succ_a)_{a \in A}$ a ***priority structure***. For a priority structure $\succ_A$, the ***FPTTC rule*** $T^{\succ_A}$ ***associated with*** $\succ_A$ is defined by an iterative procedure as follows. Consider a profile $P_N \in \mathbb{L}^n(A \cup \{a_0\})$.

Step t. Let $N_t(P_N) \subseteq N$ be the set of agents that remain after Step $t-1$ and $A_t(P_N) \subseteq A$ be the set of objects that remain after Step $t-1$.[40]

We construct a (directed) graph with the set of nodes $N_t(P_N) \cup A_t(P_N) \cup \{a_0\}$. Each agent $i \in N_t(P_N)$ points to her most-preferred element of $A_t(P_N) \cup \{a_0\}$. Each object $a \in A_t(P_N)$ points to its most-preferred agent in $N_t(P_N)$. The outside option a_0 points to each agent in $N_t(P_N)$.

There is at least one cycle.[41] Each agent in a cycle is assigned the element she is pointing to (the element might be some object or the outside option). Remove all agents and objects that appear in some cycle.

This procedure is repeated iteratively until either all agents are assigned or all objects are assigned.

The following remarks say that the assignment of an agent under an FPTTC rule will be as good as the outside option, as well as, any object for which she has the top-priority. Let $\succ_A \in \mathbb{L}^{|A|}(N)$ be a priority structure.

Remark 3.3.1. For all $P_N \in \mathbb{L}^n(A \cup \{a_0\})$ and all $i \in N$, $T_i^{\succ_A}(P_N) R_i a_0$.

Remark 3.3.2. Suppose $\tau(\succ_a) = i$ for some $a \in A$ and some $i \in N$. Then, for all $P_N \in \mathbb{L}^n(A \cup \{a_0\})$, $T_i^{\succ_A}(P_N) R_i a$.

3.4 Results

For $\succ \in \mathbb{L}(N)$ and $N' \subseteq N$, let $\tau(\succ, N')$ denote the most-preferred agent in N' according to $\succ$. For ease of presentation, we denote $\tau(\succ, N)$ by $\tau(\succ)$. For $\succ \in \mathbb{L}(N)$ and $i \in N$, let $U(i, \succ)$ denote the *upper contour set* $\{j \in N \mid j \succ i\}$ of i at $\succ$. For $P \in \mathbb{L}(A \cup \{a_0\})$, let $\tau(P)$ denote the most-preferred element of $A \cup \{a_0\}$ according to P.

Let $N' \subseteq N$, $A' \subseteq A$, and $\succ_A$ be a priority structure. The ***reduced priority structure*** $\succ_{A'}^{N'}$ is the collection $(\succ_a^{N'})_{a \in A'}$ such that for all $a \in A'$, (i) $\succ_a^{N'} \in \mathbb{L}(N')$ and (ii) for all $i, j \in N'$, $i \succ_a^{N'} j$ if and only if $i \succ_a j$. Thus, the reduced priority structure $\succ_{A'}^{N'}$ is the restriction of $\succ_A$ to the submarket (N', A').[42] Furthermore, let $\mathcal{T}(\succ_{A'}^{N'}) = \{i \mid \tau(\succ_a, N') = i \text{ for some } a \in A'\}$.

[40] Note that for all $P_N \in \mathbb{L}^n(A \cup \{a_0\})$, $N_1(P_N) = N$ and $A_1(P_N) = A$.

[41] All the cycles we consider in this chapter are assumed to be "minimal", that is, no subset of nodes of such a cycle forms another cycle. In the model without outside options, trading cycles are always minimal. However, since there can be multiple outgoing edges from the outside option a_0, non-minimal trading cycles may appear in the model with outside options.

[42] Thus, $\succ_A^N = \succ_A$.

Definition 3.4.1. (Troyan, 2019) The FPTTC rule $T^{\succ_A}$ satisfies ***dual dictatorship*** if for all $N' \subseteq N$ and $A' \subseteq A$, we have $|\mathcal{T}(\succ_{A'}^{N'})| \leq 2$.

Remark 3.4.1. If $T^{\succ_A}$ satisfies dual dictatorship, then $T^{\succ_{A'}^{N'}}$ satisfies dual dictatorship on the submarket (N', A') for all $N' \subseteq N$ and all $A' \subseteq A$.

Recall the definitions of $N_s(P_N)$ and $A_s(P_N)$ given in Section 3.3.

Definition 3.4.2. (Mandal & Roy, 2020) For a domain of profiles $\mathcal{P}_N \subseteq \mathbb{L}^n(A \cup \{a_0\})$, the FPTTC rule $T^{\succ_A}$ satisfies ***dual ownership*** on $\mathcal{P}_N$ if for all $P_N \in \mathcal{P}_N$, we have $|\mathcal{T}(\succ_{A_s(P_N)}^{N_s(P_N)})| \leq 2$ for all s.

Note 3.4.1. For an arbitrary domain of profiles $\mathcal{P}_N$, the set of FPTTC rules satisfying dual ownership is a superset of those satisfying dual dictatorship. Example 3.4.1 presents a domain of profiles on which the former set is a strict superset of the latter. This clarifies that the notions of dual dictatorship and dual ownership are different.

Example 3.4.1. Consider an allocation problem with four individuals $N = \{i, j, k, l\}$ and four objects $A = \{w, x, y, z\}$. Let $\mathcal{P} = \{P \in \mathbb{L}(A \cup \{a_0\}) \mid aPa_0 \text{ for some } a \in A\}$ be the set of preferences where the outside option is never the most-preferred choice. Let $\succ_A$ be as follows:

$\succ_w$	$\succ_x$	$\succ_y$	$\succ_z$
i	i	l	l
j	j	j	k
k	k	k	j
l	l	i	i

Table 3.1: Priority structure for Example 3.4.1

Using similar arguments as for Example C.1 in Mandal & Roy (2020), it follows that the FPTTC rule $T^{\succ_A}$ satisfies dual ownership on $\mathcal{P}^4$ but does not satisfy dual dictatorship.

Theorem 3.4.1. *An FPTTC rule on $\mathbb{L}^n(A \cup \{a_0\})$ satisfies dual dictatorship if and only if it satisfies dual ownership.*

Proof of Theorem 3.4.1. The "only-if" part of the theorem follows from respective definitions, we proceed to prove the "if" part. Assume for contradiction that $T^{\succ_A}$ does not satisfy dual dictatorship. Then, there exist $N' \subseteq N$ and $A' \subseteq A$ such that $|\mathcal{T}(\succ_{A'}^{N'})| > 2$. This implies that there exist three agents $i_1, i_2, i_3 \in N'$ and three objects $a_1, a_2, a_3 \in A'$ such that $\tau(\succ_{a_h}, N') = i_h$ for all $h = 1, 2, 3$. We distinguish the following two cases.

Case A: Suppose $\bigcup_{h=1}^{3} U(i_h, \succ_{a_h}) = \emptyset$.

Since $\bigcup_{h=1}^{3} U(i_h, \succ_{a_h}) = \emptyset$, we have $\tau(\succ_{a_h}) = i_h$ for all $h = 1, 2, 3$. Fix a profile $P_N \in \mathbb{L}^n(A \cup \{a_0\})$. By the definition of $T^{\succ_A}$, it follows that $N_1(P_N) = N$ and $A_1(P_N) = A$ (see Footnote 40). Since $\tau(\succ_{a_h}) = i_h$ for all $h = 1, 2, 3$, $N_1(P_N) = N$, and $A_1(P_N) = A$, we have $\{i_1, i_2, i_3\} \subseteq \mathcal{T}(\succ^{N_1(P_N)}_{A_1(P_N)})$. This implies $|\mathcal{T}(\succ^{N_1(P_N)}_{A_1(P_N)})| > 2$, a contradiction to the fact that $T^{\succ_A}$ satisfies dual ownership.

Case B: Suppose $\bigcup_{h=1}^{3} U(i_h, \succ_{a_h}) \neq \emptyset$.

Since $\bigcup_{h=1}^{3} U(i_h, \succ_{a_h}) \neq \emptyset$, without loss of generality, assume $U(i_1, \succ_{a_1}) \neq \emptyset$. The facts $U(i_1, \succ_{a_1}) \neq \emptyset$ and $\tau(\succ_{a_1}, N') = i_1$ together imply $\tau(\succ_{a_1}) \notin N'$. Consider the profile P_N such that $\tau(P_i) = a_1$ for all $i \in N'$ and $\tau(P_i) = a_0$ for all $i \notin N'$. Since $\tau(\succ_{a_1}) \notin N'$, it follows from the construction of P_N and the definition of $T^{\succ_A}$ that $N_2(P_N) = N'$ and $A_2(P_N) = A$. The facts $N_2(P_N) = N'$, $A_2(P_N) = A$, and $\tau(\succ_{a_h}, N') = i_h$ for all $h = 1, 2, 3$ together imply $\{i_1, i_2, i_3\} \subseteq \mathcal{T}(\succ^{N_2(P_N)}_{A_2(P_N)})$. This implies $|\mathcal{T}(\succ^{N_2(P_N)}_{A_2(P_N)})| > 2$, a contradiction to the fact that $T^{\succ_A}$ satisfies dual ownership. ■

Our next theorem provides a characterization of OSP-implementable FPTTC rules.

Theorem 3.4.2. *An FPTTC rule on $\mathbb{L}^n(A \cup \{a_0\})$ is OSP-implementable if and only if it satisfies dual ownership.*

Proof of Theorem 3.4.2. ***(If part)*** Suppose $T^{\succ_A}$ satisfies dual ownership. By Theorem 3.4.1, $T^{\succ_A}$ satisfies dual dictatorship. We show that $T^{\succ_A}$ is OSP-implementable by using induction on $|N|$, which we refer to as the *size of the market*.

Base Case: Suppose $|N| = 1$.[43] The following mechanism, labeled as G^1, implements $T^{\succ_A}$.

Step 1. Assign the (only) agent her most-preferred element of $A \cup \{a_0\}$.

It is simple to check that G^1 is OSP. Since the OSP mechanism G^1 implements $T^{\succ_A}$, it follows that $T^{\succ_A}$ is OSP-implementable. Now, we proceed to prove the induction step.

Induction Hypothesis: Assume that $T^{\succ_A}$ is OSP-implementable for $|N| \leq m$. We show $T^{\succ_A}$ is OSP-implementable for $|N| = m + 1$. Since $T^{\succ_A}$ satisfies dual dictatorship, by definition, we have $|\mathcal{T}(\succ_A)| \leq 2$. We distinguish the following two cases.

Case A: Suppose $|\mathcal{T}(\succ_A)| = 1$.

Let $\mathcal{T}(\succ_A) = \{i\}$. Define the mechanism G^{m+1} as follows:

Step 1. Assign agent i her most-preferred element of $A \cup \{a_0\}$, say a.

[43] With only one agent, $T^{\succ_A}$ vacuously satisfies dual dictatorship.

Step 2. We have a submarket $(N \setminus \{i\}, A \setminus \{a\})$ of size m. By Remark 3.4.1, $T^{\succ_{A \setminus \{a\}}^{N \setminus \{i\}}}$ satisfies dual dictatorship. By the induction hypothesis, there exists an OSP mechanism G^m that implements $T^{\succ_{A \setminus \{a\}}^{N \setminus \{i\}}}$ on the submarket $(N \setminus \{i\}, A \setminus \{a\})$. Run G^m on the submarket $(N \setminus \{i\}, A \setminus \{a\})$.

Clearly, G^{m+1} implements $T^{\succ_A}$. The mechanism G^{m+1} is OSP for agent i since she receives her top choice. For every other agent, her first decision node comes after i has been assigned, and hence, her strategic decision is equivalent to that under the OSP mechanism that implements $T^{\succ_A}$ restricted to the corresponding submarket. Thus, G^{m+1} is OSP for all agents.

Case B: Suppose $|\mathcal{T}(\succ_A)| = 2$.

Let $\mathcal{T}(\succ_A) = \{i, j\}$. Let $A_i = \{x \in A \mid \tau(\succ_x) = i\}$ and $A_j = \{x \in A \mid \tau(\succ_x) = j\}$. Define the mechanism G^{m+1} as follows:

Step 1. For each $a \in A_i \cup \{a_0\}$, ask i if her top choice is a. If i answers "Yes" for some a, assign her a, and go to Step $1(a)$. Otherwise, jump to Step 2.

Step 1(a). We have a submarket $(N \setminus \{i\}, A \setminus \{a\})$ of size m. By Remark 3.4.1, $T^{\succ_{A \setminus \{a\}}^{N \setminus \{i\}}}$ satisfies dual dictatorship. By the induction hypothesis, there exists an OSP mechanism G^m that implements $T^{\succ_{A \setminus \{a\}}^{N \setminus \{i\}}}$ on the submarket $(N \setminus \{i\}, A \setminus \{a\})$. Run G^m on the submarket $(N \setminus \{i\}, A \setminus \{a\})$.

Step 2. For each $b \in A_j \cup \{a_0\}$, ask j if her top choice is b. If j answers "Yes" for some b, assign her b, and go to Step $2(a)$. Otherwise, jump to Step 3.

Step 2(a). We have a submarket $(N \setminus \{j\}, A \setminus \{b\})$ of size m. By Remark 3.4.1, $T^{\succ_{A \setminus \{b\}}^{N \setminus \{j\}}}$ satisfies dual dictatorship. By the induction hypothesis, there exists an OSP mechanism G^m that implements $T^{\succ_{A \setminus \{b\}}^{N \setminus \{j\}}}$ on the submarket $(N \setminus \{j\}, A \setminus \{b\})$. Run G^m on the submarket $(N \setminus \{j\}, A \setminus \{b\})$.

Step 3. If the answers to both Step 1 and Step 2 are "No", then i's top choice belongs to A_j, and j's top choice belongs to A_i. Ask i for her top choice a, and j for her top choice b. Assign a to i and b to j, and go to Step $3(a)$.

Step 3(a). We have a submarket $(N \setminus \{i, j\}, A \setminus \{a, b\})$ of size $m - 1$. By Remark 3.4.1, $T^{\succ_{A \setminus \{a,b\}}^{N \setminus \{i,j\}}}$ satisfies dual dictatorship. By the induction hypothesis, there exists an OSP mechanism G^{m-1} that implements $T^{\succ_{A \setminus \{a,b\}}^{N \setminus \{i,j\}}}$ on the submarket $(N \setminus \{i, j\}, A \setminus \{a, b\})$. Run G^{m-1} on the submarket $(N \setminus \{i, j\}, A \setminus \{a, b\})$.

By definition, G^{m+1} implements $T^{\succ_A}$. Using a similar argument as for Case A, it follows from Remark 3.3.1, Remark 3.3.2, and the construction of G^{m+1} that G^{m+1} is OSP.

Since Case A and Case B are exhaustive, it follows that $T^{\succ_A}$ is OSP-implementable for $|N| = m + 1$.

(Only-if part) The proof of the "only-if" part follows by using similar arguments as for the proof of the "only-if" part of Proposition 5.1 in Mandal & Roy (2020). ■

4

Strategy-proof Allocation of Indivisible Goods when Preferences are Single-peaked

4.1 Introduction

We consider the well-known assignment problem where heterogeneous indivisible goods are to be assigned to individuals so that each individual receives at most one good. Such problems arise when, for instance, the Government wants to assign houses to the citizens, or hospitals to doctors, or a manager wants to allocate offices to employees, or tasks to workers, or a professor wants to assign projects to students. Individuals are asked to report their preferences over the goods and the designer decides the allocation based on these reports. We analyze the structure of such decision process satisfying some desirable properties such as (group) strategy-proofness, efficiency, non-bossiness, (top-)envy-proofness, and (pairwise/group-wise) reallocation-proofness.

(Group) strategy-proofness ensures that a (a group of) dishonest individual(s) cannot improve her (their) assignment(s) by misreporting her (their) preference(s).[44] Efficiency says that the assignments cannot be improved in the sense of Pareto (that is, everyone is weakly better off and someone is strictly better off). Non-bossiness says that a person cannot change the assignment of any other person without changing her own assignment. Envy-proofness says that if an individual is envious at another individual (that is, if she strictly prefers the assignment of the individual to her own assignment), then she cannot harm the individual by misreporting her preference. Top-envy-proofness, in a sense, can be viewed as envy-proofness with respect to the top-ranked object of the envious individual. Pairwise/group-wise reallocation-proofness rules out the possibility of an obvious case of manipulation where a pair/group of individuals misreport their preferences and become better off by redistributing the objects they obtain at the misreported profile.

Svensson (1999) shows that the set of strategy-proof, non-bossy, and neutral assignment rules on the unrestricted domain is the set of serial dictatorships, if every individual is assumed to be assigned an object.[45,46] Pápai (2000) characterizes strategy-proof, Pareto efficient, non-bossy, and reallocation-proof assignment rules on the unrestricted domain as *hierarchical exchange rules*. These rules can be regarded as generalizations of Gale's well-known top trading cycle (TTC) procedure.[47] Pycia & Ünver (2017) characterizes strategy-proof, Pareto efficient, and non-bossy assignment rules on the unrestricted domain as *trading cycles rules*.[48]

4.1.1 Our motivation and contribution

As we have mentioned, Svensson (1999), Pápai (2000), and Pycia & Ünver (2017) assume that the individuals can have arbitrary preferences over the goods. However, it is well-known that in many circumstances preferences of individuals are restricted in a particular way. *Single-peakedness* is known as one of the most common such restrictions. It arises when goods can be ordered based on certain criteria and individuals' preferences respect that ordering in the sense that as one moves away from her top-ranked (peak) good, her preference declines. For instance, in the problem of assigning hospitals (houses) to doctors (citizens), hospitals (houses) can be ordered based on their locations on a street and an individual may like to be assigned as close as possible to her favorite location, in the problem of assigning tasks to students, tasks can be ordered based on their technical difficulties and an individual may like to get a task that she is technically more comfortable with, etc. This motivates us to explore the structure of strategy-proof assignment rules when individuals have single-peaked

[44] A group of individuals improve their assignments if each member in it is weakly better-off and some member is strictly better-off.

[45] An assignment rule is neutral if its outcomes do not depend on the identities of the objects.

[46] Whenever it is clear from the context, we use the term "domain" to refer to a set of preferences or a set of preference profiles.

[47] Top trading cycle (TTC) is due to David Gale and discussed in Shapley & Scarf (1974).

[48] Ergin (2000) shows that an assignment rule satisfies Pareto efficiency, neutrality, and consistency if and only if it is a simple serial dictatorship rule (he uses somewhat weaker properties to show his result). Ehlers & Klaus (2006) characterize all Pareto efficient, strategy-proof, and reallocation-consistent assignment rules as *efficient priority rules*. Later, Ehlers & Klaus (2007) and Velez (2014) characterize a slightly larger class of assignment rules by weakening these characterizing properties. Karakaya et al. (2019) analyze TTC rules in the context of house allocation problem with existing tenants.

preferences. Instead of focusing only on the maximal single-peaked domain, we do our analysis on a class of single-peaked domains that we call *minimally rich*. A single-peaked domain is minimally rich if it contains all left single-peaked and all right single-peaked preferences.[49]

There are two main results in this chapter. The first one says that there is no strategy-proof, non-bossy, Pareto efficient, and strongly pairwise reallocation-proof assignment rule on a minimally rich single-peaked domain, when there are at least three individuals and three objects in the market (Theorem 4.5.1). The second result characterizes all strategy-proof, Pareto efficient, top-envy-proof, non-bossy, and pairwise reallocation-proof assignment rules on a minimally rich single-peaked domain as hierarchical exchange rules (Theorem 4.7.1). We additionally show that strategy-proofness and non-bossiness together are equivalent to group strategy-proofness on a minimally rich single-peaked domain (Proposition 4.4.1), and every hierarchical exchange rule satisfies group-wise reallocation-proofness on a minimally rich single-peaked domain (Proposition 4.7.1).[50]

Ours is not the first paper to deal with single-peaked domains, Damamme et al. (2015), Ehlers (2018), and Bade (2019) consider single-peaked domains in the context of housing markets.[51] Damamme et al. (2015) provide an algorithm which is Pareto efficient on a single-peaked domain. Ehlers (2018) shows that a Pareto efficient, strategy-proof, and individually rational rule on the maximal single-peaked domain does not necessarily coincide with Gale's TTC.[52] Bade (2019) introduces the notion of the *crawler* algorithm and shows that it is Pareto efficient, strategy-proof, and individually rational on the maximal single-peaked domain.[53] To the best of our knowledge, the present chapter is the first paper to analyze the structure of assignment rules on the single-peaked domains.

4.1.2 Organization of the chapter

The organization of this chapter is as follows. In Section 4.2, we introduce basic notions and notations that we use throughout the chapter. In Section 4.3, we define domains and discuss their properties. In Section 4.4, we define assignment rules and discuss their standard properties. We present an impossibility result (non-existence of strategy-proof, non-bossy, Pareto efficient, and strongly pairwise reallocation-proof assignment rules on a minimally rich single-peaked domain) in Section 4.5. Section 4.6 introduces the notion of hierarchical exchange rules. In Section 4.7, we present our main result: a character-

[49] A single-peaked preference is left (right) if every alternative on the left (right) of the peak is preferred to every alternative on the right (left) of the peak.

[50] This, in particular, implies that if we replace pairwise reallocation-proofness by its stronger version group-wise reallocation-proofness, the conclusion of Theorem 4.7.1 does not change.

[51] Shapley & Scarf (1974) introduce the housing market, a model (with equal number of individuals and objects) in which each individual owns a unique indivisible object (a house) initially.

[52] Gale's TTC is the unique rule to satisfy Pareto efficiency, strategy-proofness, and individual rationality on the unrestricted domain in the context of housing markets (see Ma (1994)).

[53] In fact, Bade (2019) shows that the crawler algorithm satisfies a stronger version of strategy-proofness called OSP-implementability.

ization of all strategy-proof, Pareto efficient, top-envy-proof, non-bossy, and pairwise reallocation-proof assignment rules on a minimally rich single-peaked domain as hierarchical exchange rules, and in Section 4.8, we discuss the independence of these characterizing properties.

4.2 Basic notions and notations

Let $N = \{1, \ldots, n\}$ be a (finite) set of individuals and A be a (non-empty and finite) set of objects. We denote the set of all strict linear orders over the elements of A by $\mathbb{L}(A)$.[54] An element P of $\mathbb{L}(A)$ is called a ***preference*** over A. For a preference $P \in \mathbb{L}(A)$, by R we denote the weak part of P, that is, for all $a, b \in A$, aRb if and only if $[aPb \text{ or } a = b]$. For $P \in \mathbb{L}(A)$ and non-empty $B \subseteq A$, we define $\tau(P, B) = a$ if and only if $a \in B$ and aPb for all $b \in B \setminus \{a\}$. For ease of presentation, we denote $\tau(P, A)$ by $\tau(P)$.

We introduce the notion of an ***allocation*** of a (non-empty) set of objects $B \subseteq A$ over a (non-empty) set of individuals $S \subseteq N$. If $|S| \leq |B|$, then an allocation assigns a unique object to each individual (some objects will be left unassigned if $|S| < |B|$). More formally, an allocation in this scenario is a one-to-one function $\mu : S \to B$. On the other hand, if $|B| < |S|$, then an allocation assigns each object to a unique individual (some individuals will not be assigned any object). More formally, an allocation in this scenario is an onto function $\mu : S \to B \cup \{\emptyset\}$ such that $\mu^{-1}(a)$ is singleton for all $a \in B$.

Here, $\mu(i) = a$ for some element a of A means individual i is assigned object a in allocation μ, and $\mu(i) = \emptyset$ means individual i is not assigned any object in μ. For $S \subseteq N$ and $B \subseteq A$ with $|S|, |B| \neq 0$, we denote by $\mathcal{M}(S, B)$ the set of all allocations of B over S. For ease of presentation, we denote $\mathcal{M}(N, A)$ by $\mathcal{M}$.

For ease of presentation we use the following convention throughout the chapter: for a set $\{1, \ldots, g\}$ of integers, whenever we refer to the number $g + 1$, we mean 1. For instance, if we write $s_t \geq r_{t+1}$ for all $t = 1, \ldots, g$, we mean $s_1 \geq r_2, \ldots,$ $s_{g-1} \geq r_g$, and $s_g \geq r_1$.

4.3 Domains and their properties

Each $i \in N$ has a preference $P_i \in \mathbb{L}(A)$ over A. We denote by $\mathcal{P}_i \subseteq \mathbb{L}(A)$ the set of all admissible preferences of individual i, and by $P_N = (P_1, \ldots, P_n)$ a n-vector of all the individuals' preferences, which will be referred to as a ***preference profile***. By $\mathcal{P}_N = \prod_{i=1}^{n} \mathcal{P}_i$ we denote the set of all admissible preference profiles.

Given a preference profile P_N, we denote by (P'_i, P_{-i}) the preference profile obtained from P_N by changing the preference of individual i from P_i to P'_i and keeping all other preferences unchanged.

[54] A ***strict linear order*** is a semiconnex, asymmetric, and transitive binary relation.

Definition 4.3.1. A preference $P \in \mathbb{L}(A)$ is called ***single-peaked*** with respect to an ordering $\prec \in \mathbb{L}(A)$ if

(i) for all $a_j, a_k \in A$ with $a_j \prec a_k \prec \tau(P)$, we have $a_k P a_j$, and

(ii) for all $a_j, a_k \in A$ with $\tau(P) \prec a_j \prec a_k$, we have $a_j P a_k$.

A single-peaked preference (with respect to $\prec$) is called ***left (right) single-peaked*** if for all $a_j, a_k \in A$, $a_j \prec \tau(P) \prec a_k$ implies $a_j P a_k$ $(a_k P a_j)$. A domain of preferences is called ***single-peaked*** (with respect to $\prec$) if each preference in it is single-peaked. A single-peaked domain of preferences is called ***minimally rich*** if it contains *all* left single-peaked and all right single-peaked preferences.

In the rest of the chapter we assume that for all $i \in N$, $\mathcal{P}_i$ is a minimally rich single-peaked domain (with respect to some (fixed) ordering $\prec$).

4.4 Assignment rules and their properties

In this section, we introduce the notion of assignment rules and discuss a few properties of those.

Definition 4.4.1. A function $f : \mathcal{P}_N \to \mathcal{M}$ is called an ***assignment rule*** on $\mathcal{P}_N$.

For an assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ and a preference profile $P_N \in \mathcal{P}_N$, we denote by $f_i(P_N)$ the object that is assigned to individual i by the assignment rule f at P_N.

An allocation μ ***Pareto dominates*** another allocation ν at a preference profile P_N if $\mu(i) R_i \nu(i)$ for all $i \in N$ and $\mu(j) P_j \nu(j)$ for some $j \in N$.

Definition 4.4.2. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is called *Pareto efficient at a preference profile* $P_N \in \mathcal{P}_N$ if there is no allocation that Pareto dominates $f(P_N)$ at P_N, and it is called ***Pareto efficient*** if it is Pareto efficient at every preference profile in $\mathcal{P}_N$.

Remark 4.4.1. If an assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ satisfies Pareto efficiency, then $\tau(P_j) \in \bigcup_{i \in N} \{f_i(P_N)\}$ for all $j \in N$. In other words, every object that is ranked at the top position by some individual must not be left unassigned. To see this, note that if $\tau(P_j) \notin \bigcup_{i \in N} \{f_i(P_N)\}$ for some $j \in N$, then the allocation μ defined by $\mu(j) = \tau(P_j)$ and $\mu(k) = f_k(P_N)$ for all $k \neq j$ Pareto dominates $f(P_N)$ at P_N.

