# EC Questions

Francesco Corda

September 7, 2020

# Contents

# Chapter 1

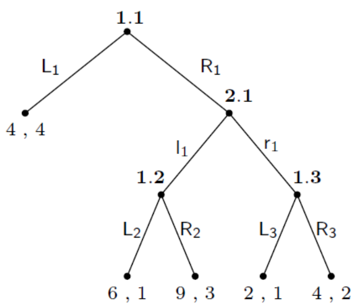
# EC1

## 1.1 EC 1.1

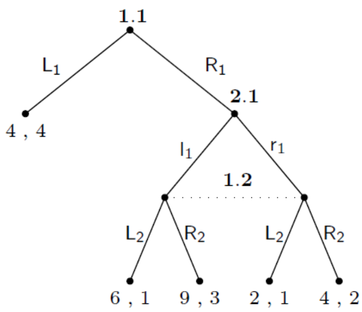
**Exercise 1.0.1** (Information set). *What is an information set?*

An information set $h$ of player $i$ is a subset of $V_i$ such that, for all $w, w' \in h$ the property $\rho(w) = \rho(w')$ holds (with an abuse of notation, we denote $\rho(h)$ as $\rho(w)$ with $w \in h$).

**Exercise 1.0.2** (Perfect vs. imperfect information). *Report an example of 2-player game in extensive-form representation with perfect information and an example of 2-player game with imperfect information.*




(a) With perfect information




(b) Without perfect information

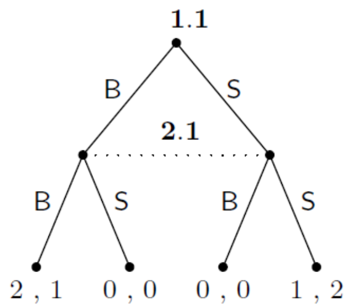
**Exercise 1.0.3** (Perfect recall). *Report the definition of perfect-recall games in terms of constraints over the information sets of the players.*

Player $i$ has perfect recall in an imperfect-information extensive-form game if for any two decision nodes $w, w'$ that are in the same information set $h$ for player $i$, for any path $\langle w_0, a_0, w_1, a_1, w_2, \ldots, w_k, a_k, w \rangle$ from the root $\bar{w}_0$ of the game to $w$ (where $w_j$ are decision nodes of player $i$ and $a_j$ are actions played at $w_j$ by player $i$) and for any path $\langle w'_0, a'_0, w'_1, a'_1, w'_2, \ldots, w'_l, a'_l, w' \rangle$ from the root of the game to $w'$ it must be the case that:

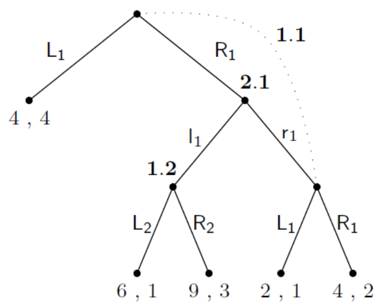
- $k = l$;
- for all $0 \leqslant j \leqslant k, w_j$ and $w'_j$ are in the same information set for player $i$;
- for all $0 \leqslant j \leqslant k$, it holds $a_j = a'_j$;

A game is with perfect recall if every player has perfect recall in it.

**Exercise 1.0.4** (Perfect vs. imperfect recall). *Report an example of 2-player game in extensive-form representation with perfect recall and an example of 2-player game with imperfect recall.*




(a) With perfect recall




(b) Without imperfect recall

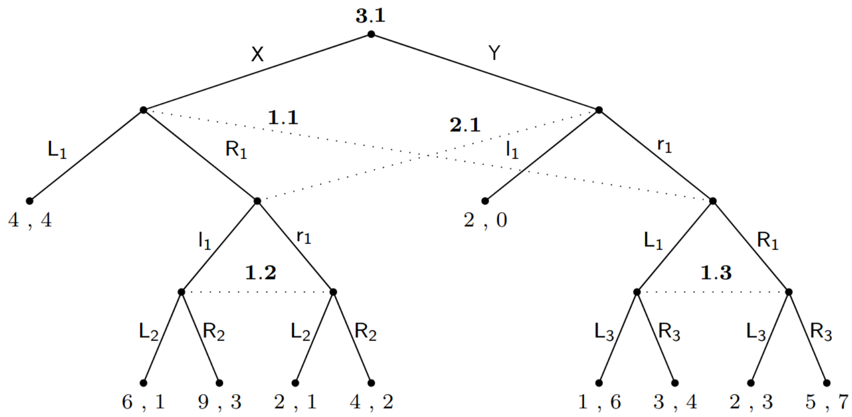
**Exercise 1.0.5** (Timeability). *Report the definition of timeable extensive-form game.*

Given an extensive-form game, the information sets are chronologically ordered when there is an assignment of labels $l \in \mathbb{N}$ to the decision nodes such that:

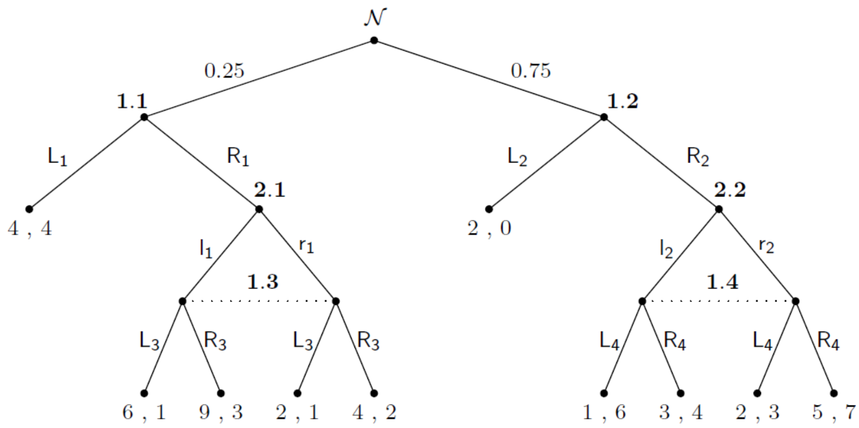
- each decision node has a label strictly larger than the label of the parent;
- all the decision nodes of the same information set have the same label.

The labels have a simple interpretation, being the time at which the information set is traversed.
A game is timeable, in the sense that all the players have the sense of time, if and only if all the information sets are chronologically ordered.

**Exercise 1.0.6** (Timeable vs. non-timeable). *Report an example of 3-player game in extensive-form representation with perfect recall in which some player has not the sense of time.*




**Exercise 1.0.7** (Game with Nature). *Report an example of 2-player game in extensive-form representation with Nature.*

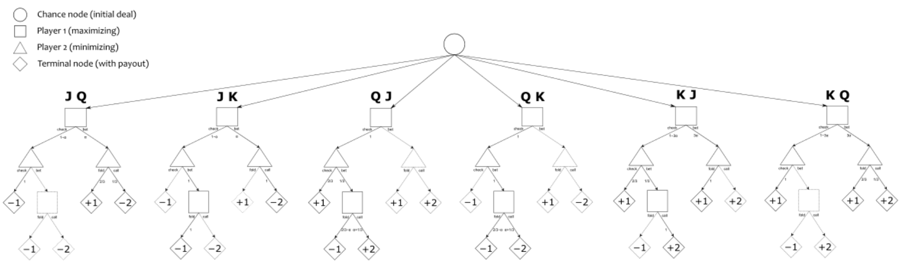



**Exercise 1.0.8** (Kuhn Poker). *Consider the Kuhn Poker game:*

- *there are two players {1, 2};*
- *there are three cards {K, Q, J};*
- *each player antes 1;*
- *each player is dealt one of the three cards, and the third is put aside unseen (no player can observe the cad of the opponent);*
- *player 1 can check or bet 1;*
  - *if player 1 checks then player 2 can check or bet 1;*
    - *if player 2 checks there is a showdown for the pot of 2;*
    - *if player 2 bets then player 1 can fold or call;*
      - *if player 1 folds then player 2 takes the pot of 3;*

