POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Matematica
Scuola di Ingegneria Industriale e dell'Informazione

Easy Sequential Games With More Than Two Players

Relatore: **Prof. Nicola Gatti**
Correlatore: **Dott. Alberto Marchesi**

Tesi di Laurea di:
Andrea Luigi Maggi
matricola 883312

Anno Accademico 2019-2020

A Nonna Eugenia,
che da lassù mi sei sempre stata vicina
e che oggi saresti stata orgogliosissima

Abstract

In Game Theory field, one of the most interesting cases of study is the model of two-player zero-sum games: there are two players and in every outcome of the game one gets the opposite of the other. The two-person strictly competitive games have a similar idea: one gets more if the other one gets less. The class of two-player zero-sum games is appealing since equilibria have an easy resolution because they can be found in polynomial time in the size of game. This property does not hold for the class of three-player zero-sum games. The aim of the work is to find and to study classes of game with generic n players that have an equilibrium strategy easy to find in the dimension of the game. We study the unilaterally competitive games and the Polymatrix games.

Sommario

In Teoria dei giochi, uno dei casi più interessanti da studiare è il modello dei giochi a somma zero: due giocatori e ad ogni possibile risultato del gioco uno ottiene l'esatto opposto dell'altro. I giochi a due giocatori strettamente competitivi hanno una idea simile: un giocatore ottiene di più se l'altro ottiene di meno. La classe dei giochi a somma zero è interessante perché di facile risoluzione dato che gli equilibri possono essere trovati in tempo polinomiale rispetto alla dimensione del gioco. Quest'ultima proprietà non vale quando si tratta di giochi a somma zero con tre giocatori. L'obiettivo di questo lavoro è trovare e studiare classi di gioco a tre o più giocatori che cui si possa trovare un equilibrio in modo facile. Studiamo quindi i giochi unilateralmente competitivi e i giochi Polimatrice.

Ringraziamenti

Il mio primo grazie va al Prof. Nicola Gatti e al Dottor Alberto Marchesi per il sostegno e per gli input fondamentali per questo lavoro, soprattutto in una fase difficile come quella del lockdown.

Un altro grazie va ai professori e alle professoresse incontrati in questo carriera universitaria: al di là delle nozioni, molti di voi me li porto nel cuore, per avermi ispirato, per avermi fatto capire cosa mi piace (e cosa non) e per avermi stimolato.

Voglio ringraziare la mia famiglia, per avermi accompagnato e sostenuto in questa fase universitaria e per ogni volta che c'è stato uno sforzo nel comprendere cosa stessi facendo.

Ringrazio chi nel mio percorso universitario mi è stato vicino. A Giulia L. per avermi fatto ragionare; ad Alessia che mi ha ricordato di essere *pazzeska* sempre; a Giulia B. e Luca, che mi sono stati vicini quotidianamente in questi ultimi due anni (definirvi preziosi è riduttivo, sappiatelo); a Guendalina, vicina di banco di ARF; ad Alice con cui ho potuto parlare delle nostre passioni politiche e non con cui non ho passato ARF ("prego"); a Luca, Valeria e Ilaria perchè mi avete sempre ascoltato; a Elena ed Eleonora, per avermi fatto sentire sempre accolto nelle mie ansie e preoccupazioni. E poi a Elisa, Samuele, Lorenzo, Davide, Giuseppe, Giulia. A Mikael, "ai" Valerio, Riccardo, Luigi, Giorgia, Vittorio, Martina, Valeria e tutte le persone conosciute in Erasmus. E, infine, un grazie a tutti quei compagni di corso che tramite una parola o una chiacchiera in pausa mi hanno saputo dare una dritta, un consiglio, che mi hanno stimolato e fatto sentire meno inadeguato.

Un grazie alle associazioni dove ho militato, Associazione Bir e PoliEdro, per avermi fatto crescere umanamente e scoprire storie a cui mai sarei arrivato.

Un grazie agli amici che hanno avuto grande pazienza, sperando di non dimenticare nessuno: a Giorgio, Emanuele e Ala che ci sono stati fin dalla scelta della facoltà; a Chiara, in te ho trovato sempre una persona intelligente e con cui confidarmi; a Francesca perchè se mi sono iscritto in questa facoltà forse lo devo soprattutto ad una birra insieme; a Sole e Anna per il divertimento; a Elisa perchè dopo più di 20 anni siamo ancora qua; a Giulia perchè dopo 20 anni siamo ancora qua; a Federico, prova vivente che i milanesi esistono; a Stefano e Angelica, con cui abbiamo ridisegnato le durate degli audio di Whatsapp e provato che si può stare vicini anche a centinaia di km di distanza; a Giovanni e Alessandro per le serate leggere; a Giulio che mi ha costantemente rallegrato con leggerezza; a Riccardo e Luca per avermi ascoltato; alle "due Silvie" per avermi scritto in pieno lockdown; grazie ad Alberto e Chiara, per me siete parte della mia famiglia; un grazie a Daniele, perchè ci sentiamo poco ma ogni volta è come ritrovarsi; a Federico, Gabriele, Matteo e tutto quel mondo sul web che ci ha unito nelle nostre passioni e nei nostri interessi.

Alla Bialetti, perchè la moka è stata una fedele compagna di viaggio; alla AS Roma per avermi fatto divertire e a volte anche arrabbiare; ai Vama, perchè la loro canzone mi ha accompagnato in tutti questi anni.

Infine, un grazie enorme a mia mamma che, anche con tantissime difficoltà (io le conosco) c'è sempre stata, e a mio fratello Simone, un amico, un compagno di scherzi, di giochi, un confidente, che mi ha sempre spronato e appoggiato in tutto.

Contents

Chapter 1

Introduction

1.1 Scientific domain

The thesis belongs to two fields: Game Theory and Algorithmic Game Theory.

A game is an elegant model of interactions between agents, called players. They can execute some actions and take a revenue during and at the end of the game. They have two basilar characteristics: they are selfish and rational. The term "selfish" means that each one will play in a way that can maximize her revenue, without caring of the satisfaction of other agent. The term "rational" means that she will take the best option for herself in every decision and she knows all the consequences of every choice. Often, a game describes situations with strategic interactions in which optimal decisions have to be taken in presence of multiple players. In order to find the optimal decision, there is the field of Algorithmic Game Theory comes to help. This field combines mathematical models and algorithmic. One of the situation mostly studied is the microeconomics scenario.

In particular, recently, a number of concrete applications based on Algorithmic Game Theory tools were deployed. We mentions just a few: in the field of physical and cyber security, non-cooperative models are commonly used to prescribe the best strategies to patrollers [5, 3, 31, 24, 1]; in the field of automatic negotiations, game theoretic models are used to find the best negotiation strategies [16, 19, 2]; in the telecommunication networks, selfish resource allocation problems are commonly orchestrated by means of game theory tools [28, 26].

1.2 Aim of the work

We focus on the study of classes of not cooperative game computationally easy to solve. We say computationally easy to solve because we focus on classes of games where getting more utility for a player means that the others get less utility so what it the best for someone, it is the worst for others. It is something related with maxmin solutions. Something very similar are the known game class of Zero-sum, Constant-sum and strictly competitive with two players. Basically these three classes are equivalent because their payoffs are obtainable through an opportune affine transformation. Moreover, strictly competitive games satisfy important properties like uniqueness of the equilibrium value and interchangeability of the equilibrium strategies. Moreover, the maxmin solution and the Nash solution coincide.

These properties do not hold in n-player zero-sum game. Thus, we aim to study the class of unilaterally competitive games that are a generalization of the previous classes and they can be used also for a situation with more than two players. We study a characterization of this class with two players. Then, we study another class of games, the Poly Sequence-matrix, and we see if this class can be represented through an unique game tree and if the equilibria of this class of games preserve the same properties of equilibria of Zero-sum Polymatrix Games.

1.3 Structure of the thesis

The thesis is structured in the following way:

- In Chapter 2 we provide the preliminaries about game theory, which are the game models that we use and the main solution concepts. Moreover we present some classes of games.

- In Chapter 3 we analyse unilaterally competitive game with two players. We study a characterization of them, starting from some property that holds in two-player zero-sum games.

- In Chapter 4 we study a general class of games with a number of players larger than two. We call this class Poly sequence-matrix games. We provide two representations and we study their relations. We also show that the equilibria of a Poly sequence-matrix game preserve the same properties satisfied by the equilibria of zero-sum polymatrix games.

- In Chapter 5 we present our conclusions and some possible paths that future work may follow.

Chapter 2

Preliminaries

2.1 Introduction to Game Theory

This thesis resorts to the mathematical field of *Game Theory*. Most of the notions discussed in this chapter can be found in [30]. A game is an elegant model of interactions between agents, called players. They can execute some actions and take a revenue during and at the end of the game. They have two basilar characteristics: they are selfish and rational. The term "selfish" means that each one will play in a way that can maximize her revenue. The term "rational" means that she will take the best option for herself in every decision and she knows all the consequences of every choice.

2.2 Game representation form

Game theory provides several representations of a game. In the following, we present some of them.

2.2.1 Extensive-form representation

We start from the most general game representation: the extensive-form. This model captures the situation in which players play sequentially on a game tree (that collects all the information about the game). Initially, we focus on the case of games with perfect information and then we generalize with the ones of imperfect information. In a perfect information game, all the players know their position in the game. This form is defined as follows:

Definition 2.2.1. The extensive-form representation of a perfect-information game is a tuple $(N, A, V, T, \iota, \rho, \chi, U)$, where:

- $N = \{1, 2, ..., n\}$ is the set of players;
- $A = \{A_1, ..., A_n\}$ is the set of actions: each $A_i = \{a_{i1}, a_{i2}, ..., a_{im_i}\}$ is the set of actions of player i;
- $V = \{V_1, ..., V_n\}$ is the set of decision node: each $V_i = \{\omega_{i1}, \omega_{i2}, ..., \omega_{ik_i}\}$ is the set of decision nodes of player i;
- T is the set of terminal nodes;
- $\iota : V \to N$ is the function returning the players that acts at a given decision node;
- $\rho : V \to \wp(A)$ is the function returning the set of actions available to player $i(\omega)$ at that decision node ω;
- $\chi : V \times A \to V \cup T$ is the function assigning the next node to each pair (ω, a) such that $a \in \rho(\omega)$. Note that the node can be a terminal or a decision node and that χ is not defined for the pairs such that $a \notin \rho(\omega)$;
- $U = (U_1, ..., U_n)$ is the set of utility functions, in which $U_i : T \to \mathbb{R}$ is the utility of player i.

In Figure 2.1, there is an example of perfect-information game with two players.

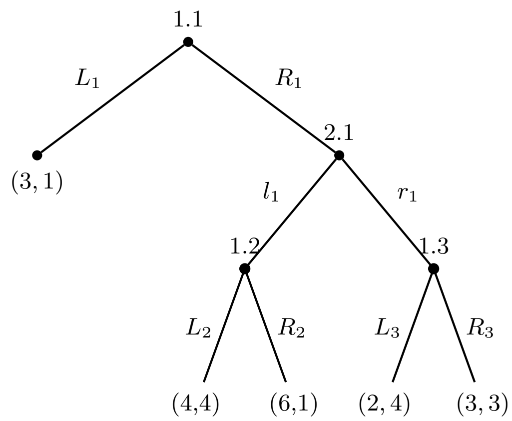

Figure 2.1: Example of perfect-information game in extensive-form representation.

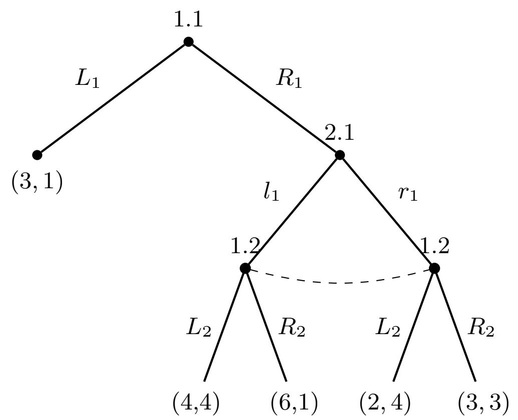

Figure 2.2: Example of imperfect-information game in extensive-form representation.

A concrete example of perfect information game is chess: the players play sequentially and all the available moves are visible by both players.

There can exist situations in which the players do not have all the information of the game (card game like Poker, for example): this is a situation in which there is not a perfect information. Let introduce the concept of *information set*, useful to capture the situation in which a player may not distinguish some nodes in which she plays:

Definition 2.2.2. An information set h of player i is a subset of V_i such that, for all ω, $\hat{\omega} \in h$, the property $\rho(\omega) = \rho(\hat{\omega})$ holds.

Now, we can define an extensive-form representation game with imperfect information:

Definition 2.2.3. The extensive-form representation of an imperfect-information game is a tuple $(N, A, V, T, \iota, \rho, \chi, U, H)$, where:

- $(N, A, V, T, \iota, \rho, \chi, U)$ is a perfect information game in extensive form;
- $H = \{H_1, ..., H_n\}$ is the set of information set: each H_i is a partition of V_i.

We provide an example of imperfect-information game in extensive-form representation in Figure 2.2: the game is very similar to the one in perfect-information of Figure 2.1 but, after the execution of R_1 by player 1, both players have to decide simultaneously which action play instead of play actions in a sequential way.

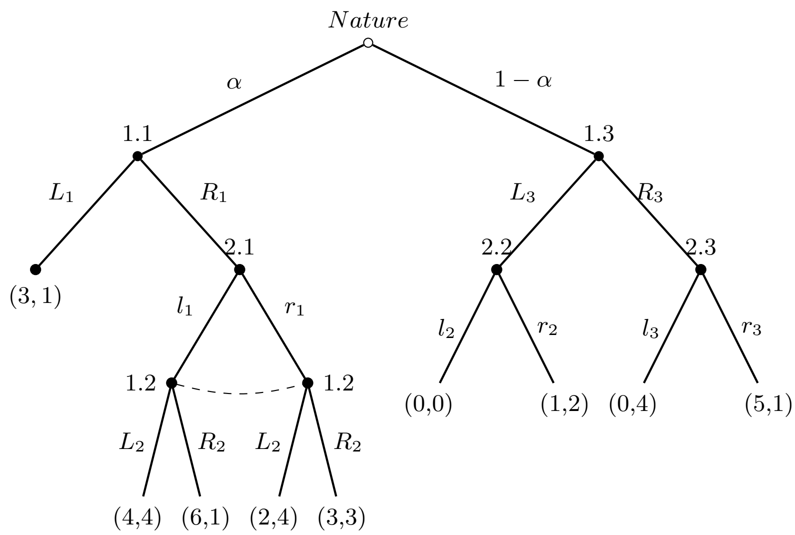

Figure 2.3: Example of game in extensive-form representation with Nature.

A game may have a player that choose her action according to a fixed probability distribution instead of following her utility (she is said *non-strategic*). This player is called Nature. A practical example of Nature is, for example, a toss of a coin.

In Figure 2.3 it is reported an example of imperfect-information game in extensive form with Nature.

Imperfect information does not mean that the the players forget the past actions and the past observations (on her own or about the others). This case is capture by *perfect-recall* property. More precisely:

Definition 2.2.4. In an imperfect-information extensive-form game, player i has perfect recall if for any pair of decision nodes ω and $\hat{\omega}$ that are in the same information set h of player i, for any path $< \omega_0, a_0, \omega_1, a_1, ..., \omega_k, a_k, \omega >$ from the root $\bar{\omega}_0$ of the game to ω and for any path $< \omega'_0, a'_0, \omega'_1, a'_1, ..., \omega'_l, a'_l, \omega' >$ from the root of the game to ω' it must be the case that:

- $k = l$;
- for all $0 \leq j \leq k$, ω_j and ω'_j are in the same information set for player i;
- for all $0 \leq j \leq k$, $a_j = a'_j$.

A game is with perfect-recall if every player has perfect recall.

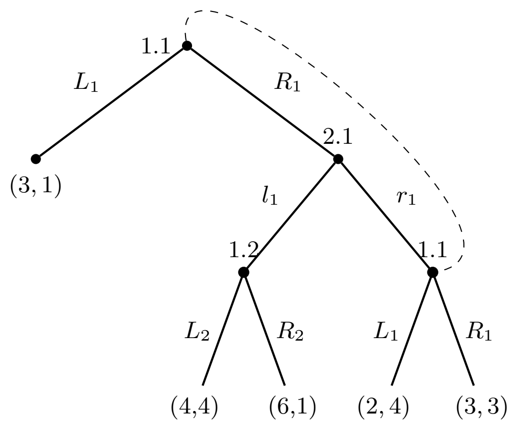

Figure 2.4: Example of game with imperfect recall.

A game that is not with perfect recall is said with *imperfect recall*.[1] All the examples seen in this chapter are with perfect-recall. In Figure 2.4, an example of game with imperfect recall is provided.

2.2.2 Normal-form representation

The extensive form is not the only available representation. When there is a 'static' situation and not a sequential game, the normal-form is a better way to represent a game (the Rock-Paper-Scissor game, for example).