Non-bossiness is a standard notion in matching theory which says that if an individual misreports her preference and her assignment does not change by the same, then the assignment of any other individual cannot change.[55]

[55] The concept of non-bossiness is due to Satterthwaite & Sonnenschein (1981).

Definition 4.4.3. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is ***non-bossy*** if for all $P_N \in \mathcal{P}_N$, all $i \in N$, and all $\tilde{P}_i \in \mathcal{P}_i$, $f_i(P_N) = f_i(\tilde{P}_i, P_{-i})$ implies $f(P_N) = f(\tilde{P}_i, P_{-i})$.

Definition 4.4.4. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is ***strategy-proof*** if for all $P_N \in \mathcal{P}_N$, all $i \in N$ and all $\tilde{P}_i \in \mathcal{P}_i$, we have $f_i(P_N) R_i f_i(\tilde{P}_i, P_{-i})$.

Note that if an assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is not strategy-proof, then there exist $P_N \in \mathcal{P}_N$, $i \in N$ and $\tilde{P}_i \in \mathcal{P}_i$ such that $f_i(\tilde{P}_i, P_{-i}) P_i f_i(P_N)$. In such cases, we say that *the individual i manipulates f at P_N via $\tilde{P}_i$*.

Definition 4.4.5. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is ***group strategy-proof*** if for all $P_N \in \mathcal{P}_N$, there do not exist a set of individuals $S \subseteq N$, and a preference profile $\tilde{P}_S$ of the individuals in S such that $f_i(\tilde{P}_S, P_{-S}) R_i f_i(P_N)$ for all $i \in S$ and $f_j(\tilde{P}_S, P_{-S}) P_j f_j(P_N)$ for some $j \in S$.

Proposition 4.4.1. *An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is group strategy-proof if and only if it is strategy-proof and non-bossy.*

The proof of this proposition is relegated to Section 4.10.

4.5 An impossibility result

We introduce the notion of *strongly pairwise reallocation-proof* assignment rules. It says that no pair of individuals can misreport their preferences and be better off redistributing their assignments ex post.[56]

Definition 4.5.1. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is *weakly manipulable through pairwise reallocation* if there exist $P_N \in \mathcal{P}_N$, distinct individuals $i, j \in N$, and $\tilde{P}_i \in \mathcal{P}_i$, $\tilde{P}_j \in \mathcal{P}_j$ such that

(i) $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i(P_N)$, and

(ii) $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$.

An assignment rule is ***strongly pairwise reallocation-proof*** if it is not weakly manipulable through pairwise reallocation.

Pápai (2000) mentions that there is no strategy-proof, non-bossy, Pareto efficient, and strongly pairwise reallocation-proof assignment rule on the unrestricted domain, where there are at least three individuals and three objects. Our next result says that the result holds if we restrict the domain to be minimally rich single-peaked.

Theorem 4.5.1. *Suppose $|N| \geq 3$ and $|A| \geq 3$. Then, there does not exist a strategy-proof, non-bossy, Pareto efficient, and strongly pairwise reallocation-proof assignment rule on $\mathcal{P}_N$.*

[56]Here, we say a group of individuals is better-off if each member in it is weakly better-off and some member is strictly better-off.

The proof of this theorem is relegated to Section 4.11.

Since group strategy-proofness is equivalent to strategy-proofness and non-bossiness (see Proposition 4.4.1), we obtain the following corollary from Theorem 4.5.1.

Corollary 4.5.1. *Suppose* $|N| \geq 3$ *and* $|A| \geq 3$*. Then, there does not exist a group strategy-proof, Pareto efficient, and strongly pairwise reallocation-proof assignment rule on* $\mathcal{P}_N$*.*

4.6 Hierarchical exchange rules

We introduce the notion of *hierarchical exchange rules* in this section. These rules are introduced in Pápai (2000) and are well-known in the literature. We present a description of these rules for the sake of completeness. The description in Section 4.6 is taken from Mandal & Roy (2020).

We introduce some basic definitions from graph theory which we will use in defining hierarchical exchange rules. We denote a rooted (directed) tree by T. For a tree T, we denote its set of nodes by $V(T)$, set of all edges by $E(T)$, and root by $r(T)$. For a node $v \in V(T)$, we denote the set of all outgoing edges from v by $E^{out}(v)$. For an edge $e \in E(T)$, we denote its source node by $s(e)$. A path in a tree is a sequence of nodes such that every two consecutive nodes form an edge.

First we explain the notion of a *TTC procedure* with respect to a given endowments of the objects over the individuals. Suppose that each object is owned by exactly one individual. Note that an individual may own more than one objects. A directed graph is constructed in the following manner. The set of nodes is the same as the set of individuals. There is a directed edge from individual i to individual j if and only if individual j owns individual i's most preferred object. Note that such a graph will have exactly one outgoing edge from every node (though possibly many incoming edges to a node). Further, there may be an edge from a node to itself. It is clear that such a graph will always have a cycle. This cycle is called a *top trading cycle (TTC)*. After forming a TTC, the individuals in the TTC are assigned their most preferred objects.

4.6.1 Verbal description of hierarchical exchange rules

The following verbal description of hierarchical exchange rules is taken from Pápai (2000). The allocation obtained by a hierarchical exchange rule can be described by the following iterative procedure. Individuals have an initial individual "endowment" of objects such that each object is exactly one individual's endowment. It is important to note that some individuals may not be endowed with any objects. Now apply the TTC procedure to this market with individual endowments. Notice that individuals who don't have endowments cannot be part of a top trading cycle, since nobody points to them, and therefore they need not point. Given that multiple endowments are allowed, after the individuals in top trading cycles leave the market with their most preferred objects, unassigned objects in the initial endowment sets of individuals who received their assignment may be left behind. These objects are reassigned as endowments to individuals who are still in the market, that is, they are "inherited" by individuals who have not yet received their assignments. Furthermore, the objects in the initial endowment sets of individuals who are still in the market remain the individual endowments of these individuals. Thus, notice that each unassigned object is the endowment of exactly one individual who is still in the market. Now apply the TTC procedure to this reduced market with the new endowments.[57] Repeat this procedure until every individual has her assignment or all the objects are assigned. Since there exists at least one top trading cycle in every stage, this procedure leads to an allocation of the objects in a finite number of stages. In particular, there are at most as many stages as there are individuals or objects, whichever number is smaller, since in each stage at least one person receives her assignment. Furthermore, for any strict preferences of the individuals, the resulting allocation is unique.

A hierarchical exchange rule is determined by the initial endowments and the hierarchical endowment inheritance in later stages. While the initial endowment sets are given a priori, the hierarchical endowment inheritance may be endogenous. In particular, the inheritance of endowments may depend on the assignments made in earlier stages.

We explain how a hierarchical exchange rule works by means of the following example.

Example 4.6.1. Suppose $N = \{1, 2, 3\}$ and $A = \{a_1, a_2, a_3, a_4\}$ with a prior order $a_1 \prec a_2 \prec a_3 \prec a_4$. A hierarchical exchange rule is based on a collection of *inheritance trees*, one tree for each object. We will define this notion formally; for the time being we explain it through the current example. Figure 4.1 presents a collection of inheritance trees $\Gamma_{a_1}, \ldots, \Gamma_{a_4}$. To understand their structure, let us look at one of them, say Γ_{a_1}. Each maximal path of this tree has $\min\{|N|, |A|\} - 1 = 2$ edges. In any maximal path, each individual appears *at most* once at the nodes. For instance, individuals $1, 2$ and 3 appear at the nodes (in that order) in the left most path of Γ_{a_1}. Each object other than a_1 appears *exactly* once at the outgoing edges from the root (thus there are three edges from the root). For every subsequent node which is not the end node of a maximal path, each object other than a_1, that has *not* already appeared in the path from the root to that node, appears *exactly* once at the outgoing edges from that node. For instance, consider the node marked with 2 in the left most path of Γ_{a_1}. Since this node is not the end node of the left most maximal path and object a_2 has already appeared at the edge from the root to this node, objects a_3 and a_4 appear exactly once at the outgoing edges from this node. Thus, each object other than a_1 appears *at most* once at the edges in any maximal path of Γ_{a_1}. For instance, objects a_2 and a_3 appear at the edges (in that order) in the left most path of Γ_{a_1}. It can be verified that other inheritance trees have the same structure.

[57] In this TTC procedure, an individual *i* point to an individual *j* if *j* owns *i*'s most preferred object among the remaining objects.

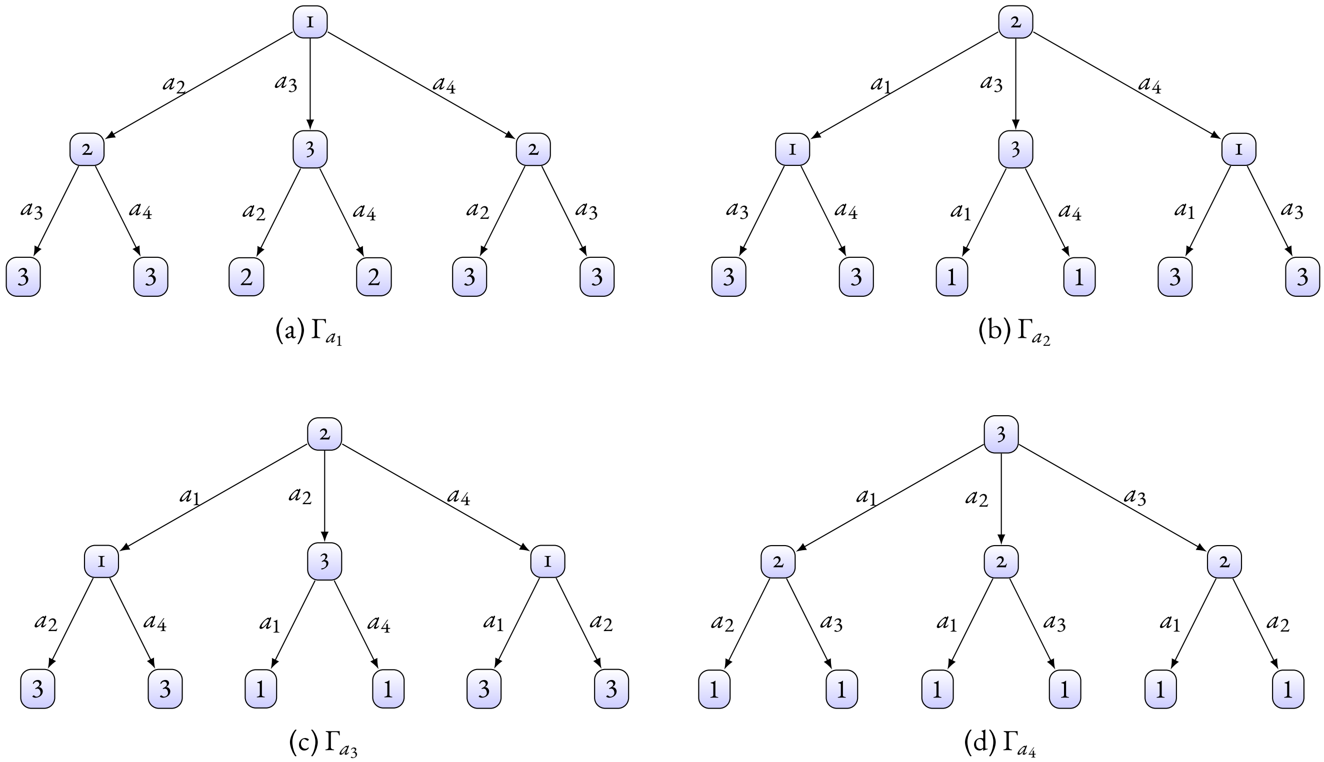


Figure 4.1: Inheritance trees for Example 4.6.1

Consider the hierarchical exchange rule based on the collection of inheritance trees given in Figure 4.1. We explain how to compute the outcome of the rule at a given preference profile. Consider the preference profile P_N as given below:

P_1	P_2	P_3
a_2	a_1	a_1
a_1	a_2	a_2
a_3	a_3	a_3
a_4	a_4	a_4

Table 4.1: Preference profile for Example 4.6.1

The outcome is computed through a number of stages. In each stage, endowments of the individuals are determined by means of the inheritance trees and TTC procedure is performed with respect to the endowments.

Stage 1.

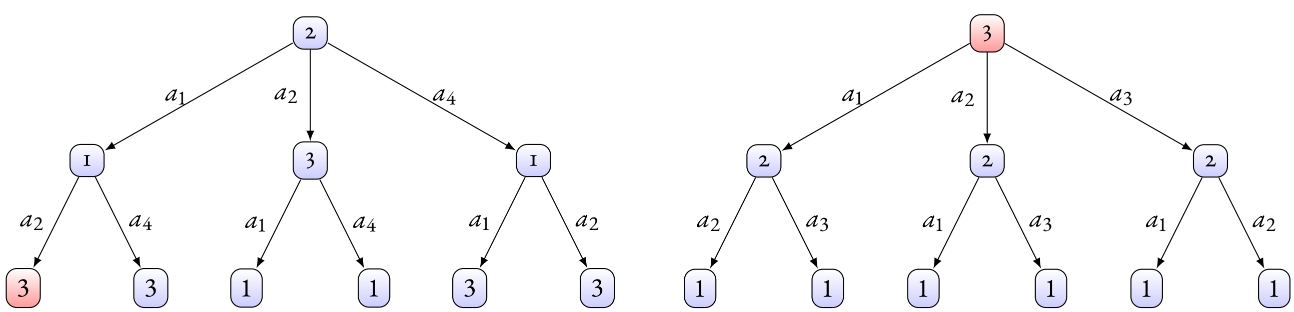
In Stage 1, the "owner" of an object a is the individual who is assigned to the root-node of the inheritance tree Γ_a. Thus, object a_1 is owned by individual 1, objects a_2 and a_3 are owned by individual 2, and object a_4 is owned by individual 3.

Once the endowments of the individuals are decided, TTC procedure is performed with respect to the endowments to decide the outcome of Stage 1. Individuals who are assigned some object in Stage 1 leave the market with the corresponding objects. It can be verified that for the preference profile P_N given in Table 4.1, individual 1 gets object a_2 and individual 2 gets object a_1 at the outcome of TTC procedure in this stage. So, individuals 1 and 2 leave the market with objects a_2 and a_1, respectively.

Stage 2.

As in Stage 1, the endowments of the individuals are decided first and then TTC procedure is performed with respect to the endowments. To decide the owner of a (remaining) object a, look at the root of the inheritance tree Γ_a. If the individual who appears there, say individual i, is remained in the market, then i becomes the owner of a. Otherwise, that is, if i is assigned an object in Stage 1, say b, then follow the edge from the root that is marked with b. If the individual appearing at the node following this edge, say j, is remained in the market, then j becomes the owner of a. Otherwise, that is, if j is assigned an object in Stage 1, say c, then follow the edge that is marked with c from the current node. As before, check whether the individual appearing at the end of this edge is remained in the market or not. Continue in this manner until an individual is found in the particular path who is not already assigned an object and decide that individual as the owner of a.

For the example at hand, the remaining market in Stage 2 consists of objects a_3 and a_4, and individual 3. Consider object a_3. Individual 2 appears at the root of Γ_{a_3}. Since individual 2 is assigned object a_1 in Stage 1, we follow the edge from the root that is marked with a_1 and come to individual 1. Since individual 1 is assigned object a_2, we follow the edge marked with a_2 from this node and come to individual 3. Since individual 3 is remained in the market, she becomes the owner of a_3. For object a_4, individual 3 appears at the root of Γ_{a_4} and she is remained in the market. So, individual 3 becomes the owner of a_4 in Stage 2. To emphasize the process of deciding the owner of an object, we have highlighted the node in red in the corresponding inheritance tree in Figure 4.2.



(a) Γ_{a_3} (b) Γ_{a_4}

Figure 4.2: Stage 2

Once the endowments are decided for Stage 2, TTC procedure is performed with respect to the endowments to decide the outcome of this stage. As in Stage 1, individuals who are assigned some object in Stage 2 leave the market with the corresponding objects. It can be verified that for the current example, individual 3 gets object a_3 in this stage. So, individual 3 leave the market with objects a_3.

Stage 3 is followed on the remaining market in a similar way as Stage 2. For the current example, everybody is assigned some object by the end of Stage 2 and hence the algorithm stops in this stage. Thus, individuals 1, 2, and 3 get objects a_2, a_1, and a_3, respectively, at the outcome of the hierarchical exchange rule.

4.6.2 Formal definition of hierarchical exchange rules

In what follows, we present a formal description of hierarchical exchange rules.

Inheritance trees

For a rooted tree T, the *level* of a node $v \in V(T)$ is defined as the number of edges appearing in the (unique) path from $r(T)$ to v.

Definition 4.6.1. For an object $a \in A$, an ***inheritance tree for*** $a \in A$ is defined as a tuple $\Gamma_a = \langle T_a, \zeta_a^{NI}, \zeta_a^{EO} \rangle$, where

(i) T_a is a rooted tree with

  (a) $\max\limits_{v \in V(T_a)} \mathit{level}(v) = \min\{|N|, |A|\} - 1$, and

  (b) $|E^{out}(v)| = |A| - \mathit{level}(v) - 1$ for all $v \in V(T_a)$ with $\mathit{level}(v) < \min\{|N|, |A|\} - 1$,

(ii) $\zeta_a^{NI} : V(T_a) \to N$ is a nodes-to-individuals function with $\zeta_a^{NI}(v) \neq \zeta_a^{NI}(\tilde{v})$ for all distinct $v, \tilde{v} \in V(T_a)$ that appear in same path, and

(iii) $\zeta_a^{EO} : E(T_a) \to A \setminus \{a\}$ is an edges-to-objects function with $\zeta_a^{EO}(e) \neq \zeta_a^{EO}(\tilde{e})$ for all distinct $e, \tilde{e} \in E(T_a)$ that appear in same path or have same source node (that is, $s(e) = s(\tilde{e})$).

In what follows, we provide two examples (for two different scenarios) of inheritance trees.

Example 4.6.2. Suppose $N = \{1, 2, 3\}$ and $A = \{a_1, a_2, a_3, a_4\}$ with a prior order $a_1 \prec a_2 \prec a_3 \prec a_4$.[58] Figure 4.3 presents an example of Γ_{a_1}.

[58] The ordering $\prec$ over A does not play any role in the definition of an inheritance tree.

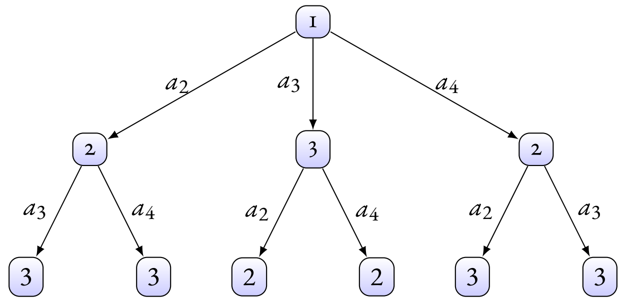


Figure 4.3: Example of Γ_{a_1}

Example 4.6.3. Suppose $N = \{1, 2, 3, 4\}$ and $A = \{a_1, a_2, a_3\}$ with a prior order $a_1 \prec a_2 \prec a_3$. Figure 4.4 presents another example of Γ_{a_1}.

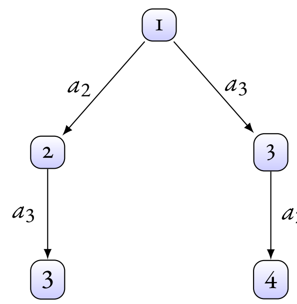


Figure 4.4: Example of Γ_{a_1}

Endowments

A hierarchical exchange rule works in several stages and in each stage, endowments of individuals are determined by using a (fixed) collection of inheritance trees.

Given a collection of inheritance trees $\Gamma = (\Gamma_a)_{a \in A}$, one for each object $a \in A$, we define a class of endowments $\mathcal{E}^\Gamma$ as follows:

(i) The ***initial endowment*** $\mathcal{E}_i^\Gamma(\emptyset)$ of individual i is given by

$$\mathcal{E}_i^\Gamma(\emptyset) = \{a \in A \mid \zeta_a^{NI}(r(T_a)) = i\}.$$

(ii) For all $S \subseteq N \setminus \{i\}$ and $B \subseteq A$ with $|S| = |B| \neq 0$, and all $\hat{\mu} \in \mathcal{M}(S, B)$, the ***endowment*** $\mathcal{E}_i^\Gamma(\hat{\mu})$ of individual i

is given by

$$\begin{aligned}\mathcal{E}_i^{\Gamma}(\hat{\mu}) = \{a \in A \setminus B \mid\ & \zeta_a^{NI}(r(T_a)) = i, \text{ or}\\ & \text{there exists a path } (v_a^1, \ldots, v_a^{r_a}) \text{ from } r(T_a) \text{ to } v_a^{r_a} \text{ in } \Gamma_a \text{ such that } \zeta_a^{NI}(v_a^{r_a}) = i\\ & \text{and for all } s = 1, \ldots, r_a - 1, \text{ we have } \zeta_a^{NI}(v_a^s) \in S \text{ and } \hat{\mu}(\zeta_a^{NI}(v_a^s)) = \zeta_a^{EO}(v_a^s, v_a^{s+1})\}.\end{aligned}$$

Iterative procedure to compute the outcome of a hierarchical exchange rule

For a given collection of inheritance trees $\Gamma = (\Gamma_a)_{a \in A}$, the ***hierarchical exchange rule*** f^{Γ} ***associated with*** Γ is defined by an iterative procedure with at most $\min\{|N|, |A|\}$ number of stages. Consider a preference profile $P_N \in \mathcal{P}_N$.

Stage 1.

Hierarchical Endowments (Initial Endowments): For all $i \in N$, $E_1(i, P_N) = \mathcal{E}_i^{\Gamma}(\emptyset)$.

Top Choices: For all $i \in N$, $T_1(i, P_N) = \tau(P_i)$.

Trading Cycles: For all $i \in N$,

$$C_1(i, P_N) = \begin{cases} \{j_1, \ldots, j_g\} & \text{if there exist } j_1, \ldots, j_g \in N \text{ such that} \\ & \text{for all } s = 1, \ldots, g,\ T_1(j_s, P_N) \in E_1(j_{s+1}, P_N), \text{ and} \\ & \text{for some } \hat{s} = 1, \ldots, g,\ j_{\hat{s}} = i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Since each individual can be in at most one trading cycle, $C_1(i, P_N)$ is well-defined for all $i \in N$. Furthermore, since both the number of individuals and the number of objects are finite, there is always at least one trading cycle. Note that $C_1(i, P_N) = \{i\}$ if $T_1(i, P_N) \in E_1(i, P_N)$.

Assigned Individuals: $W_1(P_N) = \{i \mid C_1(i, P_N) \neq \emptyset\}$.

Assignments: For all $i \in W_1(P_N)$, $f_i^{\Gamma}(P_N) = T_1(i, P_N)$.

Assigned Objects: $F_1(P_N) = \{T_1(i, P_N) \mid i \in W_1(P_N)\}$.

$\vdots$

This procedure is repeated iteratively in the remaining reduced market. For each stage t, define $W^t(P_N) = \bigcup_{u=1}^{t} W_u(P_N)$ and $F^t(P_N) = \bigcup_{u=1}^{t} F_u(P_N)$. In what follows, we present Stage $t+1$ of f^{Γ}.

Stage $t+1$.

Hierarchical Endowments (Non-initial Endowments): Let $\mu^t \in \mathcal{M}(W^t(P_N), F^t(P_N))$ such that for all $i \in W^t(P_N)$,

$$\mu^t(i) = f_i^{\Gamma}(P_N).$$

For all $i \in N \setminus W^t(P_N)$, $E_{t+1}(i, P_N) = \mathcal{E}_i^{\Gamma}(\mu^t)$.

Top Choices: For all $i \in N \setminus W^t(P_N)$, $T_{t+1}(i, P_N) = \tau(P_i, A \setminus F^t(P_N))$.

Trading Cycles: For all $i \in N \setminus W^t(P_N)$,

$$C_{t+1}(i, P_N) = \begin{cases} \{j_1, \ldots, j_g\} & \text{if there exist } j_1, \ldots, j_g \in N \setminus W^t(P_N) \text{ such that} \\ & \text{for all } s = 1, \ldots, g,\ T_{t+1}(j_s, P_N) \in E_{t+1}(j_{s+1}, P_N), \text{ and} \\ & \text{for some } \hat{s} = 1, \ldots, g,\ j_{\hat{s}} = i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Assigned Individuals: $W_{t+1}(P_N) = \{i \mid C_{t+1}(i, P_N) \neq \emptyset\}$.

Assignments: For all $i \in W_{t+1}(P_N)$, $f_i^{\Gamma}(P_N) = T_{t+1}(i, P_N)$.

Assigned Objects: $F_{t+1}(P_N) = \{T_{t+1}(i, P_N) \mid i \in W_{t+1}(P_N)\}$.

$$\vdots$$

This procedure is repeated iteratively until either all individuals are assigned or all objects are assigned. The hierarchical exchange rule f^{Γ} associated with Γ is defined as follows. For all $i \in N$,

$$f_i^{\Gamma}(P_N) = \begin{cases} T_t(i, P_N) & \text{if } i \in W_t(P_N) \text{ for some stage } t; \\ \emptyset & \text{otherwise.} \end{cases}$$

Since for every preference profile P_N and every individual i, there exists at most one stage t such that $i \in W_t(P_N)$, f^{Γ} is well-defined.

Remark 4.6.1. Note that a collection of inheritance trees do not uniquely identify a hierarchical exchange rule. More formally, two different collections of inheritance trees Γ and $\overline{\Gamma}$ may give rise to the same hierarchical exchange rule, that is, $f^{\Gamma} \equiv f^{\overline{\Gamma}}$.

We introduce the notion of *top-envy-proofness* for an assignment rule. It says that if an individual i is assigned the most preferred object of another individual j, then no matter how the individual j misreports her preference, individual i cannot be worse-off.[59] Thus, if an individual (here, j) is envious at another individual (here, i) for getting her (here, j's) top-ranked object, then the former one can never harm the latter. As the name suggests, top-envy-proofness is weaker than *envy-proofness* (that is, envy-proofness implies top-envy-proofness).[60] Loosely speaking, top-envy-proofness can be viewed as envy-proofness with respect to the top-ranked object of the envious individual.

Definition 4.7.1. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ satisfies ***top-envy-proofness*** condition if for all $P_N \in \mathcal{P}_N$ and all distinct $i, j \in N$, $\tau(P_j) = f_i(P_N)$ implies $f_i(\tilde{P}_j, P_{-j}) R_i f_i(P_N)$ for all $\tilde{P}_j \in \mathcal{P}_j$.

Next, we introduce the notion of an assignment rule being *manipulable through pairwise reallocation*. It captures the idea of manipulation where two individuals simultaneously misreport their preferences and finally benefit (with respect to their original assignments) by reshuffling their assignments that they obtain at the misreported preference profile. It further says that if any one of the two individuals misreports her preference as "planned", then her assignment will not depend whether the other individual misreports her preference as planned or reports truthfully.

Definition 4.7.2. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is *manipulable through pairwise reallocation* if there exist $P_N \in \mathcal{P}_N$, individuals $i, j \in N; i \neq j$, and $\tilde{P}_i \in \mathcal{P}_i$, $\tilde{P}_j \in \mathcal{P}_j$ such that

(i) $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i(P_N)$,

(ii) $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j(P_N)$, and

(iii) $f_i(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_i(\tilde{P}_i, P_j, P_{-i,j})$ and $f_j(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_j(P_i, \tilde{P}_j, P_{-i,j})$.