- · *if player 1 calls there is a showdown for the pot of 4;*
  - *if player 1 bets then player 2 can fold or call;*
    - *if player 2 folds then player 1 takes the pot of 3;*
    - *if player 2 calls there is a showdown for the pot of 4;*
- *in the showdown, the player with the highest card wins the pot entirely.*

*Provide the extensive-form representation of the game.*




**Exercise 1.0.9** (Simplified bargaining game). *Consider the following simplified bargaining game:*

- *there are two players, one buyer b and one seller s;*
- *the seller player is of two types: $\theta_{s.1}$ with a probability of 0.33 and $\theta_{s.2}$ with a probability of 0.67;*
- *the first player can offer either 0.33 or 0.66 and the action is perfectly observable;*
- *the second player can accept the offer x of the first player, counteroffer $x \pm 0.20$, and this action is perfectly observable;*
  - *if the second player counteroffers, then the first player can accept the offer x of the second player, counteroffer $x \pm 0.10$, and this action is perfectly observable;*
    - *if the first player counteroffers, then the second player can accept or reject;*
      - · *if the second player accepts, the game concludes with an agreement over the last offer x;*
      - · *if the second player rejects, the game concludes with a disagreement;*
    - *if the first player accepts, the game concludes with an agreement over the last offer x;*
  - *if the second player accepts, then the game concludes with an agreement over the last offer x;*
- *the utility of all the players from a disagreement is 0, while the utility of buyer from an agreement (x, t) where t is the time at which the agreement is achieved, is $(1 - x)(\delta_b)^t$, while the utility of seller s.i is $(\delta_{s.i})^t$. Assume that $\delta_b = 0.5, \delta_{s.1} = 0.4, \delta_{s.2} = 0.9$. Each action requires a unitary temporal cost.*

*Provide the extensive-form representation of the game in both cases the first player is the seller (and the second is the buyer) and the first player is the buyer (and the second is he seller).*

TODO

**Exercise 1.0.10** (Simplified patrolling game). *Consider the following simplified patrolling game:*

- *there are two players, one attacker* $a$ *and one defender* $d$*;*
- *there is a fully connected graph with four vertices labeled* $\{v_1, v_2, v_3, v_4\}$*;*
- *the defender is initially at* $v_1$ *and spends one time point to move from any vertex to any another vertex;*
- *at time 1, the attacker decides which vertex to attack among* $\{v_2, v_3, v_4\}$ *and moves to it, simultaneously the defender decides the next vertex to visit among* $\{v_2, v_3, v_4\}$*;*
- *at time 2, the defender covers the vertex which it got and decides the next vertex to visit among all the vertices not visited yet;*
- *at time 3, the defender covers he vertex which it got;*
- *if the defender covers the vertex under attack, then the utility of the defender is 1, while the utility of the attacker is 0, else the utility of the defender is* $1 - \pi(v_i)$ *where* $v_i$ *is the attacked target, while the utility of the attacker is* $\pi(v_i)$*.*

*Provide the extensive-form representation of the game.*

TODO

## 1.2 EC 1.2

**Exercise 2.0.1** (Normal-form representation of an extensive-form game). *Provide the definition of the normal-form representation of an extensive-form game (even in the case games with Nature).*

Given an extensive-form game $(N, A, V, T, \iota, \rho, \chi, U, H)$, the corresponding normal-form representation is a triplet $(N, P, U')$, in which:

- $P = \{P_1, P_2, \ldots, P_n\}$ is the set of actions (said plans) of the players and each plan $p \in P_i$ is a tuple specifying one action $a \in A_i$ per information set $h \in H_i$ of player $i$ such that $a \in \rho(h)$
- $U' = \{U_1, U_2, \ldots, U_n\}$ is the set of utility functions of the players with $U'_i : P_1 \times P_2 \times \ldots P_n \to \mathbb{R}$ such that $U'_i(p_1, p_2, \ldots, p_n) = U_i(w)$ where $w$ is the terminal node reached by applying plan profile $(p_1, p_2, \ldots, p_n)$.

**Exercise 2.0.2** (Normal-form size). *Given an extensive-form game with 2 players, h information sets per player and 2 actions available to each player at each information set, what is, in the worst case, the asymptotical size of the normal-form representation in h?*

The normal-form representation of an extensive-form game may have exponential size in the size of the game tree, where the size of the normal-form representation is $|P_1||P_2|\dots|P_n|$ and the size of the game tree is $|T|$. The size of the normal-form representation, in terms of $h$, is $|P_1||P_2| = 2^h \cdot 2^h = 2^{2h}$.

**Exercise 2.0.3** (Strategy and strategy profile). *Provide the definition of strategy and strategy profile for a game in normal-form representation.*

A normal-form strategy $\sigma_i : P_i \to [0, 1]$ with $\sigma_i \in \Delta(P_i)$ is a function returning the probability with which each plan $p_i \in P_i$ is played by player i (we denote with $\Delta(\cdot)$ the simplex over $\cdot$ ).
Strategy profile $\boldsymbol{\sigma}$ is a tuple $(\sigma_1, \sigma_2, \dots, \sigma_n)$, containing one strategy per player.
Strategy profile $\boldsymbol{\sigma}_{-i}$ is a tuple $(\sigma_1, \sigma_2, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_n)$, containing one strategy per player except for player $i$.

**Exercise 2.0.4** (Reduced normal-form representation of an extensive-form game). *Provide the definition of the reduced normal-form representation of an extensive-form game.*

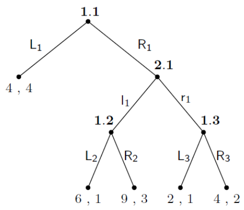
Given the normal-form representation with plans $P_1, P_2, \dots, P_n$ of an extensive-form game, the reduced normal-form representation is composed of a subset of plans $P'_1, P'_2, \dots, P'_n$ with $P'_i \subseteq P_i$ such that:

- no $p_i, p'_i \in P'_i$ with $p_i \neq p'_i$ are realization equivalent;
- every $p_i \in P_i \backslash P'_i$ is realization equivalent to some $p'_i \in P'_i$

**Exercise 2.0.5** (Reduced normal-form size). *Given an extensive-form game with 2 players, h information sets per player and 2 actions available to each player at each information set, what is, in the worst case, the asymptotical size of the reduced normal-form representation in h?*

The reduced normal-form representation of an extensive-form game may have exponential size in the size of the game tree, where the size of the normal-form representation is $|P'_1||P'_2|\dots|P'_n|$ and the size of the game tree is $|T|$. The size of the normal-form representation, in terms of $h$, is $|P'_1||P'_2| = 2^h \cdot 2^h = 2^{2h}$.

**Exercise 2.0.6** (Translation). *Given an extensive-form game (even with Nature), provide the corresponding normal-form representation and the corresponding reduced normal-form representation.*




(a) Extensive-form

| 1 \ 2 | $\mathsf{l}_1$ | $\mathsf{r}_1$ |
|---|---|---|
| $\mathsf{L}_1\mathsf{L}_2\mathsf{L}_3$ | $4,4$ | $4,4$ |
| $\mathsf{L}_1\mathsf{L}_2\mathsf{R}_3$ | $4,4$ | $4,4$ |
| $\mathsf{L}_1\mathsf{R}_2\mathsf{L}_3$ | $4,4$ | $4,4$ |
| $\mathsf{L}_1\mathsf{R}_2\mathsf{L}_3$ | $4,4$ | $4,4$ |
| $\mathsf{R}_1\mathsf{L}_2\mathsf{L}_3$ | $6,1$ | $2,1$ |
| $\mathsf{R}_1\mathsf{L}_2\mathsf{R}_3$ | $6,1$ | $4,2$ |
| $\mathsf{R}_1\mathsf{R}_2\mathsf{L}_3$ | $9,3$ | $2,1$ |
| $\mathsf{R}_1\mathsf{R}_2\mathsf{R}_3$ | $9,3$ | $4,2$ |

(b) Normal-form

| 1 \ 2 | $\mathsf{l}_1$ | $\mathsf{r}_1$ |
|---|---|---|
| $\mathsf{L}_1 * *$ | $4,4$ | $4,4$ |
| $\mathsf{R}_1\mathsf{L}_2\mathsf{L}_3$ | $6,1$ | $2,1$ |
| $\mathsf{R}_1\mathsf{L}_2\mathsf{R}_3$ | $6,1$ | $4,2$ |
| $\mathsf{R}_1\mathsf{R}_2\mathsf{L}_3$ | $9,3$ | $2,1$ |
| $\mathsf{R}_1\mathsf{R}_2\mathsf{R}_3$ | $9,3$ | $4,2$ |

(c) Reduced normal-form

**Exercise 2.0.7** (Expected utility). *Given a game in normal-form with n players, provide the formula of the expected utility for a player i.*

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

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

The degree of the polynomial is $n$ and, given player $i$, the expected utility is linear in player $i$'s strategy.