Definition 2.2.5. The normal-form representation of a game is a triple (N, A, U) where:

- $N = \{1, 2, \cdots, n\}$ is the set of players;
- $A = \{A_1, \cdots, A_n\}$ is the set of actions: each $A_i = \{a_{i1}, a_{i2}, \cdots, a_{im_i}\}$ is the set of actions of player i;
- $U = (U_1, \cdots, U_n)$ is the set of utility functions, in which $U_i : A_1 \times A_2 \times \cdots \times A_n \to \mathbb{R}$ is the utility of player i.

A classical example of game in normal form is Battle of Sexes. There are two players, a couple of lovers, that have to decide what they want

[1] We point an interested reader to this recent work on imperfect-recall games [13].

to do during evening: cinema or dinner at the restaurant. Player 1 wants to go to the cinema and player 2 prefers eating at restaurant. Each one wants to do what she prefers but doing something separately is bad for everyone. The model is:

- $N = 1, 2$ are the players (the two lovers);
- $A = \{A_1, A_2\}$ is the set of actions. A_1 and A_2 contains the same action: C (cinema) and R (restaurant);
- The utility matrix U (at the first spot there is the utility of Player 1 U_1 and at second spot the utility of Player 2 U_2) is:

		Player 2	
		C	R
Player 1	C	(2,1)	(0,0)
	R	(0,0)	(1,2)

We will use the concept of strategy to capture the behaviour of the player during game.

Definition 2.2.6. A strategy σ_i is a vector that define the probability of play an action in A_i by player i so, for each $a_{ik} \in A_i$, $\sigma_i(a_{ik}) \geq 0$ and it is a simplex over A_i so $\sum_{a_{ik} \in A_i} \sigma_i(a_{ik}) = 1$.

An example of strategy in Battle of Sexes game can be that Player 1 will play C with probability 0.9:

$$\sigma_1 = \begin{cases} 0.9 & C \\ 0.1 & R \end{cases}$$

And Player 2 the same but preferring R so:

$$\sigma_2 = \begin{cases} 0.1 & C \\ 0.9 & R \end{cases}$$

These are examples of *mixed strategy*. If there is an action $a \in A_i$ such that the probability of play it is 1 ($\sigma_i(a) = 1$), and so the probability of play all the others action is zero, the strategy is said *pure*.

The vector σ that collects the strategies of all the players is said *strategy profile* and we can call the strategy set of each player X_i. The strategy profile of all players is $X = \prod_{i \in N} X_i$.

The strategy affects the utility of player. Since a player strategy is a probability distribution over her own action, we can define the expected utility as follows:

Definition 2.2.7. The expected utility $\mathbb{E}_{\mathbf{a}\sim\sigma}[U_i(\mathbf{a})]$ returns the expected value of utility of player i given strategy profile σ. The formula can be written:

$$\mathbb{E}_{\mathbf{a}\sim\sigma}[U_i(\mathbf{a})] = \sum_{a_1\in A_1}\sum_{a_2\in A_2}...\sum_{a_n\in A_n}\sigma_1(a_1)\sigma_2(a_2)...\sigma_n(a_n)U_i(a_1,...,a_n)$$

Instead of $\mathbb{E}_{\mathbf{a}\sim\sigma}[U_i(\mathbf{a})]$, we can write $U(\sigma_1, \sigma_2, ..., \sigma_n)$.

2.2.3 Sequence-form representation

The sequence-form representation is a computationally efficient representation for extensive-form games. These notions can be found in [32] and [25].

Definition 2.2.8. Given a node ω, that belongs to player i ($\iota(\omega) = i$), a sequence q is an ordered set of actions of player i in the path $< \omega_0, a_0, \omega_1, a_1, ..., \omega_k, a_k, \omega >$ from the root $\bar{\omega}_0$ of the game to ω.

We denote the set of all sequences by Q and the set of sequences of player i with Q_i. A *sequence profile* $\mathbf{q}$ is a tuple that specify one sequence for each player.

We identify some sequences with specific properties. If no action of player i is present on the path, the sequence is said *empty* and we denote it with $q_\emptyset$. If a sequence profile is such that q_i leads player i to a terminal node, the sequence is said *terminal.*

Also over the sequences, we can define a strategy profile. Contrary on the previous ones, it is not a probability distribution over all the sequences of the player (so it is not true that $\sum_{q\in Q_i} r_i(q) = 1$).

Definition 2.2.9. A sequence-form strategy $r_i : Q_i \to [0, 1]$ is a function returning the probability with which each sequence $q \in Q_i$, is played by player i, with the constraints that $r_i(q_\emptyset) = 1$ and that $r_i(q) = \sum_{a\in\rho(h)} \text{extend}(q, a)$ [2] for each $h \in \text{lead}(q)$ [3] and for each $q \in Q_i$ that is not terminal.

[2] extend is a function $Q_i \times A_{i,h} \to Q_i$ that returns the sequence obtaining by adding the action given in input to the sequence given in input

[3] lead is a function $Q_i \to \wp(H_i)$ that returns the set of information sets directly achievable from the sequence in input

The constraints that a sequence-form strategy requires are more similar to the flow constraints. Indeed, it can be interpreted as a unitary flow, with source on the root of the game ($r_i(q_\emptyset) = 1$) that moves along the nodes in order to end to the terminal nodes. The flow can be replicated multiple times (and it happens when a sequence may lead to multiple information sets).

The constraints over the strategies can be formulated in matrix notation. We define F_i a matrix with as many rows as the cardinality of the set of information sets plus 1 ($|H_i| + 1$) and as many columns as the cardinality of her own sequences ($|Q_i|$). We define f_i as a vector of $|H_i| + 1$ positions. If we see r_i as a column vector, the constraints can be formulated as:

$$F_i r_i = f_i$$

A vector $\mathbf{r}$ that collects the sequence-strategies of all the players is said *sequence-form strategy profile*.

Now, we can give the definition of a sequence-form representation of an extensive-form game:

Definition 2.2.10. Given an extensive-form representation game $(N, A, , V, T, \iota, \rho, \chi, U, H)$, the corresponding sequence-form representation is a tuple (N, Q, U', C), where:

- N is the set of players;
- $Q = \{Q_1, ..., Q_n\}$ is the set of sequences of all the players (Q_i are the sequences of player i);
- $U' = (U'_1, ..., U'_n)$ is the set of utility functions, in which $U_i : Q_1 \times Q_2 \times ... \times Q_n \rightarrow \mathbb{R}$ is the utility of player i at node ω reached by a profile of terminal sequences (the payoff is not defined when the sequence profile contains at least a non-terminal sequence);
- $C = \{(F_1, f_1), (F_2, f_2), ..., (F_n, f_n)\}$ is the set of the matrix formulation constraints of all the players.

As an example of sequence form game we will use the one in extensive-form in Figure 2.1. The players are 1 and 2 ($N = \{1, 2\}$). The sequences of Player 1 are $Q_1 = \{q_\emptyset, L_1, R_1, R_1L_2, R_1R_2, R_1L_3, R_1R_3\}$ and the sequences of Player 2 are $Q_2 = \{q_\emptyset, l_1, r_1\}$.

The utility matrix U'_1 is:

		Player 2		
		$q_\emptyset$	l_1	r_1
Player 1	$q_\emptyset$	-	-	-
	L_1	3	-	-
	R_1	-	-	-
	R_1L_2	-	4	-
	R_1R_2	-	6	-
	R_1L_3	-	-	2
	R_1R_3	-	-	3

The utility matrix U_2' is:

		Player 2		
		$q_\emptyset$	l_1	r_1
Player 1	$q_\emptyset$	-	-	-
	L_1	1	-	-
	R_1	-	-	-
	R_1L_2	-	4	-
	R_1R_2	-	1	-
	R_1L_3	-	-	4
	R_1R_3	-	-	3

As you can see, there a lot of empty spaces due to the fact that not all the sequence profiles lead to terminal nodes.

The set C contains the strategy constraints of the players. We start from Player 1, remembering that each column represents a sequence (the order is in Q_1) and each row an information set (the first row represents the root of game, then 1.1, 1.2 and 1.3). So:

$$F_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 \end{bmatrix} f_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ where } r_1 = \begin{bmatrix} r_1(q_\emptyset) \\ r_1(L_1) \\ r_1(R_1) \\ r_1(R_1L_2) \\ r_1(R_1R_2) \\ r_1(R_1L_3) \\ r_1(R_1R_3) \end{bmatrix}$$

The constraints over Player 2 are very similar, there are less rows and less columns because of the few information sets and sequences:

$$F_2 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \end{bmatrix} f_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ where } r_2 = \begin{bmatrix} r_2(q_\emptyset) \\ r_2(l_1) \\ r_2(r_1) \end{bmatrix}$$

Another interesting example is the sequence representation of an extensive form with moves of Nature player. We are writing the utility function in sequence form for the game represented in Figure 2.3 (we consider $\alpha = 0.5$). First of all, we start by mention the sequence of each player so $Q_1 = \{q_\emptyset, L_1, R_1, L_3, R_3, R_1L_2, R_1R_2\}$ and $Q_2 = \{q_\emptyset, l_1, r_1, l_2, r_2, l_3, r_3\}$.

The utility matrices are:

$$U_1' = \begin{bmatrix} - & - & - & - & - & - & - \\ 1.5 & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & 0 & 0.5 & - & - \\ - & - & - & - & - & 0 & 2.5 \\ - & 2 & 1 & - & - & - & - \\ - & 3 & 1.5 & - & - & - & - \end{bmatrix} U_2' = \begin{bmatrix} - & - & - & - & - & - & - \\ 0.5 & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & 0 & 1 & - & - \\ - & - & - & - & - & 2 & 0.5 \\ - & 2 & 2 & - & - & - & - \\ - & 0.5 & 1.5 & - & - & - & - \end{bmatrix}$$

In the cell corresponding to the second row and first column in U_1', there is an utility of 1.5. This number is not readable by any terminal node of the game in Figure 2.3. From where does 1.5 come out? The spot in the utility matrix is the one that corresponds to $q_1 = L_1$ and $q_2 = q_\emptyset$: with this sequence profile we reach the node with payoff $(3, 1)$ but, in order to reach it, there is a move of Nature with probability value of 0.5. This move has to be considered in the computation of utility matrix (and for this reason 3 is multiplied by 0.5).

Also with the sequence-form representation, we can write the expected utility of a player, defined as follows:

Definition 2.2.11. The expected utility $\mathbb{E}_{\mathbf{q}\sim\mathbf{r}}[U_i(\mathbf{q})]$ returns the expected value of utility of player i given strategy profile $\mathbf{r}$. The formula can be written:

$$\mathbb{E}_{\mathbf{q}\sim\mathbf{r}}[U_i(\mathbf{q})] = \sum_{q_1 \in Q_1} \sum_{q_2 \in Q_2} ... \sum_{q_n \in Q_n} r_1(q_1) r_2(q_2) \cdots r_n(q_n) U_i(q_1, ..., q_n)$$

We should remember that $U_i(\mathbf{q}) = 0$ when the sequence profile $\mathbf{q}$ does not lead to a terminal node.

2.3 Solution concepts

We have not mentioned anything about the optimal strategy, that is a strategy that maximizes the player's expected payoff for a given environment in which the agent plays. There are different ways of maximizing its own utility and we call them *solution concepts*.

We start from the maxmin solution:[4]

Definition 2.3.1. The maxmin strategy is arg$max_{s_i} min_{s_{-i}} U_i(s_i, s_{-i})$ for player i, where s_i and s_{-i} are strategy of player i and of players $N \setminus \{i\}$. The maxmin value for player i is $max_{s_i} min_{s_{-i}} U_i(s_i, s_{-i})$.

We recall also the notion of Nash equilibria, that we define for simplicity for two players, as reported in [27]:[5]

Definition 2.3.2. A two-player not-cooperative game in normal form is a quadruplet $(X, Y,\ f : X \times Y \to \mathbb{R},\ g : X \times Y \to \mathbb{R})$. A **Nash equilibrium** for the game is a pair $(\bar{x}, \bar{y}) \in X \times Y$ such that:

- $f(\bar{x}, \bar{y}) \geq f(x, \bar{y})$ for all $x \in X$;
- $g(\bar{x}, \bar{y}) \geq f(\bar{x}, y)$ for all $y \in Y$;

where X and Y are the strategy spaces of the two players.

A solution concept that generalizes the Nash equilibria is the correlated equilibria. A device (an external mediator between players) draws strategy profile from a known joint probability distribution and privately communicates them to each player *ex ante* (so before the starter of the game). The probability distribution induced an equilibrium if each player gets less by deviating, assuming that everyone will follow the recommendation [11]. The correlated equilibria in normal form is defined as follow, as reported in [36]:[6]

Definition 2.3.3. Let $X = \prod_{i \in N} X_i$ and $z \in \Delta(X)$ [7] be a distribution over pure strategy profiles, where $z^{(\bar{x})}$ denotes the probability of pure

[4] We remember that the maxmin solution with two-player games can be computed efficiently in polynomial time. With three or more players, the problem is much more involved and it does not admit any polynomial-time algorithm. We mention some recent works in this field: [8, 17, 4]. Furthermore, a variant of the maxmin solution that has recently received a big attention is the concept of Stackelberg equilibrium [7, 6, 7].

[5] We recall that computing a Nash equilibrium with two-player games cannot be done in polynomial time unless the computational class PPAD is included in the computational class P, but it is unlikely that this holds. We mention some recent results on the computation of Nash equilibria [9, 21, 20, 33, 15].

[6] Recent works on the computation of correlated equilibria in extensive-form games are provided in [14, 10, 12].

[7] $z \in \Delta(X)$ is a simplex over X

strategy $\bar{x} \in S$. z is a **correlated equilibria (NFCE)** if and only if for every player i and strategies $r, t \in X_i$,

$$\sum_{\bar{x}_{-i} \in X_{-i}} U_i(r, \bar{x}_{-i}) \cdot z^{(r, \bar{x}_{-i})} \geq \sum_{\bar{x}_{-i} \in X_{-i}} U_i(t, \bar{x}_{-i}) \cdot z^{(r, \bar{x}_{-i})}$$

If we require that the suggested action is a best response in expectation before the recommended action is revealed we have a *coarse correlated equilibria*.

Definition 2.3.4. z is a **coarse correlated equilibria (NFCCE)** if and only if for every player i and strategies $t \in X_i$,

$$\sum_{\bar{x} \in X} u_i(\bar{x}) \cdot z^{(\bar{x})} \geq \sum_{\bar{x}_{-i} \in X_{-i}} p_i(t, \bar{x}_{-i}) \cdot z_{-i}^{(\bar{x}_{-i})} \tag{2.1}$$

where $z_{-i}^{(\bar{x}_{-i})} = \sum_{r \in X_i} z^{(r, \bar{x}_{-i})}$ is the marginal probability that the pure strategy profile sampled by z for players $N \setminus \{i\}$ is $\bar{x}_{-i}$.

It is known that every NFCE is also a NFCCE, as reported in [36].

The concept of correlated equilibria can be adopted even in games in extensive form. The so-called device, instead of communicate with players before the beginning of the game, a message to them is sent for each information set reached during the game [34].

Definition 2.3.5. Given a correlation device μ (so a probability distribution on set of all strategy profiles), consider the extended game in which a chance move first selects a strategy profile π according to μ. Then, whenever a player i reaches an information set h in H_i, he receives the move c at h specified in π as a signal, interpreted as a recommendation to play c. An **extensive-form correlated equilibrium (EFCE)** is a Nash equilibrium of such an extended game in which players follow their recommendation.

Also the definition of coarse correlated equilibria can be extended in extensive form [18]:

Definition 2.3.6. An **extensive-form coarse-correlated equilibrium (EFCCE)** is similar to EFCE in that each recommended move is only revealed when the players reach the decision point for which the recommendation is relevant. However, unlike EFC, the acting player must choose whether or not to commit to the recommended move before such a move is revealed to them, instead of after.

2.4 Game classes

The games can be classified with respect to their utility functions.

The most famous one is the class of *Zero-sum games*.

Definition 2.4.1. A zero-sum game is a game in which, for each terminal node ω in the game tree, the following property holds: $\sum_{i \in N} U_i(\omega) = 0$.

An easy example in normal form is the Rock-Paper-Scissors game. Two players, each one has to choose simultaneously Rock (R), Paper (P) or Scissors (S). If both choose the same action there is a tie (0 as utility for both), otherwise Rock beats Scissors that beats Paper that beats Rock. When there is no tie, the winner gets 1, the loser -1.

		Player 2		
		R	P	S
Player 1	R	(0,0)	(-1,1)	(1,-1)
	P	(1,-1)	(0,0)	(-1,1)
	S	(-1,1)	(1,-1)	(0,0)

We remark that in each spot of the matrix, the sum is equal to 0.

An equivalent class is the one of the constant-sum game.