An assignment rule is ***pairwise reallocation-proof*** if it is not manipulable through pairwise reallocation.

Our next result provides a characterization of hierarchical exchange rules.

Theorem 4.7.1. *An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is strategy-proof, Pareto efficient, top-envy-proof, non-bossy, and pairwise reallocation-proof if and only if it is a hierarchical exchange rule.*

[59] Svensson & Larsson (2005) introduce the notion of *implicit property rights* of an assignment rule. It can be verified that a strategy-proof and Pareto efficient assignment rule reveals implicit property rights if it satisfies top-envy-proofness.

[60] An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ satisfies ***envy-proofness*** condition if for all $P_N \in \mathcal{P}_N$ and all distinct $i, j \in N$, $f_i(P_N) P_j f_j(P_N)$ implies $f_i(\tilde{P}_j, P_{-j}) R_i f_i(P_N)$ for all $\tilde{P}_j \in \mathcal{P}_j$.

The proof of this theorem is relegated to Section 4.12.

Since group strategy-proofness is equivalent to strategy-proofness and non-bossiness (see Proposition 4.4.1), we obtain the following corollary from Theorem 4.7.1.

Corollary 4.7.1. *An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is group strategy-proof, Pareto efficient, top-envy-proof, and pairwise reallocation-proof if and only if it is a hierarchical exchange rule.*

We now strengthen the notion of pairwise reallocation-proof by *group-wise reallocation-proof*. As the name suggests, instead of a pair of individuals, arbitrary groups of individuals are considered in group-wise reallocation-proof. Thus, group-wise reallocation-proof ensures that no group of individuals can be better off by misreporting their preferences and redistributing the objects they obtain at the misreported preference profile. Condition (iii) in Definition 4.7.2 is suitably modified for group of individuals.

To ease our presentation, for an assignment rule f, a preference profile P_N, and a set of individuals S, we denote by $f_S(P_N)$ the allocation over S according to $f(P_N)$. More formally, $f_S(P_N)$ is the allocation μ over S such that $\mu(i) = f_i(P_N)$ for all $i \in S$. With slight abuse of notation, by $\{f_S(P_N)\}$ we denote the set of objects which are assigned to the individuals in S at P_N, that is, $\{f_S(P_N)\} := \{a \in A \mid f_i(P_N) = a \text{ for some } i \in S\}$.

Definition 4.7.3. An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is *manipulable through group-wise reallocation* if there exist $P_N \in \mathcal{P}_N$, a set of individuals $S \subseteq N$, a preference profile $\tilde{P}_S$ of the individuals in S, and an allocation $\hat{\mu}$ of $\{f_S(\tilde{P}_S, P_{-S})\}$ over S where $\hat{\mu} \neq f_S(\tilde{P}_S, P_{-S})$ such that

(i) $\hat{\mu}(i) R_i f_i(P_N)$ for all $i \in S$,

(ii) $\hat{\mu}(j) P_j f_j(P_N)$ for some $j \in S$, and

(iii) $f_i(\tilde{P}_i, \tilde{P}_{S\setminus\{i\}}, P_{-S}) = f_i(\tilde{P}_i, P_{S\setminus\{i\}}, P_{-S})$ for all $i \in S$.

An assignment rule is ***group-wise reallocation-proof*** if it is not manipulable through group-wise reallocation.

Proposition 4.7.1. *Every hierarchical exchange rule satisfies group-wise reallocation-proofness.*

The proof of this proposition is relegated to Section 4.13.

We obtain the following corollary from Theorem 4.7.1 and Proposition 4.7.1.

Corollary 4.7.2. *An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is strategy-proof, Pareto efficient, top-envy-proof, non-bossy, and group-wise reallocation-proof if and only if it is a hierarchical exchange rule.*

The next corollary is obtained by combining Corollary 4.7.1 and Proposition 4.7.1.

Corollary 4.7.3. *An assignment rule $f : \mathcal{P}_N \to \mathcal{M}$ is group strategy-proof, Pareto efficient, top-envy-proof, and group-wise reallocation-proof if and only if it is a hierarchical exchange rule.*

4.8 Independence of the conditions in Theorem 4.7.1

In this section, we show that strategy-proofness, Pareto efficiency, top-envy-proofness, non-bossiness and pairwise reallocation-proofness are all independent for a hierarchical exchange rule. In particular, we show that no four of those conditions imply the fifth one.

Example 4.8.1. In this example, we show that Pareto efficiency, top-envy-proofness, non-bossiness, and pairwise reallocation-proofness do *not* imply strategy-proofness. Consider an allocation problem with three individuals $N = \{1, 2, 3\}$ and three objects $A = \{a_1, a_2, a_3\}$ with a prior order $a_1 \prec a_2 \prec a_3$. Consider the assignment rule f such that

$$f = \begin{cases} \text{Serial dictatorship with priority } (1 \succ 3 \succ 2) & \text{if } \tau(P_1) = \tau(P_2) = a_1, \text{ and } \tau(P_3) = a_2; \\ \text{Serial dictatorship with priority } (1 \succ 2 \succ 3) & \text{otherwise.} \end{cases}$$

Consider the preference profiles $P_N = (a_1a_2a_3, a_1a_2a_3, a_2a_1a_3)$ and $\tilde{P}_N = (a_1a_2a_3, a_2a_1a_3, a_2a_1a_3)$.[61] Note that only individual 2 changes her preference from P_N to $\tilde{P}_N$. This, together with the facts $f_2(P_N) = a_3, f_2(\tilde{P}_N) = a_2$, and $a_2P_2a_3$, implies f is not strategy-proof. It can be easily verified that f is Pareto efficient, top-envy-proof, non-bossy, and pairwise reallocation-proof.

Example 4.8.2. In this example, we show that strategy-proofness, top-envy-proofness, non-bossiness, and pairwise reallocation-proofness do *not* imply Pareto efficiency. Define f such that $f_i(P_N) = \emptyset$ for all $i \in N$ and all P_N. It is easy to verify that f satisfies strategy-proofness, top-envy-proofness, non-bossiness, and pairwise reallocation-proofness. However, from Remark 4.4.1, it follows that f does *not* satisfy Pareto efficiency.

Example 4.8.3. In this example, we show that strategy-proofness, Pareto efficiency, non-bossiness, and pairwise reallocation-proofness do *not* imply top-envy-proofness condition. Consider an allocation problem with three individuals $N = \{1, 2, 3\}$ and four objects $A = \{a_1, a_2, a_3, a_4\}$ with a prior order $a_1 \prec a_2 \prec a_3 \prec a_4$. Consider the assignment rule f such that

$$f = \begin{cases} \text{Serial dictatorship with priority } (2 \succ 1 \succ 3) & \text{if } \tau(P_1) = \tau(P_2) = a_1, \text{ and } \tau(P_3) = a_4; \\ \text{Serial dictatorship with priority } (1 \succ 2 \succ 3) & \text{otherwise.} \end{cases}$$

Consider the preference profiles $P_N = (a_1a_2a_3a_4, a_1a_2a_3a_4, a_1a_2a_3a_4)$ and $\tilde{P}_N = (a_1a_2a_3a_4, a_1a_2a_3a_4, a_4a_3a_2a_1)$. Note that only individual 3 changes her preference from P_N to $\tilde{P}_N$. This, together with the facts $f_1(P_N) = a_1$, $\tau(P_3) = a_1$,

[61] Here, we denote by $(a_1a_2a_3, a_2a_3a_1, a_3a_2a_1)$ a preference profile where individuals 1, 2 and 3 have preferences $a_1a_2a_3$, $a_2a_3a_1$, and $a_3a_2a_1$, respectively.

$f_1(\tilde{P}_N) = a_2$, and $a_1 P_1 a_2$, implies f is not top-envy-proof. It can be easily verified that f is strategy-proof, Pareto efficient, non-bossy, and pairwise reallocation-proof.

Example 4.8.4. In this example, we show that strategy-proofness, Pareto efficiency, top-envy-proofness, and pairwise reallocation-proofness do *not* imply non-bossiness. Consider an allocation problem with three individuals $N = \{1, 2, 3\}$ and three objects $A = \{a_1, a_2, a_3\}$ with a prior order $a_1 \prec a_2 \prec a_3$. Consider the assignment rule f such that

$$f = \begin{cases} \text{Serial dictatorship with priority } (1 \succ 2 \succ 3) & \text{if } a_1 P_1 a_3; \\ \text{Serial dictatorship with priority } (1 \succ 3 \succ 2) & \text{if } a_3 P_1 a_1. \end{cases}$$

Consider the preference profiles $P_N = (a_2a_1a_3, a_2a_1a_3, a_2a_1a_3)$ and $\tilde{P}_N = (a_2a_3a_1, a_2a_1a_3, a_2a_1a_3)$. Note that only individual 1 changes her preference from P_N to $\tilde{P}_N$. This, together with the facts $f(P_N) = [(1, a_2), (2, a_1), (3, a_3)]$ and $f(\tilde{P}_N) = [(1, a_2), (2, a_3), (3, a_1)]$, implies f is not non-bossy. It is easy to verify that f is strategy-proof, Pareto efficient, top-envy-proof, and pairwise reallocation-proof.

Example 4.8.5. In this example, we show that strategy-proofness, Pareto efficiency, top-envy-proofness, and non-bossiness do *not* imply pairwise reallocation-proofness. Consider an allocation problem with three individuals $N = \{1, 2, 3\}$ and three objects $A = \{a_1, a_2, a_3\}$ with a prior order $a_1 \prec a_2 \prec a_3$. Consider the hierarchical exchange rule f^Γ based on the collection of inheritance trees given in Figure 4.5. Consider the assignment rule f such that

$$f = \begin{cases} \text{Serial dictatorship with priority } (2 \succ 1 \succ 3) & \text{if } \tau(P_1) = \tau(P_2) = a_3, \text{ and } \tau(P_3) = a_1; \\ f^\Gamma & \text{otherwise.} \end{cases}$$

Consider the preference profile $P_N = (a_3a_2a_1, a_3a_2a_1, a_1a_2a_3)$ and the preferences $\tilde{P}_1 \in \mathcal{P}_1$, $\tilde{P}_3 \in \mathcal{P}_3$ such that $\tau(\tilde{P}_1) = a_1$ and $\tau(\tilde{P}_3) = a_3$. It follows from the construction of f that $f(P_N) = [(1, a_2), (2, a_3), (3, a_1)]$, $f_1(\tilde{P}_1, P_2, \tilde{P}_3) = f_1(\tilde{P}_1, P_2, P_3) = a_1$, $f_3(\tilde{P}_1, P_2, \tilde{P}_3) = f_3(P_1, P_2, \tilde{P}_3) = a_3$. These facts, along with the fact $a_3 P_1 a_2$, together imply f is not pairwise reallocation-proof. It can be easily verified that f is strategy-proof, Pareto efficient, top-envy-proof, and non-bossy.

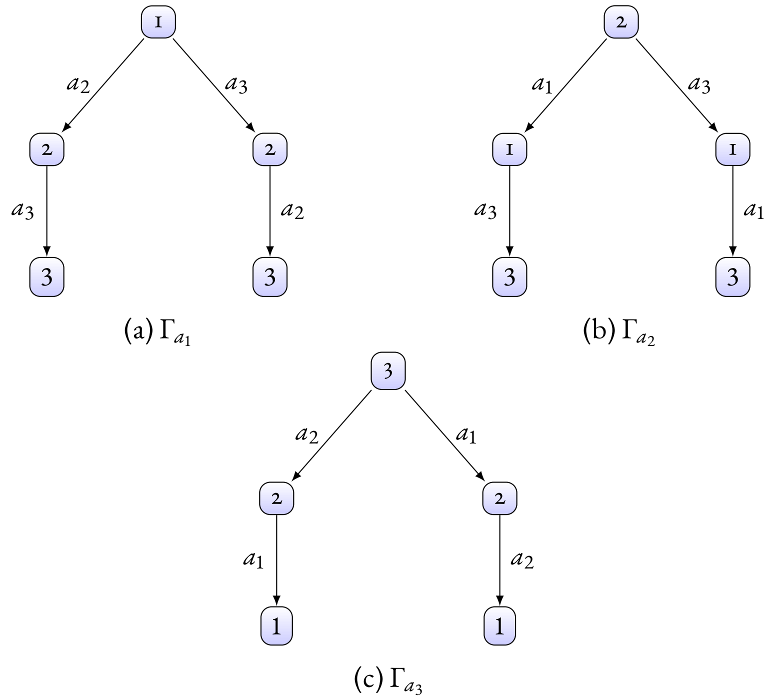


Figure 4.5: Inheritance trees for Example 4.8.5

Remark 4.8.1. The examples in this section also demonstrate that strategy-proofness, Pareto efficiency, top-envy-proofness, non-bossiness, and group-wise reallocation-proofness are all independent for a hierarchical exchange rule. To see this note that except for Example 4.8.2, all other examples deal with three individuals, and Pareto efficiency and pairwise reallocation-proofness together imply group-wise reallocation-proofness in such cases. The fact that the assignment rule in Example 4.8.2 satisfies group-wise reallocation-proofness is straightforward, and the assignment rule in Example 4.8.5 is *not* pairwise reallocation-proof (while being strategy-proof, Pareto efficient, top-envy-proof, and non-bossy), so it will not be group-wise reallocation-proof either.

4.9 Preliminaries for the proofs

For $a, b \in A$, let $P^{(a;b)}$ be a single-peaked preference (with respect to the given ordering $\prec$) such that

(i) $\tau(P^{(a;b)}) = a$, and

(ii) $P^{(a;b)}$ is a left (right) single-peaked preference if $b \preceq a$ ($a \prec b$).[62]

Remark 4.9.1. Since $\mathcal{P}_i$ is minimally rich single-peaked domain of preferences (with respect to the given ordering $\prec$) for all $i \in N$, we have $P^{(a;b)} \in \mathcal{P}_i$ for all $i \in N$ and all $a, b \in A$.

[62] By $\preceq$ we denote the weak part of $\prec$, that is, for all $a, b \in A$, $a \preceq b$ if and only if $[a \prec b \text{ or } a = b]$.

4.10 Proof of Proposition 4.4.1

(If part) Assume for contradiction that f is not group strategy-proof. Since f is not group strategy-proof, there exist $P_N \in \mathcal{P}_N$, $S \subseteq N$, and $P'_S \in \prod_{i \in S} \mathcal{P}_i$ such that $f_i(P'_S, P_{-S}) R_i f_i(P_N)$ for all $i \in S$ and $f_j(P'_S, P_{-S}) P_j f_j(P_N)$ for some $j \in S$. Consider the profile of preferences $\tilde{P}_S \in \prod_{i \in S} \mathcal{P}_i$ such that for all $i \in S$,

$$\tilde{P}_i = \begin{cases} P^{(f_i(P'_S, P_{-S}); f_i(P_N))} & \text{if } f_i(P_N) \neq \emptyset; \\ P'_i & \text{if } f_i(P_N) = \emptyset. \end{cases}$$

It follows from the construction of $\tilde{P}_S$ and Remark 4.9.1 that $\tilde{P}_S$ is well-defined.

First, we show that $f(\tilde{P}_S, P_{-S}) = f(P_N)$. Fix $j \in S$.

Claim 4.10.1. $f(\tilde{P}_j, P_{-j}) = f(P_N)$.

Proof of Claim 4.10.1. Suppose $f_j(P_N) = \emptyset$. Then, by strategy-proofness, we have $f_j(\tilde{P}_j, P_{-j}) = \emptyset$. Since $f_j(P_N) = \emptyset$ and $f_j(\tilde{P}_j, P_{-j}) = \emptyset$, by non-bossiness, we have

$$f(\tilde{P}_j, P_{-j}) = f(P_N). \tag{4.1}$$

Now, suppose $f_j(P_N) \neq \emptyset$. Then, by strategy-proofness, we have $f_j(\tilde{P}_j, P_{-j}) \tilde{R}_j f_j(P_N)$. Suppose $f_j(\tilde{P}_j, P_{-j}) \tilde{P}_j f_j(P_N)$. Since $f_j(\tilde{P}_j, P_{-j}) \tilde{P}_j f_j(P_N)$, it follows from the construction of $\tilde{P}_j$ that

$$f_j(P'_S, P_{-S}) \neq f_j(P_N), \text{ and} \tag{4.2a}$$

$$f_j(P'_S, P_{-S}) \preceq f_j(\tilde{P}_j, P_{-j}) \prec f_j(P_N) \quad \text{or} \quad f_j(P_N) \prec f_j(\tilde{P}_j, P_{-j}) \preceq f_j(P'_S, P_{-S}). \tag{4.2b}$$

Since $f_i(P'_S, P_{-S}) R_i f_i(P_N)$ for all $i \in S$, by (4.2a) we have $f_j(P'_S, P_{-S}) P_j f_j(P_N)$. This, together with (4.2b), implies $f_j(\tilde{P}_j, P_{-j}) P_j f_j(P_N)$, a contradiction to strategy-proofness. So, it must be that $f_j(\tilde{P}_j, P_{-j}) = f_j(P_N)$. By non-bossiness, the fact $f_j(\tilde{P}_j, P_{-j}) = f_j(P_N)$ implies

$$f(\tilde{P}_j, P_{-j}) = f(P_N). \tag{4.3}$$

(4.1) and (4.3) together complete the proof of Claim 4.10.1. □

Continuing in this manner, we can move the preferences of all individuals $j \in S$, from the preference P_j to $\tilde{P}_j$ one by one and obtain

$$f(\tilde{P}_S, P_{-S}) = f(P_N). \tag{4.4}$$

Next, we show that $f(\tilde{P}_S, P_{-S}) = f(P'_S, P_{-S})$. Fix $j \in S$. By strategy-proofness, we have $f_j(\tilde{P}_j, P'_{S\setminus\{j\}}, P_{-S})\tilde{R}_j f_j(P'_S, P_{-S})$. Moreover, it follows from the construction of $\tilde{P}_j$ that either $\tau(\tilde{P}_j) = f_j(P'_S, P_{-S})$ or $\tilde{P}_j = P'_j$. This, together with the fact $f_j(\tilde{P}_j, P'_{S\setminus\{j\}}, P_{-S})\tilde{R}_j f_j(P'_S, P_{-S})$, implies $f_j(\tilde{P}_j, P'_{S\setminus\{j\}}, P_{-S}) = f_j(P'_S, P_{-S})$. By non-bossiness, the fact $f_j(\tilde{P}_j, P'_{S\setminus\{j\}}, P_{-S}) = f_j(P'_S, P_{-S})$ implies

$$f(\tilde{P}_j, P'_{S\setminus\{j\}}, P_{-S}) = f(P'_S, P_{-S}).$$

Continuing in this manner, we can move the preferences of all individuals $j \in S$, from the preference P'_j to $\tilde{P}_j$ one by one and obtain

$$f(\tilde{P}_S, P_{-S}) = f(P'_S, P_{-S}). \tag{4.5}$$

However, (4.4) and (4.5) together imply $f(P'_S, P_{-S}) = f(P_N)$, a contradiction to the fact that $f_j(P'_S, P_{-S})P_j f_j(P_N)$ for some $j \in S$. This completes the proof of the "if" part of Proposition 4.4.1.

(Only-if part) It is obvious that group strategy-proofness implies strategy-proofness and non-bossiness. ■

4.11 Proof of Theorem 4.5.1

Suppose $A = \{a_1, a_2, \dots, a_m\}$ with a prior order $a_1 \prec a_2 \prec \cdots \prec a_m$, where $m \geq 3$. Assume for contradiction that there exists a strategy-proof, non-bossy, Pareto efficient, and strongly pairwise reallocation-proof assignment rule f on $\mathcal{P}_N$. Since $\mathcal{P}_i$ is minimally rich for all $i \in N$, there exists a preference profile $P^1_N \in \mathcal{P}_N$ such that $P^1_i = a_2a_1a_3\dots$ for all $i \in N$. Since $|N| \geq 3$, by Pareto efficiency, we have $\{a_1, a_2, a_3\} \subseteq \bigcup_{i\in N} \{f_i(P^1_N)\}$. Without loss of generality, assume $f_1(P^1_N) = a_1$, $f_2(P^1_N) = a_2$, and $f_3(P^1_N) = a_3$.

Since $\mathcal{P}_i$ is minimally rich for all $i \in N$, we can construct the preference profiles presented in Table 4.2. Here, l denotes an individual other than 1, 2, 3 (if any). Note that such an individual does not change her preference across the mentioned preference profiles.

Preference profiles	Individual 1	Individual 2	Individual 3	...	Individual l
P^2_N	$a_2\dots a_ma_1$	$a_1a_2a_3\dots$	$a_2a_1a_3\dots$	...	$a_2a_1a_3\dots$
P^3_N	$a_2\dots a_ma_1$	$a_2a_1a_3\dots$	$a_2a_1a_3\dots$	...	$a_2a_1a_3\dots$

Table 4.2: Preference profiles for Theorem 4.5.1

Since $f_1(P^1_N) = a_1$ and $f_2(P^1_N) = a_2$, it follows from strong pairwise reallocation-proofness of f that

$$f_1(P^2_N) = a_2 \text{ and } f_2(P^2_N) = a_1. \tag{4.6}$$

By (4.6) we have $f_2(P^2_N) = a_1$. This, together with strategy-proofness of f, implies $f_2(P^3_N) \in \{a_1, a_2\}$. Suppose $f_2(P^3_N) = a_1$. Since $f_2(P^2_N) = a_1$ and $f_2(P^3_N) = a_1$, by non-bossiness and (4.6), we have $f_1(P^3_N) = a_2$. However, since $a_2 P^1_1 a_1$, the facts $f_1(P^1_N) = a_1$ and $f_1(P^3_N) = a_2$ together contradict strategy-proofness of f. So, it must be that

$$f_2(P^3_N) = a_2. \tag{4.7}$$

Since $f_1(P^1_N) = a_1$ and $f_3(P^1_N) = a_3$, (4.7) together with strong pairwise reallocation-proofness of f, implies that

$$f_1(P^3_N) = a_3, f_2(P^3_N) = a_2, \text{ and } f_3(P^3_N) = a_1. \tag{4.8}$$

By (4.8) we have $f_2(P^3_N) = a_2$ and $f_3(P^3_N) = a_1$. Combining these facts with strong pairwise reallocation-proofness of f, we have $f_2(P^2_N) = a_1$ and $f_3(P^2_N) = a_2$. However, the fact that $f_3(P^2_N) = a_2$ contradicts (4.6). This completes the proof of Theorem 4.5.1. ■

4.12 Proof of Theorem 4.7.1

To prove Theorem 4.7.1, we use the notations introduced in Section 4.6. Furthermore, for a preference profile $P_N \in \mathcal{P}_N$ and a hierarchical exchange rule, we assume $F^0(P_N) = \emptyset$ and $W^0(P_N) = \emptyset$.

The following lemma is taken from Pápai (2000). She proves this lemma for the unrestricted domain. Since $\mathcal{P}_N$ is a subset of the unrestricted domain, the result holds for $\mathcal{P}_N$ as well.

Lemma 4.12.1 (Lemma 4 in Pápai (2000)). *Let f^Γ be a hierarchical exchange rule, $P_N \in \mathcal{P}_N$, and $i, j \in N$. Suppose $i \in W_s(P_N)$ and $f^\Gamma_j(P_N) \neq f^\Gamma_j(\tilde{P}_i, P_{-i})$ for some $\tilde{P}_i \in \mathcal{P}_i$. Then, either $j \in C_s(i, P_N)$ or $j \notin W^s(P_N)$.*

We obtain the following lemma from Lemma 4.12.1.

Lemma 4.12.2. *Let f^Γ be a hierarchical exchange rule and $P_N \in \mathcal{P}_N$. Suppose $i \in W_{s_i}(P_N)$, $j \in W_{s_j}(P_N)$ and $s_i < s_j$. Then, $f^\Gamma_i(\bar{P}_j, P_{-j}) = f^\Gamma_i(P_N)$ for all $\bar{P}_j \in \mathcal{P}_j$.*

Lemma 4.12.3 establishes a property which says that if an individual j prefers the assignment of another individual i of a hierarchical exchange rule, then it must be that i is assigned before j.

Lemma 4.12.3. *Let f^Γ be a hierarchical exchange rule and $P_N \in \mathcal{P}_N$. Suppose $i \in W_{s_i}(P_N)$ and $j \in W_{s_j}(P_N)$ such that $f^\Gamma_i(P_N) P_j f^\Gamma_j(P_N)$. Then, $s_i < s_j$.*

Proof of Lemma 4.12.3. Assume for contradiction that $s_j \leq s_i$. Since $j \in W_{s_j}(P_N)$, by the definition of f^Γ, we have $f_j^\Gamma(P_N) = \tau(P_j, A \setminus F^{j-1}(P_N))$. Furthermore, the fact $i \in W_{s_i}(P_N)$ together with the definition of f^Γ, implies that $f_i^\Gamma(P_N) \in A \setminus F^{s_i-1}(P_N)$. This, together with the fact $s_j \leq s_i$, yields $f_i^\Gamma(P_N) \in A \setminus F^{s_j-1}(P_N)$. However, the facts that $f_j^\Gamma(P_N) = \tau(P_j, A \setminus F^{s_j-1}(P_N))$ and $f_i^\Gamma(P_N) \in A \setminus F^{s_j-1}(P_N)$ together contradict the fact $f_i^\Gamma(P_N) P_j f_j^\Gamma(P_N)$. This completes the proof of Lemma 4.12.3. ■

4.12.1 Proof of the "if" part of Theorem 4.7.1

It follows from Pápai (2000) that every hierarchical exchange rule satisfies strategy-proofness, Pareto efficiency, top-envy-proofness, and non-bossiness on the unrestricted domain.[63] Since $\mathcal{P}_N$ is a subset of the unrestricted domain, it follows that every hierarchical exchange rule satisfies strategy-proofness, Pareto efficiency, top-envy-proofness, and non-bossiness on $\mathcal{P}_N$. In what follows, we show that every hierarchical exchange rule satisfies pairwise reallocation-proofness on $\mathcal{P}_N$.

Let f^Γ be a hierarchical exchange rule on $\mathcal{P}_N$. Assume for contradiction that f^Γ does not satisfy pairwise reallocation-proofness. Then, there must exist $P_N \in \mathcal{P}_N$, distinct $i, j \in N$, and $\tilde{P}_i \in \mathcal{P}_i, \tilde{P}_j \in \mathcal{P}_j$ such that

(i) $f_j^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i^\Gamma(P_N)$,

(ii) $f_i^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j^\Gamma(P_N)$, and

(iii) $f_i^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_i^\Gamma(\tilde{P}_i, P_{-i})$ and $f_j^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_j^\Gamma(\tilde{P}_j, P_{-j})$.