## 1.3 EC 1.4

**Exercise 4.0.1** (Sequence-form representation of an extensive-form game). *Provide the definition of the sequence-form representation of an extensive-form game (even in the case games with Nature).*

Given an extensive-form 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 agents;
- $Q = \{Q_1, Q_2, \ldots, Q_n\}$ is the set of sequences of all the players and $Q_i$ is the set of sequences of player $i$
- $U' = \{U'_1, U'_2, \ldots, U'_n\}$ is the set of utility functions of all the players where $U'_i : Q_1 \times Q_2 \times \ldots \times Q_n \rightarrow \mathbb{R}$ returns the utility $U_i(w)$ of the terminal node $w$ reached by a profile of terminal sequences, while it is not defined if the profile of sequences contains at least a non-terminal sequence;
- $C = \{(F_1, f_1), (F_2, f_2), \ldots, (F_n, f_n)\}$ is the set of the constraints over the sequence-form strategies of all the players.

**Exercise 4.0.2** (Sequence-form size). *Given an extensive-form game with 2 players, h information sets per player and 2 actions available to each player at* *each information set, what is, in the worst case, the asymptotical size of the sequence-form representation in h?*

The sequence-form representation has a size linear in the size of the game tree, containing a number of sequences that is $O(|V|)$. The size of the normal-form representation, in terms of $h$, is $O(2h)$.

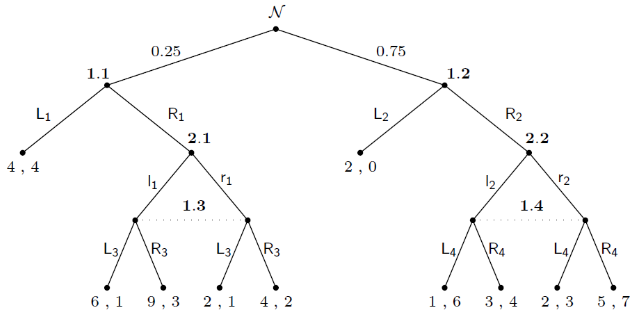
**Exercise 4.0.3** (Strategy and strategy profile). *Provide the definition of strategy and strategy profile for a game in sequence-form representation.*

A sequence-form strategy (said realization plan) $r_i : Q_i \to [0,1]$, with the constraints that $r_i(q_\varnothing) = 1$ and that $r_i(q) = \sum_{a\in\rho(h)} \mathsf{extend}(q,a)$ for each $h \in \mathsf{lead}(q)$ and for every $q \in Q_i$ that is not terminal, is a function returning the probability with which each sequence $q \in Q_i$ is played by player $i$.
Differently from strategies in the normal-form representation, in the sequence-form representation, strategies may not satisfy the condition that $\sum_{g\in Q_t} r_i(q) = 1$ (that is, a realization plan may be not a probability distribution). Indeed, the sequence-form representation requires different constraints.
The constraints over the strategies are linear in the strategies and, by using matrix-based notation, such constraints can be formulated as $F_i r_i = f_i$, where $F_i$ is a matrix $\mathcal{M}$ with size $(|H_i|+1) \times |Q_i|$ , $r_i$ is here intended as a column vector, and $f_i$ is a column vector of $|H_i|+1$ positions.

Strategy profile $\mathbf{r}$ is a tuple $(r_1, r_2, \ldots, r_n)$ , containing one strategy per player. Strategy profile $\mathbf{r}_{-i}$ is a tuple $(r_1, r_2, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n)$ , containing one strategy per player except for player $i$.

**Exercise 4.0.4** (Translation). *Given an extensive-form game (even with Nature), provide the corresponding sequence-form representation.*

(Sequence-form representation with Nature). An example of extensive-form game with Nature and its sequence-form representation is reported below (we use symbol '×' to denote the cases in which the utility functions are not defined).




$$U_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ 1.00 & \times & \times & \times & \times \\ 1.50 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & 1.50 & 0.50 & \times & \times \\ \times & 2.25 & 0.75 & \times & \times \\ \times & \times & \times & 0.75 & 1.50 \\ \times & \times & \times & 2.25 & 3.75 \end{bmatrix}, \quad U_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ 1.00 & \times & \times & \times & \times \\ 0.00 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & 0.25 & 1.50 & \times & \times \\ \times & 0.75 & 1.00 & \times & \times \\ \times & \times & \times & 4.50 & 2.25 \\ \times & \times & \times & 3.00 & 5.25 \end{bmatrix}.$$

Notice that the sequence-form representation has a size, in terms of variables and constraints, that is linear in the size, in terms of terminal nodes, of the game tree.

**Exercise 4.0.5** (Expected utility). *Given a game in sequence-form with $n$ players, provide the formula of the expected utility for a player $i$.*

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

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

once assigned $U_i(\mathbf{q}) = 0$ for all the profile $\mathbf{q}$ containing at least one non-terminal sequence. The degree of the polynomial is $n$ and, given player $i$, the expected utility is linear in player $i$'s strategy.

## 1.4 EC 1.5

**Exercise 5.0.1** (Kuhn theorem). *Provide the statement of the Kuhn theorem.*

Given every game with perfect recall and every normal-form strategy, there always exists at least a realization-equivalent agent-form strategy and vice versa.

## 1.5 EC 1.6

**Exercise 6.0.1** (Realization equivalence). *Provide the conditions under which two strategies (even defined over different representations of the same game) are realization equivalent.*

(Realization equivalence). Given an extensive-form game, two strategies, even defined on different representations, are realization equivalent if, for every strategy of the opponents, they induce the same probability distribution over the outcomes of the game.

(Sequence/normal-form strategies realization equivalence). Given a game and a player $i$, a sequence-form strategy $r_i$ and a normal-form strategy $\sigma_i$ are realization equivalent if and only if the following property holds for every $q \in Q_i$:

$$r_i(q) = \sum_{p \in P_i : \mathsf{action}(q) \in p} \sigma_i(p)$$

The above relation provides a direct way to find a sequence-form strategy given a normal-form strategy. The reverse is more involved, since every sequence-form strategy may be realization equivalent to many normal-form strategies and therefore it is necessary the resolution of a linear equation system.

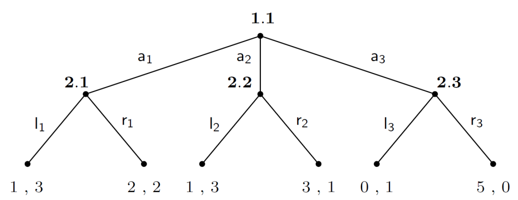
(Normal-form strategy derivation from a sequence-form strategy). Given a game and a sequence-form strategy $r_i$, the reduced normal-form strategy $\sigma_i$ defined as

$$\sigma_i(p) = \prod_{q \in \bar{Q}_i : \mathsf{action}(q) \in p} r_i(q)$$

for every $p \in P_i$, is realization equivalent to $r_i^1$.