Definition 2.4.2. A constant-sum game is a game in which, for each terminal node ω in the game tree, the following property holds: $\sum_{i \in N} U_i(\omega) = constant$.

An example in normal form of a constant sum game is the Rock-paper-Scissors game (shown before) once a constant is added to the payoffs of each player:

		Player 2		
		R	P	S
Player 1	R	(1,1)	(1,1)	(2,0)
	P	(2,0)	(1,1)	(1,1)
	S	(1,1)	(2,0)	(1,1)

Some two-player games can have the following property. For every change of strategy profile, if a player increases her expected utility, the other reduces hers. This class of games are called *strictly competitive* game. There will be reported the definition in [23]:

Definition 2.4.3. The two-person game Γ is called strictly competitive (SC) if for each $i, j = 1, 2$ $i \neq j$ and for all strategy profile σ' σ'', we have $U_i(\sigma'_1, \sigma'_2) \geq U_i(\sigma''_1, \sigma''_2)$ if and only if $U_j(\sigma'_1, \sigma'_2) \leq U_j(\sigma''_1, \sigma''_2)$.

An example in normal-form is the following:

		Player 2		
		R	P	S
Player 1	R	(3,1)	(1,2)	(5,0)
	P	(5,0)	(3,1)	(1,2)
	S	(1,2)	(5,0)	(3,1)

A candidate type of game suitable for our goal are the *Unilaterally Competitive* games. They can have more than two players and any unilateral change of strategy take a vantage (disadvantage, respectively) in terms of utility to her, take a disadvantage (vantage, respectively) to the others.

Definition 2.4.4. The game Γ is called Unilaterally Competitive (UC) if for each $i \in N$ for all strategy profile σ'_i σ''_i of player i and σ'_{-i} of player $N \setminus \{i\}$ we have $U_i(\sigma''_i, \sigma'_{-i}) \geq U_i(\sigma'_i, \sigma'_{-i})$ if and only if $U_j(\sigma''_i, \sigma'_{-i}) \leq U_j(\sigma'_i, \sigma'_{-i})$, for all $j \in N$, $j \neq i$.

These classes of game will be studied in detail in Chapter 3.

Another class of game that we will use in the thesis is the one of the *Polymatrix games*. In [36], they are defined as follows:

Definition 2.4.5. A **Polymatrix game** G consists of the following:

- A finite set $V = \{1, ..., n\}$ of players (sometimes called nodes), and a finite set E of edges, which are taken to be unordered pairs $[i, j]$ of players, $i \neq j$;
- for each player $i \in V$, a finite set of strategies S_i;
- for each edge $[i, j] \in E$, a two-person game (p^{ij}, p^{ji}) where the players are i,j, the strategy set S_i, S_j, respectively, and the payoffs $p^{ij} : S_i \times S_j \mapsto \mathbb{R}$, and similarly for p^{ij};
- for each player $i \in V$ and strategy profile $\bar{s} = (s_1, ...s_n) \in \prod_{j \in V} S_j$, the payoff of player i under $\bar{s}$ is $p_i(\bar{s}) = \sum_{[i,j] \in E} p^{ij}(s_i, s_j)$

The Polymatrix games can be represented through a Graph (in Figure 2.5 we have an example with 4 players) where A_i is the space of

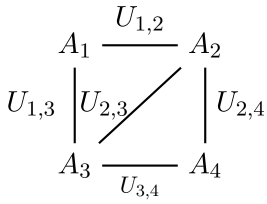

Figure 2.5: Graphical example representation of Polymatrix game: each edge is a two-player game. Here, player 1 (with a set action A_1) plays against player 2 and player 3 but not against player 4.

actions of each player. Any player can play potentially against all the players and for each game has a different utility function. Every edge is a two-person game. The player's strategy is unique, the games are different.

Finally, we introduce the definition of treeplex [29]:

Definition 2.4.6. A treeplex can be see as a tree whose nodes are simplexes. The tree structure endows the complex with a certain kind of sequential characteristic. In particular, treeplexes include the types of polytopes that arise in the computation of Nash equilibria of sequential games. The class of complexes is recuversely defined as follows:

1. Basic sets: Every standard simplex $\Delta_m := \{\mathbf{x} \in [0,1]^m : \sum_{j=1}^{k} x_j = 1\}$ is a treeplex.

2. Cartesian product: if $Q_1, ..., Q_k$ are treeplexes, then $Q_1 \times ... \times Q_k$ is a treeplex.

3. Branching: if $P \subseteq [0,1]^p$ and $Q \subseteq [0,1]^q$ are treeplexes and $i \in \{1, ..., p\}$ then:

$$P \boxed{\mathrm{i}} Q := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{p+q} : \mathbf{x} \in P, \mathbf{y} \in x_i \cdot Q\}$$

   is a treeplex.

Chapter 3

UC games with two players

In this chapter, we study the two-player UC games.

3.1 Characterization in Utility Space

The UC games constitute a class of games where the utilities of players are somehow restricted. When the number of players is two, a game is UC if any unilateral change of strategy by one player results in a increase in that player's payoff if and only if this change in strategy results in a decline in the payoffs of the other player. This is not the same property of SC games and in the next section we will show the difference.

3.1.1 Zero-sum, Constant-Sum and Strictly Competitive games

The classes of Zero-sum games, the Constant-sum games and SC games are restricted in their utilities and they follow the same idea of UC games (when one improves after her strategy change, the others get less utility).

It is known that a zero-sum game is equivalent to a constant-sum game (once a constant has been subtracted from the utility of a player) and any SC game is equivalent to a zero-sum game once an affine transformation, in principle different for any player, is applied. The property that defines the SC games is different to the one that defines the UC games. In fact, in a SC game the relative changes in the payoffs have to be satisfied for any pair of strategies and not only for the unilateral changes. The condition in SC games is more restrictive.

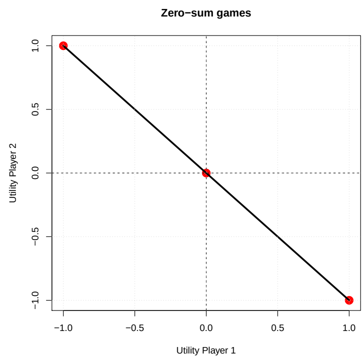

Figure 3.1: Graphical representation of Zero-sum game's payoff with two players: the points lie on $U_1 + U_2 = 0$.

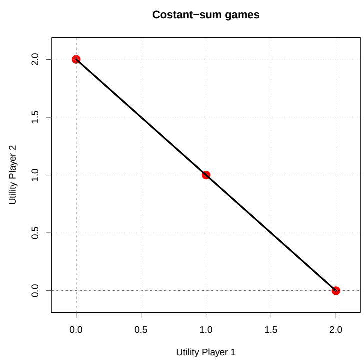

Figure 3.2: Graphical representation of Constant-sum game's payoff with two players and $constant = 2$.

In [23], it is claimed that the class of SC games is a subset of the class of UC games. How these game are represented in the Utility Space?

Given the space of players' utilities $(U_1, U_2, \cdots, U_n)$ in $\mathbb{R}^n$, each terminal node of a zero-sum game can be mapped as a point in such a space and all these points lie on a hyperplane $\sum_{i \in N} U_i = 0$. All the points of segment in Figure 3.1 are reachable by mixed strategies.

In a constant-sum game, since they are equivalent to a zero-sum game, given the space of players' utilities $(U_1, U_2, ..., U_n)$ in $\mathbb{R}^n$, each terminal node can be mapped as a point in such a space and all these points lie on a hyperplane $\sum_{i \in N} U_i = constant$. All the points of the segment in Figure 3.2 are reachable by mixed strategies.

In a two-person SC game, given the space (U_1, U_2) in $\mathbb{R}^2$, each terminal node can be mapped as a point in such a space and all these

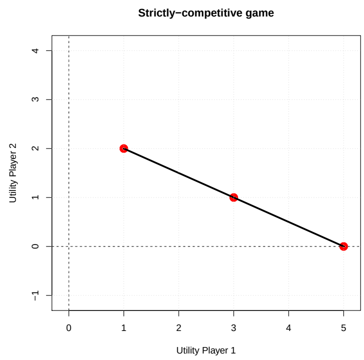

Figure 3.3: Graphical representation of SC game's payoff.

points lie on a hyperplane $\sum_{i \in N} \alpha_i U_i = constant$ where $\alpha_i \in (0, 1]$. Also in this case, the line has negative slope as in Figure 3.3.

3.1.2 Two-player UC games

The characterization idea for UC games is very similar: every change of strategy of a player has to be represented by a negative line slope. In this class of games, we do not have a unique line (as in the previous classes) but a bundle of straight line: each line is a *unilateral change* of strategy by one single player while it is fixed the opponents' strategy. These relative changes in the payoff have to be satisfied only for unilateral changes and not for all pairs of strategies (as underlined in [23]).

If all the straight lines have negative slope, we have exactly the meaning of the definition: the change of strategy of a single player takes an advantage (disadvantage, respectively) to her and a disadvantage (advantage, resp.) to the others. This is true because a positive straight line means an improvement for both player as a consequence of change of strategy profile (and this is against the definition of UC game).

To verify this condition, we write the utilities of each player in terms of their mixed strategies and study the rate with respect to the change of strategy (and this can be done through the analysis of derivative's sign). In this chapter, we are studying the case with only two players so we are requiring that, while a utility of a specific player increases (decreases, resp.) its value thanks to a change of strategy, the utility of the other player decreases (increases, resp.). So the derivatives of the utility function with respect to the strategy profile must have different

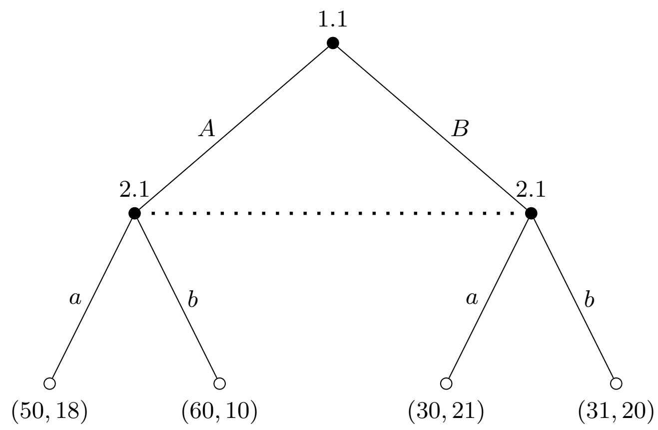

Figure 3.4: Extensive-form of the Example in 3.1.3.

signs.

An example is shown below.

3.1.3 Example

Let the strategy set of player 1 be $X_1 = \{A, B\}$ and the strategy set of player 2 be $X_2 = \{a, b\}$; the payoff matrix is:

		Player 2	
		a	b
Player 1	A	(50,18)	(60,10)
	B	(30,21)	(31,20)

Each player has an unique information set. In Figure 3.4, is shown the game in extensive form.

We define the mixed strategy of each player. We denote with variable z the probability that player 1 will play action A and the her strategy is:

$$\sigma_1 = \begin{cases} z & A \\ 1-z & B \end{cases}$$

And with variable x the probability that player 2 will play action a and her strategy is:

$$\sigma_2 = \begin{cases} x & a \\ 1-x & b \end{cases}$$

x and z are real numbers contained in $[0, 1]$. Defining the strategies in this way helps us in capturing any payoff, given any strategy profile.

Once defined the strategies, the utility functions of each player in terms of mixed strategy are:

$$U_1 = 50xz + 60(1-x)z + 30x(1-z) + 31(1-x)(1-z)$$

$$U_2 = 18xz + 10(1-x)z + 21x(1-z) + 20(1-x)(1-z)$$

Now, we study how the unilateral change of strategies affects the utility function in terms of increasing and decreasing their payoff. To do this, we study the derivative of U with respect to x and z. In fact, the derivative is the measure of the increase (or decrease) owing to the change of strategy. We are interested only on its sign.

First of all, we analyse the derivative of U_1 with respect to z: study this derivative means study how the change of strategy in terms of increasing the probability of play A by player 1 affects her own utility while player 2 fixes her strategy. If the derivative is positive, it means that the change of strategy improves her utility, if it is negative she gets worse. The same computation has to be done for U_2 with respect to z.

If a game is UC, we expect to have derivatives of opposite sign for the two players while they are referring to the same derivation variable. If they have the same sign, this means that, owing to a change of strategy of one specific player, both increase or decrease their utility (increase if both positive, decrease if both negative).

We compute them, starting from variable z for the first player we have:

$$\begin{aligned}\frac{\partial U_1}{\partial z} &= 50x + 60(1-x) - 30x - 31(1-x) = \\ &= 50x + 60 - 60x - 30x - 31 + 31x = -9x + 29\end{aligned}$$

and for the second player:

$$\begin{aligned}\frac{\partial U_2}{\partial z} &= 18x + 10(1-x) - 21x - 20(1-x) = \\ &= 18x + 10 - 10x - 21x - 20 + 20x = 7x - 10\end{aligned}$$

Fixed x and for any of its possible value, $\frac{\partial U_1}{\partial z}$ is positive and $\frac{\partial U_2}{\partial z}$ is negative. So, after a change of strategy by Player 1, one improves her utility and the other not. In Figure 3.5, it is represented the unilateral

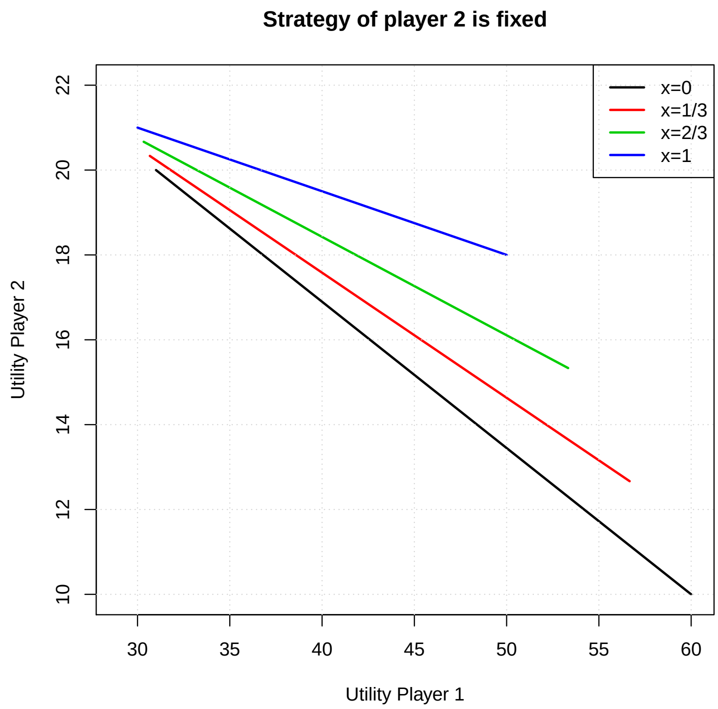

Figure 3.5: This picture represents the utility payoff of games, when player 2's strategy is fixed. For a better representation, we put only some value of x: the straight lines have negative slope.

change of strategy of Player 1 for different value of x. It is clear that all the lines have negative slope.

Now, we should verify the condition also for the change of strategy of Player 2 and so we compute the derivative of U_1 and U_2 with respect to x:

$$\frac{\partial U_1}{\partial x} = 50z - 60z + 30(1-z) - 31(1-z) = \\ = 50z - 60z + 30 - 30z - 31 + 31z = -9z - 1;$$

$$\frac{\partial U_2}{\partial x} = 18z - 10z + 21(1-z) - 20(1-z) = \\ = 18z - 10z + 21 - 21z - 20 + 20z = 7z + 1;$$

Fixed z and for any of its possible value, $\frac{\partial U_2}{\partial x}$ is positive while $\frac{\partial U_1}{\partial x}$ is negative. The graphical representation is on Figure 3.6 and the interpretation is the same of the previous case (all lines have negative slope).

Now, we want to show a counterexample that can be easily mistaken with a UC game but it is not. We will show why.

We have the same action space of the previous example but U_1 and U_2 are different. In Figure 3.7 it is represented in extensive form representation, while in the normal-form is:

		Player 2	
		a	b
Player 1	A	(10,20)	(12,10)
	B	(9,24)	(7,26)

Looking at the pure strategies, the game seems UC. For example, consider that Player 2 will play b and Player 1 changes from B to A: the latter will improve her utility (from 7 to 12) while the other will get worse (from 26 to 10). The same argument can be repeated for the other pure strategies.