Claim 4.12.1. *$f_i^\Gamma(P_N)$ and $f_j^\Gamma(P_N)$ are distinct objects.*

Proof of Claim 4.12.1. Suppose $f_i^\Gamma(P_N) = \emptyset$. Since f^Γ is strategy-proof, $f_i^\Gamma(P_N) = \emptyset$ implies $f_i^\Gamma(\tilde{P}_i, P_{-i}) = \emptyset$. However, the facts that $f_i^\Gamma(\tilde{P}_i, P_{-i}) = \emptyset$ and $f_i^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_i^\Gamma(\tilde{P}_i, P_{-i})$ together imply $f_i^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = \emptyset$, a contradiction to the fact $f_i^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j^\Gamma(P)$. So, it must be that

$$f_i^\Gamma(P_N) \neq \emptyset. \tag{4.9}$$

Since $f_j^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i^\Gamma(P_N)$, (4.9) implies $f_j^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) \neq \emptyset$. This, together with the fact $f_j^\Gamma(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_j^\Gamma(\tilde{P}_j, P_{-j})$, implies $f_j^\Gamma(\tilde{P}_j, P_{-j}) \neq \emptyset$. Since f^Γ is strategy-proof, $f_j^\Gamma(\tilde{P}_j, P_{-j}) \neq \emptyset$ implies

$$f_j^\Gamma(P_N) \neq \emptyset. \tag{4.10}$$

(4.9) and (4.10) together complete the proof of Claim 4.12.1. □

[63] For details see Lemma 1, Lemma 7, and the main theorem of Pápai (2000).

It follows from Claim 4.12.1 that there exist stages s_i and s_j of f^T at P_N such that $i \in W_{s_i}(P_N)$ and $j \in W_{s_j}(P_N)$. Now, we complete the proof by distinguishing two cases.

Case 1: Suppose $s_j \leq s_i$.

Since f^T is Pareto efficient, $f_i^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j^T(P_N)$ implies that there exists $k \in N \setminus \{j\}$ such that $f_k^T(P_N) = f_i^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$. The facts $f_i^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_j f_j^T(P_N)$ and $f_k^T(P_N) = f_i^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ together imply $f_k^T(P_N) P_j f_j^T(P_N)$ and $f_k^T(P_N) \in A$. It follows from the fact $f_k^T(P_N) \in A$ that there exists a stage s_k of f^T at P_N such that $k \in W_{s_k}(P_N)$. Since $j \in W_{s_j}(P_N)$, $k \in W_{s_k}(P_N)$, and $f_k^T(P_N) P_j f_j^T(P_N)$, by Lemma 4.12.3, we have $s_k < s_j$. This, together with the fact $s_j \leq s_i$, implies $s_k < s_i$. Since $i \in W_{s_i}(P_N)$, $k \in W_{s_k}(P_N)$, and $s_k < s_i$, by Lemma 4.12.2, we have

$$f_k^T(P_N) = f_k^T(\tilde{P}_i, P_{-i}). \tag{4.11}$$

Furthermore, the facts $i \in W_{s_i}(P_N)$, $k \in W_{s_k}(P_N)$, and $s_k < s_i$ together imply $i \neq k$. Since $f_k^T(P_N) \in A$ and $i \neq k$, (4.11) implies

$$f_k^T(P_N) \neq f_i^T(\tilde{P}_i, P_{-i}). \tag{4.12}$$

However, the facts $f_i^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_i^T(\tilde{P}_i, P_{-i})$ and $f_k^T(P_N) = f_i^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$ together contradict (4.12).

Case 2: Suppose $s_i < s_j$.

If $f_j^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) P_i f_i^T(P_N)$, then the proof follows using a similar logic as for Case 1. Since $f_j^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) R_i f_i^T(P_N)$, let us assume

$$f_j^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j}) = f_i^T(P_N). \tag{4.13}$$

Since $i \in W_{s_i}(P_N)$, $j \in W_{s_j}(P_N)$, and $s_i < s_j$, by Lemma 4.12.2, we have

$$f_i^T(\tilde{P}_j, P_{-j}) = f_i^T(P_N). \tag{4.14}$$

Furthermore, since $f_j^T(\tilde{P}_j, P_{-j}) = f_j^T(\tilde{P}_i, \tilde{P}_j, P_{-i,j})$, by (4.13) and (4.14), we have

$$f_i^T(\tilde{P}_j, P_{-j}) = f_j^T(\tilde{P}_j, P_{-j}) = f_i^T(P_N). \tag{4.15}$$

However, by Claim 4.12.1, we have $f_i^T(P_N) \in A$. Since $f_i^T(P_N) \in A$ and $i \neq j$, (4.15) implies that $f^T(\tilde{P}_j, P_{-j})$ is not an allocation, a contradiction.

Since Cases 1 and 2 are exhaustive, it follows that f^T satisfies pairwise reallocation-proofness on $\mathcal{P}_N$.

4.12.2 Proof of the "only-if" part of Theorem 4.7.1

Let f be a strategy-proof, Pareto efficient, top-envy-proof, non-bossy, and pairwise reallocation-proof assignment rule. We will show that f is a hierarchical exchange rule.

Construction of the inheritance trees based on f

Fix $a \in A$. We proceed to construct an inheritance tree $\Gamma_a = \langle T_a, \zeta_a^{NI}, \zeta_a^{EO} \rangle$ for $a \in A$. Let T_a be a rooted tree that satisfies Condition (i) of Definition 4.6.1. Let $\zeta_a^{EO} : E(T_a) \to A \setminus \{a\}$ be an edges-to-objects function that satisfies Condition (iii) of Definition 4.6.1. We will define $\zeta_a^{NI} : V(T_a) \to N$, a nodes-to-individuals function, in accordance with property Condition (ii) of Definition 4.6.1 based on f.

Let $\mathcal{P}_N^0 \subseteq \mathcal{P}_N$ be the set of all preference profiles P_N such that $\tau(P_i) = a$ for all $i \in N$.

Lemma 4.12.4. *There exists $k \in N$ such that $f_k(P_N) = a$ for all $P_N \in \mathcal{P}_N^0$.*

Proof of Lemma 4.12.4. By Remark 4.4.1, for every given $P_N \in \mathcal{P}_N^0$, there exists an individual $k \in N$ such that $f_k(P_N) = a$. It remains to show that this individual is unique for all preference profile in $\mathcal{P}_N^0$, that is, $f_k(P_N) = f_k(P'_N) = a$ for all $P_N, P'_N \in \mathcal{P}_N^0$. Assume for contradiction that $f_j(P_N) = f_{j'}(P'_N) = a$ for some $P_N, P'_N \in \mathcal{P}_N^0$ and $j, j' \in N$ such that $j \neq j'$.

Since $f_j(P_N) = a$, $\tau(P_j) = a$, and $aP_kf_k(P_N)$ for all $k \neq j$, by moving the preferences of the individuals $k \neq j$ one by one from P_k to P'_k, and by applying top-envy-proofness condition every time, we obtain $f_j(P_j, P'_{-j}) = a$. Moreover, since $f_{j'}(P'_N) = a$ and $j \neq j'$, we have $f_j(P'_N) \neq a$. This, together with the fact $\tau(P'_j) = a$, implies $aP'_jf_j(P'_N)$. However, the facts $f_j(P_j, P'_{-j}) = a$ and $aP'_jf_j(P'_N)$ together contradict strategy-proofness of f. This completes the proof of Lemma 4.12.4. ■

By Lemma 4.12.4, there exists $i_1 \in N$ such that $f_{i_1}(P_N) = a$ for all $P_N \in \mathcal{P}_N^0$. Define $\zeta_a^{NI}(v_a^1) = i_1$ where v_a^1 is the root-node of T_a. Let $(v_a^1, \dots, v_a^r)$ with $r \geq 2$ be a path from v_a^1 to v_a^r in T_a. We define ζ_a^{NI} on $\{v_a^s \mid 1 \leq s \leq r\}$ in a recursive manner.

Assume that ζ_a^{NI} is defined on $\{v_a^s \mid 1 \leq s \leq r-1\}$. Let $\zeta_a^{NI}(v_a^s) = i_s$ for all $s = 1, \dots, r-1$. We proceed to define ζ_a^{NI} on v_a^r. Let $\mathcal{P}_N^{r-1} \subseteq \mathcal{P}_N$ be the set of all preference profiles P_N such that $P_{i_s} = P^{(\zeta_a^{EO}(v_a^s, v_a^{s+1}); a)}$ for all $s = 1, \dots, r-1$, and $\tau(P_i) = a$ otherwise. Note that for all $P_N \in \mathcal{P}_N^{r-1}$ and all $s, s' \in \{1, \dots, r-1\}$, $\tau(P_{i_s}) \neq \tau(P_{i_{s'}})$ if $s \neq s'$.

Lemma 4.12.5. *There exists $k \in N \setminus \{i_1, \dots, i_{r-1}\}$ such that $f_k(P_N) = a$ for all $P_N \in \mathcal{P}_N^{r-1}$.*

Proof of Lemma 4.12.5. We first prove two claims that we will use to complete the proof of Lemma 4.12.5.

Claim 4.12.2. *Let $S = \{h_1, \ldots, h_m\} \subsetneq N$ be a set of distinct individuals with $m < |A|$ and let $\{b_1, \ldots, b_m\} \in A \setminus \{a\}$ be a set of distinct objects. Consider the preference profile P_N such that $\tau(P_{h_u}) = b_u$ for all $u = 1, \ldots, m$ and $\tau(P_i) = a$ for all $i \notin S$. Then, there exists $j \in N \setminus S$ such that $f_j(P_N) = a$.*

Proof of Claim 4.12.2. By Remark 4.4.1, for all $c \in \{a, b_1, \ldots, b_m\}$, there exists $j_c \in N$ such that $f_{j_c}(P_N) = c$. It remains to show $j_a \notin S$. Assume for contradiction that $j_a \in S$. Let $\{j_1, \ldots, j_{t-1}\} \subseteq S$ and $j_t \notin S$ be such that $j_1 = j_a$, $f_{j_{s+1}}(P_N) = \tau(P_{j_s})$ for all $1 \leq s \leq t-1$. Since S is finite, to show such a sequence must exist, it is sufficient to show that $j_1, \ldots, j_{t-1}$ are all distinct. We show this in what follows. Assume for contradiction that l is the first index in the ordering $1, \ldots, t-1$ for which there exists $l < l' \leq t-1$ such that $j_l = j_{l'}$. Suppose $l = 1$. The facts $l = 1, j_l = j_{l'}$, $j_1 = j_a, f_{j_a}(P_N) = a$ and $f_{j_{l'}}(P_N) = \tau(P_{j_{l'-1}})$ together imply $\tau(P_{j_{l'-1}}) = a$. This is a contradiction since $j_{l'-1} \in S$, which in particular means $\tau(P_{j_{l'-1}}) \in \{b_1, \ldots, b_m\}$. Now, suppose $l > 1$. Then $j_l = j_{l'}, f_{j_l}(P_N) = \tau(P_{j_{l-1}})$ and $f_{j_{l'}}(P_N) = \tau(P_{j_{l'-1}})$ together imply

$$\tau(P_{j_{l-1}}) = \tau(P_{j_{l'-1}}). \tag{4.16}$$

However, by our assumption on l, $j_{l-1} \neq j_{l'-1}$. Because $j_{l-1}, j_{l'-1} \in S$ and $j_{l-1} \neq j_{l'-1}$, by the construction of P_N, $\tau(P_{j_{l-1}}) \neq \tau(P_{j_{l'-1}})$, a contradiction to (4.16). This shows that $j_1, \ldots, j_{t-1}$ are all distinct.

By the construction of $\{j_1, \ldots, j_t\}$, $\{f_{j_s}(P_N) \mid s = 1, \ldots, t\} = \{\tau(P_{j_s}) \mid s = 1, \ldots, t\}$. Define the allocation μ such that $\mu(i) = \tau(P_i)$ for all $i \in \{j_1, \ldots, j_t\}$ and $\mu(i) = f_i(P_N)$ for all $i \in N \setminus \{j_1, \ldots, j_t\}$. Clearly μ Pareto dominates $f(P_N)$ at P_N, which violates Pareto efficiency of f at P_N. This completes the proof of Claim 4.12.2. □

Claim 4.12.3. *For all $P_N \in \mathcal{P}_N^{r-1}$ and all $s = 1, \ldots, r-1$, we have $f_{i_s}(P_N) = \tau(P_{i_s})$.*

Proof of Claim 4.12.3. Fix $P_N \in \mathcal{P}_N^{r-1}$. We prove this in two steps.

Step 1. In this step, we show that $f_{i_s}(P_N) P_{i_s} a$ for all $s = 1, \ldots, r-1$. Assume for contradiction that $a R_{i_{s^*}} f_{i_{s^*}}(P_N)$ for some $s^* \in \{1, \ldots, r-1\}$. Consider the preference profile $\tilde{P}_N$ such that $\tilde{P}_{i_t} = P_{i_t}$ for all $t = 1, \ldots, s^*-1$ and $\tau(\tilde{P}_i) = a$, otherwise. By the recursive definition of ζ_a^{NI},

$$f_{i_{s^*}}(\tilde{P}_N) = a. \tag{4.17}$$

Since $\tau(\tilde{P}_i) = a$ for all $i \in N \setminus \{i_1, \ldots, i_{s^*-1}\}$, (4.17) implies that $f_{i_{s^*}}(\tilde{P}_N) = \tau(\tilde{P}_{i_{s^*}})$ and $f_{i_{s^*}}(\tilde{P}_N) \tilde{P}_i f_i(\tilde{P}_N)$ for all $i \in N \setminus \{i_1, \ldots, i_{s^*}\}$. Therefore, by moving the preferences of all the individuals $i \in N \setminus \{i_1, \ldots, i_{s^*}\}$ from $\tilde{P}_i$ to P_i, and by applying top-envy-proofness condition every time, it follows from the construction of $\tilde{P}_N$ that

$$f_{i_{s^*}}(\tilde{P}_{i_{s^*}}, P_{-i_{s^*}}) = a. \tag{4.18}$$

By strategy-proofness, (4.18) implies

$$f_{i_{s^*}}(P_N) R_{i_{s^*}} a. \tag{4.19}$$

By Claim 4.12.2, there exists $j \in N \setminus \{i_1, \dots, i_{r-1}\}$ such that $f_j(P_N) = a$. Since $j \in N \setminus \{i_1, \dots, i_{r-1}\}$ and $f_j(P_N) = a$, (4.19) implies $f_{i_{s^*}}(P_N) P_{i_{s^*}} a$, a contradiction to our assumption. This proves $f_{i_s}(P_N) P_{i_s} a$ for all $s = 1, \dots, r-1$.

Step 2. In this step, we show that $f_{i_s}(P_N) = \tau(P_{i_s})$ for all $s = 1, \dots, r-1$. Assume for contradiction that $f_{i_{s_1}}(P_N) \neq \tau(P_{i_{s_1}})$ for some $s_1 \in \{1, \dots, r-1\}$. Let $s_1, \dots, s_u$ be the *maximal* sequence of distinct elements such that $\{s_1, \dots, s_u\} \subseteq \{1, \dots, r-1\}$ and $f_{i_{s_{t+1}}}(P_N) = \tau(P_{i_{s_t}})$ for all $t = 1, \dots, u-1$. Let $j \in N$ be such that $f_j(P_N) = \tau(P_{i_{s_u}})$. By the maximality assumption of $s_1, \dots, s_u$, either $j \in N \setminus \{i_1, \dots, i_{r-1}\}$ or $j = i_{s_1}$. We distinguish the following two cases.

Case 1: Suppose $j \in N \setminus \{i_1, \dots, i_{r-1}\}$.

By the construction of s_u, we have $f_{i_{s_u}}(P_N) \neq \tau(P_{i_{s_u}})$. Also, since $s_u \in \{1, \dots, r-1\}$, by Step 1, $f_{i_{s_u}}(P_N) P_{i_{s_u}} a$. Combining the facts $f_{i_{s_u}}(P_N) \neq \tau(P_{i_{s_u}})$ and $f_{i_{s_u}}(P_N) P_{i_{s_u}} a$, we have

$$\tau(P_{i_{s_u}}) \; P_{i_{s_u}} \; f_{i_{s_u}}(P_N) \; P_{i_{s_u}} \; a. \tag{4.20}$$

Also, since $s_u \in \{1, \dots, r-1\}$, by the construction of P_N, we have $P_{i_{s_u}} = P^{(\tau(P_{i_{s_u}});a)}$. This, together with (4.20), implies

$$\tau(P_{i_{s_u}}) \prec f_{i_{s_u}}(P_N) \prec a \quad \text{or} \quad a \prec f_{i_{s_u}}(P_N) \prec \tau(P_{i_{s_u}}). \tag{4.21}$$

Since $j \in N \setminus \{i_1, \dots, i_{r-1}\}$, by the construction of P_N, we have $\tau(P_j) = a$. This, together with (4.21), implies

$$a \; P_j \, f_{i_{s_u}}(P_N) \; P_j \; \tau(P_{i_{s_u}}). \tag{4.22}$$

Since $f_j(P_N) = \tau(P_{i_{s_u}})$, (4.20) implies $f_j(P_N) P_{i_{s_u}} f_{i_{s_u}}(P_N)$. Furthermore, since $f_j(P_N) = \tau(P_{i_{s_u}})$, (4.22) implies $f_{i_{s_u}}(P_N) \; P_j \, f_j(P_N)$. However, the facts $f_j(P_N) P_{i_{s_u}} f_{i_{s_u}}(P_N)$ and $f_{i_{s_u}}(P_N) \; P_j \, f_j(P_N)$ together contradict Pareto efficiency of f at P_N.

Case 2: Suppose $j = i_{s_1}$.

By the construction of $\{s_1, \dots, s_u\}$ and j, we have $\{f_{i_{s_t}}(P_N) \mid t = 1, \dots, u\} = \{\tau(P_{i_{s_t}}) \mid t = 1, \dots, u\}$. Let μ be the allocation such that $\mu(i) = \tau(P_i)$ for all $i \in \{i_{s_t} \mid t = 1, \dots, u\}$ and $\mu(i) = f_i(P_N)$ for all $i \in N \setminus \{i_{s_t} \mid t = 1, \dots, u\}$. Clearly, μ Pareto dominates $f(P_N)$ at P_N, which violates Pareto efficiency of f at P_N.

Case 1 and Case 2 together complete Step 2, and Step 1 and Step 2 together complete the proof of Claim 4.12.3. □

Now we complete the proof of Lemma 4.12.5. By Claim 4.12.2, for every given $P_N \in \mathcal{P}_N^{r-1}$, there exists an individual

$k \in N \setminus \{i_1, \ldots, i_{r-1}\}$ such that $f_k(P_N) = a$. It remains to show that this individual is unique for all preference profile in $\mathcal{P}_N^{r-1}$, that is, $f_k(P_N) = f_k(\tilde{P}_N) = a$ for all $P_N, \tilde{P}_N \in \mathcal{P}_N^{r-1}$. Assume for contradiction that $f_j(P_N) = f_{\tilde{j}}(\tilde{P}_N) = a$ for some $P_N, \tilde{P}_N \in \mathcal{P}_N^{r-1}$ and $j, \tilde{j} \in N \setminus \{i_1, \ldots, i_{r-1}\}$ such that $j \neq \tilde{j}$.

Consider the preference profile $(\tilde{P}_{i_1}, P_{-i_1}) \in \mathcal{P}_N^{r-1}$. Since $P_N, (\tilde{P}_{i_1}, P_{-i_1}) \in \mathcal{P}_N^{r-1}$, by Claim 4.12.3, we have $f_{i_1}(P_N) = f_{i_1}(\tilde{P}_{i_1}, P_{-i_1})$. Using non-bossiness, $f_{i_1}(P_N) = f_{i_1}(\tilde{P}_{i_1}, P_{-i_1})$ implies

$$f(P_N) = f(\tilde{P}_{i_1}, P_{-i_1}).$$

Continuing in this manner, we can move the preferences of all individuals i_s, $s = 0, \ldots, r-1$, from the preference P_{i_s} to $\tilde{P}_{i_s}$ one by one and obtain

$$f(P_N) = f(\tilde{P}_{i_1}, \ldots, \tilde{P}_{i_{r-1}}, P_{-\{i_1,\ldots,i_{r-1}\}}). \tag{4.23}$$

The fact $f_j(P_N) = a$, together with (4.23), implies $f_j(\tilde{P}_{i_1}, \ldots, \tilde{P}_{i_{r-1}}, P_{-\{i_1,\ldots,i_{r-1}\}}) = a$. Since $j \in N \setminus \{i_1, \ldots i_{r-1}\}$ and $\tau(P_i) = a$ for all $i \in N \setminus \{i_1, \ldots i_{r-1}\}$, it follows from the fact $f_j(\tilde{P}_{i_1}, \ldots, \tilde{P}_{i_{r-1}}, P_{-\{i_1,\ldots,i_{r-1}\}}) = a$ that $f_j(\tilde{P}_{i_1}, \ldots, \tilde{P}_{i_{r-1}}, P_{-\{i_1,\ldots,i_{r-1}\}}) = \tau(P_j)$ and $f_j(\tilde{P}_{i_1}, \ldots, \tilde{P}_{i_{r-1}}, P_{-\{i_1,\ldots,i_{r-1}\}}) P_i f_i(\tilde{P}_{i_1}, \ldots, \tilde{P}_{i_{r-1}}, P_{-\{i_1,\ldots,i_{r-1}\}})$ for all $i \in N \setminus \{i_1, \ldots i_{r-1}, j\}$. Therefore, by moving the preferences of all the individuals $i \in N \setminus \{i_1, \ldots i_{r-1}, j\}$ from P_i to $\tilde{P}_i$, and by applying top-envy-proofness condition every time, we obtain

$$f_j(P_j, \tilde{P}_{-j}) = a. \tag{4.24}$$

Since $f_{\tilde{j}}(\tilde{P}_N) = a$ and $j \neq \tilde{j}$, we have $f_j(\tilde{P}_N) \neq a$. Moreover, $j \in N \setminus \{i_1, \ldots i_{r-1}\}$ implies $\tau(\tilde{P}_j) = a$. Combining the facts $f_j(\tilde{P}_N) \neq a$ and $\tau(\tilde{P}_j) = a$, we obtain $a \tilde{P}_j f_j(\tilde{P}_N)$. However, this, together with (4.24), contradicts strategy-proofness of f. This completes the proof of Lemma 4.12.5. ∎

By Lemma 4.12.5, there exists $i_r \in N \setminus \{i_1, \ldots i_{r-1}\}$ such that $f_{i_r}(P_N) = a$ for all $P_N \in \mathcal{P}_N^{r-1}$. Define $\zeta_a^{NI}(v_a^r) = i_r$. This completes the recursive definition of ζ_a^{NI}, and thereby completes the construction of Γ_a.

Similarly for each object, an inheritance tree is constructed. Thus, we have constructed a collection of inheritance trees Γ, based on the assignment rule f.

Now, we prove $f(P_N) = f^{\Gamma}(P_N)$ for all $P_N \in \mathcal{P}_N$, where f^{Γ} is the hierarchical exchange rule associated with Γ.

$f(P_N) = f^{\Gamma}(P_N)$ for all $P_N \in \mathcal{P}_N$

Fix $P_N \in \mathcal{P}_N$. We show $f(P_N) = f^{\Gamma}(P_N)$. We prove this by induction on the stages of f^{Γ} at P_N.

Base Case: *Assignments in Stage 1*.

(i) $f_i(P_N) = f_i^{\Gamma}(P_N)$ for all $i \in W^1(P_N)$, and

(ii) $f_i(P'_N) = f_i(P_N)$ for all $i \in W^1(P_N)$, where $P'_N \in \mathcal{P}_N$ is such that for all $i \in W^1(P_N)$ either $\tau(P'_i) = f_i(P_N)$ or $P'_i = P_i$.

Proof of the Base Case. First, we prove a claim that we use in the proof of the Base Case.

Claim 4.12.4. *Let $i \in N$ and let $a \in E_1(i, P_N)$. Suppose $\tilde{P}_N \in \mathcal{P}_N$ is such that $\tau(\tilde{P}_i) = a$. Then $f_i(\tilde{P}_N) = a$.*

Proof of Claim 4.12.4. By the definition of f^{Γ}, $a \in E_1(i, P_N)$ implies $\zeta_a^{NI}(v_a^1) = i$ where v_a^1 is the root-node of T_a.[64] By the construction of Γ_a, $\zeta_a^{NI}(v_a^1) = i$ implies that

$$f_i(\bar{P}_N) = a \text{ for all } \bar{P}_N \in \mathcal{P}_N \text{ with } \tau(\bar{P}_j) = a \text{ for all } j \in N. \tag{4.25}$$

Now we show $f_i(\tilde{P}_N) = a$ for all $\tilde{P}_N$ with $\tau(\tilde{P}_i) = a$. Consider the preference profile $(\tilde{P}_i, \hat{P}_{-i})$ such that $\tau(\hat{P}_j) = a$ for all $j \neq i$. By (4.25), we have $f_i(\tilde{P}_i, \hat{P}_{-i}) = a$. Since $\tau(\tilde{P}_i) = a$, $f_i(\tilde{P}_i, \hat{P}_{-i}) = a$, and $\tau(\hat{P}_j) = a$ for all $j \neq i$, we have $f_i(\tilde{P}_i, \hat{P}_{-i}) = \tau(\tilde{P}_i)$ and $f_i(\tilde{P}_i, \hat{P}_{-i}) \hat{P}_j f_j(\tilde{P}_i, \hat{P}_{-i})$ for all $j \neq i$. Therefore, by moving the preferences of all the individuals $j \neq i$ from $\hat{P}_j$ to $\tilde{P}_j$, and by applying top-envy-proofness condition every time, we have $f_i(\tilde{P}_N) = a$. This completes the proof of Claim 4.12.4. ☐

Now, we proceed to prove the Base Case. First we show (i) of the Base Case. Fix $i \in W^1(P_N)$. We complete the proof for (i) of the Base Case by using another level of induction on the number of individuals in $C_1(i, P_N)$.

Base Case (for (i) of the Base Case). Suppose $|C_1(i, P_N)| = 1$. It follows from the definition of f^{Γ} that $T_1(i, P_N) \in E_1(i, P_N)$ and $T_1(i, P_N) = \tau(P_i)$. Therefore, by Claim 4.12.4, we have

$$f_i(P_N) = T_1(i, P_N). \tag{4.26}$$

By the definition of f^{Γ}, $|C_1(i, P_N)| = 1$ means

$$f_i^{\Gamma}(P_N) = T_1(i, P_N). \tag{4.27}$$

By (4.26) and (4.27), we have $f_i(P_N) = f_i^{\Gamma}(P_N)$. This completes the proof of Base Case (for (i) of the Base Case). Note that since $P_N \in \mathcal{P}_N$ and $i \in W^1(P_N)$ are chosen arbitrarily, using similar logic as above, we have $f_j(\tilde{P}_N) = f_j^{\Gamma}(\tilde{P}_N)$ for all $\tilde{P}_N \in \mathcal{P}_N$ and all $j \in W^1(\tilde{P}_N)$ with $|C_1(j, \tilde{P}_N)| = 1$.

[64] Recall that $\Gamma_a = \langle T_a, \zeta_a^{NI}, \zeta_a^{EO} \rangle$.

Induction Hypothesis (for (i) of the Base Case). Let $u \geq 2$. Assume that $f_i(P_N) = f_i^\Gamma(P_N)$ for $|C_1(i, P_N)| = u - 1$. Assume, furthermore, that for all $\tilde{P}_N \in \mathcal{P}_N$ and all $j \in W^1(\tilde{P}_N)$ such that $|C_1(j, \tilde{P}_N)| = u - 1$, we have $f_j(\tilde{P}_N) = f_j^\Gamma(\tilde{P}_N)$.

We show $f_i(P_N) = f_i^\Gamma(P_N)$ for $|C_1(i, P_N)| = u$. Let $C_1(i, P_N) = \{j_1, \ldots, j_u\}$ such that for all $l = 1, \ldots, u$, $T_1(j_l, P_N) \in E_1(j_{l+1}, P_N)$, where $i = j_1$. Assume for contradiction that $f_{j_1}(P_N) \neq f_{j_1}^\Gamma(P_N)$.