**Exercise 6.0.2** (From sequence-form to normal-form). *Given an extensive-form game, its normal-form and sequence-form representations, and a strategy profile defined over the sequence-form, provide a strategy defined over the normal-form that is realization equivalent. Is the latter unique?*

Consider the following game in extensive form




its corresponding normal-form representation (the reduced normal form is the same):

2

| 1 | | $l_1l_2l_3$ | $l_1l_2r_3$ | $l_1r_2l_3$ | $l_1r_2r_3$ | $r_1l_2l_3$ | $r_1l_2r_3$ | $r_1r_2l_3$ | $r_1r_2r_3$ |
|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | 1, 3 | 1, 3 | 1, 3 | 1, 3 | 2, 2 | 2, 2 | 2, 2 | 2, 2 |
| | $a_2$ | 1, 3 | 3, 1 | 3, 1 | 3, 1 | 1, 3 | 1, 3 | 3, 1 | 3, 1 |
| | $a_3$ | 0, 1 | 5, 0 | 0, 1 | 5, 0 | 0, 1 | 5, 0 | 0, 1 | 5, 0 |

6-1

and its corresponding sequence-form representation:

2

| 1 | | $l_1$ | $r_1$ | $l_2$ | $r_2$ | $l_3$ | $r_3$ |
|---|---|---|---|---|---|---|---|
| | $a_1$ | 1, 3 | 2, 2 | ×, × | ×, × | ×, × | ×, × |
| | $a_2$ | ×, × | ×, × | 1, 3 | 3, 1 | ×, × | ×, × |
| | $a_3$ | ×, × | ×, × | ×, × | ×, × | 0, 1 | 5, 0 |

(Normal-form strategy derivation from a sequence-form strategy). Consider the following sequence-form strategy:

$$r_2 = \begin{bmatrix} 1.00 \\ 0.50 \\ 0.50 \\ 0.30 \\ 0.70 \\ 0.25 \\ 0.75 \end{bmatrix}$$

In the realization-equivalent strategy $\sigma_2$, we have, e.g., plan $p = \mathsf{l}_1\mathsf{l}_2\mathsf{l}_3$ played with probability $\sigma_2(\mathsf{l}_1\mathsf{l}_2\mathsf{l}_3) = 0.50 \cdot 0.30 \cdot 0.25$. This derivation does not work for the non-reduced normal form and is not unique.

**Exercise 6.0.3** (From normal-form to sequence-form). *Given an extensive-form game, its normal-form and sequence-form representations, and a strategy profile defined over the normal-form, provide a strategy defined over the sequence-form that is realization equivalent. Is the latter unique?*

(Sequence/normal-form strategies realization equivalence). Consider the example above. Strategies $r_2$ and $\sigma_2$ are realization equivalent if the following equations are satisfied:

$$\begin{aligned}
r_2(\mathrm{l}_1) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{l}_3) &+ \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{r}_3) &+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{l}_3) &+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{r}_3)\\
r_2(r_1) &= \sigma_2(r_1l_2\mathrm{l}_3) &+ \sigma_2(r_1l_2r_3) &+ \sigma_2(r_1r_2\mathrm{l}_3) &+ \sigma_2(r_1r_2r_3)\\
r_2(\mathrm{l}_2) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{l}_3) &+ \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{r}_3) &+ \sigma_2(r_1\mathrm{l}_2\mathrm{l}_3) &+ \sigma_2(r_1\mathrm{l}_2r_3)\\
r_2(r_2) &= \sigma_2(\mathrm{l}_1r_2\mathrm{l}_3) &+ \sigma_2(\mathrm{l}_1r_2r_3) &+ \sigma_2(r_1r_2\mathrm{l}_3) &+ \sigma_2(r_1r_2r_3)\\
r_2(\mathrm{l}_3) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{l}_3) &+ \sigma_2(\mathrm{r}_1\mathrm{l}_2\mathrm{l}_3) &+ \sigma_2(\mathrm{r}_1\mathrm{r}_2\mathrm{l}_3) &+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{l}_3)\\
r_2(\mathrm{r}_3) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{r}_3) &+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{r}_3) &+ \sigma_2(\mathrm{r}_1\mathrm{r}_2\mathrm{r}_3) &+ \sigma_2(\mathrm{r}_1\mathrm{l}_2\mathrm{r}_3)
\end{aligned}$$

Notice that $r_2(q) = \frac{1}{2}$ for every $q$ is realization equivalent to $\sigma_2(p) = \frac{1}{8}$ for every $p$, but it is realization equivalent also to $\sigma_2(l_1l_2l_3) = \sigma_2(r_1r_2r_3) = \frac{1}{2}$ and $\sigma_2(p) = 0$ for all the other $p$. The derivation is unique.

**Exercise 6.0.4** (Sequence-form and reduced normal-form). *What is the relation* *between sequence-form representation and reduced normal-form representation?*

(Sequence/reduced-normal-form strategies realization equivalence). Every pure sequence-form strategy corresponds to a single plan in the reduced normal form.

The observation above shows that the number of pure realization plans is exponentially large in the size of the game tree, the number of plans of the reduced normal-form being exponentially large. When using the reduced normal-form representation, we have a direct way to derive a strategy that is realization equivalent to a given realization plan.

## 1.6 EC 1.7

**Exercise 7.0.1** (Zero-sum/constant-sum/strictly competitive games). *Provide the definition of:*

- *2-player zero-sum game;*
- *2-player constant-sum game;*
- *2-player strictly-competitive game.*

*Furthermore, describe their relationships, and their properties in the players utility space.*

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

(Zero-sum game and utility space). Given the space of players' utilities $(U_1, U_2, \ldots, U_n)$ in $\mathbb{R}^n$, each terminal node can be mapped as a point in such a space and, in a zero-sum game, all these points lie on a hyperplane $\sum_{i \in N} U_i = 0$.

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

(Constant-sum game and utility space). Given the space of players' utilities $(U_1, U_2, \ldots, U_n)$ in $\mathbb{R}^n$, each terminal node can be mapped as a point in such a space and, in a constant-sum game, all these points lie on a hyperplane $\sum_{i \in N} U_i = constant$.

(Strictly competitive game). A strictly competitive game is a 2 -player game in which, for any pair of strategy profiles $\sigma, \sigma', \mathbb{E}_{\mathbf{a} \sim \sigma}[U_1(\mathbf{a})] < \mathbb{E}_{\mathbf{a} \sim \sigma'}[U_1(\mathbf{a})]$ if and only if $\mathbb{E}_{\mathbf{a} \sim \sigma}[U_2(\mathbf{a})] > \mathbb{E}_{\mathbf{a} \sim \sigma'}[U_2(\mathbf{a})]$.
Strictly-competitive games capture the situation in which, for every given strategy profile and for any perturbation of it, a player increases her expected utility if and only if the opponent reduces her expected utility. It is known that a strictly-competitive game is just a zero-sum game once an affine transformation

is applied.

(Strictly competitive game and utility space). Given the space of players' utilities $(U_1, U_2)$ in $\mathbb{R}^2$, each terminal node can be mapped as a point in such a space and, in a strictly competitive game, all these points lie on a hyperplane $\sum_{i \in N} \alpha_i U_i = constant$, where $\alpha_i \in (0, 1]$.

(Constant-sum game as zero-sum game). A constant-sum game is equivalent to a zero-sum game, once the constant has been subtracted from the utility of a player.

(Strictly competitive and zero-sum games). Any strictly-competitive game is equivalent to a zero-sum game once an affine transformation, in principle different for each player, is applied.

**Exercise 7.0.2** (Polymatrix games). *Provide the definition of Polymatrix games and the formula of players expected utility. Furthermore, provide the conditions under which a Polymatrix game is zero sum.*

A Polymatrix game in normal-form representation is a tuple $(N, A, U)$ where:

- the definitions of $N$ and $A$ are standard as in normal-form games;
- $U = \{U_1, U_2, \ldots, U_n\}$ and each utility function $U_i$ is defined as $\sum_{j \in N \setminus \{i\}} U_{i,j}$ where each $U_{i,j}$ is defined as $U_{i,j} : A_i \times A_j \to \mathbb{R}$ and therefore each $U_{i,j}$ can be described as a matrix.