Taking the same mixed strategy profile for each player of the previous example, the utility functions are:

$$U_1 = 10xz + 9x(1-z) + 12z(1-x) + 7(1-z)(1-x)$$

$$U_2 = 20xz + 24x(1-z) + 10z(1-x) + 26(1-z)(1-x)$$

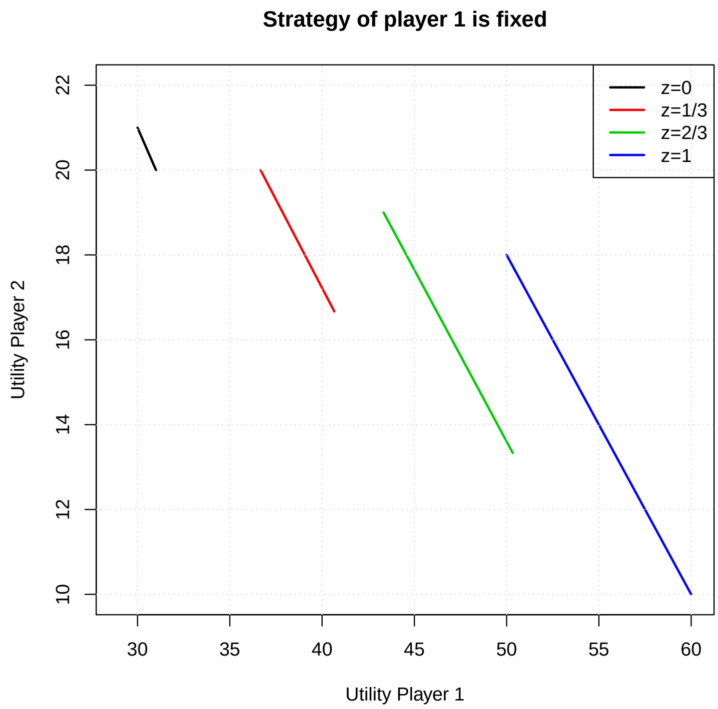

Figure 3.6: This picture represents the utility payoff of games, when Player 1's strategy is fixed. For a better representation, we put only some value of z. The straight lines have negative slope.

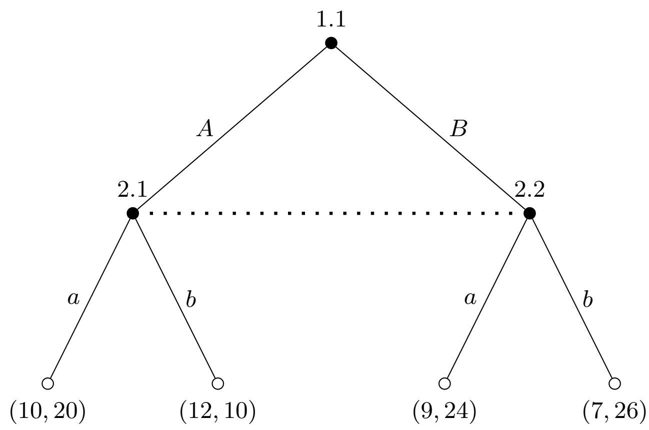

Figure 3.7: Counterexample in Section 3.1.3.

The game is not UC because of the following strategy profile: suppose Player 1 plays $(\frac{1}{3},\frac{2}{3})$ while Player 2 moves from profile $(\frac{1}{3},\frac{2}{3})$ to $(\frac{1}{2},\frac{1}{2})$. In the first scenario, the payoff is $(\frac{80}{9}, \frac{64}{3})$ while in the second one is $(9, \frac{65}{3})$: these last values are both greater than the first ones so we have found a strategy profile that is in contradiction with the definition of UC. The graphical representation in utility space is on Figure 3.8. This is sufficient to prove the non-unilaterally competitive of the game.

It is also interesting see the geometrically representation in Utility space (Figure 3.9) for more values of x. Graphically, it is clear that there are a lot of positive slope lines and so, a changing of strategy of Player 2 for some specific value of Player 1's strategy, take an increase of payoff for both players.

3.2 Modularity between two games

The previous method is not easy to verify and we are looking for some easier and more immediate properties in order to verify that a game is UC. One is the *Modularity between two games*: taking two games with same restriction in utility function, we unify them through a node of Nature, do they preserve the payoff matrix constraints?

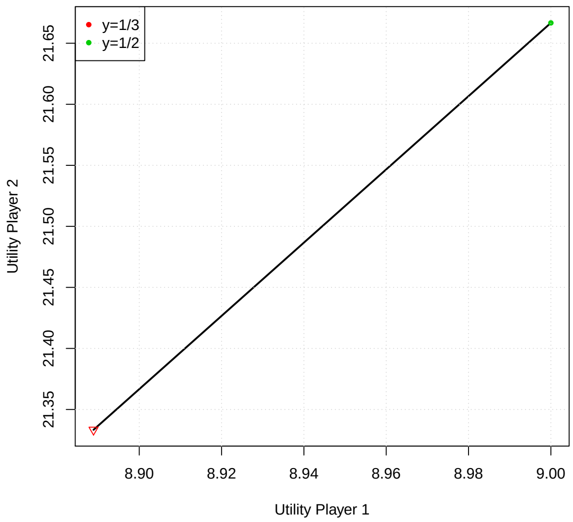

Figure 3.8: This is the representation of the positive slope line in the change of strategy of Player 2 from $(\frac{1}{3}, \frac{2}{3})$ to $(\frac{1}{2}, \frac{1}{2})$.

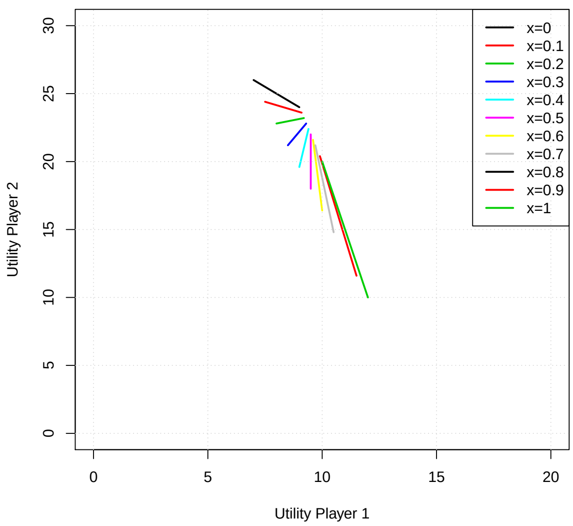

Figure 3.9: Graphical representation of different strategy profiles of Player 2: it is clear that in the rate from 0 to 1 there are some positive slope lines.

Let define two general games (Γ_1 and Γ_2) with two players and a strategy set of two actions for each one described by:

Γ_1 :

		Player 2	
		r_1	s_1
Player 1	R_1	(a_{11},b_{11})	(a_{12},b_{12})
	S_1	(a_{13},b_{13})	(a_{14},b_{14})

Γ_2 :

		Player 2	
		r_2	s_2
Player 1	R_2	(a_{21},b_{21})	(a_{22},b_{22})
	S_2	(a_{23},b_{23})	(a_{24},b_{24})

We denote with U_{11} and U_{12} the utility matrices of player 1 and player 2 regarding the game Γ_1 and U_{21} and U_{22} regarding Γ_2. The action space of player 1 and player 2 in Γ_1 is X_1 and Y_1 respectively. In Γ_2 the spaces are X_2 for player 1 and Y_2 for player 2. The probability given by the Nature is α for the first game and $1-\alpha$ for the other one.

First of all, we need to understand the meaning of *Modularity*. We can identify two cases.

The first one is when the action's space are equal between Γ_1 and Γ_2 ($X_1 \equiv X_2$ and $Y_1 \equiv Y_2$). The payoff matrix is a weighted sum between the utility matrices of games. In Figure 3.10, the extensive-form representation while in the following the normal-form (we remove the subscript by calling, for sake of simplicity, S instead of S_1 and S_2 and the same for R):

		Player 2	
		r	s
Pl.1	R	$(\alpha a_{11} + (1-\alpha)a_{21},$ $\alpha b_{11} + (1-\alpha)b_{21})$	$(\alpha a_{12} + (1-\alpha)a_{22},$ $\alpha b_{12} + (1-\alpha)b_{22})$
	S	$(\alpha a_{13} + (1-\alpha)a_{23},$ $\alpha b_{13} + (1-\alpha)b_{23})$	$(\alpha a_{14} + (1-\alpha)a_{24},$ $\alpha b_{14} + (1-\alpha)b_{24})$

The second case is when the actions' space between games are not equal. So the the root of the game (in extensive form) is a Nature node that gives a probability α to play game Γ_1 and $1-\alpha$ to play Γ_2, with $\alpha \in [0,1]$ (as shown in Figure 3.11). In this way, we are going to modify the action's space of each player: after the combination, the actions' space of each player is the Cartesian product of the previous ones (so $X = X_1 \times X_2$ and $Y = Y_1 \times Y_2$).

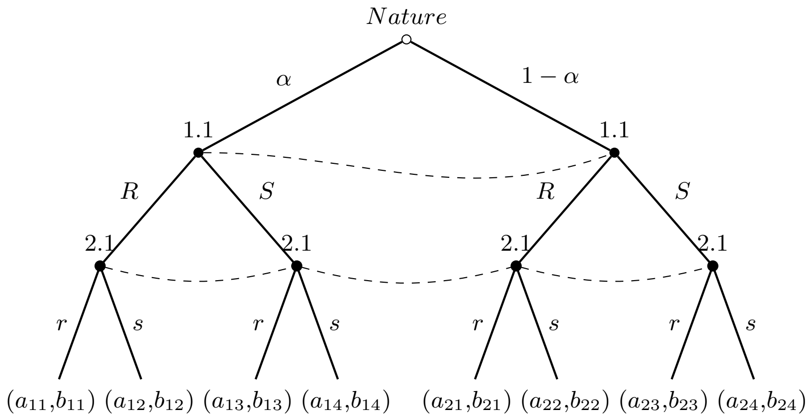

Figure 3.10: Composition of a game when the actions' spaces of the players coincide.

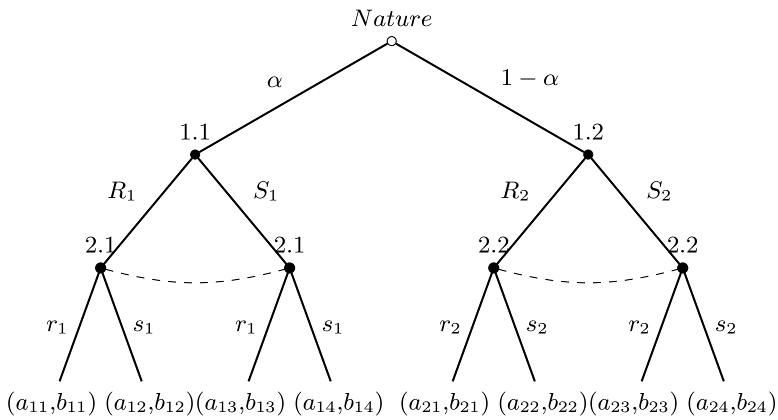

Figure 3.11: Composition of a game when the actions' spaces of the players do not coincide beetween games.

This way of combining two games is representable in normal form with the following utility matrix:

		Player 2			
		r_1r_2	r_1s_2	s_1r_2	s_1s_2
Pl.1	R_1R_2	$(\alpha a_{11} + (1-\alpha)a_{21}, \alpha b_{11} + (1-\alpha)b_{21})$	$(\alpha a_{11} + (1-\alpha)a_{22}, \alpha b_{11} + (1-\alpha)b_{22})$	$(\alpha a_{12} + (1-\alpha)a_{21}, \alpha b_{12} + (1-\alpha)b_{21})$	$(\alpha a_{12} + (1-\alpha)a_{22}, \alpha b_{12} + (1-\alpha)b_{22})$
	R_1S_2	$(\alpha a_{11} + (1-\alpha)a_{23}, \alpha b_{11} + (1-\alpha)b_{21})$	$(\alpha a_{11} + (1-\alpha)a_{24}, \alpha b_{11} + (1-\alpha)b_{24})$	$(\alpha a_{12} + (1-\alpha)a_{23}, \alpha b_{12} + (1-\alpha)b_{23})$	$(\alpha a_{12} + (1-\alpha)a_{24}, \alpha b_{12} + (1-\alpha)b_{24})$
	S_1R_2	$(\alpha a_{13} + (1-\alpha)a_{21}, \alpha b_{13} + (1-\alpha)b_{21})$	$(\alpha a_{13} + (1-\alpha)a_{22}, \alpha b_{13} + (1-\alpha)b_{22})$	$(\alpha a_{14} + (1-\alpha)a_{21}, \alpha b_{14} + (1-\alpha)b_{21})$	$(\alpha a_{14} + (1-\alpha)a_{22}, \alpha b_{14} + (1-\alpha)b_{22})$
	S_1S_2	$(\alpha a_{13} + (1-\alpha)a_{23}, \alpha b_{13} + (1-\alpha)b_{23})$	$(\alpha a_{13} + (1-\alpha)a_{24}, \alpha b_{13} + (1-\alpha)b_{24})$	$(\alpha a_{14} + (1-\alpha)a_{23}, \alpha b_{14} + (1-\alpha)b_{23})$	$(\alpha a_{14} + (1-\alpha)a_{24}, \alpha b_{14} + (1-\alpha)b_{24})$

3.2.1 Zero-sum games

If Γ_1 and Γ_2 are zero-sum games, is the combination of the two games a zero-sum game? In other words, does the combination of the two games preserve the property? We study the answers at these questions in both cases.

Considering the first case. Each outcome is at zero-sum: in fact, each node of game Γ_1 is $U_{11} + U_{12} = 0$ and for each node of game Γ_2 we have $U_{21} + U_{22} = 0$. Also the convex combination of them:

$$\alpha \cdot (U_{11} + U_{12}) + (1-\alpha) \cdot (U_{21} + U_{22}) = \alpha \cdot 0 + (1-\alpha) \cdot 0 = 0$$

is equal to 0.

Also in the second case it remains a zero-sum game because we are combining through a weighted average two results that summed together are at zero-sum. More specifically, let consider the sum of a general payoff of the previous utility function:

$$\begin{aligned} \alpha \cdot a_{1j} + (1-\alpha) \cdot a_{2i} + \alpha \cdot b_{1j} + (1-\alpha) \cdot b_{2i} &= \\ = \alpha(a_{1j} + b_{1j}) + (1-\alpha)(a_{2i} + b_{2i}) &= \\ = \alpha \cdot 0 + (1-\alpha) \cdot 0 = 0 & \end{aligned}$$

for any value of α from 0 to 1. So the combination of two different zero-sum games is a zero sum game since all the spots in the utility matrix have sum at zero.

3.2.2 Constant-sum games

And what if the Γ_1 and Γ_2 are constant-sum games? Suppose K_1 and K_2 the constants of the Γ_1 and Γ_2.

If $K_1 = K_2$, it is trivial because the case is very similar to the zero sum case.

If $K_1 \neq K_2$, the new game is a game with constant $\alpha K_1 + (1-\alpha)K_2$.

For what concerns the first case, in each node of Γ_1 we have $U_{11} + U_{12} = K_1$ and in each node of Γ_2 we have $U_{21} + U_{22} = K_2$: also the convex combination of them $(\alpha \cdot (U_{11} + U_{12}) + (1-\alpha) \cdot (U_{21} + U_{22}))$ is equal to a constant (that is $\alpha \cdot (K_1) + (1-\alpha) \cdot (K_2)$).

For the second case, for each terminal node ω we can say:

$$\sum_{j=1,2} \sum_{i \in N} U_{ij}(\omega) = constant \Rightarrow \alpha \cdot a_{1i} + (1-\alpha) \cdot a_{2j} + \alpha \cdot b_{1i} + (1-\alpha) \cdot b_{2j} =$$

$$= \alpha \cdot (a_{1i} + b_{1i}) + (1-\alpha) \cdot (a_{2j} + b_{2j}) =$$
$$= \alpha \cdot (K_1) + (1-\alpha) \cdot (K_2) = constant$$

So:

$$constant = \alpha \cdot (K_1) + (1-\alpha) \cdot (K_2)$$

3.2.3 Strictly Competitive games

Arguing about combination of SC games, it is less immediate. We know that (from [22]), mathematically, a game $(A, -B)$ (where A and $-B$ are the payoffs of the players) is strictly competitive if for any two pairs of mixed strategies (x, y) and (x', y'), $x^T Ay - x'^T Ay'$ and $x^T By - x'^T By'$ have the same sign: the characterization is that B is an affine variant of A, so there exist some $\lambda > 0$ and μ real number such that:

$$B = \lambda A + \mu U$$

where U is a matrix of 1.

For simplicity, we suppose that each player has two actions and the payoffs for Γ_1 are $(A_1, -B_1)$, defined as follows:

$A_1 =$

	r_1	s_1
R_1	a_{11}	a_{12}
S_1	a_{13}	a_{14}

$B_1 =$

	r_1	s_1
R_1	b_{11}	b_{12}
S_1	b_{13}	b_{14}

And for Γ_2 are $(A_2, -B_2)$, defined as follows:

$$A_2 = \begin{array}{c|c|c|} & r_1 & s_1 \\ \cline{2-3} R_1 & a_{21} & a_{22} \\ \cline{2-3} S_1 & a_{23} & a_{24} \\ \cline{2-3} \end{array} \qquad B_2 = \begin{array}{c|c|c|} & r_2 & s_2 \\ \cline{2-3} R_2 & b_{21} & b_{22} \\ \cline{2-3} S_2 & b_{23} & b_{24} \\ \cline{2-3} \end{array}$$

for Γ_2.