Take $\hat{P}_{j_1} = P_{j_u}$ and $\hat{P}_{j_u} = P_{j_1}$. By the construction of $\hat{P}_{j_1}$ and the definition of f^Γ, it follows that $\tau(\hat{P}_{j_1}) \in E_1(j_1, P_N)$. Since $\tau(\hat{P}_{j_1}) \in E_1(j_1, P_N)$, by Claim 4.12.4, we have

$$f_{j_1}(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}) = f_{j_1}(\hat{P}_{j_1}, P_{-j_1}) = \tau(\hat{P}_{j_1}). \tag{4.28}$$

By the definition of $C_1(i, P_N)$ and the construction of $\hat{P}_{j_u}$, it follows that $|C_1(j_u, (\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}))| = |C_1(j_u, (\hat{P}_{j_u}, P_{-j_u}))| = u - 1$. Therefore, by Induction Hypothesis (for (i) of the Base Case), we have

$$f_{j_u}(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}) = f_{j_u}^\Gamma(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}), \text{ and} \tag{4.29a}$$

$$f_{j_u}(\hat{P}_{j_u}, P_{-j_u}) = f_{j_u}^\Gamma(\hat{P}_{j_u}, P_{-j_u}). \tag{4.29b}$$

By the definition of f^Γ, we have

$$f_{j_1}^\Gamma(P_N) = \tau(P_{j_1}), \text{ and} \tag{4.30a}$$

$$f_{j_u}^\Gamma(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}) = f_{j_u}^\Gamma(\hat{P}_{j_u}, P_{-j_u}) = \tau(\hat{P}_{j_u}). \tag{4.30b}$$

Since $\hat{P}_{j_u} = P_{j_1}$, combining (4.29) and (4.30b), we obtain

$$f_{j_u}(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}) = f_{j_u}(\hat{P}_{j_u}, P_{-j_u}) = \tau(P_{j_1}). \tag{4.31}$$

Since $f_{j_1}(P_N) \neq f_{j_1}^\Gamma(P_N)$ by our assumption, (4.30a) and (4.31) together imply

$$f_{j_u}(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}) P_{j_1} f_{j_1}(P_N). \tag{4.32}$$

By (4.28) and (4.31), we have

$$f_h(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1,j_u}) = f_h(\hat{P}_h, P_{-h}) \text{ for all } h = j_1, j_u. \tag{4.33}$$

Since $\hat{P}_{j_1} = P_{j_u}$, by (4.28), we have $f_{j_1}(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1 j_u}) = \tau(P_{j_u})$, which in particular means

$$f_{j_1}(\hat{P}_{j_1}, \hat{P}_{j_u}, P_{-j_1 j_u}) R_{j_u} f_{j_u}(P_N). \tag{4.34}$$

However, (4.32), (4.33) and (4.34) together contradict pairwise reallocation-proofness of f. This completes the proof of (i) of the Base Case. Note, furthermore, that since $P_N \in \mathcal{P}_N$ and $i \in W^1(P_N)$ are chosen arbitrarily, using similar logic as above, we have

$$f_j(\tilde{P}_N) = f^{\mathcal{T}}_j(\tilde{P}_N) \text{ for all } \tilde{P}_N \in \mathcal{P}_N \text{ and all } j \in W^1(\tilde{P}_N). \tag{4.35}$$

Now we show (ii) of the Base Case. Fix $P'_N \in \mathcal{P}_N$ such that for all $i \in W^1(P_N)$ either $\tau(P'_i) = f_i(P_N)$ or $P'_i = P_i$. From (i) of the Base Case, we have $f_i(P_N) = f^{\mathcal{T}}_i(P_N)$ for all $i \in W^1(P_N)$. This, together with the definition of $f^{\mathcal{T}}$, implies

$$f_i(P_N) = \tau(P_i) \text{ for all } i \in W^1(P_N). \tag{4.36}$$

It follows from the construction of P'_N and (4.36) that $\tau(P'_i) = \tau(P_i)$ for all $i \in W^1(P_N)$. This, together with the definition of $f^{\mathcal{T}}$, implies

$$W^1(P_N) \subseteq W^1(P'_N), \text{ and} \tag{4.37a}$$

$$f^{\mathcal{T}}_i(P'_N) = f^{\mathcal{T}}_i(P_N) \text{ for all } i \in W^1(P_N). \tag{4.37b}$$

(4.37) and (4.35) together complete the proof of (ii) of the Base Case. This completes the proof of the Base Case. □

Now, we proceed to prove the induction step.

Induction Hypothesis: Fix a stage $t \geq 2$. Assume that

(i) $f_i(P_N) = f^{\mathcal{T}}_i(P_N)$ for all $i \in W^{t-1}(P_N)$, and

(ii) $f_i(P'_N) = f_i(P_N)$ for all $i \in W^{t-1}(P_N)$, where P'_N is such that for all $i \in W^{t-1}(P_N)$ either $\tau(P'_i) = f_i(P_N)$ or $P'_i = P_i$.

We show

(i) $f_i(P_N) = f^{\mathcal{T}}_i(P_N)$ for all $i \in W^t(P_N)$, and

(ii) $f_i(P'_N) = f_i(P_N)$ for all $i \in W^t(P_N)$, where P'_N is such that for all $i \in W^t(P_N)$ either $\tau(P'_i) = f_i(P_N)$ or $P'_i = P_i$.

First, we prove a claim.

Claim 4.12.5. *Let $i \in N \setminus W^{t-1}(P_N)$ and let $a \in E_t(i, P_N)$. Suppose $\tilde{P}_N \in \mathcal{P}_N$ is such that $\tilde{P}_j = P_j$ for all $j \in W^{t-1}(P_N)$ and $\tau(\tilde{P}_i, A \setminus F^{t-1}(P_N)) = a$. Then, $f_i(\tilde{P}_N) = a$.*

Proof of Claim 4.12.5. Since $i \in N \setminus W^{t-1}(P_N)$ and $a \in E_t(i, P_N)$, it follows from the definition of f^Γ that there exists $r \geq 1$ such that there is a path $(v_a^1, \ldots, v_a^r)$ in T_a from v_a^1 (root-node of T_a) to v_a^r such that $\zeta_a^{NI}(v_a^r) = i$ and for all $s = 1, \ldots, r-1$, we have $\zeta_a^{NI}(v_a^s) \in W^{t-1}(P_N)$ and $f^\Gamma_{\zeta_a^{NI}(v_a^s)}(P_N) = \zeta_a^{EO}(v_a^s, v_a^{s+1})$. Note that for all $s = 1, \ldots, r-1$, by (i) of the Induction Hypothesis, $f_{\zeta_a^{NI}(v_a^s)}(P_N) = f^\Gamma_{\zeta_a^{NI}(v_a^s)}(P_N)$.

First, we show that $f_i(\bar{P}_N) = a$ for all $\bar{P}_N \in \mathcal{P}_N$ such that $\bar{P}_j = P_j$ for all $j \in W^{t-1}(P_N)$ and $\tau(\bar{P}_j) = a$ for all $j \in N \setminus W^{t-1}(P_N)$. Fix $\bar{P}_N \in \mathcal{P}_N$ such that $\bar{P}_j = P_j$ for all $j \in W^{t-1}(P_N)$ and $\tau(\bar{P}_j) = a$ for all $j \in N \setminus W^{t-1}(P_N)$. If $r = 1$, then $a \in E_1(i, P_N)$, and hence by Claim 4.12.4, we have $f_i(\bar{P}_N) = a$. Suppose $r > 1$. Let $S = \{\zeta_a^{NI}(v_a^s) \mid s = 1, \ldots, r-1\}$. By construction, $S \subseteq W^{t-1}(P_N)$. Consider the preference profile $\hat{P}_N$ such that $\hat{P}_j = P^{(f_j(P_N);a)}$ for all $j \in S$, $\tau(\hat{P}_j) = a$ for all $j \in W^{t-1}(P_N) \setminus S$, and $\hat{P}_j = \bar{P}_j$ for all $j \in N \setminus W^{t-1}(P_N)$. Since $f^\Gamma_{\zeta_a^{NI}(v_a^s)}(P_N) = \zeta_a^{EO}(v_a^s, v_a^{s+1})$ and $f_{\zeta_a^{NI}(v_a^s)}(P_N) = f^\Gamma_{\zeta_a^{NI}(v_a^s)}(P_N)$, by the construction of Γ_a, we have

$$f_i(\hat{P}_N) = a. \tag{4.38}$$

By the construction of $\hat{P}_N$, $\tau(\hat{P}_j) = a$ for all $j \in N \setminus S$. Since $i \in N \setminus W^{t-1}(P_N)$, $S \subseteq W^{t-1}(P_N)$, and $\tau(\hat{P}_j) = a$ for all $j \in N \setminus S$, by (4.38), we have $f_i(\hat{P}_N) = \tau(\hat{P}_i)$ and $f_i(\hat{P}_N)\hat{P}_j f_j(\hat{P}_N)$ for all $j \in W^{t-1}(P_N) \setminus S$. Therefore, by moving the preferences of all the individuals $j \in W^{t-1}(P_N) \setminus S$ from $\hat{P}_j$ to P_j, and by applying top-envy-proofness condition every time, we have

$$f_i(\underline{P}_N) = a, \tag{4.39}$$

where $\underline{P}_j = \hat{P}_j$ for all $j \notin W^{t-1}(P_N) \setminus S$ and $\underline{P}_j = P_j$ for all $j \in W^{t-1}(P_N) \setminus S$. By the construction of $\underline{P}_N$, for all $j \in W^{t-1}(P_N)$, either $\tau(\underline{P}_j) = f_j(P_N)$ or $\underline{P}_j = P_j$. Therefore, by (ii) of the Induction Hypothesis, we obtain

$$f_j(\underline{P}_N) = f_j(P_N) \text{ for all } j \in W^{t-1}(P_N). \tag{4.40}$$

Take $j \in S$. Consider the preference profile P''_N, where $P''_j = P_j$ and $P''_k = \underline{P}_k$ for all $k \neq j$. Since for all $k \in W^{t-1}(P_N)$, either $\tau(P''_k) = f_k(P_N)$ or $\underline{P}_k = P_k$, by (ii) of the Induction Hypothesis, $f_j(P''_N) = f_j(P_N)$. By (4.40), this means $f_j(P''_N) = f_j(\underline{P}_N)$. Since only individual j changes her preference from $\underline{P}_N$ to P''_N and $f_j(P''_N) = f_j(\underline{P}_N)$, by non-bossiness, we have $f(P''_N) = f(\underline{P}_N)$. By moving the preferences of all individuals $j \in S$ from $\underline{P}_j$ to P_j one by one and every time applying a similar logic, we conclude

$$f(\bar{P}_N) = f(\underline{P}_N). \tag{4.41}$$

Combining (4.39) and (4.41), we have

$$f_i(\bar{P}_N) = a. \tag{4.42}$$

Now we complete the proof of Claim 4.12.5. Take $\tilde{P}_N$ such that $\tilde{P}_j = P_j$ for all $j \in W^{t-1}(P_N)$ and $\tau(\tilde{P}_i, A \setminus F^{t-1}(P_N)) = a$. By (4.42) and the construction of $\bar{P}_N$, we have $f_i(\bar{P}_N) = \tau(\bar{P}_i)$ and $f_i(\bar{P}_N)\bar{P}_j f_j(\bar{P}_N)$ for all $j \notin W^{t-1}(P_N) \cup \{i\}$. Therefore, by moving the preferences of all the individuals $j \notin W^{t-1}(P_N) \cup \{i\}$ from $\bar{P}_j$ to $\tilde{P}_j$, and by applying top-envy-proofness condition every time, we obtain

$$f_i(\bar{P}_i, \tilde{P}_{-i}) = a. \tag{4.43}$$

Since f is strategy-proof, (4.43) implies

$$f_i(\tilde{P}_N)\tilde{R}_i a. \tag{4.44}$$

By the choice of $\tilde{P}_N$, we have $\tilde{P}_j = P_j$ for all $j \in W^{t-1}(P_N)$. By (ii) of the Induction Hypothesis

$$f_j(\tilde{P}_N) = f_j(P_N) \text{ for all } j \in W^{t-1}(P_N). \tag{4.45}$$

Since $\tau(\tilde{P}_i, A \setminus F^{t-1}(P_N)) = a$, (4.44) and (4.45) together imply $f_i(\tilde{P}_N) = a$. This completes the proof of Claim 4.12.5. □

Now the proof of the induction step follows by using similar logic as for the proof of the Base Case with Claim 4.12.5 in place of Claim 4.12.4. ■

4.13 Proof of Proposition 4.7.1

Let f^{Γ} be a hierarchical exchange rule on $\mathcal{P}_N$. Assume for contradiction that f^{Γ} does not satisfy group-wise reallocation-proofness. Then, there must exist $P_N \in \mathcal{P}_N$, a set of individuals $S \subseteq N$, a preference profile $\tilde{P}_S$ of the individuals in S, and an allocation $\hat{\mu}$ of $\{f^{\Gamma}_S(\tilde{P}_S, P_{-S})\}$ over S where $\hat{\mu} \neq f^{\Gamma}_S(\tilde{P}_S, P_{-S})$ such that

(i) $\hat{\mu}(i) R_i f^{\Gamma}_i(P_N)$ for all $i \in S$,

(ii) $\hat{\mu}(j) P_j f^{\Gamma}_j(P_N)$ for some $j \in S$, and

(iii) $f^{\Gamma}_i(\tilde{P}_i, \tilde{P}_{S\setminus\{i\}}, P_{-S}) = f^{\Gamma}_i(\tilde{P}_i, P_{S\setminus\{i\}}, P_{-S})$ for all $i \in S$.

Condition (ii) implies that there exists $i^* \in S$ such that $\hat{\mu}(i^*) P_{i^*} f^{\Gamma}_{i^*}(P_N)$. Moreover, it follows from the definition of $\hat{\mu}$ that there exists a set of individuals $\{i_1 = i^*, \ldots, i_m\} \subseteq S$ such that $\hat{\mu}(i_h) = f^{\Gamma}_{i_{h+1}}(\tilde{P}_S, P_{-S})$ for all $h = 1, \ldots, m$.

Since $\hat{\mu}(i^*) P_{i^*} f^{\Gamma}_{i^*}(P_N)$, this, together with Condition (iii) and strategy-proofness of f^{Γ}, implies $m \geq 2$. Combining all these observations with Condition (i), we have

$$f^{\Gamma}_{i_{h+1}}(\tilde{P}_S, P_{-S}) R_{i_h} f^{\Gamma}_{i_h}(P_N) \text{ for all } h = 2, \ldots, m, \text{ and} \tag{4.46a}$$

$$f^{\Gamma}_{i_2}(\tilde{P}_S, P_{-S}) P_{i_1} f^{\Gamma}_{i_1}(P_N). \tag{4.46b}$$

Claim 4.13.1. *$f^{\Gamma}_{i_h}(P_N) \in A$ for all $h = 1, \ldots, m$.*

Proof of Claim 4.13.1. Suppose $f^{\Gamma}_{i_2}(P_N) = \emptyset$. Since f^{Γ} is strategy-proof, $f^{\Gamma}_{i_2}(P_N) = \emptyset$ implies $f^{\Gamma}_{i_2}(\tilde{P}_{i_2}, P_{-i_2}) = \emptyset$. This, together with Condition (iii), yields $f^{\Gamma}_{i_2}(\tilde{P}_S, P_{-S}) = \emptyset$, a contradiction to (4.46b). So, it must be that

$$f^{\Gamma}_{i_2}(P_N) \neq \emptyset. \tag{4.47}$$

Combining (4.46a) and (4.47), we have $f^{\Gamma}_{i_3}(\tilde{P}_S, P_{-S}) \neq \emptyset$. This, together with Condition (iii), yields $f^{\Gamma}_{i_3}(\tilde{P}_{i_3}, P_{-i_3}) \neq \emptyset$. Since f^{Γ} is strategy-proof, $f^{\Gamma}_{i_3}(\tilde{P}_{i_3}, P_{-i_3}) \neq \emptyset$ implies

$$f^{\Gamma}_{i_3}(P_N) \neq \emptyset. \tag{4.48}$$

Continuing in this manner, we obtain

$$f^{\Gamma}_{i_h}(P_N) \neq \emptyset \text{ for all } h = 1, \ldots, m. \tag{4.49}$$

(4.49) completes the proof of Claim 4.13.1. □

It follows from Claim 4.13.1 that for all $h = 1, \ldots, m$, there exists a stage s_h of f^{Γ} at P_N such that $i_h \in W_{s_h}(P_N)$.

Claim 4.13.2. *$s_{h+1} \leq s_h$ for all $h = 2, \ldots, m$.*

Proof of Claim 4.13.2. Assume for contradiction that there exists a $h^* \in \{2, \ldots, m\}$ such that $s_{h^*} < s_{h^*+1}$. By (4.46a), we have $f^{\Gamma}_{i_{h^*+1}}(\tilde{P}_S, P_{-S}) R_{i_{h^*}} f^{\Gamma}_{i_{h^*}}(P_N)$. We complete the proof of Claim 4.13.2 by distinguishing two cases.

Case 1: Suppose $f^{\Gamma}_{i_{h^*+1}}(\tilde{P}_S, P_{-S}) P_{i_{h^*}} f^{\Gamma}_{i_{h^*}}(P_N)$.

Since f^{Γ} is Pareto efficient, $f^{\Gamma}_{i_{h^*+1}}(\tilde{P}_S, P_{-S}) P_{i_{h^*}} f^{\Gamma}_{i_{h^*}}(P_N)$ implies that there exists $k \in N \setminus \{i_{h^*}\}$ such that $f^{\Gamma}_k(P_N) = f^{\Gamma}_{i_{h^*+1}}(\tilde{P}_S, P_{-S})$. The facts $f^{\Gamma}_{i_{h^*+1}}(\tilde{P}_S, P_{-S}) P_{i_{h^*}} f^{\Gamma}_{i_{h^*}}(P_N)$ and $f^{\Gamma}_k(P_N) = f^{\Gamma}_{i_{h^*+1}}(\tilde{P}_S, P_{-S})$ together imply $f^{\Gamma}_k(P_N) P_{i_{h^*}} f^{\Gamma}_{i_{h^*}}(P_N)$ and $f^{\Gamma}_k(P_N) \in A$. It follows from the fact $f^{\Gamma}_k(P_N) \in A$ that there exists a stage s_k of f^{Γ} at P_N such that $k \in W_{s_k}(P_N)$. Since $i_{h^*} \in W_{s_{h^*}}(P_N)$, $k \in W_{s_k}(P_N)$, and $f^{\Gamma}_k(P_N) P_{i_{h^*}} f^{\Gamma}_{i_{h^*}}(P_N)$, by Lemma 4.12.3, we have $s_k < s_{h^*}$. This, together with the fact that $s_{b^*} < s_{b^*+1}$, implies $s_k < s_{b^*+1}$. Since $i_{b^*+1} \in W_{s_{b^*+1}}(P_N)$, $k \in W_{s_k}(P_N)$, and $s_k < s_{b^*+1}$, by Lemma 4.12.2, we have

$$f_k^\Gamma(P_N) = f_k^\Gamma(\tilde{P}_{i_{b^*+1}}, P_{-i_{b^*+1}}). \tag{4.50}$$

Furthermore, the facts $i_{b^*+1} \in W_{s_{b^*+1}}(P_N)$, $k \in W_{s_k}(P_N)$, and $s_k < s_{b^*+1}$ together imply $i_{b^*+1} \neq k$. Since $f_k^\Gamma(P_N) \in A$ and $i_{b^*+1} \neq k$, (4.50) implies

$$f_k^\Gamma(P_N) \neq f_{i_{b^*+1}}^\Gamma(\tilde{P}_{i_{b^*+1}}, P_{-i_{b^*+1}}). \tag{4.51}$$

However, the fact $f_k^\Gamma(P_N) = f_{i_{b^*+1}}^\Gamma(\tilde{P}_S, P_{-S})$ and Condition (iii) together contradict (4.51).

Case 2: Suppose $f_{i_{b^*+1}}^\Gamma(\tilde{P}_S, P_{-S}) = f_{i_{b^*}}^\Gamma(P_N)$.

Since $i_{b^*} \in W_{s_{b^*}}(P_N)$, $i_{b^*+1} \in W_{s_{b^*+1}}(P_N)$, and $s_{b^*} < s_{b^*+1}$, by Lemma 4.12.2, we have

$$f_{i_{b^*}}^\Gamma(\tilde{P}_{i_{b^*+1}}, P_{-i_{b^*+1}}) = f_{i_{b^*}}^\Gamma(P_N). \tag{4.52}$$

Furthermore, since $f_{i_{b^*+1}}^\Gamma(\tilde{P}_S, P_{-S}) = f_{i_{b^*}}^\Gamma(P_N)$, Condition (iii) and (4.52) together imply

$$f_{i_{b^*}}^\Gamma(\tilde{P}_{i_{b^*+1}}, P_{-i_{b^*+1}}) = f_{i_{b^*+1}}^\Gamma(\tilde{P}_{i_{b^*+1}}, P_{-i_{b^*+1}}) = f_{i_{b^*}}^\Gamma(P_N). \tag{4.53}$$

However, by Claim 4.13.1, we have $f_{i_{b^*}}^\Gamma(P_N) \in A$. Since $f_{i_{b^*}}^\Gamma(P_N) \in A$ and $i_{b^*} \neq i_{b^*+1}$, (4.53) implies that $f^\Gamma(\tilde{P}_{i_{b^*+1}}, P_{-i_{b^*+1}})$ is not an allocation, a contradiction.

Since Cases 1 and 2 are exhaustive, this completes the proof of Claim 4.13.2. □

Now, we complete the proof of Proposition 4.7.1. By Claim 4.13.2, we have $s_1 \leq s_2$. Moreover, by (4.46b), we have $f_{i_2}^\Gamma(\tilde{P}_S, P_{-S}) P_{i_1} f_{i_1}^\Gamma(P_N)$. Since $s_1 \leq s_2$ and $f_{i_2}^\Gamma(\tilde{P}_S, P_{-S}) P_{i_1} f_{i_1}^\Gamma(P_N)$, using a similar logic as for Case 1 in Claim 4.13.2, we get a contradiction. This completes the proof of Proposition 4.7.1. ■

5

Matchings under Stability, Minimum Regret, and Forced and Forbidden Pairs in Marriage Problem

5.1 Introduction

This chapter explores the possibilities of designing mechanisms satisfying properties such as (pairwise) stability, minimum regret, and forced and forbidden pairs in case of two-sided one-to-one matching problem (marriage problem).

(Pairwise) stability is a well-known property of a matching. Gale & Shapley (1962) provide an algorithm called *men-proposing/women-proposing deferred acceptance (MPDA/WPDA)* algorithm that produces a stable matching at every preference profile. It is well-known that the outcome of the MPDA (WPDA) algorithm is (i) men-maximal (women-maximal), that is, such an outcome maximizes the match of each man (woman) over all stable matchings, and (ii) women-pessimal (men-pessimal), that is, such an outcome minimizes the match of each woman (man) over all stable matchings.[65]

[65] See Gale & Shapley (1962), McVitie & Wilson (1971), Knuth (1976), and Abdulkadiroglu & Sönmez (2013) for details.

The main motivation of this chapter is to provide an algorithmic characterization of all stable matchings at every preference profile. The other motivation is to provide algorithms to construct stable matchings with additional desirable properties such as minimum regret and forced/forbidden pairs. The importance of a characterization of all stable matchings is well-established in the literature. McVitie & Wilson (1971) provide an iterative procedure to compute all stable matchings for the marriage problem and Martınez et al. (2004) extend that algorithm to two-sided many-to-many matching problem with *substitutable* preferences.[66] Irving & Leather (1986) provide an alternative method of computing all stable matchings for the marriage problem by using the lattice structure of the set of stable matchings. To the best of our knowledge, apart from *Gale-Shapley algorithm*, no direct algorithm that produces stable matching is introduced to the literature.[67] However, as discussed earlier, stable matchings produced by Gale-Shapley algorithm (Gale & Shapley, 1962) suffer from the problem that they are either extremely biased against men (in case of WPDA algorithm) or that towards women (in case of MPDA algorithm).

We present a class of algorithms that we call *men-women proposing deferred acceptance (MWPDA)* algorithms which can produce all stable matchings at every preference profile. Such an algorithm is based on a given collection of cut-off parameters one for each man. A cut-off parameter κ_m for a man m is an arbitrary integer between 1 and the number of women plus one. For a given collection of cut-off parameters the algorithm works in a sequence of stages as follows. At the beginning of Stage 1, each man m proposes each acceptable woman who appears in top κ_m positions according to his preference, and then WPDA algorithm is performed with respect to the proposals that the women receive. From a given stage we go to the subsequent stage if there is a man who (i) has not yet proposed all acceptable women according to his preference, and (ii) is unmatched at that given stage. Moreover, in any stage, if a man m was matched in the previous stage, then he proposes the same set of women as he did in the previous stage, otherwise he proposes the remaining set of acceptable women (that is, the acceptable women who do not appear in top κ_m positions according to his preference).

Theorem 5.3.1 of this chapter shows that the outcome of an MWPDA algorithm is stable at every preference profile for any cut-off vector. Theorem 5.3.2 shows that for any stable matching at a preference profile, there is a cut-off vector such that the MWPDA algorithm with respect to it will produce that stable matching. Theorem 5.3.3 provides a necessary and sufficient condition on the cut-off vectors so that the MWPDA algorithms with those cut-off vectors will converge at the first stage. We also discuss that these algorithms can be extended to produce all stable matchings in a two-sided many-to-one

[66] Kelso Jr & Crawford (1982) are the first to use the substitutability property to show the existence of stable matchings in a many-to-one model with money.

[67] McVitie & Wilson (1971) provide a method to compute all stable matchings at a preference profile. However, their method is lengthy in the sense that every time one needs to produce some particular stable matching, he/she has to start from the men-maximal (or women-maximal) stable matching and keep on producing all stable matchings that come in the process before he/she arrives at the intended stable matching. Another problem with this method is that it is not structured enough to produce stable matching with additional desirable properties.

matching problem (college admissions problem) in a way mentioned in Roth & Sotomayor (1989).

The notion of *minimum regret under stability* is introduced in Knuth (1976). It captures the idea of a Rawlsian welfare function. The regret of an agent in a matching is defined as the rank of his/her match according to his/her preference, and the regret of a matching is defined as the highest regret (over all agents) at that matching. A stable matching satisfies *minimum regret stable* property at a preference profile if it has the minimum regret among all the stable matchings at that preference profile.[68] Both MPDA and WPDA algorithms are far from satisfying the minimum regret under stability as their outcomes are either women-pessimal or men-pessimal. We provide a direct algorithm called the *sequential MWPDA* algorithm that produces a minimum regret stable matching at every preference profile.[69] We further show that the outcome of the sequential MWPDA algorithm is women-optimal in the set of all minimum regret stable matchings.

For practical reasons, sometimes one needs to construct stable matching with additional constraints. The notion of stable matching with *forced pairs* is introduced in Knuth (1976), and that with *forbidden pairs* is introduced in Dias et al. (2003). To the best of our knowledge, there is no direct algorithm that produces stable matching with these properties.[70] We provide an algorithm called the *conditional MWPDA algorithm* that produces stable matching with given sets of forced and forbidden pairs, whenever such a matching exists. We further show that whenever the conditional MWPDA algorithm produces such a matching, the outcome is women-optimal in the set of all stable matchings with given sets of forced and forbidden pairs.