(Polymatrix game and expected utility). The formula $\mathbb{E}_{\mathbf{a} \sim \mathbf{s}}[U_i(\mathbf{a})]$ in a Polymatrix game can be written as:

$$\mathbb{E}_{\mathbf{a} \sim \sigma}[U_i(\mathbf{a})] = \sum_{j \in N \setminus \{i\}} \sum_{a_i \in A_i} \sum_{a_j \in A_j} \sigma_i(a_i) \sigma_j(a_j) U_{i,j}(a_i, a_j)$$

The degree of the polynomial is 2 (independently from n) and, given player i, the expected utility is linear in player $i$'s strategy.

(Zero-sum Polymatrix games). A Polymatrix game with players $N$ is zero sum if for every $a \in A_1 \times A_2 \times \ldots \times A_n$ we have $\sum_{i \in N} U_i(a) = 0$.

(Zero-sum and zero-sum pairwise Polymatrix games). Any zero-sum Polymatrix game is equivalent to a Polymatrix game where $U_{i,j} + U_{i,j}^T = 0$. These last games are called zero-sum pairwise Polymatrix games.

## 1.7 EC 1.8

**Exercise 8.0.1** (Bayesian games). *Given a Bayesian game in epistemic-form, provide the representation in extensive-form.*

A simultaneous-moves Bayesian game in epistemic-form representation is a tuple $(N, A, \Theta, \Omega, U)$ where:

- $N = \{1, 2, \ldots, n\}$ is the set of players;
- $A = \{A_1, A_2, \ldots, A_n\}$ is the set of the actions of all the players and $A_i = \{a_1, a_2, \ldots, a_{m_i}\}$ is the set of player $i$'s actions;
- $\Theta = \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n$ is the set of the types of all the players and $\Theta_i = \{\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,k_i}\}$ is the set of the types of player $i$
- $\Omega : \Theta \to \Delta(\Theta)$ returns the probability associated with each $(\theta_1, \theta_2, \ldots, \theta_n)$ where $\theta_i \in \Theta_i$
- $U = \{U_1, U_2, \ldots, U_n\}$ is the set of the utility functions of all the players and $U_i : A_1 \times A_2 \times \ldots \times A_n \times \Theta \to \mathbb{R}$ is the utility function of player $i$.

Epistemic-form:
(Bayesian Battle of sexes). Consider a Bayesian game composed of:

- $N = \{1, 2\}$
- $A = \{A_1, A_2\}$ with $A_1 = \{\text{B}, \text{S}\}, A_2 = \{\text{B}, \text{S}\}$
- $\Theta = \{\Theta_1, \Theta_2\}$ with $\Theta_1 = \{\theta_{1,1}, \theta_{1,2}\}, \Theta_2 = \{\theta_{2,1}, \theta_{2,2}\}$
- $\Omega = \{\Omega_1, \Omega_2\}$ with:

$$\Omega_1 = \left\{ \begin{array}{ll} 0.2 & \theta_{1.1} \\ 0.8 & \theta_{1.2} \end{array} \right. , \quad \Omega_2 = \left\{ \begin{array}{ll} 0.3 & \theta_{2.1} \\ 0.7 & \theta_{2.2} \end{array} \right.$$

- utility functions are represented by means of the following bi-matrices:

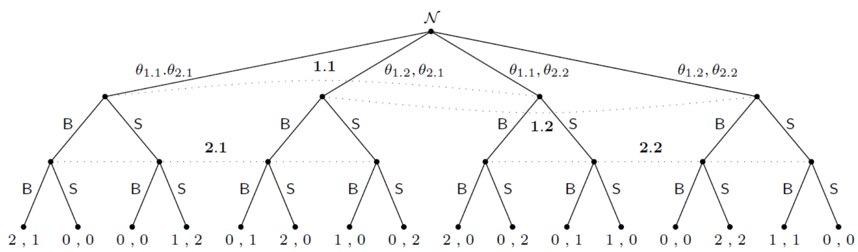
| $\theta_{1.1}$ / $\theta_{2.1}$ | B | S |
|---|---|---|
| B | 2, 1 | 0, 0 |
| S | 0, 0 | 1, 2 |

| $\theta_{1.1}$ / $\theta_{2.2}$ | B | S |
|---|---|---|
| B | 2, 0 | 0, 2 |
| S | 0, 1 | 1, 0 |

| $\theta_{1.2}$ / $\theta_{2.1}$ | B | S |
|---|---|---|
| B | 0, 1 | 2, 0 |
| S | 1, 0 | 0, 2 |

| $\theta_{1.2}$ / $\theta_{2.2}$ | B | S |
|---|---|---|
| B | 0, 0 | 2, 2 |
| S | 1, 1 | 0, 0 |

Extensive-form:



# Chapter 2

# EC2

## 2.1 EC 2.7

**Exercise 7.0.1** (Maxmin strategy definition). *Provide the definition of Maxmin strategy in 2–player normal–form games.*

The problem of finding a Maxmin strategy of player $i$ against player $-i$ is formulated as:

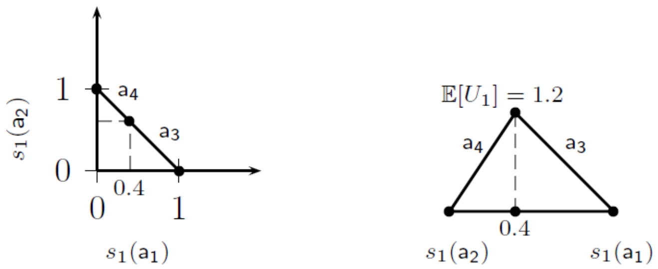
$$
\begin{aligned}
\arg\max_{\sigma_i} \quad & \min_{\sigma_{-i}} \quad \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} \left[ U_i\left(a_i, a_{-i}\right) \sigma_i\left(a_i\right) \sigma_{-i}\left(a_{-i}\right) \right] \\
& \quad s.t. \quad \sum_{a_{-i} \in A_{-i}} \sigma_{-i}\left(a_{-i}\right) = 1 \\
& \qquad\qquad \sigma_{-i}\left(a_{-i}\right) \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
s.t. \quad & \sum_{a_i \in A_i} \sigma_i\left(a_i\right) = 1 \\
& \sigma_i\left(a_i\right) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

**Exercise 7.0.2** (Maxmin strategy example). *Given a 2–player normal–form games with 2 actions per player, find a Maxmin strategy.*

Consider the following 2-player matrix game in which player 1 plays as max player and player 2 plays as min player:

| 1 \ 2 | $a_3$ | $a_4$ |
|---|---|---|
| $a_1$ | 0 | 3 |
| $a_2$ | 2 | 0 |

We report the strategy simplex of player 1 partitioned in subareas with the corresponding action player 2 would play and subsequently how the expected utility of player 1 varies in the simplex given the strategy of player 2. The Maxmin value of player 1 is 1.2.




**Exercise 7.0.3** (Complexity). *What is the computational complexity of finding the Maxmin/Minmax strategy with 2–player normal–form games?*

The Maxmin strategy and the corresponding Maxmin value of player $i$ against player $-i$ can be found in polynomial time in the size of the game, where the size is given by the number of actions of the players. Therefore the Maxmin finding problem is in $\mathsf{FP}$ class.