If these two games are SC, there exist λ_i and μ_i such that: $B_i = \lambda_i A_i + \mu_i U$ for $i = 1, 2$.

If the players have the same action's space in the two games, it is necessary arguing about the payoff matrices, that are:

$$A_\alpha = \alpha A_1 + (1 - \alpha) A_2$$

$$B_\alpha = \alpha B_1 + (1 - \alpha) B_2$$

If there exist λ_α and μ_α such that:

$$B_\alpha = \lambda_\alpha A_\alpha + \mu_\alpha U$$

also the composition of two games is a SC game. We do some computation:

$$B_\alpha = \alpha B_1 + (1 - \alpha) B_2 = \alpha(\lambda_1 A_1 + \mu_1 U) + (1 - \alpha)(\lambda_2 A_2 + \mu_2 U) =$$
$$= \lambda_1(\alpha A_1 + (1 - \alpha)\frac{\lambda_2}{\lambda_1} A_2) + (\alpha \mu_1 + (1 - \alpha)\mu_2) U$$

So, if $\lambda_1 = \lambda_2$, the composition of two SC games is an SC game and

$$\lambda_\alpha = \lambda_1$$

$$\mu_\alpha = \alpha \mu_1 + (1 - \alpha) \mu_2$$

If we are in second case, the payoff matrix A of player 1 is:

		Player 2			
		$r_1 r_2$	$r_1 s_2$	$s_1 r_2$	$s_1 s_2$
Pl.1	$R_1 R_2$	$\alpha a_{11} + (1-\alpha) a_{21}$	$\alpha a_{11} + (1-\alpha) a_{22}$	$\alpha a_{12} + (1-\alpha) a_{21}$	$\alpha a_{12} + (1-\alpha) a_{22}$
	$R_1 S_2$	$\alpha b_{11} + (1-\alpha) b_{23}$	$\alpha a_{11} + (1-\alpha) a_{24}$	$\alpha a_{12} + (1-\alpha) a_{23}$	$\alpha a_{12} + (1-\alpha) a_{24}$
	$S_1 R_2$	$\alpha a_{13} + (1-\alpha) a_{21}$	$\alpha a_{13} + (1-\alpha) a_{22}$	$\alpha a_{14} + (1-\alpha) a_{21}$	$\alpha a_{14} + (1-\alpha) a_{22}$
	$S_1 S_2$	$\alpha a_{13} + (1-\alpha) a_{23}$	$\alpha a_{13} + (1-\alpha) a_{24}$	$\alpha a_{14} + (1-\alpha) a_{23}$	$\alpha a_{14} + (1-\alpha) a_{24}$

And the payoff matrix of Player 2 is $-B$, where B is:

		Player 2			
		$r_1 r_2$	$r_1 s_2$	$s_1 r_2$	$s_1 s_2$
Pl.1	$R_1 R_2$	$\alpha b_{11} + (1-\alpha) b_{21}$	$\alpha b_{11} + (1-\alpha) b_{22}$	$\alpha b_{12} + (1-\alpha) b_{21}$	$\alpha b_{12} + (1-\alpha) b_{22}$
	$R_1 S_2$	$\alpha b_{11} + (1-\alpha) b_{21}$	$\alpha b_{11} + (1-\alpha) b_{24}$	$\alpha b_{12} + (1-\alpha) b_{23}$	$\alpha b_{12} + (1-\alpha) b_{24}$
	$S_1 R_2$	$\alpha b_{13} + (1-\alpha) b_{21}$	$\alpha b_{13} + (1-\alpha) b_{22}$	$\alpha b_{14} + (1-\alpha) b_{21}$	$\alpha b_{14} + (1-\alpha) b_{22}$
	$S_1 S_2$	$\alpha b_{13} + (1-\alpha) b_{23}$	$\alpha b_{13} + (1-\alpha) b_{24}$	$\alpha b_{14} + (1-\alpha) b_{23}$	$\alpha b_{14} + (1-\alpha) b_{24}$

So, a general outcome in A is:

$$\alpha \cdot a_{1j} + (1-\alpha) \cdot a_{2i}$$

and a general outcome in B is:

$$\alpha \cdot b_{1j} + (1-\alpha) \cdot b_{2i}$$

where i and j are natural numbers from 1 to 4.

We know, since Γ_1 and Γ_2 are SC games, that there exist λ_1 and μ_1 such that $b_{1j} = \lambda_1 a_{1j} + \mu_1$ and λ_2 and μ_2 such that $b_{2j} = \lambda_2 a_{2j} + \mu_2$.

So:

$$\begin{aligned}\alpha \cdot b_{1j} + (1-\alpha) \cdot b_{2i} = \\ &= \alpha \cdot (\lambda_1 a_{1j} + \mu_1) + (1-\alpha) \cdot (\lambda_2 a_{2j} + \mu_2) = \\ &= \lambda_1 (\alpha \cdot a_{1j} + (1-\alpha) \frac{\lambda_2}{\lambda_1} a_{2i}) + \alpha\mu_1 + (1-\alpha)\mu_2\end{aligned}$$

Also in this case the composition between different games preserves the property if $\lambda_1 = \lambda_2$ and:

$$\lambda_\alpha = \lambda_1$$

$$\mu_\alpha = \alpha\mu_1 + (1-\alpha)\mu_2$$

Finally, only a little class of SC game can be composed by Nature in a stochastic way and its composition is a SC game too.

3.2.4 UC games: counterexample

In fact, in general, the composition of two UC games is not an UC game and now we show a counterexample.

We compose through nature, two UC games with the same action space (one sub-game is the one that we see in Section 3.1.3):

The game linked to the move with probability p is a UC game and the game linked to the move with probability $1-p$ too. The modularity is not UC. In fact, considering $p = 0.5$ and as first scenario player 1 plays A and player 2 plays a. Their utility functions are:

$$U_1 = 50p + 50(1-p) \xrightarrow{p=0.5} 50$$

$$U_2 = 18p + 46(1-p) \xrightarrow{p=0.5} 32$$

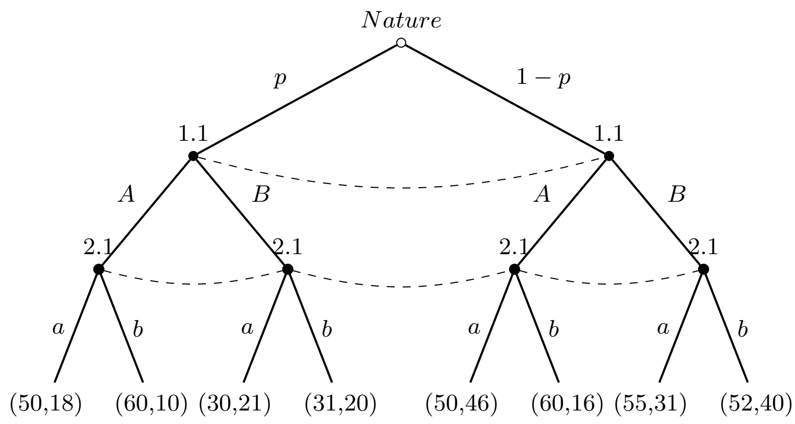

Figure 3.12: Extensive representation of the counterexample.

If player 1 decides to change her strategy and play B (instead of A), the utility functions of the players are:

$$U_1 = 30p + 55(1-p) \xrightarrow{p=0.5} 42.5\,[< 50]$$

$$U_2 = 21p + 31(1-p) \xrightarrow{p=0.5} 26\,[< 32]$$

So both players have less utility, so this is not a UC game.

3.2.5 Decomposability

If we decompose a game into two games, these new games preserve the property of the "total" game? A zero-sum game can be decomposed into two zero-sum games? About zero-sum and constant-sum games the answer is obvious and it is yes. About SC and UC games the answer is yes, since we are restricting the constraints about the strategy. So, if the property holds in a bigger group of constraints, it continues to hold in a more restricted group.

3.3 Perturbation of a zero-sum game

Since composing together two UC games gives not a UC Games, we would like to understand how to obtain a UC game. If we start from a game that is zero-sum, with x_1, x_2 and x_3 known, which properties has to have x_4 in order to obtain a UC game?

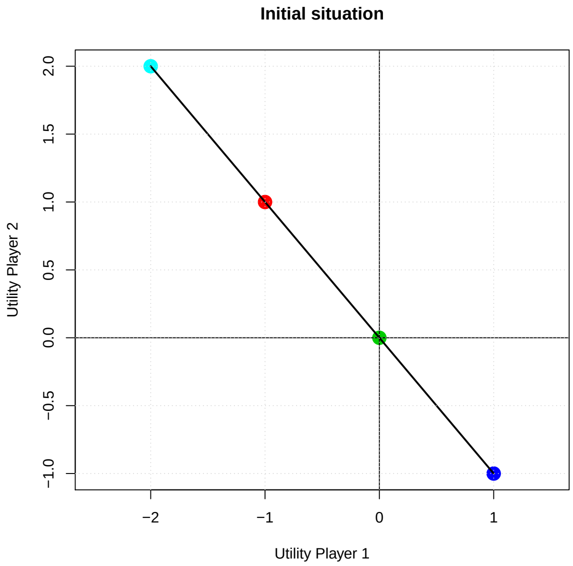

Figure 3.13: Graphical representation of example payoff.

		Player 2	
		a_1	b_1
Player 1	A_1	x_1	x_2
	B_1	x_3	x_4 ?

In order to obtain a zero-sum game, it is obvious that the outcome x_4 should lie on the bisector of the second and the fourth quarter. What we can looking for is, more in general, try to understand which condition the unknown payoff x_4 has to respect in order to have a SC or a UC game. We can start with the following example:

	a	b
A	$(0, 0)$	$(1, -1)$
B	$(-1, 1)$	$(-2, 2 - \epsilon)$

In Figure 3.14 there is the graphical representation and how to ϵ effects the payoff.

Therefore, we perturbate one payoff of one player of a zero-sum game. Is it still a zero-sum game? No. Is it a UC game? No and we are going to show a strategy profile that is a counterexample: suppose that Player 1 plays indifferently A or B (so her profile is $(1/2, 1/2)$) and we denote with p the probability of action b to be played and $1 - p$

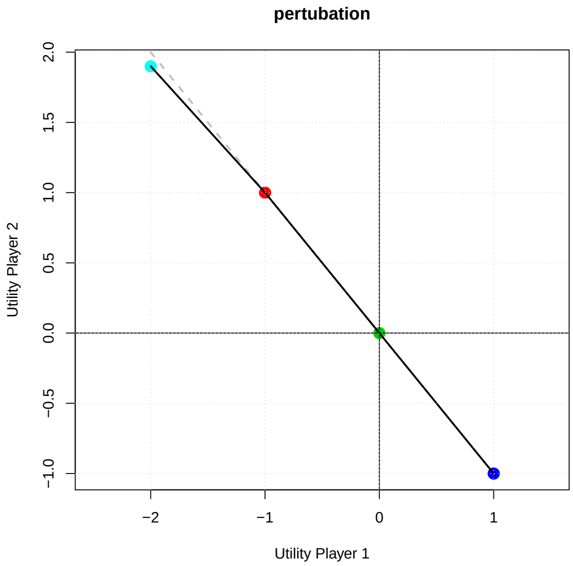

Figure 3.14: It is the game in the example but with the payoff $(-2, 2-\epsilon)$ *perturbed.*

the probability of action a. The utilities of each player are:

$$U_1 = \frac{1}{2}p - \frac{1}{2}(1-p) - 2(\frac{1}{2}p) = -\frac{1}{2}$$

$$U_2 = -\frac{1}{2}p + \frac{1}{2}(1-p) + \frac{1}{2}p(2-\epsilon) = \frac{1}{2} - \frac{\epsilon p}{2} = \frac{1}{2}(1-\epsilon p)$$

The utility of Player 1 is constant while the Player 2's utility can increase or decrease changing the value of p. There is no unilateral change.

Another example is that player 1 plays $(\frac{1}{3}, \frac{2}{3})$. So:

$$U_1 = \frac{1}{3}p - \frac{2}{3}(1-p) - \frac{4}{3}p = -\frac{2}{3} - \frac{1}{3}p$$

$$U_2 = -\frac{1}{3}p + \frac{2}{3}(1-p) + \frac{2}{3}p(2-\epsilon) = \frac{2}{3} + p(\frac{1}{3} - \frac{2}{3}\epsilon)$$

And it is clear that increasing p does not take to a unilateral change of payoff (with ϵ small enough).

3.4 Add a player

In this section we want to extend the number of players of this class of games.

To do that, we start from a two-player Strictly Competitive Game and we add a dummy player to it. A dummy player is a player with no action that participates to the game without doing anything. How the utility of the third player has to be in order to have a UC game? The answer is that it is not possible build that type of game, especially when there is no dominated actions.

Let see this through an example.

The initial SC game is the following one:

		Player 2	
		a	b
Player 1	A	(1,2)	(0,4)
	B	(0,4)	(2,0)

The third player is *dummy* so she has no action and the game, in three-player version, is:

		Player 2	
		a	b
Player 1	A	(1,2,x)	(0,4,y)
	B	(0,4,z)	(2,0,w)

where we just add the payoff of the third player to the existing one.

Which values of x, y, z and w are valid in order to make the SC game into an UC game?

First of all, let see the pure strategy profiles. Suppose player 1 will play B: if player 2 moves from playing a to playing b, she is reducing her utility (while player 1 is improving). Also player 3 has to improve her utility so $z < w$. For the same reason, $y < x$ while player 1 plays A. Now, suppose that player 2 will play b and player 1 changes from A to B. Since player 1 will improve her utility, both players 2 and 3 have to decrease their utility so $0 < 4$ and $w < y$. So the following chain is formed:

$$z < w < y < x \tag{3.1}$$

But z has to be greater than x: in fact, if Player 2 plays a and Player 1 switch from A to B, the latter is losing utility and so Player 2 and 3 have to gain from this change. So $z > x$ but this is a contradiction with Equation 3.1.

So we cannot construct, from a two-person SC game, a UC game even if the third player is dummy.

One of the goal is find classes of game solvable computationally easily. Initially we thought that the only class was the one of the UC games. But the games of this class are very hard to find. Also in [35] it is claimed that the *competitive property* is very restrictive when randomized strategies are permitted.

Chapter 4

Poly Sequence-Matrix Games

In this chapter we will present a class of game suitable for our goal with a generic number n of players.

4.1 Definition

We have to find and see if there is a class of game with a general number of players.

A zero-sum game with more than two players is something with no meaning. In fact, consider a game with a generic number $n-1$ of player: it can be easily transformed in a zero-sum game by taking the n^{th} player dummy and with payoff, for each terminal node ω, equals to $-\sum_{1\leq i\leq n-1} U_i(\omega)$.

We can look for the zero-sum Polymatrix games. A normal-form game has a utility matrix that is a tensor with dimension equal to the number of players. If we are considering a Polymatrix game, the dimensionality of utility matrix is reduced since, instead of a tensor, we have a list of two-dimension matrices.

We introduce the following class of games:

Definition 4.1.1. A Poly sequence-matrix game G consists of the following:

- A finite set $V = \{1, \ldots, n\}$ (where $n \in \mathbb{N}$) of players and a collection of treeplexes for each one;
- for each player $i \in V$, a finite set of sequence strategy r_i;

- A finite set E of unordered pairs $[ij]$ with $i \neq j$ and a collection of sparse utility functions $\{U_{ij}\}_{[ij]\in E}$;
- For each player i and strategy profile $r^* = (r_1, ..., r_n)$, the payoff of player i under r^* is $p_i(r^*) = \sum_{[i,j]\in E} U_{ij}(r_i, r_j)$;

Furthemore G is zero-sum game if for all strategy profiles $r^* = (r_1, ..., r_n)$, $\sum_{i\in V} p_i(r^*) = 0$.

A treeplex defines the order and the sequences of each player. A polymatrix game is a list of two-player games: the idea is the same so a player plays against not all the players (just a few) and the list of two-player games is contained in E. Each player will play the same strategy in each game.

Now we ask if the Poly sequence-matrix game is a unique game. If yes, we should able to extend it as a unique game in extensive-form game: the resulting tree has to have the property for which, in each path from the root of the game to a terminal node, it plays at most two players and Nature.

The idea is something similar to the one in Figure 4.1.