5.1.1 Organization of the chapter

The chapter is organized as follows. The marriage problem framework is presented in Section 5.2. In Section 5.3, we present MWPDA algorithms and show that they produce all stable matchings at every preference profile for the marriage problem. We also provide a necessary and sufficient condition for the convergence of these algorithms at the first stage, and discuss how these algorithms can be used to construct all stable matchings for the college admissions problem. In Section 5.4, we present an algorithm that produces a minimum regret stable matching at every preference profile, and in Section 5.5, we present an algorithm that produces a stable matching with given sets of forced and forbidden pairs.

[68] Note that the regret of an unstable matching can be strictly less than the minimum regret under stability.

[69] Knuth (1976) provides an algorithm with runtime of the order $O(n^4)$ to find a minimum regret stable matching where n is the number of men, as well as the number of women. The algorithm given in Knuth (1976) is attributed to Alan Selkow. Later, Gusfield (1987) provide an algorithm that terminates in $O(n^2)$ time.

[70] Knuth (1976) provides an algorithm that produces a stable matching with a given set of forced pairs or reports that none exists, in $O(n^2)$ time, where n is the number of men, as well as the number of women. Later, Gusfield & Irving (1989) provide an algorithm that terminates in $O(|Q_1|^2)$ time, after pre-processing the preference lists in $O(n^4)$ time, where Q_1 is the set of given forced pairs. Dias et al. (2003) provide a computer algorithm that produces a stable matching with a given set of forced pairs Q_1 and a given set of forbidden pairs Q_2 in $O((|Q_1| + |Q_2|)^2)$ time, after pre-processing the preference lists in $O(n^4)$ time.

5.2 Model

For a finite set A, let $\mathbb{L}(A)$ denote the set of all strict linear orders over A.[71] An element P of $\mathbb{L}(A)$ is called a ***preference*** over A. For a preference $P \in \mathbb{L}(A)$, let R denote the weak part of P, that is, for all $a, b \in A$, aRb if and only if $[aPb$ or $a = b]$.

For $P \in \mathbb{L}(A)$ and $1 \leq k \leq |A|$, we define $T_k(P) := \{b \in A : |\{a : aRb\}| \leq k\}$. So, $T_k(P)$ is the set of top k elements of A according to P. Moreover, for $P \in \mathbb{L}(A)$ and $a \in A$, we define $rank(P, a) = k$ if $\big|\{b \in A : bPa\}\big| = k - 1$.

We introduce a specialized model of the two-sided matching problem, which will turn out to be sufficiently general to explore the general problem. The simplest two-sided matching problem to model is the "marriage problem", which consists of two (finite) sets of agents $M = \{m_1, \ldots, m_p\}$ and $W = \{w_1, \ldots, w_q\}$ ("men" and "women"). Throughout this chapter, we assume $p, q \geq 2$. We denote by $N = M \cup W$. Each $m \in M$ has a preference $P_m \in \mathbb{L}(W \cup \{\emptyset\})$ and each $w \in W$ has a preference $P_w \in \mathbb{L}(M \cup \{\emptyset\})$. A man m (woman w) is called ***acceptable*** for a woman w (man m) at a preference P_w (P_m) if $mP_w\emptyset$ ($wP_m\emptyset$). For $m \in M$ ($w \in W$), we denote by $\mathcal{A}(P_m)$ ($\mathcal{A}(P_w)$) the set of acceptable women (men) for m (w) at a preference P_m (P_w). By $P_N = (P_{m_1}, \ldots, P_{m_p}, P_{w_1}, \ldots P_{w_q})$, we denote a vector of all the agents' preferences, which will be referred to as a ***preference profile***.

Definition 5.2.1. A ***matching*** between M and W is a function $\mu : N \to N \cup \{\emptyset\}$ such that

(i) $\mu(m) \in W \cup \{\emptyset\}$ for all $m \in M$,

(ii) $\mu(w) \in M \cup \{\emptyset\}$ for all $w \in W$, and

(iii) $\mu(m) = w$ if and only if $\mu(w) = m$.

Definition 5.2.2. A matching $\mu : N \to N \cup \{\emptyset\}$ is ***individually rational*** at a preference profile P_N if $\mu(a)R_a\emptyset$ for all $a \in N$.

Definition 5.2.3. A pair $(m, w) \in M \times W$ is called a *blocking pair* of a matching $\mu : N \to N \cup \{\emptyset\}$ at a preference profile P_N if $wP_m\mu(m)$ and $mP_w\mu(w)$.

A matching $\mu : N \to N \cup \{\emptyset\}$ is called ***pairwise stable*** at a preference profile P_N if it is individually rational and has no blocking pairs at P_N.

Definition 5.2.4. A coalition $N' \subseteq N$ is called a *blocking coalition* of a matching $\mu : N \to N \cup \{\emptyset\}$ at a preference profile P_N if there exists another matching $\mu' : N \to N \cup \{\emptyset\}$ such that

[71] A ***strict linear order*** is a semiconnex, asymmetric, and transitive binary relation.

(i) $\mu'(a) \in N' \cup \{\emptyset\}$ for all $a \in N'$, and

(ii) $\mu'(a) P_a \mu(a)$ for all $a \in N'$.

If a matching $\mu : N \to N \cup \{\emptyset\}$ has no blocking coalition at a preference profile P_N, then it is called ***stable*** at P_N.

Remark 5.2.1. It is well-known that pairwise stability and stability are equivalent.[72] For this reason, we will say a matching is stable at a preference profile if and only if it is pairwise stable at that preference profile.

We denote by $\mathcal{C}(P_N)$ the set of all stable matchings at a preference profile P_N. It is well-known that $\mathcal{C}(P_N) \neq \emptyset$ for every preference profile P_N (see Gale & Shapley (1962) for details).

Definition 5.2.5. For a preference profile P_N and a set of matchings $\mathcal{M}$, a matching $\mu \in \mathcal{M}$ is ***women-optimal in*** $\mathcal{M}$ at P_N if $\mu(w) R_w \mu'(w)$ for all $w \in W$ and all $\mu' \in \mathcal{M}$. Similarly, one can define the notion a men-optimal matching in a set of matchings.[73]

A matching $\mu \in \mathcal{C}(P_N)$ is ***men-optimal (women-optimal) stable matching*** at P_N if μ is men-optimal (women-optimal) in $\mathcal{C}(P_N)$ at P_N.

It is well-known that a men-optimal (women-optimal) stable matching exists at every preference profile (see Gale & Shapley (1962) for details).

5.3 Algorithms for producing all stable matchings at a preference profile

An ***algorithm*** is a procedure that produces a matching at any preference profile. In this section, we provide a class of algorithms, called ***men-women proposing deferred acceptance (MWPDA)*** algorithms, which can produce every stable matching at a preference profile. These algorithms are built on well-known *deferred acceptance (DA)* algorithms. For the sake of completeness, we begin with a description (that is suitable for our purpose) of DA algorithms.

5.3.1 Deferred Acceptance algorithm

There are two types of deferred acceptance algorithms: women-proposing deferred acceptance (WPDA) and men-proposing deferred acceptance (MPDA). In the following, we provide a description of the WPDA algorithm at a preference profile P_N. The same of the MPDA algorithm can be obtained by interchanging the roles of women and men in the WPDA algorithm.

[72] See Roth & Sotomayor (1992) for details.

[73] Women-optimal (men-optimal) matching in an arbitrary set of matchings may not exist.

Step 1. Every woman w proposes her top-ranked acceptable man according to P_w[74]. Then, every man m who has at least one proposal keeps (tentatively) the top acceptable woman according to P_m among these proposals and rejects the rest. Denote the tentative matching thus obtained by μ_1.

Step 2. Every woman w who was rejected in the previous step, proposes the top acceptable man among those men who have not rejected her in earlier steps. Then, every man m who has at least one proposal, including any proposal tentatively kept from earlier steps, keeps (tentatively) the top acceptable woman among these proposals and rejects the rest. Denote the tentative matching thus obtained by μ_2.

$$\vdots$$

The process is then repeated from Step 2 till a step such that for each woman one of the following two happens: (i) she has proposed all acceptable men, (ii) she is accepted by some man who is acceptable to her. At this point, the tentative proposal accepted by a man becomes permanent. Call this the outcome of the WPDA algorithm at P_N.

Remark 5.3.1. Gale & Shapley (1962) show that at every preference profile P_N, there exists a unique men-optimal stable matching that is produced by the MPDA algorithm and a unique women-optimal stable matching that is produced by the WPDA algorithm.

Throughout this chapter, we denote the men-optimal and the women-optimal stable matching at a preference profile P_N by $\mu_M(P_N)$ and $\mu_W(P_N)$, respectively. Moreover, whenever the preference profile P_N is clear from the context, we drop it from these notations, that is, we write μ_M for $\mu_M(P_N)$, etc.

Remark 5.3.2. For all $\mu \in \mathcal{C}(P_N)$, $\mu_M(m) R_m \mu(m) R_m \mu_W(m)$ for all $m \in M$, and $\mu_W(w) R_w \mu(w) R_w \mu_M(w)$ for all $w \in W$.[75]

5.3.2 MWPDA algorithms

We begin with introducing a piece of notation that will simplify the presentation of our algorithm. For a preference $P_w \in \mathbb{L}(M \cup \{\emptyset\})$ and $M' \subseteq M$, define $P_w^{M'}$ as the preference that is obtained by moving the elements of $M' \cup \{\emptyset\}$ to the top of P_w maintaining their relative ordering. More formally, $P_w^{M'}$ is such that (i) for all $x, y \in M' \cup \{\emptyset\}$, $x P_w^{M'} y$ if and only if $x P_w y$, and (ii) for all $x \in M' \cup \{\emptyset\}$ and $y \in M \setminus M'$, we have $x P_w^{M'} y$.[76]

[74] That is, if the top-ranked man of a woman is acceptable, then she proposes him, otherwise she does not propose anybody.
[75] See Gale & Shapley (1962), McVitie & Wilson (1971), Knuth (1976), and Abdulkadiroglu & Sönmez (2013) for details.
[76] Note that such a preference $P_w^{M'}$ may not be unique since it does not specify the relative ranking of the elements of $M \setminus M'$.

An MWPDA algorithm is parameterized by a *cut-off vector*. A ***cut-off vector*** is defined as $\kappa = (\kappa_{m_1}, \ldots, \kappa_{m_p})$, where for all $m \in M$, $\kappa_m \in \{1, \ldots, q+1\}$ is the cut-off parameter of man m. An MWPDA algorithm involves a sequence of stages. At the beginning of a stage, say Stage s, each man m proposes a set of women (which is determined by the parameters). We denote this set by $W^s(m)$. The set of proposals that each $w \in W$ receives in that stage is denoted by $M^s(w)$, that is, $M^s(w) = \{m : w \in W^s(m)\}$.

Below, we present a detailed description (using the notations introduced above) of the MWPDA algorithm with cut-off vector κ at a preference profile P_N.

Stage 1. Take $W^1(m) = T_{\kappa_m}(P_m) \cap \mathcal{A}(P_m)$ for all $m \in M$. Perform the WPDA algorithm at the preference profile $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$. Let μ^1 be the outcome. If $W^1(m) = \mathcal{A}(P_m)$ for all $m \in M$ with $\mu^1(m) = \emptyset$, then conclude that the algorithm converges and define μ^1 as the outcome of the algorithm. Otherwise, go to Stage 2.

Stage 2. For all $m \in M$, take $W^2(m)$ such that

$$W^2(m) = \begin{cases} W^1(m) & \text{if } \mu^1(m) \neq \emptyset; \\ \mathcal{A}(P_m) \setminus W^1(m) & \text{if } \mu^1(m) = \emptyset \text{ and } W^1(m) \subsetneq \mathcal{A}(P_m); \\ \emptyset & \text{if } \mu^1(m) = \emptyset \text{ and } W^1(m) = \mathcal{A}(P_m).^{77} \end{cases}$$

Perform the WPDA algorithm at the preference profile $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^2(w_1)}, \ldots, P_{w_q}^{M^2(w_q)})$. Let μ^2 be the outcome. If $W^1(m) \cup W^2(m) = \mathcal{A}(P_m)$ for all $m \in M$ with $\mu^2(m) = \emptyset$, then conclude that the algorithm converges and define μ^2 as the outcome of the algorithm. Otherwise, go to Stage 3.

Stage 3. For all $m \in M$, take $W^3(m)$ such that

$$W^3(m) = \begin{cases} W^2(m) & \text{if } \mu^2(m) \neq \emptyset; \\ \mathcal{A}(P_m) \setminus \big(\bigcup_{s \leq 2} W^s(m)\big) & \text{if } \mu^2(m) = \emptyset \text{ and } \bigcup_{s \leq 2} W^s(m) \subsetneq \mathcal{A}(P_m); \\ \emptyset & \text{if } \mu^2(m) = \emptyset \text{ and } \bigcup_{s \leq 2} W^s(m) = \mathcal{A}(P_m). \end{cases}$$

Perform the WPDA algorithm at the preference profile $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^3(w_1)}, \ldots, P_{w_q}^{M^3(w_q)})$. Let μ^3 be the outcome. If $\bigcup_{s \leq 3} W^s(m) = \mathcal{A}(P_m)$ for all $m \in M$ with $\mu^3(m) = \emptyset$, then conclude that the algorithm converges and define μ^3 as the outcome of the algorithm. Otherwise, go to Stage 4.

[77] It follows from the definition of $W^1(m)$ that $W^1(m) \subseteq \mathcal{A}(P_m)$ for all $m \in M$. Therefore, the cases considered in this definition are exhaustive.

⋮

We continue this till a stage t^* such that $\bigcup_{s \leq t^*} W^s(m) = \mathcal{A}(P_m)$ for all $m \in M$ with $\mu^{t^*}(m) = \emptyset$. Since both the number of men and the number of women are finite, such a stage t^* must exist. At this stage, define the matching μ^{t^*} as the outcome the algorithm.

Remark 5.3.3. If $\kappa_m = q + 1$ for all $m \in M$, then the MWPDA algorithm with κ boils down to the WPDA algorithm.

We illustrate MWPDA algorithm by means of the following example.

Example 5.3.1. Let $M = \{m_1, m_2, m_3, m_4, m_5\}$ and $W = \{w_1, w_2, w_3, w_4, w_5\}$. Consider the preference profile P_N as given below:

P_{m_1}	P_{m_2}	P_{m_3}	P_{m_4}	P_{m_5}	P_{w_1}	P_{w_2}	P_{w_3}	P_{w_4}	P_{w_5}
w_1	w_1	w_2	w_1	w_1	m_2	m_4	m_5	m_2	m_3
w_2	w_3	w_1	w_2	w_2	m_5	m_5	m_2	m_3	m_1
w_3	w_2	w_3	w_5	w_3	m_1	m_2	m_4	m_1	m_5
w_4	w_4	w_4	w_4	w_4	$\emptyset$	m_1	m_3	m_5	$\emptyset$
w_5	w_5	w_5	w_3	w_5	m_3	m_3	$\emptyset$	m_4	m_2
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	m_4	$\emptyset$	m_1	$\emptyset$	m_4

Table 5.1: Preference profile for Example 5.3.1

Let the cut-off vector κ be such that $\kappa_{m_1} = 2$, $\kappa_{m_2} = 4$, $\kappa_{m_3} = 3$, $\kappa_{m_4} = 1$ and $\kappa_{m_5} = 2$. The MWPDA algorithm with κ at the preference profile P_N given in Table 5.1 works as follows.

Stage 1. Perform the WPDA algorithm at the preference profile $(P_{m_1}, \ldots, P_{m_5}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_5}^{M^1(w_5)})$ given in Table 5.2. The dots in Table 5.2 indicate that all preferences for the corresponding parts are irrelevant and can be chosen arbitrarily. To emphasize the process at Stage 1, for each man m we have highlighted the women in P_m in blue that m proposes, and for each woman w we have highlighted the men in P_w in blue who propose her.

P_{m_1}	P_{m_2}	P_{m_3}	P_{m_4}	P_{m_5}	P_{w_1}	P_{w_2}	P_{w_3}	P_{w_4}	P_{w_5}	$P_{w_1}^{M^1(w_1)}$	$P_{w_2}^{M^1(w_2)}$	$P_{w_3}^{M^1(w_3)}$	$P_{w_4}^{M^1(w_4)}$	$P_{w_5}^{M^1(w_5)}$
w_1	w_1	w_2	w_1	w_1	m_2	m_4	m_5	m_2	m_3	m_2	m_5	m_2	m_2	$\emptyset$
w_2	w_3	w_1	w_2	w_2	m_5	m_5	m_2	m_3	m_1	m_5	m_2	m_3	$\emptyset$	$\vdots$
w_3	w_2	w_3	w_5	w_3	m_1	m_2	m_4	m_1	m_5	m_1	m_1	$\emptyset$	$\vdots$	
w_4	w_4	w_4	w_4	w_4	$\emptyset$	m_1	m_3	m_5	$\emptyset$	$\emptyset$	m_3	$\vdots$		
w_5	w_5	w_5	w_3	w_5	m_3	m_3	$\emptyset$	m_4	m_2	m_3	$\emptyset$			
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	m_4	$\emptyset$	m_1	$\emptyset$	m_4	m_4	$\vdots$			

Table 5.2: Updated preference profile at Stage 1

The outcome of the WPDA algorithm at Stage 1 is $[(m_1,\emptyset),(m_2,w_1),(m_3,w_3),(m_4,\emptyset),(m_5,w_2)]$. Since $\mu^1(m_1)=\emptyset$ with $W^1(m_1)\subsetneq \mathcal{A}(P_{m_1})$, we go to Stage 2.

Stage 2. Perform the WPDA algorithm at the preference profile $(P_{m_1},\ldots,P_{m_5},P_{w_1}^{M^2(w_1)},\ldots,P_{w_5}^{M^2(w_5)})$ given in Table 5.3.

P_{m_1}	P_{m_2}	P_{m_3}	P_{m_4}	P_{m_5}	P_{w_1}	P_{w_2}	P_{w_3}	P_{w_4}	P_{w_5}	$P_{w_1}^{M^2(w_1)}$	$P_{w_2}^{M^2(w_2)}$	$P_{w_3}^{M^2(w_3)}$	$P_{w_4}^{M^2(w_4)}$	$P_{w_5}^{M^2(w_5)}$
w_1	w_1	w_2	w_1	w_1	m_2	m_4	m_5	m_2	m_3	m_2	m_4	m_2	m_2	m_1
w_2	w_3	w_1	w_2	w_2	m_5	m_5	m_2	m_3	m_1	m_5	m_5	m_4	m_1	$\emptyset$
w_3	w_2	w_3	w_5	w_3	m_1	m_2	m_4	m_1	m_5	$\emptyset$	m_2	m_3	m_4	m_4
w_4	w_4	w_4	w_4	w_4	$\emptyset$	m_1	m_3	m_5	$\emptyset$	m_3	m_3	$\emptyset$	$\emptyset$	$\vdots$
w_5	w_5	w_5	w_3	w_5	m_3	m_3	$\emptyset$	m_4	m_2	$\vdots$	$\emptyset$	m_1	$\vdots$	
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	m_4	$\emptyset$	m_1	$\emptyset$	m_4		$\vdots$	$\vdots$		

Table 5.3: Updated preference profile at Stage 2

The outcome of the WPDA algorithm at Stage 2 is $[(m_1,w_4),(m_2,w_1),(m_3,w_3),(m_4,w_2),(m_5,\emptyset)]$. Since $\mu^2(m_5)=\emptyset$ with $W^1(m_5)\cup W^2(m_5)\subsetneq \mathcal{A}(P_{m_5})$, we go to Stage 3.

Stage 3. Perform the WPDA algorithm at the preference profile $(P_{m_1},\ldots,P_{m_5},P_{w_1}^{M^3(w_1)},\ldots,P_{w_5}^{M^3(w_5)})$ given in Table 5.4.

P_{m_1}	P_{m_2}	P_{m_3}	P_{m_4}	P_{m_5}	P_{w_1}	P_{w_2}	P_{w_3}	P_{w_4}	P_{w_5}	$P_{w_1}^{M^3(w_1)}$	$P_{w_2}^{M^3(w_2)}$	$P_{w_3}^{M^3(w_3)}$	$P_{w_4}^{M^3(w_4)}$	$P_{w_5}^{M^3(w_5)}$
w_1	w_1	w_2	w_1	w_1	m_2	m_4	m_5	m_2	m_3	m_2	m_4	m_5	m_2	m_1
w_2	w_3	w_1	w_2	w_2	m_5	m_5	m_2	m_3	m_1	$\emptyset$	m_2	m_2	m_1	m_5
w_3	w_2	w_3	w_5	w_3	m_1	m_2	m_4	m_1	m_5	m_3	m_3	m_4	m_5	$\emptyset$
w_4	w_4	w_4	w_4	w_4	$\emptyset$	m_1	m_3	m_5	$\emptyset$	$\vdots$	$\emptyset$	m_3	m_4	m_4
w_5	w_5	w_5	w_3	w_5	m_3	m_3	$\emptyset$	m_4	m_2		$\vdots$	$\emptyset$	$\emptyset$	$\vdots$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	m_4	$\emptyset$	m_1	$\emptyset$	m_4			m_1	$\vdots$	

Table 5.4: Updated preference profile at Stage 3

The outcome of the WPDA algorithm at Stage 3 is $[(m_1, w_4), (m_2, w_1), (m_3, \emptyset), (m_4, w_2), (m_5, w_3)]$. Since $\mu^3(m_3) = \emptyset$ with $W^1(m_3) \cup W^2(m_3) \cup W^3(m_3) \subsetneq \mathcal{A}(P_{m_3})$, we go to Stage 4.

Stage 4. Perform the WPDA algorithm at the preference profile $(P_{m_1}, \dots, P_{m_5}, P_{w_1}^{M^4(w_1)}, \dots, P_{w_5}^{M^4(w_5)})$ given in Table 5.5.

P_{m_1}	P_{m_2}	P_{m_3}	P_{m_4}	P_{m_5}	P_{w_1}	P_{w_2}	P_{w_3}	P_{w_4}	P_{w_5}	$P_{w_1}^{M^3(w_1)}$	$P_{w_2}^{M^3(w_2)}$	$P_{w_3}^{M^3(w_3)}$	$P_{w_4}^{M^3(w_4)}$	$P_{w_5}^{M^3(w_5)}$
w_1	w_1	w_2	w_1	w_1	m_2	m_4	m_5	m_2	m_3	m_2	m_4	m_5	m_2	m_3
w_2	w_3	w_1	w_2	w_2	m_5	m_5	m_2	m_3	m_1	$\emptyset$	m_2	m_2	m_3	m_1
w_3	w_2	w_3	w_5	w_3	m_1	m_2	m_4	m_1	m_5	$\vdots$	$\emptyset$	m_4	m_1	m_5
w_4	w_4	w_4	w_4	w_4	$\emptyset$	m_1	m_3	m_5	$\emptyset$		$\vdots$	$\emptyset$	m_5	$\emptyset$
w_5	w_5	w_5	w_3	w_5	m_3	m_3	$\emptyset$	m_4	m_2			m_1	m_4	m_4
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	m_4	$\emptyset$	m_1	$\emptyset$	m_4			$\vdots$	$\emptyset$	$\vdots$

Table 5.5: Updated preference profile at Stage 4

The outcome of the WPDA algorithm at Stage 4 is $[(m_1, w_5), (m_2, w_1), (m_3, w_4), (m_4, w_2), (m_5, w_3)]$. Since $\mu^4(m) \neq \emptyset$ for all $m \in M$, the outcome of MWPDA algorithm with the cut-off vector κ is $[(m_1, w_5), (m_2, w_1), (m_3, w_4), (m_4, w_2), (m_5, w_3)]$.

5.3.3 MWPDA algorithms produce all stable matchings

In this section, we explore the stability of the outcome of an MWPDA algorithm. We also provide a sufficient condition on an MWPDA algorithm to produce a specific stable matching at the first step of the WPDA algorithm at Stage 1 of the mentioned MWPDA algorithm. Our next theorem shows that the outcome of an MWPDA algorithm at any preference profile and with any cut-off vector is stable.

Theorem 5.3.1. *For every preference profile P_N and every cut-off vector κ, the MWPDA algorithm with κ produces a stable matching at P_N.*

The proof of this theorem is relegated to Section 5.6; here we provide the idea of it. By Observation 5.6.1, the match of each man (weakly) improves (according to his preference) over the steps of the WPDA algorithm at any given stage. Next, we show the match of each woman (weakly) improves over the stages (Lemma 5.6.1). Finally, we combine these two facts to prove Theorem 5.3.1.

Now, we present the main result of this section. It says that every stable matching at any preference profile can be produced by an MWPDA algorithm with some cut-off vector. However, we prove a stronger version of this, which says that every stable matching at a preference profile can be produced at the first step of the WPDA algorithm at Stage 1 of an MWPDA algorithm by using a *suitable* cut-off vector.

Theorem 5.3.2. *Let P_N be a preference profile and let $\mu \in \mathcal{C}(P_N)$. Suppose the cut-off vector κ is such that $\kappa_m = rank(P_m, \mu(m))$ for all $m \in M$. Then, the MWPDA algorithm with cut-off vector κ produces μ at P_N. Furthermore, μ is produced at the first step of the WPDA algorithm at Stage 1 (of the mentioned MWPDA algorithm).*

The proof of this theorem is relegated to Section 5.7.2. It is worth mentioning that the cut-off vector κ defined in Theorem 5.3.2 is *not* the unique cut-off vector that produces μ at the first step of the WPDA algorithm at Stage 1.

In view of Theorem 5.3.2, one may think that if every stable matching can be produced at the first step of the WPDA algorithm at Stage 1 of an MWPDA algorithm, then why do we need a sequence of stages and a sequence of steps of the WPDA algorithm at each stage? The answer to this question is as follows. As it is evident from Theorem 5.3.2, the 'suitable' cut-off vector for a given stable matching that produces it at the first step of the WPDA algorithm at the first stage *cannot* be identified without using complete knowledge of that stable matching. Thus, in order to find *all* stable matchings at a preference profile, one needs to use MWPDA algorithm with arbitrary cut-off vectors (and consequently needs to go through several stages).

5.3.4 Convergence of MWPDA algorithms at the first stage

In this section, we discuss the convergence of an MWPDA algorithm. As we have mentioned in Section 5.3.3, for every stable matching there exists a cut-off vector so that the MWPDA algorithm with that converges at the first step of the WPDA algorithm at Stage 1 producing the stable matching. However, identifying such a cut-off vector requires complete knowledge of the stable matching. In view of this, we provide a necessary and sufficient condition on the cut-off vectors so that the MWPDA algorithms with those cut-off vectors converge at the first stage.

Recall that, we denote the men-optimal stable matching at a preference profile P_N by $\mu_M(P_N)$. Moreover, whenever the preference profile P_N is clear from the context, we drop it from this notation, that is, we write μ_M for $\mu_M(P_N)$.

Theorem 5.3.3. *Let P_N be a preference profile. The MWPDA algorithm with a cut-off vector κ at P_N converges at Stage 1 if and only if $\kappa_m \geq \min\left\{rank(P_m, \mu_M(m)), \max\{|\mathcal{A}(P_m)|, 1\}\right\}$ for all $m \in M$.*

The proof of this theorem is relegated to Section 5.7.1.

Remark 5.3.4. A cut-off vector κ with $\kappa_m \geq \min\left\{rank(P_m, \mu_M(m)), \max\{|\mathcal{A}(P_m)|, 1\}\right\}$ for all $m \in M$ does *not* guarantee the convergence of the MWPDA algorithm *at the first step* of the WPDA algorithm at the first stage, it might take several steps to converge.