**Exercise 7.0.4** (Mathematical programming formulation). *Given a 2–player normal–form game, provide the mathematical programming formulation for finding a Maxmin/Minmax strategy and show its derivation.*

The Maxmin strategy and the corresponding value of player $i$ against player $-i$ can be found by solving the following Linear Program (LP):

$$
\begin{aligned}
\arg\max_{\sigma_i, v_j} \quad & v_i \\
\text{s.t.} \quad & v_i - \sum_{a_i \in A_i} \left[ U_i\left(a_i, a_{-i}\right) \sigma_i\left(a_i\right) \right] && \leqslant 0 \quad \forall a_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i\left(a_i\right) && = 1 \\
& \sigma_i\left(a_i\right) && \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

*Proof.* Given the definition of the Maxmin optimization problem:

$$\begin{aligned}
\arg\max_{\sigma_i} \quad & \min_{\sigma_{-i}} \quad \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} \Big[ U_i(a_i, a_{-i})\, \sigma_i(a_i)\, \sigma_{-i}(a_{-i}) \Big] && \\
& \quad \text{s.t.} \quad \sum_{a_{-i} \in A_{-i}} \sigma_{-i}(a_{-i}) && = 1 \\
& \quad \qquad \sigma_{-i}(a_{-i}) && \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
\text{s.t.} \quad & \sum_{a_i \in A_i} \sigma_i(a_i) && = 1 \\
& \sigma_i(a_i) && \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

we substitute the inner maximization problem with its dual problem that, by strong duality, returns the same optimal value of objective function, obtaining:

$$\begin{aligned}
\arg\max_{\sigma_i} \quad & \max_{v_i} \quad v_i && \\
& \quad \text{s.t.} \quad v_i - \sum_{a_i \in A_i} \Big[ U_i(a_i, a_{-i})\, \sigma_i(a_i) \Big] && \leqslant 0 \quad \forall a_{-i} \in A_{-i} \\
\text{s.t.} \quad & \sum_{a_i \in A_i} \sigma_i(a_i) && = 1 \\
& \sigma_i(a_i) && \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

Finally, joining the two max operators we obtain:

$$\begin{aligned}
\arg\max_{\sigma_i, v} \quad & v_i && \\
\text{s.t.} \quad & v_i - \sum_{a_i \in A_i} \Big[ U_i(a_i, a_{-i})\, \sigma_i(a_i) \Big] && \leqslant 0 \quad \forall a_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i(a_i) && = 1 \\
& \sigma_i(a_i) && \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

□

The Minmax strategy player $i$ against player $-i$ and the corresponding value of player $-i$ can be found by solving the following Linear Program:

$$\begin{aligned}
\arg\min_{\sigma_i, v_j} \quad & v_{-i} && \\
\text{s.t.} \quad & v_{-i} - \sum_{a_i \in A_i} \left[ U_{-i}\left(a_i, a_{-i}\right) \sigma_i\left(a_i\right) \right] && \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i\left(a_i\right) && = 1 \\
& \sigma_i\left(a_i\right) && \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

*Proof.* Given the definition of the Minmax optimization problem:

$$
\begin{aligned}
\arg\min_{\sigma_i} \quad & \max_{\sigma_{-i}} \quad \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} \left[ U_{-i}(a_i, a_{-i})\, \sigma_i(a_i)\, \sigma_{-i}(a_{-i}) \right] \\
& \quad \text{s.t.} \quad \sum_{a_{-i} \in A_{-i}} \sigma_{-i}(a_{-i}) = 1 \\
& \quad \qquad \sigma_{-i}(a_{-i}) \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
\text{s.t.} \quad & \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

we substitute the inner maximization problem with its dual problem that, by strong duality, returns the same optimal value of objective function, obtaining:

$$
\begin{aligned}
\arg\min_{\sigma_i} \quad & \min_{v} \quad v_{-i} \\
& \quad \text{s.t.} \quad v_{-i} - \sum_{a_i \in A_i} \Big[ U_{-i}(a_i, a_{-i})\, \sigma_i(a_i) \Big] \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
\text{s.t.} \quad & \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

Finally, we obtain:

$$
\begin{aligned}
\arg\min_{s_i, v_{-i}} \quad & v_{-i} \\
\text{s.t.} \quad & v_{-i} - \sum_{a_i \in A_i} \Big[ U_{-i}(a_i, a_{-i})\, \sigma_i(a_i) \Big] \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

□

**Exercise 7.0.5** (Solving a game). *Given a normal–form game with 2 players, find a Maxmin strategy by means of AMPL + GUROBI.*

TODO

## 2.2 EC 2.8

**Exercise 8.0.1** (Maxmin strategy definition). *Provide the definition of Maxmin strategy in 2–player sequence– form games.*

The Maxmin strategy of player i against player $-i$ in sequence form can be found by solving the following linear mathematical programming problem:

$$
\begin{aligned}
\arg\max_{r_i, v_i} \quad & \sum_{h_{-i} \in H_{-i} \cup \{h_\varnothing\}} f_{-i}(h_{-i})\, v_i(h_{-i}) \\
\text{s.t.} \quad & \sum_{h_{-i} \in H_{-i} \cup \{h_\varnothing\}} F_{-i}(h_{-i}, q_{-i})\, v_i(h_{-i}) - \sum_{q_i \in Q_i} \left[ U_i(q_i, q_{-i})\, r_i(q_i) \right] \leqslant 0 \quad \forall q_{-i} \in Q_{-i} \\
& \sum_{q_i \in Q_i} F_i(h_i, q_i)\, r_i(q_i) = f_i(h_i) \quad \forall h_i \in H_i \cup \{h_\varnothing\} \\
& r_i(q_i) \geqslant 0 \quad \forall q_i \in Q_i
\end{aligned}
$$

**Exercise 8.0.2** (Complexity). *What is the computational complexity of finding the Maxmin/Minmax strategy with 2–player sequence–form games?*

The Maxmin strategy and the corresponding Maxmin value of player $i$ against player $-i$ in an extensive-form game can be found, by means of the sequence-form representation, in polynomial time in the size of the game, where the size is given by the number of nodes of the game tree. Therefore the Maxmin finding problem is in FP class.

**Exercise 8.0.3** (Mathematical programming formulation). *Given a 2–player sequence–form game, provide the mathematical programming formulation for finding a Maxmin/Minmax strategy and show its derivation.*

The Minmax strategy of player i against player $-i$ in sequence form can be found by solving the following linear mathematical programming problem:

$$
\begin{aligned}
\arg\min_{r_i, v_{-i}} \quad & \sum_{h_{-i} \in H_{-i} \cup \{\mathrm{h}_\varnothing\}} f_{-i}(h_{-i})\, v_i(h_{-i}) \\
\text{s.t.} \quad & \sum_{h_{-i} \in H_{-i} \cup \{\mathrm{h}_\varnothing\}} F_{-i}(h_{-i}, q_{-i})\, v_i(h_{-i}) - \sum_{q_i \in Q_i} \left[U_{-i}(q_i, q_{-i})\, r_i(q_i)\right] \leqslant 0 && \forall q_{-i} \in Q_{-i} \\
& \sum_{q_i \in Q_i} F_i(h_i, q_i)\, r_i(q_i) = f_i(h_i) && \forall h_i \in H_i \cup \{\mathrm{h}_\varnothing\} \\
& r_i(q_i) \geqslant 0 && \forall q_i \in Q_i
\end{aligned}
$$

**Exercise 8.0.4** (Solving a game). *Given a sequence–form game with 2 players, find a Maxmin strategy by means of AMPL + GUROBI.*

TODO

## 2.3 EC 2.9

**Exercise 9.0.1** (Team Maxmin strategy definition). *Provide the definition of the Maxmin strategy when there are multiple max players and a single min player.*

**Many-max player/single-min players.** Now, we focus on the case in which there are multiple max players and only one min player. Initially, we remark that this situation makes sense only when the utility functions of all the max players are the same, otherwise it would not be clear which objective function the max players should maximize. Therefore, in this case, the max players are forming a team and searching for the Maxmin strategy is equivalent to searching for a Team-Maxmin strategy. Also here, we can distinguish the case in which the team plays in correlated strategies from the case in which the team plays in mixed strategies. We initially focus on the first case.