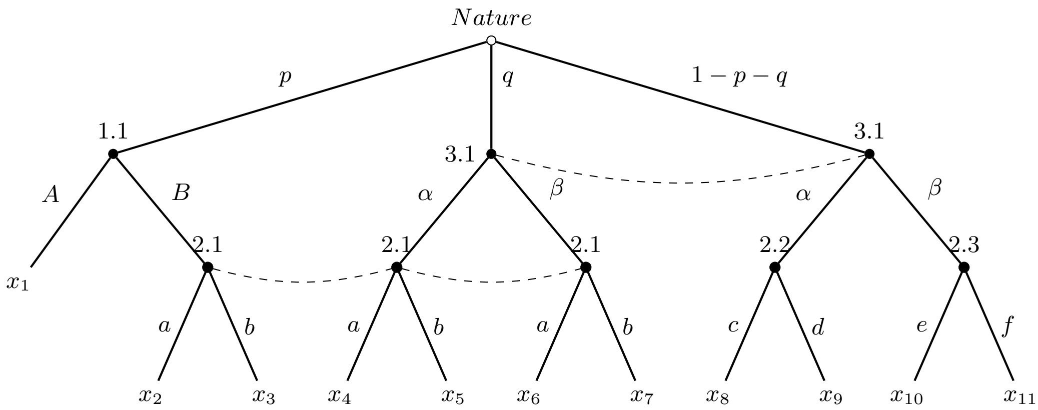

Figure 4.1: Each payoff x_i is reachable from information sets of at most two players.

As a real-world scenario, we can imagine a multinational corporation that has to decide his economic strategy in different countries. For a sake of simplicity, suppose that in each country can co-exist at most two companies that produce a specific item. Suppose there are 3 countries (for example, Italy, Brazil and US). In each one of these countries, there is a national company that produces the item made also by the multinational corporation. The multinational company is in competition in these countries against each one of the national company but they have to decide a unique strategy (that is the business plan). The Italian company is not influenced in what happens in US and Brazil: the same applies for the others national companies.

In order to prove that playing a poly sequence-matrix game is equivalent to playing a unique extensive-form game, we need to find an algorithm that maps a representation of game into the other representation type of game in equivalent way. It is what we will do in the continuation of the chapter.

4.2 Algorithms that maps a poly sequence-matrix game into a tree

4.2.1 Finding a poly sequence-matrix games

First of all, we want to be able to obtain the treeplex of player, knowing her strategy constraints (F_i, f_i) from a sequence representation. Here it is presented the algorithm:

Algorithm 1 From Matrix to Treeplex

1: **procedure** FROM MATRIX TO TREEPLEX(F_i, f_i)
2: Check the feasibility of (F_i, f_i)
3: **for all** row of F_i **do**
4: the column-sequence with -1 correspond to the one father-node (call it P)
5: the column-sequence with 1 correspond to the one son-node (call them $S = (S_1, S_2, ...)$)
6: generate sons S from parent P (as in figure 4.2)

Some comments about the algorithm: with feasibility of matrix and vector, we mean the property that F_i and f_i have to have in order to respect a strategy profile $F_i r_i = f_i$. The tree structure of each player has to be in perfect recall in such a way as to be able to built the order of the sequence.

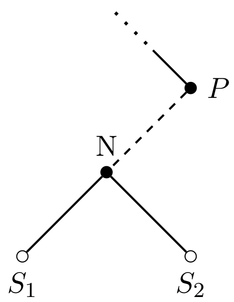

Figure 4.2: Representation of how from a node can be generated the son nodes in a treeplex.

4.2.2 Mapping any poly sequence-matrix game into an extensive-form game

In order to prove that playing a poly sequence-matrix game is equivalent to play a unique extensive-form game, we need to find an algorithm that map a representation of game into the other representation of game. We start from casting a two-player game into an extensive-form game. This initial work will be used then for the poly sequence-matrix games with more than two players.

As input we have a poly sequence-matrix game. We know that a treeplex defines the sequences of each player. We should also have an element called "layout": given two sequences of two different players, "layout" defines the order of information sets. Moreover we will use a Pointer: for each payoff of matrix U, we go through the tree from root to the leaf. The pointer is useful because it is a way to add nodes and actions to the tree. Every movement of the pointer stops on a Nature Node (it is useful in the case a sequence takes to a leaf but it is not necessarily a terminal sequence).

Here the algorithm:

Algorithm 2 From Poly sequence To Extensive Form

```
procedure FROM POLY SEQUENCE TO EXTENSIVE FORM(treeplex_1, treeplex_2, U_12, layout_12)
    T = {}                                              ▷ T is the tree of the extensive game
    N = root node                                       ▷ A Nature Node
    T = T + AddRoot(N)
    for all row i of U_12 do
        for all column j of U_12 do
            if U_12[i, j] ≠ ∅ then
                seq_1 = seq(i)
                seq_2 = seq(j)
                P = path(seq_1, seq_2)                  ▷ layout_12 is needed
                pointer = root of T                     ▷ pointer starts from the root
                for all action a ∈ P do
                    info = information set of action a
                    if pointer can go on node info then
                        Move pointer on action a after info
                    else
                        createinfoset(info, a)
                        Move pointer on action a after info
                    if a is the last action of path P then
                        Add a move of Nature from pointer and put U_12[i, j] as the payoff of the leaf
    Add 0 to the missed leaves
    For each Nature node, if there is only one outgoing move, cut it and unify
    Link all the nodes that belongs to the same information set
    Return T
```

When we said that "pointer can go on node $info$" we mean that a node that belongs to information set $info$ is reachable by the pointer: if yes, the pointer can go on it from its position, if not we have to add the a node that belongs to information set $info$ from the position of the pointer.

Adding 0 to the missed leaves means that the algorithm can generate some uncompleted information sets: thanks to the sparse matrix U, at the end of for-cycle, it is not necessary true that all the information sets in the tree have all the outgoing actions. So we have to add them in order to have a well-defined tree. Since every path has to take to a terminal node, we call these added leaves as *payoff 0*. We say 0 but can be also $-M$ (with M big enough). It is necessary that these terminal nodes are not considered as a solution, because are not-defined nodes.

We provide an example to understand this phenomena. Suppose that player 2 has only one sequence (the empty one), and player 1 has the following treeplex:

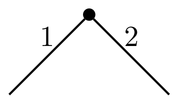

Figure 4.3: Treeplex of player 1.

so the sequences of player 1 are $Q_1 = \{\emptyset, 1, 2\}$. The utility matrix is:

		Player 2		
		$\emptyset$	a	b
Player 1	$\emptyset$	-	x_1	-

Following the algorithm, the output after the for-cycle through the row and columns of U is in Figure 4.4.

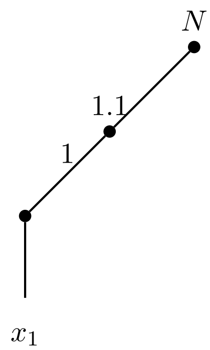

Figure 4.4: Output after the for-cycle

The game tree cannot have this form. In fact, the information set of player 1 is uncompleted and, in order to obtain a valid game, all the information set's actions have to be present at the game. So a payoff 0 has to be add at the game and at move corresponding to action 2.

So the resulting game is shown in Figure 4.5 (and we cut all the redundant nature moves).

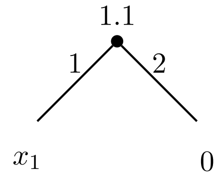

Figure 4.5: Final output of counterexample.

From the previous example, it is clear that, if the utility matrix defines completely a game, not any U can be a representation of a well-defined two-person game in extensive-form.

The U matrix has to have some properties in such a way that all the information sets in the generated tree are complete. Practically, verifying this property is very complex. In fact, for each payoff in the matrix, it has to be verify that one of all the possible combinations of payoff that realize a well-defined game is present on the game. In the previous example, since there is x_1, also a payoff in $(\emptyset, 2)$ has to be present. The example is simple and immediate but if there is a treeplex with more than one level, this condition is really difficult to be checked. In fact, for each payoff in the matrix, every possible combination of other payoffs (in such a way to have a well-defined game) has to be found and it has to be verified that one of them is in the matrix. This is too complex and not useful.

So it is not true that any U of a two-player game and any treeplex represents a game in extensive form. We can construct the tree only if we add a-posteriori the leaves with *payoff 0*. In the following we are going to propose a different solution.

4.2.3 Another point of view

We can try to see the problem from another point of view. In fact, we can try to build the game tree starting from the full payoff matrix: "full" means that every combination of sequences of the two players, correspond to a payoff. Once we get the tree with all the results, we can start to *pruning* the tree in such a way to cut and delete the not-defined parts. The Nature will do the role of generating sub-games so the distribution probability of outgoing moves will be uniform (if there are N outgoing moves, the probability will be $\frac{1}{N}$ for each move).

So we should find an algorithm that firstly will build the tree from a complete matrix, and then a pruning algorithm.

Constructing a tree from a full matrix

We start from constructing a tree with all the possible results. We can see the Nature's move as something that, with a positive probability, activates a sub-game and with 0 probability disables a sub-game.

Algorithm 3 Get Full tree

procedure GET FULL TREE($treeplex_1$, $treeplex_2$,U_{12})
  $T = \{\}$
  N = root node
  $T = T + AddRoot(N)$
  InformationSetDrawn = [ ]
  for all row i of U_{12} **do**
    for all column j of U_{12} **do**
      $seq_1 = seq(i)$
      $seq_2 = seq(j)$
      I = info(seq_1, seq_2) ▷ list of inf.sets met by the two sequences
      if $I \in$ InformationSetDrawn **then**
        Go to the leaf that corresponds to $seq_1 \cup seq_2$
        Call p the value of probability met from the root to ω
        Put as leaf $\frac{1}{p}U_{12}[i, j]$
      else
        From root, draw a Nature's move with probability p_{i_I}
        From that move, draw all the information sets with all the outgoing moves of sequence $seq_1 \cup seq_2$
        Go to the leaf that corresponds to $seq_1 \cup seq_2$
        Put as leaf $\frac{1}{p_{i_I}}U_{12}[i, j]$
        Put I in InformationSetDrawn
  For each Nature node, if there is only one outgoing move, cut it and unify.
  From each Nature node, equalize the outgoing moves with the same probability.
  Link all the nodes that belong to the same information set

Before going into a concrete example, we underline the role of set *InformationSetDrawn*: it is a way to put all together the terminal nodes reachable by the same information sets (but different actions). This is a convenient way of representing sub-games. In the next example, the benefit will be more clear.

Example

Suppose we have two treeplexes as in Figure 4.6.

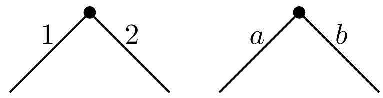

Figure 4.6: Treeplexes of player 1 and player 2.

Each player has a unique information set (1.1 and 2.1). From them, we obtain the sequences: $Q_1 = \{\emptyset, 1, 2\}$ and $Q_2 = \{\emptyset, a, b\}$. We construct the utility matrix: any row corresponds to a sequence of player 1, any column to the one of player 2. The resulting *full* matrix is:

		Player 2		
		$\emptyset$	a	b
Player 1	$\emptyset$	x_0	x_1	x_2
	1	x_3	x_4	x_5
	2	x_6	x_7	x_8

Now, we apply the algorithm:

Step 0 T is the tree, initially empty. N is the root of the game.

Step 1: We start from reading the utility matrix U. In the first row and first column we read x_0. The two correspondent sequences are $(\emptyset, \emptyset)$ and there are no information set needed to reach x_0. So, from node N, a move of Nature is drawn with probability p_1 (Figure 4.7).

Step 2: Now we can go to the next column so we read x_1 and the two correspondent sequences are $(\emptyset, a)$: the information set is 1.1. It is the first time that we meet it. So, from root N we draw a Nature's move (with probability p_2) and from this move we draw all the information sets of $(\emptyset, a)$ (so only 1.1) and we attack x_1 to the correspondent leaf multiplied by $\frac{1}{p_2}$. We add 1.1 in the list $InformationSetDrawn$, useful to have memory of which information sets are present and which not.

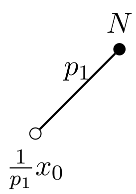

Figure 4.7: Step 1.

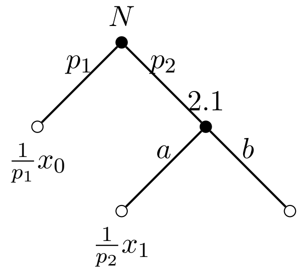

Figure 4.8: Step 2.

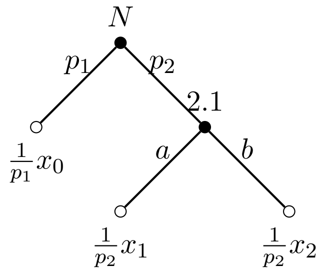

Figure 4.9: Step 3.

From Figure 4.8 you can see that, after action b, there is no payoff: with the following step we will fill that spot.

Step 3: In fact, we can go to the next column and we read x_2: the corresponding sequences are $(\emptyset, b)$ and the information set is 2.1 and we already have it in our tree ($InformationSetDrawn = \{[\emptyset], [2.1]\}$). The probability to go in the single information set 2.1 is p_2. We go in the corresponding leaf and attach the payoff x_2 multiplied by $\frac{1}{p_2}$ (Figure 4.9).

Step 4: all the elements of the first row have been read so we can go to the second one. We read x_3 and the sequence profile is $(1, \emptyset)$. The information set is only 1.1, that is not in the set $InformationSetDrawn$: we draw it from the root and we add x_3 with the corrective factor to the corresponding spot (Figure 4.10).

Step 5: In the following column we read x_4: the sequences are $(1, a)$ and the information sets met to reach the node are 1.1 and 2.1. The set $InformationSetDrawn$ is composed by $\{\emptyset, [2.1], [1.1]\}$: separately, the information sets 1.1 and 2.1 are in the list but not as a couple. So, in order to make the moves of Nature something that active or deactivate sub-games, we drawn from the root N a move of Nature with probability p_4 that take to the sub-game with simultaneous actions

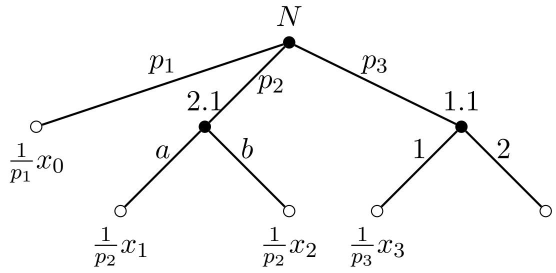

Figure 4.10: Step 4.

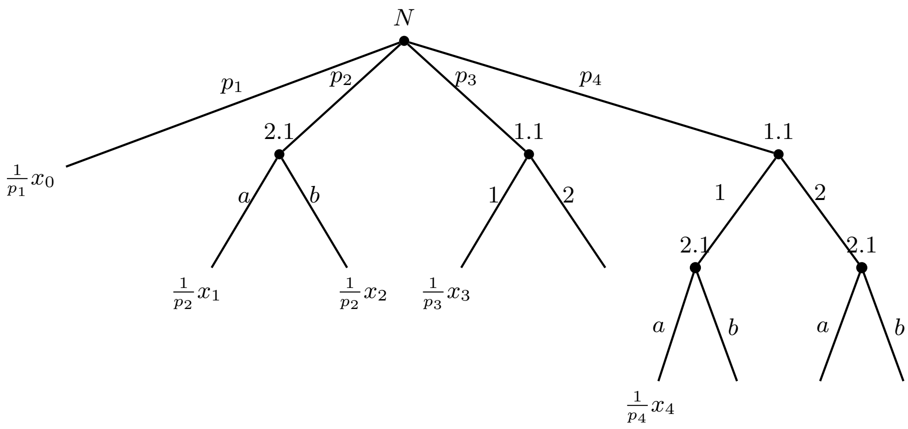

Figure 4.11: Step 5.

with the two information sets already mentioned. Then we fill the spot $(1, a)$ with x_4 (Figure 4.11). Since insert the other payoffs is similar to the previous steps, we go directly to the end of for-cycle.

Final for-cycle step: when we finish to read the utility matrix, the tree is presented as in Figure 4.12.

Final step: The last step is cut the single outgoing moves of Nature (and the only Nature node has four moves so we do nothing) and we connect all the node that belong to the same information set. Moreover, the only Nature node is the root that is connected to four nodes: so $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$. So the final game is in Figure 4.13.

4.2.4 Pruning algorithm

Until now, we see how to built a tree from a full matrix U of two players. In sequence form, we know that not all the combination of sequences between different players are defined, so the U matrix has

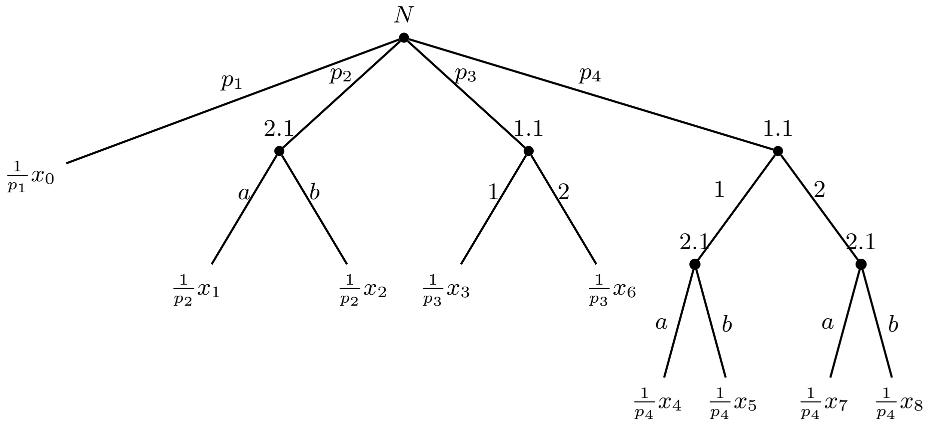

Figure 4.12: Step at the end of the for-cycle.