5.3.5 Application to the college admissions problem

The "college admissions problem" is a many-to-one generalization of the marriage problem.[78] Every (many-to-one) stable matching in the college admissions problem where colleges' preferences satisfy *responsiveness* can be obtained from Theorem 5.3.2 in the following way.[79]

(i) Construct a marriage problem for the given college admissions problem (see Roth (1985) and Roth & Sotomayor (1989) for details on how to construct a related marriage problem).

(ii) Apply MWPDA algorithms to obtain all (one-to-one) stable matchings of the marriage problem.

(iii) Transform all (one-to-one) stable matchings of the marriage problem to their many-to-one versions by using a transformation as defined in Roth & Sotomayor (1989).

It follows from Lemma 1 in Roth & Sotomayor (1989) that the many-to-one matchings of the college admissions problem constructed in this manner will be the only pairwise stable matchings, and from Proposition 1 in Roth & Sotomayor (1989), that they will also be the only stable matchings.

5.4 A minimum regret stable algorithm

In this section, we present an algorithm which produces a stable matching at every preference profile with an additional desirable property, namely minimum regret. As we have mentioned in Remark 5.3.1, the outcome of the WPDA algorithm

[78] See Abdulkadiroglu & Sönmez (2013) for a formal description of the college admissions problem.
[79] The notion of responsiveness is due to Roth (1985), see Abdulkadiroglu & Sönmez (2013) for a formal definition of the same.

is women-optimal stable matching and that of the MPDA algorithm is men-optimal stable matching. In other words, both these algorithms are extremely biased.[80] However, as the following example demonstrates, MWPDA algorithms with suitable cut-off vectors can produce stable matchings that are not so biased.

Example 5.4.1. Let $M = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$. Consider the preference profile P_N given in Table 5.6.

P_{m_1}	P_{m_2}	P_{m_3}	P_{w_1}	P_{w_2}	P_{w_3}
w_1	w_2	w_3	m_2	m_3	m_1
w_2	w_3	w_1	m_3	m_1	m_2
w_3	w_1	w_2	m_1	m_2	m_3
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Table 5.6: Preference profile for Example 5.4.1

The outcome of the MPDA algorithm at P_N is

$$\mu_M = [(m_1, w_1), (m_2, w_2), (m_3, w_3)],$$

and that of the WPDA algorithm is

$$\mu_W = [(m_1, w_3), (m_2, w_1), (m_3, w_2)].$$

However, the outcome of the MWPDA algorithm with $\kappa = (2, 2, 2)$ is

$$\mu = [(m_1, w_2), (m_2, w_3), (m_3, w_1)].$$

Note that in μ_M, each man gets his best choice whereas each woman gets her worst, and conversely, in μ_W, each woman gets her best choice whereas each man gets his worst. However, in μ, all men and women get their second-best choices.

In view of this example, we define the notion of *minimum regret under stability*. This notion is introduced in Knuth (1976) as a desirable property of a matching.

Definition 5.4.1. Let P_N be a preference profile and let μ be a matching at P_N. Then, the ***regret*** of μ at P_N is defined as $\alpha(\mu, P_N) = \max_{a \in N} \; rank(P_a, \mu(a))$.

The ***minimum regret under stability*** at P_N is defined as $\alpha(P_N) = \min_{\mu \in \mathcal{C}(P_N)} \alpha(\mu, P_N)$.

[80] See Remark 5.3.2 for details.

It is worth mentioning that the regret of an unstable matching can be strictly less than the minimum regret under stability.

Definition 5.4.2. (Knuth, 1976) A matching μ^* is ***minimum regret stable*** at a preference profile P_N if it is stable at P_N and its regret is same as minimum regret under stability at P_N, that is, $\alpha(\mu^*, P_N) = \alpha(P_N)$.

An algorithm is called ***minimum regret stable*** if it produces a minimum regret stable matching at every preference profile.

It is worth noting that the minimum regret property has a close resemblance with a Rawlsian welfare function. Roughly speaking, this property tries to improve the outcome of the 'poorest of the poor' agent. Clearly, both WPDA and MPDA algorithms do not satisfy this property in general since these algorithms always maximize the matches of one side of the market (women or men), and consequently maximizes the regret of the other side. For instance, consider Example 5.4.1. The regret of the both outcomes of the WPDA and MPDA algorithms is 3. However, the same of the outcome of the MWPDA algorithm with $\kappa = (2, 2, 2)$ is 2.

5.4.1 Sequential MWPDA algorithm

In this section, we present an algorithm that is minimum regret stable. We call this the ***sequential MWPDA*** algorithm. It involves a sequence of rounds. At every round, it performs an MWPDA algorithm with a cut-off vector. Below, we present a formal description of this algorithm at a preference profile P_N. Let $\kappa^* = \max_{m \in M} rank(P_m, \mu_M(m))$.

Round 1. Perform the MWPDA algorithm with κ such that $\kappa_m = \kappa^*$ for all $m \in M$. Let μ_1^* be the outcome of the MWPDA algorithm at Round 1. If $rank(P_m, \emptyset) \leq \kappa^*$ for all $m \in M$ or $rank(P_w, \mu_1^*(w)) \leq \kappa^*$ for all $w \in W$, then conclude that the algorithm converges and define μ_1^* as the outcome of the sequential MWPDA algorithm. Else, go to Round 2.

Round 2. Perform the MWPDA algorithm with κ such that $\kappa_m = \kappa^* + 1$ for all $m \in M$. Let μ_2^* be the outcome of the MWPDA algorithm at Round 2. If $rank(P_m, \emptyset) \leq \kappa^* + 1$ for all $m \in M$ or $rank(P_w, \mu_2^*(w)) \leq \kappa^* + 1$ for all $w \in W$, then conclude that the algorithm converges and define μ_2^* as the outcome of the sequential MWPDA algorithm. Else, go to Round 3.

$$\vdots$$

Continue this till a round k such that either we have $rank(P_m, \emptyset) \leq \kappa^* + k - 1$ for all $m \in M$ or $rank(P_w, \mu_k^*(w)) \leq \kappa^* + k - 1$ for all $w \in W$ for the *first time* at Round k.[81] In other words, k is such that for all round $l < k$, there exists

[81] Since κ_m cannot be bigger than $q + 1$, such a round must exist.

$m \in M$ with $rank(P_m, \emptyset) > \kappa^* + l - 1$ and $w \in W$ with $rank(P_w, \mu_l^*(w)) > \kappa^* + l - 1$. Define μ_k^* as the outcome of the sequential MWPDA algorithm.

Remark 5.4.1. It is worth noting that in order to execute the sequential MWPDA algorithm at a preference profile P_N, first one needs to compute the men-optimal stable matching at P_N.

Remark 5.4.2. By Theorem 5.3.3, the MWPDA algorithm used at every round of the sequential MWPDA algorithm converges at Stage 1. This ensures quick convergence of the sequential MWPDA algorithm.

Our next result says that the sequential MWPDA algorithm produces the women-optimal matching in the set of all minimum regret stable matchings.

Theorem 5.4.1. *The sequential MWPDA algorithm is minimum regret stable. Furthermore, the outcome of the sequential MWPDA algorithm is women-optimal in the set of all minimum regret stable matchings.*

The proof of this theorem is relegated to Section 5.8.

5.5 Stable matching with forced and forbidden pairs

The notion of stable matching with *forced pairs* is introduced in Knuth (1976), and that with *forbidden pairs* is introduced in Dias et al. (2003). In this section, we provide an algorithm that produces stable matching with forced and forbidden pairs, whenever such a matching exists.

Definition 5.5.1. Given a set of pairs $Q_1 \subseteq M \times W$, we say a matching μ is with ***forced pairs*** Q_1 if every pair in Q_1 is matched in μ, that is, $\mu(m) = w$ for all $(m, w) \in Q_1$.

Definition 5.5.2. Given a set of pairs $Q_2 \subseteq M \times W$, we say a matching μ is with ***forbidden pairs*** Q_2 if no pair in Q_2 is matched in μ, that is, $\mu(m) \neq w$ for all $(m, w) \in Q_2$.

5.5.1 Conditional MWPDA algorithm

Consider a preference profile P_N and let Q_1 be a set of forced pairs and Q_2 be a set of forbidden pairs. Note that for all $(m, w), (m', w') \in Q_1$ with $(m, w) \neq (m', w')$, we have $m \neq m'$ and $w \neq w'$.[82] For $m \in M$, with slight abuse of notation, we say $m \in Q_1$, if there exists $w \in W$ such that $(m, w) \in Q_1$.

In what follows, we present an algorithm, called ***conditional MWPDA algorithm given*** (Q_1, Q_2), that produces a stable matching with forced pairs Q_1 and forbidden pairs Q_2, whenever such a matching exists. The algorithm involves a

[82] Otherwise there will be no stable matching with forced pairs Q_1.

sequence of rounds. At every round, an MWPDA algorithm is performed with a cut-vector κ such that $\kappa_m = rank(P_m, w)$ for all $m \in Q_1$ with $(m, w) \in Q_1$. The cut-off parameters for other men may change over rounds; they are defined at the beginning of each round of the conditional MWPDA algorithm.

Round 1. Define κ^1 such that for all $m \notin Q_1$, $\kappa^1_m = rank(P_m, \emptyset)$. Perform the MWPDA algorithm with κ^1. Let μ^*_1 be the outcome of the MWPDA algorithm at Round 1.

(i) If μ^*_1 is with forced pairs Q_1 and forbidden pairs Q_2, then conclude that the algorithm converges and define μ^*_1 as the outcome of the algorithm.

(ii) Else, if there exists a pair $(m, w) \in Q_1$ such that $\mu^*_1(m) \neq w$, then conclude that the algorithm STOPS.

(iii) Else, go to Round 2.

Round 2. Define κ^2 such that for all $m \notin Q_1$,

$$\kappa^2_m = \begin{cases} rank(P_m, \mu^*_1(m)) & \text{if } (m, \mu^*_1(m)) \notin Q_2; \\ rank(P_m, \mu^*_1(m)) - 1 & \text{if } (m, \mu^*_1(m)) \in Q_2. \end{cases}$$

Perform the MWPDA algorithm with κ^2. Let μ^*_2 be the outcome of the MWPDA algorithm at Round 2.

(i) If μ^*_2 is with forced pairs Q_1 and forbidden pairs Q_2, then conclude that the algorithm converges and define μ^*_2 as the outcome of the algorithm.

(ii) Else, if there exists a pair $(m, w) \in Q_1$ such that $\mu^*_2(m) \neq w$ or if there exists $m \in M$ such that $rank(P_m, \mu^*_2(m)) > \kappa^2_m$, then conclude that the algorithm STOPS.

(iii) Else, go to Round 3.

⋮

Note that for any two consecutive rounds r and $r+1$, for each $m \notin Q_1$, we have $\kappa^r_m \leq \kappa^{r+1}_m$, and for at least one $m \notin Q_1$, we have $\kappa^r_m < \kappa^{r+1}_m$. Therefore, if the algorithm does not converge or STOP at any round, then there will come a round r where some $m \notin Q_1$ will have $\kappa^r_m = 0$. In that case too, conclude that the algorithm STOPS.

5.5.2 Conditional MWPDA algorithm produces stable matching with forced and forbidden pairs

The following result says that a stable matching with given forced and forbidden pairs exists at a preference profile only if the conditional MWPDA algorithm converges at that preference profile. It further says that whenever the conditional MWPDA algorithm converges, it produces a stable matching with given forced and forbidden pairs, which is also women-optimal in the set of all stable matchings with the given forced and forbidden pairs. Thus, if at a preference profile, the conditional MWPDA algorithm STOPS at any round, then it must be that there is no stable matching with the corresponding forced and forbidden pairs at that preference profile.

Theorem 5.5.1. *A stable matching with forced pairs Q_1 and forbidden pairs Q_2 exists at a preference profile P_N if and only if the conditional MWPDA algorithm given (Q_1, Q_2) converges at P_N. Further, whenever this algorithm converges, the outcome is women-optimal in the set of all stable matchings with forced pairs Q_1 and forbidden pairs Q_2.*

The proof of this theorem is relegated to Section 5.9.

By the construction of the conditional MWPDA algorithm, we obtain the following corollary from Theorem 5.5.1. It says that whenever there is no forbidden pair, the conditional MWPDA algorithm will come to a conclusion at the first round itself: either it will converge or it will STOP. If it converges at this round, then a stable matching with given forced pairs is produced as the outcome which is also women-optimal in the set of all such stable matchings. If it STOPS, then that means there is no such a stable matching.

Corollary 5.5.1. *Let P_N be a preference profile and let Q_1 be a set of forced pairs.*

(i) If there exists a stable matching with forced pairs Q_1 at P_N, then the conditional MWPDA algorithm given $(Q_1, \emptyset)$ at P_N converges at Round 1. Furthermore, the outcome is women-optimal in the set of all stable matchings with forced pairs Q_1.

(ii) If there is no stable matching with forced pairs Q_1 at P_N, then the conditional MWPDA algorithm given $(Q_1, \emptyset)$ at P_N STOPS at Round 1.

5.6 Proof of Theorem 5.3.1

In all our proofs, for a given MWPDA algorithm at a preference profile P_N, we use the notation μ_k^s to denote the outcome obtained at Step k of the WPDA algorithm at Stage s of the given MWPDA algorithm, and the notation t^* to denote the last stage of the MWPDA algorithm. We make two observations which we will use in our proofs.

Observation 5.6.1. *Consider a stage, say s, and two steps l and k with $l \leq k$ of the WPDA algorithm at Stage s of an MWPDA algorithm at a preference profile P_N. Then, it follows from the property of the WPDA algorithm that for all $m \in M$, we have $\mu^s_k(m) R_m \mu^s_l(m)$.*

Observation 5.6.2. *Consider a stage, say s, of an MWPDA algorithm at a preference profile P_N. It follows from the property of the WPDA algorithm that μ^s is stable at the preference profile $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^s(w_1)}, \ldots, P_{w_q}^{M^s(w_q)})$.*[83]

Fix a preference profile P_N. Take an arbitrary cut-off vector κ and consider the MWPDA algorithm with κ at P_N. First, we prove a lemma that says that the match of a woman gets better over stages.

Lemma 5.6.1. *For all $r \leq s \leq t^*$ and all $w \in W$, $\mu^s(w) R_w \mu^r(w)$.*

Proof of Lemma 5.6.1. By the definition of the MWPDA algorithm, we have $\mu^s(w) R_w^{M^s(w)} \emptyset$ for all $w \in W$. This, together with the construction of $M^s(w)$, implies that $\mu^s(w) R_w \emptyset$ for all $w \in W$. So, if $\mu^r(w) = \emptyset$ for some $w \in W$, then there is nothing to show for that w. Take $w \in W$ such that $\mu^r(w) = m \in M$ and take $r < t^*$. It is enough to show that $\mu^{r+1}(w) R_w \mu^r(w)$. Assume for contradiction that $m P_w \mu^{r+1}(w)$.

Because $\mu^r(m) = w$, by the definition of the MWPDA algorithm, we have $W^r(m) = W^{r+1}(m)$ and $w \in W^r(m)$. Combining all these, we have $w \in W^{r+1}(m)$, which implies $m \in M^{r+1}(w)$. Since $m P_w \mu^{r+1}(w)$ and $m \in M^{r+1}(w)$, we have $m P_w^{M^{r+1}(w)} \mu^{r+1}(w)$. By the definition of the MWPDA algorithm, there must be some step l of the WPDA algorithm at Stage $r+1$ where m rejects w to be tentatively matched with some $w' \in W^{r+1}(m)$ whom he prefers to w. This means

$$w' P_m w, \text{ and} \tag{5.1a}$$

$$m P_{w'}^{M^{r+1}(w')} \emptyset. \tag{5.1b}$$

Moreover, since $w' \in W^{r+1}(m)$ and $W^r(m) = W^{r+1}(m)$, we have $w' \in W^r(m)$.

Assume that Step l of the WPDA algorithm at Stage $r+1$ has the property that there is no $\hat{w} \in W$ with $\mu^r(\hat{w}) \neq \emptyset$ and $\mu^r(\hat{w}) P_{\hat{w}} \mu^{r+1}(\hat{w})$ such that man $\mu^r(\hat{w})$ rejects woman $\hat{w}$ at some step $l' < l$ of the WPDA algorithm at Stage $r+1$. This is without loss of generality because, if there is such woman $\hat{w}$, then we can take $w = \hat{w}$.

Suppose $m P_{w'} \mu^r(w')$. Because $w' \in W^r(m)$, we have $m \in M^r(w')$. Since $m P_{w'} \mu^r(w')$ and $m \in M^r(w')$, it follows from the construction of $P_{w'}^{M^r(w')}$ that $m P_{w'}^{M^r(w')} \mu^r(w')$. This, together with (5.1a) and the fact $\mu^r(m) = w$, implies that (m, w') blocks μ^r at $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^r(w_1)}, \ldots, P_{w_q}^{M^r(w_q)})$, which is a contradiction to Observation 5.6.2. So, it must be that $\mu^r(w') R_{w'} m$. Because $\mu^r(w) = m$, $w \neq w'$, and $\mu^r(w') R_{w'} m$, we have $\mu^r(w') P_{w'} m$. Moreover, it follows from (5.1b)

[83] See Section 5.3.2 for the definition of the notation $P_w^{M^s(w)}$.

and the construction of $P_{w'}^{M^{r+1}(w')}$ that $mP_{w'}\emptyset$. Combining the facts that $\mu^r(w')P_{w'}m$ and $mP_{w'}\emptyset$, we have

$$\mu^r(w')P_{w'}mP_{w'}\emptyset. \tag{5.2}$$

Now, we complete the proof of the lemma. Because $w' \in W^{r+1}(m)$, we have $m \in M^{r+1}(w')$. Furthermore, (5.2) implies $\mu^r(w') \in M$. This, together with the definition of the MWPDA algorithm, yields $\mu^r(w') \in M^{r+1}(w')$. Since $m, \mu^r(w') \in M^{r+1}(w')$, it follows from (5.2) that $\mu^r(w')P_{w'}^{M^{r+1}(w')}mP_{w'}^{M^{r+1}(w')}\emptyset$. This, together with the fact that woman w' is tentatively matched with man m at Step l of the WPDA algorithm at Stage $r+1$, implies that $\mu^r(w')$ rejects w' at some step $l' < l$ of the WPDA algorithm at Stage $r+1$. However, this contradicts our assumption on Step l of the WPDA algorithm at Stage $r+1$, which completes the proof of Lemma 5.6.1. ■

Completion of the proof of Theorem 5.3.1. In view of Remark 5.2.1, we show that the outcome of the MWPDA algorithm is pairwise stable. Note that by the definition of the MWPDA algorithm, its outcome is always individually rational. We show that no pair can block its outcome. Let μ be the outcome of the MWPDA algorithm. Assume for contradiction that a pair $(m, w) \in M \times W$ blocks μ at P_N.

Since μ is individually rational at P_N and (m, w) is a blocking pair of μ at P_N, we have $wP_m\mu(m)R_m\emptyset$ and $mP_w\mu(w)R_w\emptyset$. Because $wP_m\mu(m)$, there must be some stage, say r^*, at which m proposes w for the first time. If $\mu^{r^*}(w)R_wm$, then by Lemma 5.6.1, we have $\mu(w)R_wm$, which contradicts the fact $mP_w\mu(w)R_w\emptyset$. So, assume $mP_w\mu^{r^*}(w)$. Since $w \in W^{r^*}(m)$ and $mP_w\mu^{r^*}(w)$, w proposes m and gets rejected at some step, say l, of the WPDA algorithm at Stage r^*. Since $wP_m\emptyset$, by Observation 5.6.1, this means

$$\mu^{r^*}(m)P_mwP_m\emptyset. \tag{5.3}$$

If $r^* = t^*$, then (5.3) implies $\mu(m)P_mw$, which contradicts the fact $wP_m\mu(m)R_m\emptyset$. So, assume $r^* < t^*$. By (5.3), we have $\mu^{r^*}(m) \neq \emptyset$. Since $r^* < t^*$ and $\mu^{r^*}(m) \neq \emptyset$, m proposes the women in $W^{r^*}(m)$ at the beginning of Stage r^*+1. Then, using a similar argument as for the derivation of (5.3), we have $\mu^{r^*+1}(m)P_mwP_m\emptyset$. Continuing in this manner, it follows that $\mu(m)P_mwP_m\emptyset$, which contradicts the fact $wP_m\mu(m)R_m\emptyset$. This completes the proof of Theorem 5.3.1. ■

5.7 Proofs of Theorem 5.3.2 and Theorem 5.3.3

In this section, we prove Theorem 5.3.2 and Theorem 5.3.3. We prove Theorem 5.3.3 first since we use that in the proof of Theorem 5.3.2.

5.7.1 Proof of Theorem 5.3.3

We prove Theorem 5.3.3 using the following lemmas. Our first lemma is taken from McVitie & Wilson (1970). It says that the set of unmatched men or women stays the same in all stable matchings.

Lemma 5.7.1. *(McVitie & Wilson, 1970) Let P_N be a preference profile and let $\mu, \mu' \in \mathcal{C}(P_N)$. Then, for all $a \in N$, $\mu(a) = \emptyset$ implies $\mu'(a) = \emptyset$.*

Our next lemma provides a sufficient condition on κ such that a given stable matching at a preference profile P_N remains stable at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$.

Lemma 5.7.2. *Let P_N be a preference profile and let $\mu \in \mathcal{C}(P_N)$. Then, μ is stable at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$ if $\kappa_m \geq \min \left\{ rank(P_m, \mu(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$ for all $m \in M$.*

Proof of Lemma 5.7.2. Suppose $\kappa_m \geq \min \left\{ rank(P_m, \mu(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$ for all $m \in M$. In view of Remark 5.2.1, we show that μ is pairwise stable at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$. First note that since $\kappa_m \geq \min \left\{ rank(P_m, \mu(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$ for all $m \in M$, we have $\mu(w) \in M^1(w) \cup \{\emptyset\}$ for all $w \in W$. Moreover, since $\mu(w) \in M^1(w) \cup \{\emptyset\}$ for all $w \in W$, we have for all $w \in W$ and all $m \in M$, $m R_w^{M^1(w)} \mu(w)$ implies $m R_w \mu(w)$. Further note that the preferences of the men are unchanged from P_N to $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$. Therefore, if (m, w) blocks μ at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$, then they also block μ at P_N contradicting the fact that μ is stable at P_N. Hence, μ cannot have a blocking pair at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$. Using a similar logic, it follows that μ is individually rational at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$. ■

Completion of the proof of Theorem 5.3.3. (If part) Take a cut-off vector κ such that $\kappa_m \geq \min \left\{ rank(P_m, \mu_M(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$ for all $m \in M$. We show the MWPDA algorithm with κ at P_N converges at Stage 1. By the definition of the algorithm, it converges at Stage 1 if $W^1(m) = \mathcal{A}(P_m)$ for all $m \in M$ with $\mu^1(m) = \emptyset$. Take $m \in M$. If $\mu_M(m) = \emptyset$, then by the definition of κ, m proposes all acceptable women at the beginning of Stage 1, and hence $W^1(m) = \mathcal{A}(P_m)$. Suppose $\mu_M(m) \neq \emptyset$. It is enough to show that $\mu^1(m) \neq \emptyset$. Because $\kappa_m \geq \min \left\{ rank(P_m, \mu_M(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$ for all $m \in M$, by Lemma 5.7.2, μ_M is stable at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$. Furthermore, by Observation 5.6.2, μ^1 is stable at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$. Since μ^1 and μ_M both are stable at $(P_{m_1}, \dots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \dots, P_{w_q}^{M^1(w_q)})$, by Lemma 5.7.1, we have $\mu^1(m) \neq \emptyset$.

(Only-if part) Take a cut-off vector κ such that $\kappa_m < \min \left\{ rank(P_m, \mu_M(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$ for some $m \in M$. Assume for contradiction that the MWPDA algorithm with κ at P_N converges at Stage 1. Since $\kappa_m < \min \left\{ rank(P_m, \mu_M(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$, this means $\mu^1(m) \neq \emptyset$ and $rank(P_m, \mu^1(m)) \leq \kappa_m$. Combining the facts $rank(P_m, \mu^1(m)) \leq \kappa_m$ and $\kappa_m < \min \left\{ rank(P_m, \mu_M(m)), \max \left\{ |\mathcal{A}(P_m)|, 1 \right\} \right\}$, we have $rank(P_m, \mu^1(m)) < rank(P_m, \mu_M(m))$.

This, along with Remark 5.3.2, implies μ^1 is not stable at P_N, which contradicts Theorem 5.3.1. This completes the proof of Theorem 5.3.3. ■

5.7.2 Proof of Theorem 5.3.2

Let μ_M be the men-optimal stable matching at P_N. Because $\mu \in \mathcal{C}(P_N)$, by Remark 5.3.2, we have $rank(P_m, \mu(m)) \geq rank(P_m, \mu_M(m))$ for all $m \in M$. This, together with the fact that $\kappa_m = rank(P_m, \mu(m))$ for all $m \in M$, means $\kappa_m \geq \min\left\{rank(P_m, \mu_M(m)), \max\left\{|\mathcal{A}(P_m)|, 1\right\}\right\}$ for all $m \in M$. Therefore, by Theorem 5.3.3, the MWPDA algorithm with κ converges at Stage 1.

Now, we show $\mu^1 = \mu$. Since $\kappa_m = rank(P_m, \mu(m))$ for all $m \in M$, we have $\kappa_m \geq \min\left\{rank(P_m, \mu(m)), \max\left\{|\mathcal{A}(P_m)|, 1\right\}\right\}$ for all $m \in M$. This, together with Lemma 5.7.2, implies that μ is stable at $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$. Moreover, by the definition of the MWPDA algorithm, μ^1 is women-optimal stable matching at $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$. Since $\mu \in \mathcal{C}(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$ and μ^1 is women-optimal stable matching at $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$, by Remark 5.3.2, it follows that

$$\mu(m) R_m \mu^1(m) \text{ for all } m \in M. \tag{5.4}$$

Since $\mu, \mu^1 \in \mathcal{C}(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$, by Lemma 5.7.1, we have

$$\mu^1(m) = \mu(m) \text{ for all } m \in M \text{ with } \mu^1(m) = \emptyset. \tag{5.5}$$

By the definition of the MWPDA algorithm, $rank(P_m, \mu^1(m)) \leq \kappa_m$ for all $m \in M$ with $\mu^1(m) \neq \emptyset$. This, together with definition of κ and (5.4), implies that

$$\mu^1(m) = \mu(m) \text{ for all } m \in M \text{ with } \mu^1(m) \neq \emptyset. \tag{5.6}$$

(5.5) and (5.6) together imply $\mu^1 = \mu$.

It remains to show that the MWPDA algorithm with κ converges at the first step of the WPDA algorithm at Stage 1. Suppose not. Then, there exists a pair (m, w) such that at the first step of the WPDA algorithm at Stage 1, w proposes m and gets rejected. By the definition of the MWPDA algorithm, this means $w \in W^1(m)$ and $m P_w^{M^1(w)} \mu^1(w)$. Moreover, since $\mu^1 = \mu$ and $m P_w^{M^1(w)} \mu^1(w)$, we have $\mu(m) \neq w$. The facts $\kappa_m = rank(P_m, \mu(m))$, $w \in W^1(m)$, and $w \neq \mu(m)$ together imply $w P_m \mu(m)$. Because $\mu^1 = \mu$, this, together with the fact $m P_w^{M^1(w)} \mu^1(w)$, implies (m, w) blocks μ^1 at $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$, a contradiction to Observation 5.6.2. This completes the proof of Theorem 5.3.2. ■

5.8 Proof of Theorem 5.4.1

We prove a sequence of lemmas that we use in the proof of Theorem 5.4.1.