(Maxmin optimization problem (with correlated min players)). The problem of finding a Maxmin strategy of a team of $n-1$ players, playing in correlated fashion, against player $i$ is formulated as:

$$
\begin{aligned}
\arg\max_{\boldsymbol{\sigma}_{-i}} \quad & \min_{\sigma_i} \quad \sum_{\mathbf{a}_{-i}\in A_{-i}} \sum_{a_i\in A_i} \left[U_{-i}\left(a_i,\mathbf{a}_{-i}\right)\sigma_i\left(a_i\right)\boldsymbol{\sigma}_{-i}\left(\mathbf{a}_{-i}\right)\right] \\
& \quad s.t. \quad \sum_{a_i\in A_i}\sigma_i\left(a_i\right)=1 \\
& \qquad\quad \sigma_i\left(a_i\right)\geqslant 0 \qquad \forall a_i\in A_i \\
s.t. \quad & \sum_{\mathbf{a}_{-i}\in A_{-i}} \boldsymbol{\sigma}_i\left(\mathbf{a}_{-i}\right) = 1 \\
& \boldsymbol{\sigma}_i\left(\mathbf{a}_{-i}\right) \geqslant 0 \quad \forall \mathbf{a}_{-i}\in A_{-i}
\end{aligned}
$$

(Maxmin optimization problem (with non-correlated min players)). The problem of finding a Maxmin strategy of a team of $n-1$ players, playing in mixed strategies, against player $i$ is formulated as:

$$
\begin{aligned}
\arg\max_{\boldsymbol{\sigma}_{-i}} \quad & \min_{\sigma_i} \quad \sum_{a_i\in A_i} \sum_{\mathbf{a}_{-i}\in A_{-i}} \left[U_{-i}\left(a_i,\mathbf{a}_{-i}\right)\prod_{j\in N}\sigma_j(a_j)\right] \\
& \quad s.t. \quad \sum_{a_i\in A_i}\sigma_i\left(a_i\right)=1 \\
& \qquad\quad \sigma_i\left(a_i\right)\geqslant 0 \qquad \forall a_i\in A_i \\
s.t. \quad & \sum_{a_j\in A_j} \sigma_i\left(a_j\right) = 1 \qquad \forall j\in N\setminus\{i\} \\
& \sigma_j\left(a_j\right) \geqslant 0 \quad \forall j\in N\setminus\{i\}, \forall a_j\in A_j
\end{aligned}
$$

**Exercise 9.0.2** (Complexity). *What is the computational complexity of finding the Maxmin strategy with n–max players and 1 min player in normal–form games (equivalently finding a Minmax strategy with n–min player and 1 max player)?*

The Maxmin strategy and the corresponding Maxmin value of $n$ min-players against 1 max-player can be found in polynomial time in the size of the game, where the size is given by the number of actions of the players. Therefore the Maxmin finding problem is in $\mathsf{FP}$ class.

**Exercise 9.0.3** (Mathematical programming formulation). *Given an n–player normal–form game, provide the mathematical programming formulation for finding a Maxmin strategy and show its derivation when there are* $n-1$ *max players and* 1 *min players.*

(Maxmin strategy formulation (with correlated min players)). The Maxmin strategy and the corresponding value of a team of $n-1$ players, playing in correlated fashion, against player i can be found by solving the following Linear Program:

$$
\begin{aligned}
\arg\max_{\boldsymbol{\sigma}_{-i}, v_{-i}} \quad & v_{-i} \\
s.t. \quad & v_{-i} - \sum_{\mathbf{a}_{-i} \in A_{-i}} \Big[ U_{-i}(a_i, \mathbf{a}_{-i})\, \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \Big] \leqslant 0 \quad \forall a_i \in A_i \\
& \sum_{\mathbf{a}_{-i} \in A_{-i}} \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) = 1 \\
& \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \geqslant 0 \quad \forall \mathbf{a}_{-i} \in A_{-i}
\end{aligned}
$$

**Exercise 9.0.4** (Solving a game). *Given a normal–form game with n players among which $n-1$ are maximizers and 1 is a minimizer, find a Maxmin strategy by means of AMPL + BARON.*

**Exercise 9.0.5** (L'ho aggiunta io). *Provide the definition and the mathematical programming formulation of the Maxmin strategy when there is a single max player and multiple min players.*

(Maxmin optimization problem (with correlated min players)). The problem of finding a Maxmin strategy of player i against $n-1$ players, playing in correlated fashion, is formulated as:

$$
\begin{aligned}
\arg\max_{\sigma_i} \quad & \min_{\boldsymbol{\sigma}_{-i}} \sum_{a_i \in A_i} \sum_{\mathbf{a}_{-i} \in A_{-i}} \Big[ U_i(a_i, \mathbf{a}_{-i})\, \sigma_i(a_i)\, \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \Big] \\
& \quad s.t. \quad \sum_{\mathbf{a}_{-i} \in A_{-i}} \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) = 1 \\
& \quad \phantom{s.t.} \quad \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \geqslant 0 \quad \forall \mathbf{a}_{-i} \in A_{-i} \\
s.t. \quad & \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

We can easily derive the mathematical program solving the above problem.
(Maxmin strategy formulation (with correlated min players)). The Maxmin strategy and the corresponding value of player i against $n-1$ correlated players can be found by solving the following Linear Program:

$$
\begin{aligned}
\arg\max_{\sigma_i, v_i} \quad & v_i \\
s.t. \quad & v_i - \sum_{a_i \in A_i} \Big[ U_i(a_i, \mathbf{a}_{-i})\, \sigma_i(a_i) \Big] \leqslant 0 \quad \forall \mathbf{a}_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

(Maxmin strategy formulation (with correlated min players)). The problem of finding a Maxmin strategy with multiple min players playing correlated strategies is equivalent to the problem of finding a Maxmin strategy with a single min players, in which the actions available to this player are all the possible joint actions of all the min players.
Let us consider the case in which the min players play mixed strategies and

therefore they cannot correlate.
(Maxmin optimization problem (with non-correlated min players)). The problem of finding a Maxmin strategy of player i against $n-1$ players, each playing independently from the others, is formulated as:

$$\begin{array}{llll}
\arg\max\limits_{\sigma_i} & \min\limits_{\boldsymbol{\sigma}_{-i}} \sum\limits_{a_i \in A_i} \sum\limits_{\mathbf{a}_{-i} \in A_{-i}} \left[ U_i(a_i, \mathbf{a}_{-i}) \prod\limits_{j \in N} \sigma_j(a_j) \right] & & \\
& \quad s.t. \quad \sum\limits_{a_j \in A_j} \sigma_j(a_j) & = 1 & \forall j \in N \backslash \{i\} \\
& \quad \sigma_j(a_j) & \geqslant 0 & \forall j \in N \backslash \{i\}, \forall a_j \in A_j \\
s.t. & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & = 1 & \\
& \sigma_i(a_i) & \geqslant 0 & \forall a_i \in A_i
\end{array}$$

In this case, the problem of deriving a finite (in terms of variables and constraints) mathematical program is open. The crucial issue concerns the impossibility to reformulate the minimization problem since it is not linear and therefore strong duality cannot be applied.

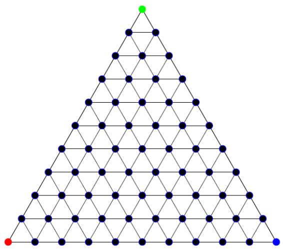
## 2.4 EC 2.10

**Exercise 10.0.1** (Sperner's labeling). *Provide the definition of Sperner's labeling.*

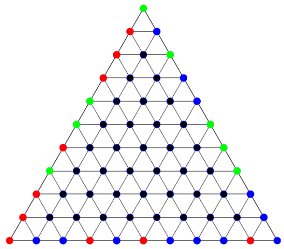
(Simplicial subdivision). A simplicial subdivision of an n-dimensional simplex $\Delta^n$ is a finite set of subsimplices $\mathcal{D} = \{\Delta_i^n\}$ for which $\bigcup_{\Delta_i^n \in \mathcal{D}} \Delta_i^n = \Delta^n$, and for any $\Delta_i^n, \Delta_j^n \in \mathcal{D}$ the set $\Delta_i^n \cap \Delta_j^n$ is either empty or equal to a common facet.
(Sperner's labeling). Given a n-dimensional simplex and a simplicial subdivion $\mathcal{D}$, a Sperner's labeling is a labeling of the vertices V of the subsimplices with labels $L = \{0, 1, \ldots, n\}$ is a function $l : V \rightarrow L$ satisfying the following conditions:

- each one of the $n+1$ vertices of the simplex is labeled with a label different from the label of the others;
- the vertices of a subsimplex that lie on the facet of the simplex whose vertices have labels in $\{1, \ldots, n\}$ have label in $\{1, \ldots, n\}$ (a similar condition holds for the vertices on the other facets);
- there is no restriction on the labeling of the vertices in the interior of the simplex.