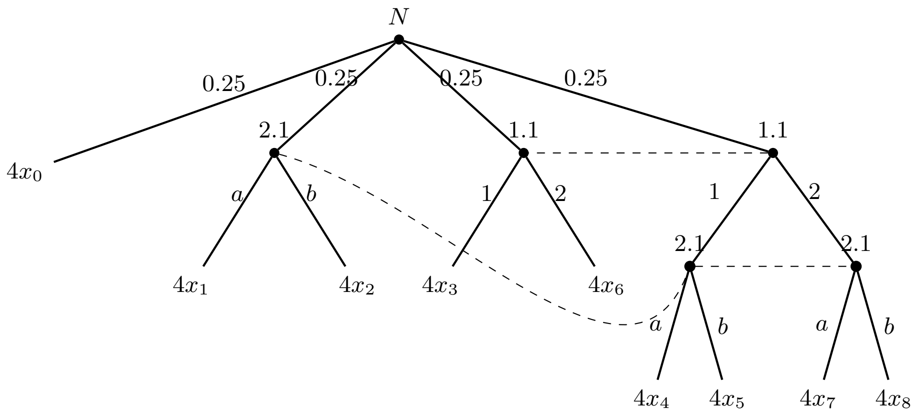

Figure 4.13: Final output.

many empty slot that has to be removed by the game tree. We will do it by a so-called *pruning algorithm*.

We substitute the leaves not-defined with a *payoff 0* and, if a node of nature take only to these types of payoffs, cut it (or put the probability of that move to 0). We remember that the Nature has the role of activate or de-activate the sub-games. Of course, if we cut one of them, all the corrective factors must be fixed.

Before to present the algorithm, we define a useful statistics over node: the *order* of a node is the number of actions played by someone (nature or players) from the root of the game to the node. This is useful in order to have an ordered way to read the tree.

Algorithm 4 Pruning

```
procedure PRUNING(T, U)
    Substitute all the payoffs that are not defined in U with a 0
    Sort the tree nodes in ascending order
    for all node ω ∈ T do
        if From node ω you can reach only payoffs 0 then
            Cut all the outgoing moves and substitute the node with a payoff 0
    End the for-cycle when for each node you can reach at least a payoff different from 0
    Rebalance all the probability of Nature moves
    Rebalance all the payoff with the right corrective factor
    If from a Nature node, there is only one outgoing move, cut it and unify it with the following node.
```

In the following, we provide an example.

Example

We want to apply the "pruning algorithm" to the tree provided in Figure 4.13 and with utility table:

		Player 2		
		$\emptyset$	a	b
Player 1	$\emptyset$	-	-	-
	1	-	x_4	x_5
	2	-	x_7	x_8

Initially, we substitute all the nodes not defined in the table with "payoff 0" (Figure 4.14)

Now we read the tree in order to delete some unnecessary parts of it. The node of information set 2.1 leads with both action to payoff 0 since x_1 and x_2 are not defined in utility matrix. So we can delete the information set and put a 0 (as in Figure 4.15).

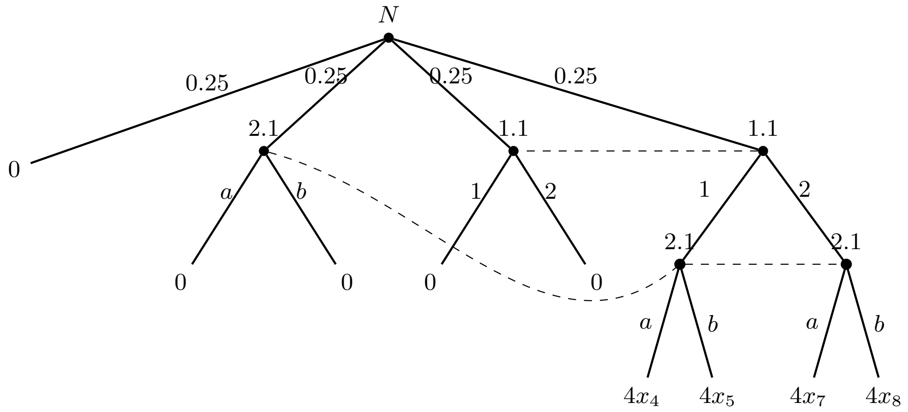

Figure 4.14: All the nodes not-defined in the utility matrix are substituted by payoff 0

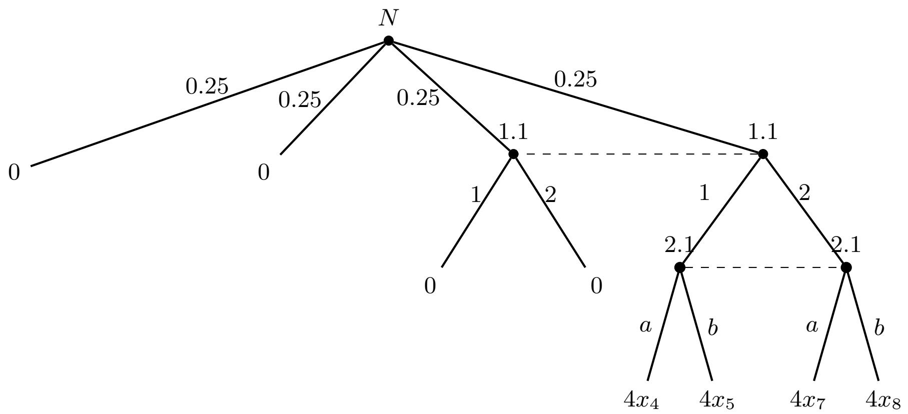

Figure 4.15: The node of information set 2.1 leads only to un-defined payoff so we substitute that node with 0.

The same happens to information set 1.1. At this point, in the tree, every node leads at least to a defined payoff. We can cut the three moves of Nature since they lead only to payoff 0 (Figure 4.16).

Since the Node N has only one move outgoing, we can cut it and there are no more needed to adjust with a corrective factor the payoff in the terminal nodes. The resulting game (Figure 4.17) is the game in normal form:

		Player 2	
		a	b
Player 1	1	x_4	x_5
	2	x_7	x_8

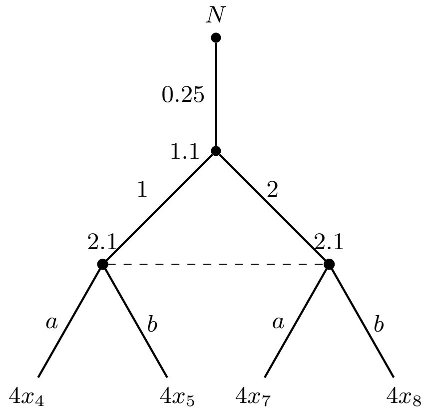

Figure 4.16: The remaining tree after the elimination of the three Nature's moves.

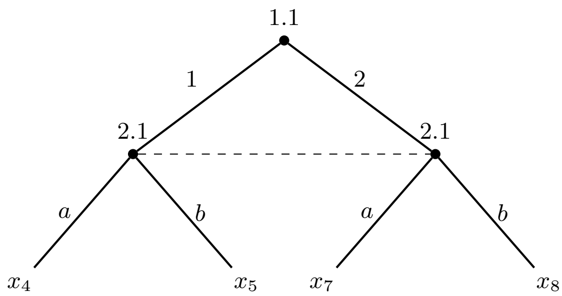

Figure 4.17: Final tree after the execution of Pruning algorithm of the example.

4.2.5 The final algorithms

Now, we have all the issues to obtain a unique game tree (so a game in extensive-form) from a Poly sequence-matrix game with generic n players. First of all, we formalize the version with two players and then we will use it for the n-player version.

Since a Poly sequence-matrix game can be view, like polymatrix class, as a list of two-player games, we start from finding the extensive version of it. The algorithm is the union of the two procedures provided before:

Algorithm 5 Get a two-player Tree

procedure GET A TWO-PLAYER TREE($treeplex_i, treeplex_j, U_{ij}$)
  U is the not-empty spot version of U_{ij}
  T = Get Full Tree($treeplex_i, treeplex_j, U$)
  T = Pruning(T,U_{ij})
  Return T

The goal is to obtain a unique game tree also when the players are more than two. The next algorithm unifies all the two-players trees thanks to a new root of the game, a Nature Node. In this way, the resulting tree preserves the property of nodes (each one is reachable by action of at most two players and Nature) and it is a unique game.

Moreover, each game tree obtained by the treeplexes and utility matrix U_{ij}, has terminal nodes defined for the player i and j and not for the others. Each terminal nodes has to be extended in a way that, for each terminal node, the utility is defined not only for i and j but also for the other players, even if they are not involved in the game. In this way, the resulting game tree will be composed by multiple sub-tree, each one where 2 players interact and have some different utilities depending on their strategy and the others have a constant payoff because they are not involved. In absence of other information, the agents not involved in the sub-tree will receive 0 as utility. So every terminal nodes ω has to be transformed from a $\mathbb{R}^2$ vector into a $\mathbb{R}^{|V|}$, adding a constant for the other players. This is what we do in the next pseudocode:

Algorithm 6 Extend dimension of terminal nodes

```
procedure EXTEND DIMENSION OF TERMINAL NODES(ω_i, i, ω_j, j, N)
    for k = 1, .., N do
        if k == i then
            ω̂_k = ω_i
        else if k == j then
            ω̂_k = ω_j
        else
            ω̂_k = 0
    Return ω̂
```

Now we have all the instruments in order to write an unique game tree:

Algorithm 7 From poly sequence-matrix game to extensive-form game

procedure FROM POLY SEQUENCE-MATRIX GAME TO EXTENSIVE-FORM GAME($\{treeplex\}_{i \in V}, \{U_{ij}\}_{[i,j] \in E}$)
  for all $[i, j] \in E$ **do**
    T_{ij} = Get a two-player tree $[treeplex_i, treeplex_j, U_{ij}]$
    R_{ij} =root of T_{ij}
    for all terminal node $\omega \in T_{ij}$ **do**
      ω = Extend dimension of terminal node $[\omega_i, i, \omega_j, j, |V|]$
  $T = \{\ \}$
  Call N the Nature root node
  Add N to tree T
  Call K the cardinality of set E
  Link the $\{R_{ij}\}_{[i,j] \in E}$ to N through moves of Nature (probability $\frac{1}{K}$)
  Multiply each payoff by a corrective factor of K
  Link the nodes belonging to the same information set
  Return T

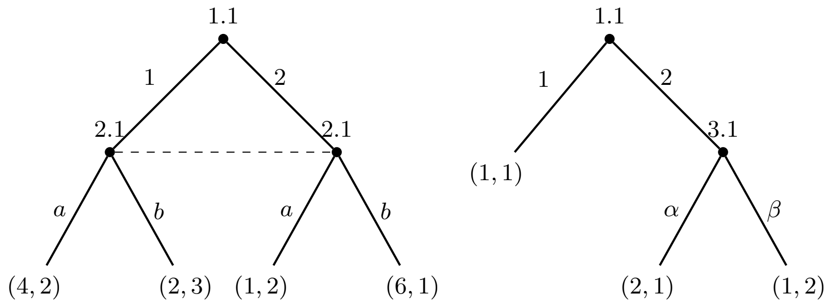

Figure 4.18: Example: on the left the tree T_{12} and on the right the tree T_{23}.

Example

Suppose to have the game trees in Figure 4.18.

In one game tree there is Player 1 that plays against Player 2 (and we denote it with T_{12}) and in the second one Player 1 is against Player 3 (tree T_{13}). If we want to unify them into a unique game of tree, the first step is modified the payoff of terminal nodes into $\mathbb{R}^3$ vectors. In the game where Player 3 does not play, we add a constant (say 0) at the end of the node, while in the second tree we add a constant for the second player at the second slot (Figure 4.19).

Now, going on with the algorithm, we collect them through Nature. We identify the root of the two games as R_{12} and R_{13}. It is clear

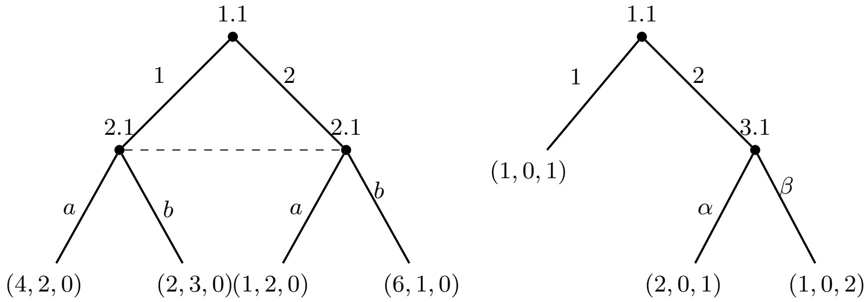

Figure 4.19: We add the player's missed payoff to the terminal nodes.

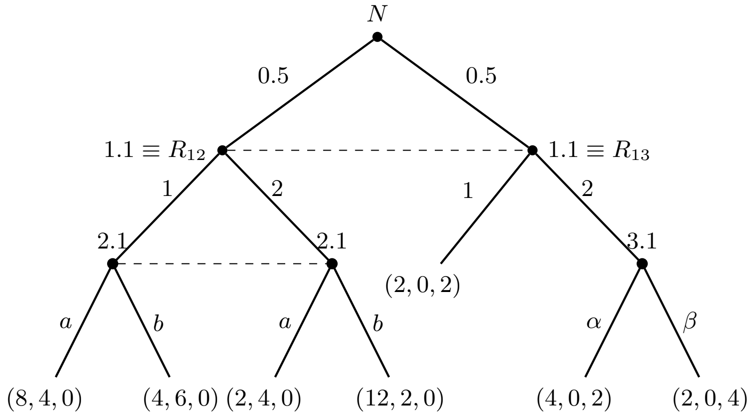

Figure 4.20: The two trees in Figure 4.18 are represented as a unique game tree.

that the set E is composed by $[1, 2]$ and $[1, 3]$ (Player 2 and Player 3 do not play together) and so the cardinality is 2 ($K = 2$). So from root of the game we will start two moves, each one with probability $\frac{1}{K} = 0.5$: one will take to tree T_{12} and one to tree T_{23}. Since we add a probability distribution, we need to adjust the payoffs, multiply them by the inverse of the probability.

The resulting output (Figure 4.20) represents an unique game tree starting from two different two-player games. Each path from the root to the terminal nodes preserves the property in which there are at most actions of two players.

4.3 Properties of Poly sequence-matrix games

In [36] is presented a way to find Nash equilibria with linear programming for zero-sum polymatrix games and some properties around the

Nash equilibria in this class of game. Do these properties hold also for the Poly sequence-matrix games?

4.3.1 Finding Nash Equilibria

First of all we say that a game G in poly sequence matrix form is at **zero-sum** if for all strategy profile $\bar{r} = (r_1, r_2, ..., r_n)$ is such that $\sum_{i \in V} p_i(\bar{r}) = 0$

Then fix a zero-sum poly sequence matrix game. We shall formulate a linear program which captures G.