Lemma 5.8.1. *Let P_N be a preference profile and let κ be such that $\kappa_m \geq rank(P_m, \mu_M(m))$ for all $m \in M$. Suppose μ is the outcome of the MWPDA algorithm with κ at P_N. Then, $rank(P_m, \mu(m)) \leq \kappa_m$ for all $m \in M$.*

Proof of Lemma 5.8.1. By Theorem 5.3.1, $\mu \in \mathcal{C}(P_N)$. Since $\mu, \mu_M \in \mathcal{C}(P_N)$, by Lemma 5.7.1, we have $\mu(m) = \mu_M(m)$ for all $m \in M$ with $\mu(m) = \emptyset$. This, together with the definition of κ, implies

$$rank(P_m, \mu(m)) \leq \kappa_m \text{ for all } m \in M \text{ with } \mu(m) = \emptyset. \tag{5.7}$$

By the definition of κ, we have $\kappa_m \geq \min\left\{rank(P_m, \mu_M(m)), \max\left\{|\mathcal{A}(P_m)|, 1\right\}\right\}$ for all $m \in M$. Therefore, by Theorem 5.3.3, the MWPDA algorithm with κ at P_N converges at Stage 1 producing μ. This, together with the definition of the MWPDA algorithm, implies

$$rank(P_m, \mu(m)) \leq \kappa_m \text{ for all } m \in M \text{ with } \mu(m) \neq \emptyset. \tag{5.8}$$

The proof of Lemma 5.8.1 follows from (5.7) and (5.8). ■

The implication of our next lemma is as follows. Let μ be the outcome of the MWPDA algorithm with cut-off vector κ where κ is such that every man gets to propose the woman (together with other women) who he would be matched with in the men-optimal stable matching (if a man is unmatched in the men-optimal stable matching, then he proposes all acceptable women). Let μ' be another stable matching where the rank of the match of every man m (the match might be some woman or $\emptyset$) according to P_m is less than or equal to κ_m. Then, for every woman, the match in μ must be at least as good as that in μ'.

Lemma 5.8.2. *Let P_N be a preference profile and let κ be such that $\kappa_m \geq rank(P_m, \mu_M(m))$ for all $m \in M$. Let μ be the outcome of the MWPDA algorithm with κ at P_N. Suppose $\mu' \in \mathcal{C}(P_N)$ is such that $rank(P_m, \mu'(m)) \leq \kappa_m$ for all $m \in M$. Then, $\mu(w) R_w \mu'(w)$ for all $w \in W$.*

Proof of Lemma 5.8.2. Suppose μ and μ' are as defined in Lemma 5.8.2. Since $\kappa_m \geq rank(P_m, \mu_M(m))$ for all $m \in M$, we have $\kappa_m \geq \min\left\{rank(P_m, \mu_M(m)), \max\left\{|\mathcal{A}(P_m)|, 1\right\}\right\}$ for all $m \in M$. This, along with Theorem 5.3.3, implies that the MWPDA algorithm with κ at P_N converges at Stage 1 producing μ. By Observation 5.6.2, this means μ is stable at

$(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$. Also, since $rank(P_m, \mu'(m)) \leq \kappa_m$ for all $m \in M$, we have $\kappa_m \geq \min \Big\{ rank(P_m, \mu'(m)), \max \{ |\mathcal{A}(P_m)|, 1 \} \Big\}$ for all $m \in M$. This, along with Lemma 5.7.2, implies that μ' is stable at $(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$. Because $\mu, \mu' \in \mathcal{C}(P_{m_1}, \ldots, P_{m_p}, P_{w_1}^{M^1(w_1)}, \ldots, P_{w_q}^{M^1(w_q)})$ and μ is the outcome of the WPDA algorithm at Stage 1 of the MWPDA algorithm, by Remark 5.3.2, we have $\mu(w) R_w^{M^1(w)} \mu'(w)$ for all $w \in W$. By the definition of the MWPDA algorithm, $\mu(w) \in M^1(w) \cup \{\emptyset\}$. As $rank(P_m, \mu'(m)) \leq \kappa_m$ for all $m \in M$, we have $\mu'(w) \in M^1(w) \cup \{\emptyset\}$ for all $w \in W$. Since for all $w \in W$, we have $\mu(w), \mu'(w) \in M^1(w) \cup \{\emptyset\}$ and $\mu(w) R_w^{M^1(w)} \mu'(w)$, by the construction of $P_w^{M^1(w)}$, we have $\mu(w) R_w \mu'(w)$ for all $w \in W$. This completes the proof of Lemma 5.8.2. ■

Completion of the proof of Theorem 5.4.1. By Theorem 5.3.1, it is straightforward that the sequential MWPDA algorithm is stable. We proceed to show that the sequential MWPDA algorithm produces a minimum regret stable matching at every preference profile. Take a preference profile P_N. Let κ be the cut-off vector that is used at the terminal round of the sequential MWPDA algorithm at P_N and μ be the outcome of the sequential MWPDA algorithm at P_N. It follows from the definition of the sequential MWPDA algorithm that $\kappa_m \geq rank(P_m, \mu_M(m))$ for all $m \in M$. Therefore, by Lemma 5.8.1 along with the definition of the sequential MWPDA algorithm, we have

$$rank(P_m, \mu(m)) \leq \kappa_m \text{ for all } m \in M. \tag{5.9}$$

Claim 5.8.1. *$\kappa_m \leq \alpha(P_N)$ for all $m \in M$.*

Proof of Claim 5.8.1. Assume for contradiction that $\kappa_m > \alpha(P_N)$ for some (and hence, all) $m \in M$. Consider the round of the sequential MWPDA algorithm where the MWPDA algorithm is performed with $\hat{\kappa}$ where $\hat{\kappa}_m = \alpha(P_N)$ for all $m \in M$. Let $\hat{\mu}$ be the outcome of that round. By the definition of $\alpha(P_N)$, there must exist $\mu' \in \mathcal{C}(P_N)$ such that $\alpha(\mu', P_N) = \alpha(P_N)$. Because $\alpha(\mu', P_N) = \alpha(P_N)$, we have $rank(P_m, \mu'(m)) \leq \alpha(P_N)$ for all $m \in M$. By Lemma 5.8.2, this means $\hat{\mu}(w) R_w \mu'(w)$ for all $w \in W$. Therefore, $\max\limits_{w \in W} rank(P_w, \hat{\mu}(w)) \leq \max\limits_{w \in W} rank(P_w, \mu'(w)) \leq \alpha(P_N)$. By the definition of the sequential MWPDA algorithm, this means that the algorithm cannot go for another round, which contradicts the fact that $\kappa_m > \alpha(P_N)$ for all $m \in M$. This completes the proof of Claim 5.8.1. □

Since κ is the cut-off vector that is used at the terminal round of the sequential MWPDA algorithm at P_N and μ is the outcome of the sequential MWPDA algorithm at P_N, one of the following two statements must hold.

(1) $rank(P_m, \emptyset) \leq \kappa_m$ for all $m \in M$.

(2) $rank(P_w, \mu(w)) \leq \kappa_m$ for all $w \in W$ and for some (and hence, all) $m \in M$.

We distinguish the following two cases.

Case 1: Suppose $rank(P_m, \emptyset) \leq \kappa_m$ for all $m \in M$.

Since $rank(P_m, \emptyset) \leq \kappa_m$ for all $m \in M$ and μ is the outcome of the sequential MWPDA, it is easy to verify that μ is the women-optimal stable matching at P_N. By the definition of $\alpha(P_N)$, there must exist $\mu' \in \mathcal{C}(P_N)$ such that $\alpha(\mu', P_N) = \alpha(P_N)$. Since μ is the women-optimal stable matching, we have $rank(P_w, \mu(w)) \leq rank(P_w, \mu'(w)) \leq \alpha(P_N)$ for all $w \in W$. Moreover, by Claim 5.8.1 along with (5.9), we have $rank(P_m, \mu(m)) \leq \alpha(P_N)$ for all $m \in M$. Combining the facts that $rank(P_m, \mu(m)) \leq \alpha(P_N)$ for all $m \in M$ and $rank(P_w, \mu(w)) \leq \alpha(P_N)$ for all $w \in W$, we have $\alpha(\mu, P_N) \leq \alpha(P_N)$. By the definition of $\alpha(P_N)$, this means $\alpha(\mu, P_N) = \alpha(P_N)$. So, μ is a minimum regret stable matching at P_N. Because μ is the women-optimal stable matching at P_N, this implies that μ is women-optimal in the set of all minimum regret stable matchings at P_N.

Case 2: Suppose $rank(P_w, \mu(w)) \leq \kappa_m$ for all $w \in W$ and for some (and hence, all) $m \in M$.

Since $rank(P_w, \mu(w)) \leq \kappa_m$ for all $w \in W$ and for some $m \in M$, it follows from (5.9) and the definition of the sequential MWPDA algorithm that $\alpha(\mu, P_N) \leq \kappa_m$ for all $m \in M$. This, together with Claim 5.8.1, implies that $\alpha(\mu, P_N) \leq \kappa_m \leq \alpha(P_N)$ for all $m \in M$. By the definition of $\alpha(P_N)$, this means

$$\alpha(\mu, P_N) = \kappa_m = \alpha(P_N) \text{ for all } m \in M. \tag{5.10}$$

By (5.10), we have $\alpha(\mu, P_N) = \alpha(P_N)$. So, μ is a minimum regret stable matching at P_N.

Let μ' be a minimum regret stable matching at P_N. Clearly, $rank(P_m, \mu'(m)) \leq \alpha(P_N)$ for all $m \in M$. This, together with (5.10), implies that $rank(P_m, \mu'(m)) \leq \kappa_m$ for all $m \in M$. Furthermore, it follows from the definition of the sequential MWPDA algorithm that μ is the outcome of the MWPDA algorithm with κ at P_N. Since $\kappa_m \geq rank(P_m, \mu_M(m))$ for all $m \in M$, μ is the outcome of the MWPDA algorithm with κ, and μ' is a stable matching with $rank(P_m, \mu'(m)) \leq \kappa_m$ for all $m \in M$, by Lemma 5.8.2, we have $\mu(w) R_w \mu'(w)$ for all $w \in W$. Since μ is a minimum regret stable matching at P_N, this implies that μ is women-optimal in the set of all minimum regret stable matchings at P_N.

Since Case 1 and Case 2 are exhaustive, it follows that the outcome of the sequential MWPDA algorithm is women-optimal in the set of all minimum regret stable matchings. This completes the proof of Theorem 5.4.1. ■

5.9 Proof of Theorem 5.5.1

The following lemma follows from Lemma 1 in Gale & Sotomayor (1985), which establishes a relationship between two stable matchings at a preference profile.

Lemma 5.9.1. *Let P_N be a preference profile and let $\mu, \mu' \in \mathcal{C}(P_N)$. Then, $\mu(m) R_m \mu'(m)$ for all $m \in M$ if and only if $\mu'(w) R_w \mu(w)$ for all $w \in W$.*

Let us first recall some of the notations used in the context of the conditional MWPDA algorithm. For a preference profile P_N, a set of forced pairs Q_1, and a set of forbidden pairs Q_2, κ^r is the cut-off vector associated with the MWPDA algorithm at Round r of the conditional MWPDA algorithm given (Q_1, Q_2) and μ_r^* is the outcome of the MWPDA algorithm at Round r.

Completion of the proof of Theorem 5.5.1. It is obvious that if the conditional MWPDA algorithm given (Q_1, Q_2) converges at P_N, then there exists a stable matching with forced pairs Q_1 and forbidden pairs Q_2. We proceed to prove the rest of the theorem. Suppose there exists a stable matching with forced pairs Q_1 and forbidden pairs Q_2 at P_N. Let $\bar{\mathcal{C}}(P_N)$ be the set of all stable matchings at P_N with forced pairs Q_1 and forbidden pairs Q_2. Clearly, $\bar{\mathcal{C}}(P_N) \neq \emptyset$. Define the mapping $\mu^* : N \to N \cup \{\emptyset\}$ such that

(i) for all $m \in M$, $\mu^*(m) = x$ if and only if there exists a $\mu \in \bar{\mathcal{C}}(P_N)$ such that $\mu(m) = x$ and $\mu'(m) R_m x$ for all $\mu' \in \bar{\mathcal{C}}(P_N)$, and

(ii) for all $w \in W$, $\mu^*(w) = y$ if and only if there exists a $\mu \in \bar{\mathcal{C}}(P_N)$ such that $\mu(w) = y$ and $y R_w \mu'(w)$ for all $\mu' \in \bar{\mathcal{C}}(P_N)$.

It follows from the construction of μ^* that it is women-optimal in $\bar{\mathcal{C}}(P_N)$ (see Knuth (1976) for details). We show that the conditional MWPDA algorithm given (Q_1, Q_2) converges at P_N producing μ^* as the outcome.

If $\mu_1^* = \mu^*$, then we are done. Suppose $\mu_1^* \neq \mu^*$.

Claim 5.9.1. *For all $m \in M$, we have*

(i) $rank(P_m, \mu_1^*(m)) \leq \kappa_m^1$, *and*

(ii) $\mu^*(m) R_m \mu_1^*(m)$.

Proof of Claim 5.9.1. By the definition of κ^1, we have $\kappa_m^1 \geq rank(P_m, \mu^*(m))$ for all $m \in M$. Since $\mu^* \in \mathcal{C}(P_N)$, by Remark 5.3.2, we have $rank(P_m, \mu^*(m)) \geq rank(P_m, \mu_M(m))$ for all $m \in M$. Combining the facts that $\kappa_m^1 \geq rank(P_m, \mu^*(m))$ for all $m \in M$ and $rank(P_m, \mu^*(m)) \geq rank(P_m, \mu_M(m))$ for all $m \in M$, we have $\kappa_m^1 \geq rank(P_m, \mu_M(m))$ for all $m \in M$. Therefore, by Lemma 5.8.1, $rank(P_m, \mu_1^*(m)) \leq \kappa_m^1$ for all $m \in M$. This proves (i) in Claim 5.9.1.

By Lemma 5.8.2, $\kappa_m^1 \geq rank(P_m, \mu^*(m))$ for all $m \in M$ implies $\mu_1^*(w) R_w \mu^*(w)$ for all $w \in W$. By Lemma 5.9.1, this implies $\mu^*(m) R_m \mu_1^*(m)$ for all $m \in M$. This proves (ii) in Claim 5.9.1. □

Claim 5.9.2. *$\mu_1^*(m) = \mu^*(m) = w$ for all $(m, w) \in Q_1$.*

Proof of Claim 5.9.2. Since $\kappa_m^1 = rank(P_m, w)$ for all $(m, w) \in Q_1$, $\mu^*(m) = w$ for all $(m, w) \in Q_1$, by Claim 5.9.1, we have $\mu_1^*(m) = w$ for all $(m, w) \in Q_1$, which completes the proof of Claim 5.9.2. ☐

By Claim 5.9.2, it follows that the conditional MWPDA algorithm given (Q_1, Q_2) will not stop at Round 1, and because it does not converge either at Round 1, it will go to Round 2.

Claim 5.9.3. *$\kappa_m^2 \geq rank(P_m, \mu^*(m))$ for all $m \in M$.*

Proof of Claim 5.9.3. By the definition of κ^2, we have $\kappa_m^2 = rank(P_m, \mu^*(m))$ for all $m \in Q_1$. Take $m \notin Q_1$. If $(m, \mu_1^*(m)) \notin Q_2$, then by the definition of κ^2 and (ii) in Claim 5.9.1, we have $\kappa_m^2 \geq rank(P_m, \mu^*(m))$. On the other hand, if $(m, \mu_1^*(m)) \in Q_2$, which in particular means $\mu^*(m) \neq \mu_1^*(m)$, then by (ii) in Claim 5.9.1, it must be that $\mu^*(m) P_m \mu_1^*(m)$. Therefore, by the definition of κ^2 and (ii) in Claim 5.9.1, we have $\kappa_m^2 \geq rank(P_m, \mu^*(m))$. This completes the proof of Claim 5.9.3. ☐

Using similar logic as for Claims 5.9.1 and 5.9.2, it follows that

$$rank(P_m, \mu_2^*(m)) \leq \kappa_m^2 \text{ for all } m \in M, \tag{5.11a}$$

$$\mu^*(m) R_m \mu_2^*(m) \text{ for all } m \in M, \text{ and} \tag{5.11b}$$

$$\mu_2^*(m) = \mu^*(m) = w \text{ for all } (m, w) \in Q_1. \tag{5.11c}$$

Claim 5.9.4. *$\mu_2^*(m) R_m \mu_1^*(m)$ for all $m \in M$ and there exists $m' \notin Q_1$ such that $\mu_2^*(m') P_{m'} \mu_1^*(m')$.*

Proof of Claim 5.9.4. By the definition of κ^2, (5.11a) implies $\mu_2^*(m) R_m \mu_1^*(m)$ for all $m \notin M$. Moreover, as $\mu_1^* \neq \mu^*$, there must exist $m' \notin Q_1$ such that $(m', \mu_1^*(m')) \in Q_2$. This, together with the definition of κ^2 and (5.11a), yields $\mu_2^*(m') P_{m'} \mu_1^*(m')$. ☐

By Claim 5.9.4, (5.11a), and (5.11c), it follows that the conditional MWPDA algorithm given (Q_1, Q_2) either converges at Round 2 or goes to Round 3. If it goes to Round 3, then using similar logic as for Claim 5.9.2, we have $\mu_3^*(m) = \mu^*(m) = w$ for all $(m, w) \in Q_1$, and that for Claim 5.9.4, we have $\mu_3^*(m) R_m \mu_2^*(m)$ for all $m \in M$ and there exists $\bar{m} \notin Q_1$ such that $\mu_3^*(\bar{m}) P_{\bar{m}} \mu_2^*(\bar{m})$.

We argue that the conditional MWPDA algorithm given (Q_1, Q_2) must converge at some round.[84] Suppose not. Then, we will get a sequence of stable matchings $\mu_1^*, \mu_2^*, \ldots$ such that $\mu^*(m) R_m \ldots R_m \mu_2^*(m) R_m \mu_1^*(m)$ for all $m \in M$. Because

[84] Recall that the conditional MWPDA algorithm always terminates, that is, either converges or STOPS at every preference profile (see Section 5.5.1 for details).

$\mu_1^*, \mu_2^*, \ldots$ are all distinct and the number of stable matchings is finite, it follows that there must be a round where μ^* will be produced, and hence the conditional MWPDA algorithm will converge.

Now, we show that the outcome of the conditional MWPDA algorithm given (Q_1, Q_2) is always μ^*. Let $\tilde{r}$ be the terminal round of the conditional MWPDA algorithm given (Q_1, Q_2). Using similar logic as for Claim 5.9.1, we have $\mu^*(m) R_m \mu_{\tilde{r}}^*(m)$ for all $m \in M$. Since $\mu^*, \mu_{\tilde{r}}^* \in \mathcal{C}(P_N)$, by Lemma 5.9.1, $\mu_{\tilde{r}}^*(w) R_w \mu^*(w)$ for all $w \in W$. Moreover, since the conditional MWPDA algorithm converges, it must be that $\mu_{\tilde{r}}^* \in \bar{\mathcal{C}}(P_N)$. Since $\mu_{\tilde{r}}^* \in \bar{\mathcal{C}}(P_N)$ and $\mu_{\tilde{r}}^*(w) R_w \mu^*(w)$ for all $w \in W$, by the definition of μ^*, we have $\mu^* = \mu_{\tilde{r}}^*$. This completes the proof of Theorem 5.5.1. ■

References

Abdulkadiroglu, A. & Sönmez, T. (2013). Matching markets: Theory and practice. *Advances in Economics and Econometrics*, 1, 3–47.

Arribillaga, R. P., Massó, J., & Neme, A. (2020). On obvious strategy-proofness and single-peakedness. *Journal of Economic Theory*, 186, 104992.

Ashlagi, I. & Gonczarowski, Y. A. (2018). Stable matching mechanisms are not obviously strategy-proof. *Journal of Economic Theory*, 177, 405–425.

Bade, S. (2019). Matching with single-peaked preferences. *Journal of Economic Theory*, 180, 81–99.

Bade, S. & Gonczarowski, Y. A. (2017). Gibbard-satterthwaite success stories and obvious strategyproofness. In *Proceedings of the 2017 ACM Conference on Economics and Computation* (pp. 565–565).: ACM.

Chen, Y. & Sönmez, T. (2006). School choice: an experimental study. *Journal of Economic theory*, 127(1), 202–231.

Damamme, A., Beynier, A., Chevaleyre, Y., & Maudet, N. (2015). The power of swap deals in distributed resource allocation. https://hal.archives-ouvertes.fr/hal-01212603.

Dias, V. M., Da Fonseca, G. D., De Figueiredo, C. M., & Szwarcfiter, J. L. (2003). The stable marriage problem with restricted pairs. *Theoretical Computer Science*, 306(1-3), 391–405.

Ehlers, L. (2002). Coalitional strategy-proof house allocation. *Journal of Economic Theory*, 105(2), 298–317.

Ehlers, L. (2018). Strategy-proofness and essentially single-valued cores revisited. *Journal of Economic Theory*, 176, 393–407.

Ehlers, L. & Klaus, B. (2006). Efficient priority rules. *Games and Economic Behavior*, 55(2), 372–384.

Ehlers, L. & Klaus, B. (2007). Consistent house allocation. *Economic Theory*, 30(3), 561–574.

Ergin, H. I. (2000). Consistency in house allocation problems. *Journal of mathematical economics*, 34(1), 77–97.

Gale, D. & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1), 9–15.

Gale, D. & Sotomayor, M. (1985). Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3), 223–232.

Gusfield, D. (1987). Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1), 111–128.

Gusfield, D. & Irving, R. W. (1989). *The stable marriage problem: structure and algorithms*. MIT press.

Hassidim, A., Marciano, D., Romm, A., & Shorrer, R. I. (2017). The mechanism is truthful, why aren't you? *American Economic Review*, 107(5), 220–24.

Hassidim, A., Romm, A., & Shorrer, R. I. (2016). " strategic" behavior in a strategy-proof environment. In *Proceedings of the 2016 ACM Conference on Economics and Computation* (pp. 763–764).

Irving, R. W. & Leather, P. (1986). The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3), 655–667.

Karakaya, M., Klaus, B., & Schlegel, J. C. (2019). Top trading cycles, consistency, and acyclic priorities for house allocation with existing tenants. *Journal of Economic Theory*, 184, 104948.

Kelso Jr, A. S. & Crawford, V. P. (1982). Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, (pp. 1483–1504).

Kesten, O. (2006). On two competing mechanisms for priority-based allocation problems. *Journal of Economic Theory*, 127(1), 155–171.

Knuth, D. E. (1976). Mariages stables et leurs relations avec d'autres problèmes combinatoires. English translation in "Stable marriage and its relation to other combinatorial problems", Volume 10 of CRM Proceedings and Lecture Notes, American Mathematical Society, 1997. *Les Presses de l'Université de Montréal*.

Li, S. (2017). Obviously strategy-proof mechanisms. *American Economic Review*, 107(11), 3257–87.

Ma, J. (1994). Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory*, 23(1), 75–83.

Mackenzie, A. (2020). A revelation principle for obviously strategy-proof implementation. *Games and Economic Behavior*, 124, 512–533.

Mandal, P. & Roy, S. (2020). Obviously strategy-proof implementation of assignment rules: A new characterization. *MPRA Paper 108591*. https://mpra.ub.uni-muenchen.de/108591/.

Mandal, P. & Roy, S. (2021a). Matchings under stability, minimum regret, and forced and forbidden pairs in marriage problem. *MPRA Paper 107213*. https://mpra.ub.uni-muenchen.de/107213/.

Mandal, P. & Roy, S. (2021b). On obviously strategy-proof implementation of fixed priority top trading cycles with outside options. *Working Paper*.

Mandal, P. & Roy, S. (2021c). Strategy-proof allocation of indivisible goods when preferences are single-peaked. *MPRA Paper 105320*. https://mpra.ub.uni-muenchen.de/105320/.

Martınez, R., Massó, J., Neme, A., & Oviedo, J. (2004). An algorithm to compute the full set of many-to-many stable matchings. *Mathematical Social Sciences*, 47(2), 187–210.

McVitie, D. & Wilson, L. B. (1970). Stable marriage assignment for unequal sets. *BIT Numerical Mathematics*, 10(3), 295–309.

McVitie, D. G. & Wilson, L. B. (1971). The stable marriage problem. *Communications of the ACM*, 14(7), 486–490.

Pápai, S. (2000). Strategyproof assignment by hierarchical exchange. *Econometrica*, 68(6), 1403–1433.

Pycia, M. & Troyan, P. (2019). A theory of simplicity in games and mechanism design. *Available at SSRN 2853563*.

Pycia, M. & Ünver, M. U. (2017). Incentive compatible allocation and exchange of discrete resources. *Theoretical Economics*, 12(1), 287–329.

Rees-Jones, A. (2018). Suboptimal behavior in strategy-proof mechanisms: Evidence from the residency match. *Games and Economic Behavior*, 108, 317–330.

Roth, A. E. (1985). The college admissions problem is not equivalent to the marriage problem. *Journal of economic Theory*, 36(2), 277–288.

Roth, A. E. & Sotomayor, M. (1989). The college admissions problem revisited. *Econometrica: Journal of the Econometric Society*, (pp. 559–570).

Roth, A. E. & Sotomayor, M. (1992). Two-sided matching. *Handbook of game theory with economic applications*, 1, 485–541.

Satterthwaite, M. A. & Sonnenschein, H. (1981). Strategy-proof allocation mechanisms at differentiable points. *The Review of Economic Studies*, 48(4), 587–597.

Shapley, L. & Scarf, H. (1974). On cores and indivisibility. *Journal of mathematical economics*, 1(1), 23–37.

Shorrer, R. I. & Sóvágó, S. (2018). Obvious mistakes in a strategically simple college admissions environment: Causes and consequences. *Available at SSRN 2993538*.

Svensson, L.-G. (1999). Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4), 557–567.

Svensson, L.-G. & Larsson, B. (2005). Strategy-proofness, core, and sequential trade. *Review of Economic Design*, 9(2), 167–190.

Thomas, C. (2020). Classification of priorities such that deferred acceptance is obviously strategyproof. *arXiv preprint arXiv:2011.12367*.

Troyan, P. (2019). Obviously strategy-proof implementation of top trading cycles. *International Economic Review*, 60(3), 1249–1261.

Velez, R. A. (2014). Consistent strategy-proof assignment by hierarchical exchange. *Economic Theory*, 56(1), 125–156.

6
List of Articles

- "Obviously Strategy-proof Implementation of Assignment Rules: A New Characterization"
  International Economic Review, https://doi.org/10.1111/iere.12538
- "On Obviously Strategy-proof Implementation of Fixed Priority Top Trading Cycles with Outside Options"
  Economics Letters, https://doi.org/10.1016/j.econlet.2021.110239
- "Strategy-proof Allocation of Indivisible Goods when Preferences are Single-peaked"
  (MPRA: 105320)
- "Matchings under Stability, Minimum Regret, and Forced and Forbidden Pairs in Marriage Problem"
  (MPRA: 107213)