Since the pairwise game in a poly sequence-matrix game are in sequence form, we start from Best Response problem of player i against strategy $\mathbf{r}_{-i}$ of players $N \setminus \{i\}$:

$$\begin{aligned}
\max_{r_i} \quad & \sum_{q_i \in Q_i} \sum_{\mathbf{q}_{-i} \in Q_i} U_i(q_i, \mathbf{q}_{-i}) \prod_{j \in V} r_j(q_j) \\
\text{s.t.} \quad & \sum_{q_i \in Q_i} F_i(h, q_i) r(q_i) = f_i(h), \qquad \forall h \in H_i \cup \{h_\emptyset\} \\
& r_i(q_i) \geq 0, \qquad \forall q_i \in Q_i
\end{aligned}$$

The variable of the problem is the strategy $\mathbf{r}_i$ of player i. By using strong duality, the same problem can be written in the dual form:

$$\begin{aligned}
\min_{v_i} \quad & \sum_{h_i \in H_i \cup \{h_\emptyset\}} f_i(h_i) v_i(h_i) \\
\text{s.t.} \quad & \sum_{h_i \in H_i \cup \{h_\emptyset\}} F_i(h_i, q_i) v_i(h_i) - \sum_{\mathbf{q}_{-i} \in Q_{-i}} \left[U_i(q_i, \mathbf{q}_{-i}) \prod_{j \in V \setminus \{i\}} r_j(q_j) \right] \geq 0 \quad \forall q_i \in Q_i
\end{aligned}$$

These two problems are referring to player i: we know that the Best response problem of player i against strategy $\mathbf{r}_{-i}$ of players $N \setminus \{i\}$ in sequence form can be formulated by complementarity slackness theorem as:

$$r_i(q_i)\left(\sum_{h_i\in H_i\cup\{h_\emptyset\}} F_i(h_i,q_i)v_i(h_i) - \sum_{\mathbf{q}_{-i}\in Q_{-i}}\left[U_i(q_i,\mathbf{q}_{-i})\prod_{j\in V\setminus\{i\}} r_j(q_j)\right]\right) = 0 \qquad \forall q_i \in Q_i$$

$$\sum_{h_i\in H_i\cup\{h_\emptyset\}} F_i(h_i,q_i)v_i(h_i) - \sum_{\mathbf{q}_{-i}\in Q_i}\left[U_i(q_i,\mathbf{q}_{-i})\prod_{j\in V\setminus\{i\}} r_j(q_j)\right] \geq 0 \qquad \forall q_i \in Q_i$$

$$\sum_{q_i\in Q_i} F_i(h,q_i)r(q_i) = f_i(h) \qquad \forall h \in H_i\cup\{h_\emptyset\}$$

$$r_i(q_i) \geq 0 \qquad \forall q_i \in Q_i$$

The problem of finding a Nash equilibrium in sequence from can be formulated as a mathematical programming problem whose nature is NLCP (Non-Linear Complementary Constraint).

$$r_i(q_i)\left(\sum_{h_i\in H_i\cup\{h_\emptyset\}} F_i(h_i,q_i)v_i(h_i) - \sum_{\mathbf{q}_{-i}\in Q_{-i}}\left[U_i(q_i,\mathbf{q}_{-i})\prod_{j\in V\setminus\{i\}} r_j(q_j)\right]\right) = 0 \qquad \forall i\in N, \forall q_i \in Q_i$$

$$\sum_{h_i\in H_i\cup\{h_\emptyset\}} F_i(h_i,q_i)v_i(h_i) - \sum_{\mathbf{q}_{-i}\in Q_i}\left[U_i(q_i,\mathbf{q}_{-i})\prod_{j\in V\setminus\{i\}} r_j(q_j)\right] \geq 0 \qquad \forall i\in N, \forall q_i \in Q_i$$

$$\sum_{q_i\in Q_i} F_i(h,q_i)r(q_i) = f_i(h) \qquad \forall i\in N, \forall h \in H_i\cup\{h_\emptyset\}$$

$$r_i(q_i) \geq 0 \qquad \forall i\in N, \forall q_i \in Q_i$$

In the case of poly sequence-matrix we can remove the productory since the payoff of each player is the sum of utility of different two-player games ($p_i(\bar{s}) = \sum_{[i,j]\in E} p^{ij}(s_i,s_j) = \sum_{[i,j]\in E} U_{ij}(q_i,q_j)r_i(q_i)r_j(q_j)$). So the problem can be written as follows:

$$r_i(q_i)\left(\sum_{h_i\in H_i\cup\{h_\emptyset\}} F_i(h_i,q_i)v_i(h_i) - \sum_{[ij]\in E}\sum_{q_j\in Q_j} U_i(q_i,q_j)r_j(q_j)\right) = 0 \qquad \forall i\in N, \forall q_i \in Q_i$$

$$\sum_{h_i\in H_i\cup\{h_\emptyset\}} F_i(h_i,q_i)v_i(h_i) - \sum_{[ij]\in E}\sum_{q_j\in Q_j} U_i(q_i,q_j)r_j(q_j) \geq 0 \qquad \forall i\in N, \forall q_i \in Q_i$$

$$\sum_{q_i\in Q_i} F_i(h,q_i)r(q_i) = f_i(h) \qquad \forall i\in N, \forall h \in H_i\cup\{h_\emptyset\}$$

$$r_i(q_i) \geq 0 \qquad \forall i\in N, \forall q_i \in Q_i$$

Looking at the first equation of the problem, fixed player i and sequence q_i, if $r_i(q_i) > 0$ the expression inside the parenthesis has to be equal to 0. So, in a feasible solution, if the sequence q_i can be played with strictly positive probability only if the expected utility given by playing such sequence equals te maximum expected utility that player can get ($\sum_{j \in V \setminus \{i\}} \sum_{q_j \in Q_j} U_i(q_i, q_j) r_j(q_j) = \sum\limits_{h_i \in H_i \cup \{h_\emptyset\}} F_i(h_i, q_i) v_i(h_i)$) so the second equation has to be equal to 0.

4.3.2 Properties

The following question is understand which of the properties of zero-sum two-person games also generalize to zero-sum poly-sequence matrix games. We consider the following properties of zero-sum two-person games:

1. Each player has a unique payoff value in all Nash equilibria, known as her value in the game.

2. Equilibrium strategies are max-min strategies, i.e., each player uses a strategy that maximizes her worst-case payoff (with respect to her opponent's strategies).

3. Equilibrium strategies are exchangeable, i.e., if (x_1, x_2) and (y_1, y_2) are equilibria, then so are (x_1, y_2) and (y_1, x_2). In particular, the set of equilibrium strategies of each player is convex, and the set of equilibria is the corresponding product set.

4. There are no correlated equilibria (or even coarse correlated equilibria, see definition in Preliminaries) whose marginals with respect to the players do not constitute a Nash equilibrium.

Value of a player

We show a counterexample in which two different Nash equilibria take to different value of the game. We start with a Poly sequence-matrix game with three players. Player 1 has no move (so only the $q_\emptyset$ sequence), while from the treeplexes of Player 2 and Player 3 we obtain three sequences: $q_\emptyset$, H and T. The set E contains the couple of player $\{1,2\}$ and $\{2,3\}$ and the payoffs are:

$\{1,2\}$: If Player 2 chooses T receives 1 while Player 1 receives -1, if Player 2 choose H, she receives -1 while Player 1 get -1:

		Player 2		
		∅	H	T
Player 1	∅	-	(1,-1)	(-1,1)

$\{2,3\}$: If Player 2 chooses the same strategy of Player 3, she will receives 1 and the other -1, vice-versa Player 2 receives -1 and the other 1.

		Player 3		
		∅	H	T
Player 2	∅	-	-	-
	H	-	(1,-1)	(-1,1)
	T	-	(-1,1)	(1,-1)

One Nash equilibrium profile is $(q_{\emptyset}, T, H)$ [1] and the value is $(-1, 0, 1)$. First of all, we notice that Player 1 participates at the game with Player 2 without do anything so she has only one strategy available. Of course, Player 2 will play T because in the first game obtains more and Player 3 will choose H. There is no incentive to deviate for Player 3 that has to choose a pure strategy different from player 2. Also Player 2 has no incentive to deviate from her strategy, given the strategy profile of the others players. So, $(q_{\emptyset}, T, H)$ is an equilibrium.

But also $(q_{\emptyset}, \frac{1}{2}T + \frac{1}{2}H, H)$ is a Nash equilibrium with value $(0, 0, 0)$. In fact, Player 2 can mix her strategy and she obtains 0 as before. Player 1 is obliged to have that strategy and Player 3 has no incentive to move from play H.

So different equilibria assign different payoff and the property does not hold.

Max-min strategy

A max-min strategy, in a n-player game (with $n > 2$), is a strategy that maximizes her worst-case payoff for any strategy of the opponents. In the game presented in the previous section, Player 3 max-min strategy is play with the same probability H and T. But we have seen that in the Nash Equilibrium, she plays a different strategy profile and there are no Nash equilibrium where she plays this one. In fact, Player 2 will maximizes her payoff by playing strategy T (also because Player 1 has

[1] With an abuse of notation we write the strategy as in normal form but, since they are sequence games, we should define the vector r, as shown in Preliminaries chapter.

only $r_1(q_\emptyset) = 1$) and so Player 3 can improve her value by playing pure strategy H instead of the maxmin strategy $(\frac{1}{2}T, \frac{1}{2}H)$.

So there can exist equilibria that are not max-min strategy.

Exchangeability

Take as a poly sequence-matrix a game with three players, every one with sequence $q_\emptyset$, H and T, and all players play against all. For each two-player game we have:

	∅	H	T
∅	-	-	-
H	-	(1,-1)	(-1,1)
T	-	(-1,1)	(1,-1)

where the rows are associated to Player 1, Player 2 and Player 3 and the columns to Player 2, Player 3 and Player 1 respectively. Two Nash equilibrium of this game are (H, H, H) and (T, T, T). But if we take (T, H, H) is not more a Nash equilibrium since Player 3 will get -2 and deviating to T can receive more (by getting 2). Hence, the exchangeability property does not hold.

Correlated equilibria

Let Γ be a perfect-recall extensive-form game. In [18] a preposition says that the correlated equilibria follow the inclusion:

$$EFCE \subseteq EFCCE \subseteq NFCCE$$

So every EFCE and EFCCE is also a NFCCE equilibrium. In [36] the following theorem is shown:

Theorem 1. *If z is a coarse correlated equilibrium, then $\hat{x}$ is a Nash equilibrium where, for every player i, $\hat{x}_i$ is the marginal probability distribution $\hat{x}_i^r = \sum_{\bar{s}_{-i} \in Q_{-i}} z^{(r, \bar{s}_{-i})}$ for all $r \in S_i$*

And so the result is extended to the equilibria in a game in extensive-form. Since a poly sequence-matrix game is representable through a game in extensive form, the extensive (coarse) correlated equilibria are Nash equilibria of the game.

Chapter 5

Conclusions and future works

In our work, we study the classes of non cooperative games computationally easy to solve because their resolution is polynomial time in the size of game. First of all, we introduce all the concepts known in the literature that we use. The different representations of a model game, the different kind of equilibria and some of classes of games, like Zero-sum, Constant-sum and strictly competitive games with two players. Motivated to the fact that the two-strictly competitive games have some useful properties as uniqueness of the equilibrium value, interchangeability of the equilibria and the maxmin strategy equilibrium that are Nash equilibrium of the game, we want to extend the number of players. With three players or more, these classes of games lose their property. A candidate class of game is the one of the unilaterally competitive (UC) games. This class of games is interesting because to every unilateral change of strategy of one player, if she gets more in terms of utility, all the others get less. We study the UC games for two players and we find a characterization of these. Since the characterization can be computationally difficult to be verified because the derivates are involved, we try to see if, in some intuitive way, the UC games could be recognize. When we study the case of three players we realize that the UC games are really difficult to find and to construct. Moreover the *competitive* property is very restrictive when mixed strategies are involved. We look for another class of games with a generic number of players. We formalize the concept of Poly sequence-matrix games and see if this class of game can be represented as a unique game in extensive-form. We provide an algorithm to make one representation into the other. Motived to the results in [36] for the Zero-sum Polymatrix games, we analyse if all the properties that hold for the equilibria of a zero-sum Polymatrix game also hold for the equilibria of zero-sum Poly sequence-matrix game.

At the end of this thesis there are still some open questions.

In our work, we study the characterization of Unilaterally competitive games with two players. We test only easy cases and we do not talk about how much is computationally difficult the derivatives' computation. Maybe it can be develop some Matlab program in order to have a quantity way of computing.

In the poly sequence-matrix games, we suppose to have games with perfect recall because the sequence-form representation is defined only for games with perfect recall. We do not discuss about the case of games with imperfect recall and if there is a possible representation for this class of games.

Another thing that can be done in the future is a generalization of the extensive-form into a poly sequence-matrix game. We have not study a way to represent an extensive form game into a poly sequence-matrix game and so it can be interesting see if, given some properties of the tree, there can be a way to represent it into a poly sequence-matrix game.

Bibliography

[1] Bo An, Manish Jain, Milind Tambe, and Christopher Kiekintveld. Mixed-initiative optimization in security games: A preliminary report. In *Help Me Help You: Bridging the Gaps in Human-Agent Collaboration, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-05, Stanford, California, USA, March 21-23, 2011*, 2011.

[2] Bo An, Nicola Gatti, and Victor R. Lesser. Bilateral bargaining with one-sided uncertain reserve prices. *Auton. Agents Multi Agent Syst.*, 26(3):420–455, 2013.

[3] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artif. Intell.*, 184-185:78–123, 2012.

[4] Nicola Basilico, Andrea Celli, Giuseppe De Nittis, and Nicola Gatti. Team-maxmin equilibrium: Efficiency bounds and algorithms. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 356–362. AAAI Press, 2017.

[5] Nicola Basilico, Giuseppe De Nittis, and Nicola Gatti. Adversarial patrolling with spatially uncertain alarm signals. *Artif. Intell.*, 246: 220–257, 2017.

[6] Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Be a leader or become a follower: The strategy to commit to with multiple leaders. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence,*

IJCAI 2019, Macao, China, August 10-16, 2019, pages 123–129. ijcai.org, 2019.

[7] Matteo Castiglioni, Alberto Marchesi, Nicola Gatti, and Stefano Coniglio. Leadership in singleton congestion games: What is hard and what is easy. *Artif. Intell.*, 277, 2019.

[8] Andrea Celli and Nicola Gatti. Computational results for extensive-form adversarial team games. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 965–972. AAAI Press, 2018.

[9] Andrea Celli, Alberto Marchesi, and Nicola Gatti. On the complexity of nash equilibrium reoptimization. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[10] Andrea Celli, Stefano Coniglio, and Nicola Gatti. Computing optimal ex ante correlated equilibria in two-player sequential games. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 909–917. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[11] Andrea Celli, Stefano Coniglio, and Nicola Gatti. Computing optimal coarse correlated equilibria in sequential games. *arXiv preprint arXiv:1901.06221*, 2019.

[12] Andrea Celli, Alberto Marchesi, Tommaso Bianchi, and Nicola Gatti. Learning to correlate in multi-player general-sum sequential games. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019,*

NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pages 13055–13065, 2019.

[13] Andrea Celli, Giulia Romano, and Nicola Gatti. Personality-based representations of imperfect-recall games. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 1868–1870. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[14] Andrea Celli, Alberto Marchesi, Gabriele Farina, and Nicola Gatti. No-regret learning dynamics for extensive-form correlated and coarse correlated equilibria. *CoRR*, abs/2004.00603, 2020.

[15] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272. IEEE, 2006.

[16] Vincent Conitzer and Tuomas Sandholm. Expressive negotiation in settings with externalities. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 255–260, 2005.

[17] Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Ex ante coordination and collusion in zero-sum multi-player extensive-form games. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 9661–9671, 2018.

[18] Gabriele Farina, Tommaso Bianchi, and Tuomas Sandholm. Coarse correlation in extensive-form games. *arXiv preprint*, 2019.

[19] Nicola Gatti. Extending the alternating-offers protocol in the presence of competition: models and theoretical analysis. *Ann. Math. Artif. Intell.*, 55(3-4):189–236, 2009.

[20] Nicola Gatti, Giorgio Patrini, Marco Rocco, and Tuomas Sandholm. Combining local search techniques and path following for bimatrix games. In Nando de Freitas and Kevin P. Murphy, editors, *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pages 286–295. AUAI Press, 2012.

[21] Nicola Gatti, Marco Rocco, and Tuomas Sandholm. On the verification and computation of strong nash equilibrium. In Maria L. Gini, Onn Shehory, Takayuki Ito, and Catholijn M. Jonker, editors, *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pages 723–730. IFAAMAS, 2013.

[22] C. Daskalaskis I.Adler and C. Papadimitriou. A note on strictly competitive game. *Springer-Verlag Berlin Heidelberg*, page 472, 2009.

[23] A. Kats and J-F Thisse. Unilaterally competitive games. *International Journal of Game Theory*, 21(3):291–299, 1992.

[24] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1*, pages 689–696, 2009.

[25] Daphne Koller, Nimrod Megiddo, and Bernhard Von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and economic behavior*, 14(2):247–259, 1996.

[26] Samson Lasaulce and Hamidou Tembine. Chapter 9 - medium access control games. In *Game Theory and Learning for Wireless Networks*, pages 245 – 273. Academic Press, Oxford, 2011.

[27] R. Lucchetti. *A Primer in Game Theory.* Società Editrice Esculapio, 2011.

[28] Ilaria Malanchini, Matteo Cesana, and Nicola Gatti. Network selection and resource allocation games for wireless access networks. *IEEE Trans. Mob. Comput.*, 12(12):2427–2440, 2013.

[29] A. Gilpin S. Hoda and J. Peña. Smoothing techiques for computing nash equilibria of sequential games. *Mathematics of Operations Research*, page 497, May 2010.

[30] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[31] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: Looking beyond a decade of success. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 5494–5501, 2018.

[32] Bernhard Von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

[33] Bernhard Von Stengel. Computing equilibria for two-person games. *Handbook of game theory with economic applications*, 3: 1723–1759, 2002.

[34] Bernhard Von Stengel and Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

[35] O. De Wolf. Optimal strategies in n-person unilaterally competitive games. Technical report, Université Catholique de Louvain. Center for Operations Research, 1999.

[36] C. Daskalakis Y. Cai, O. Candogan and C. Papadimitriou. Zero-sum polymatrix games: A generalization of minmax. *Institute for Operations Research and the Management Sciences (INFORMS)*, 2016.