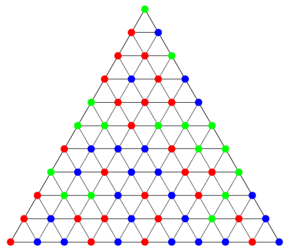
An example of Sperner's labeling of the above simplex $\Delta^2$ is as follows:



*Labeling simplex vertices*



*Labeling vertices on the simplex facets*



*Labeling inner vertices*

**Exercise 10.0.2** (Panchromatic and quasi–panchromatic subsimplex). *Provide the definitions of panchromatic and quasi–panchromatic n–dimension subsimplex.*

(Panchromatic (completely labeled) subsimplex). A subsimplex is called completely labeled or panchromatic if each vertex has a label different from the one of all the other vertices of the subsimplex. Therefore, a panchromatic n-dimensional subsimplex has $n + 1$ different labels.

(Quasi-panchromatic (almost completely labeled) subsimplex). An n dimensional subsimplex is called almost completely labeled or quasi panchromatic if each vertex has a label different from the one of all the other vertices of the subsimplex except one. Therefore, a panchromatic n-dimensional subsimplex has n different labels among which one label is present twice.

**Exercise 10.0.3** (Sperner's Lemma). *Provide the statement of the Sperner's Lemma.*

(Sperner's Lemma). Given a n-dimensional simplex, a simplicial subdivion $\mathcal{D}$, and a Sperner's labeling $l$, there exists an odd number of panchromatic n-dimensional subsimplices. Therefore, there exists at least one panchromatic subsimplex.

**Exercise 10.0.4** (Sperner's walk). *Provide the definition of Spener's walk and its characterization.*

(Sperner's walk). Given a n-dimensional simplex $\Delta^n$, a simplicial subdivion $\mathcal{D}$, and a Sperner's labeling $l$, a Sperner's walk is a sequence of adjacent subsimplices such that each common facet traversed to move from a subsimplex to the adjacent subsimplex has the same n different labels, that is $L \setminus \{s\}$ for some $s \in L$. Therefore, each subsimplex of the walk is either quasi panchromatic or panchromatic.

(Sperner's walk characterization). There are four main classes of Sperner's walks:

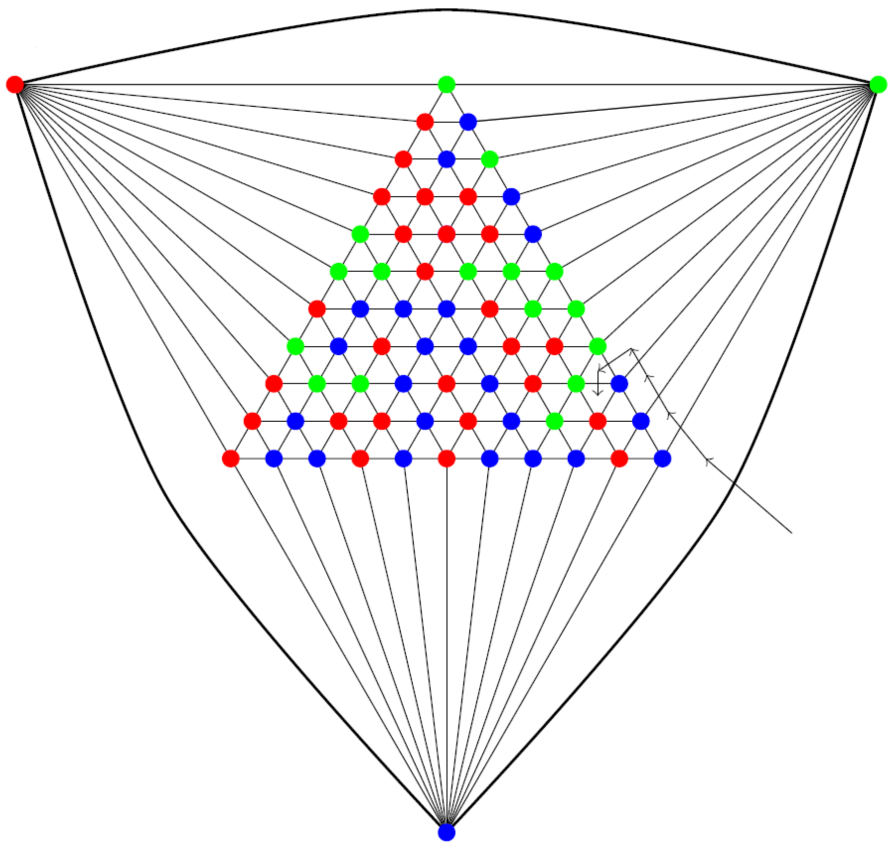
- a sequence of subsimplices starting from a subsimplex with a facet on a facet of the simplex and ending in a subsimplex with a face on the same facet of the simplex without traversing any panchromatic subsimplex or any subsimplex with a facet on of the other facets of the simplex;

- a sequence of subsimplices starting from a subsimplex with a facet on a facet of simplex and ending in a panchromatic subsimplex;
- a cycle of subsimplices without any panchromatic subsimplex;
- a sequence of subsimplices starting from a panchromatic subsimplex and ending in a panchromatic subsimplex.

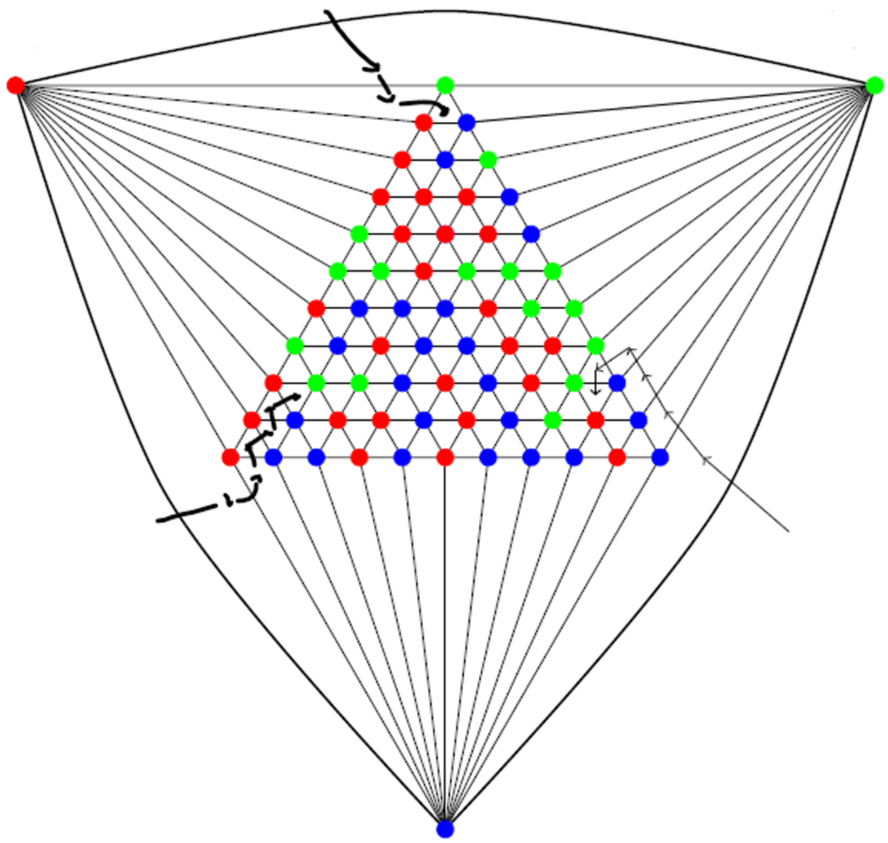
Except walks strictly contained in the walks of the above four classes, no other Sperner's walks are possible.

**Exercise 10.0.5** (Scarf's algorithm). *Describe the Scarf's algorithm and its properties.*

(Scarf's algorithm). We describe the algorithm for the 2-dimensional case. Given a 2-dimensional simplex $\Delta^2$ and a simplicial subdivion $\mathcal{D}$, we embed $\Delta^2$ in a larger simplex $\bar{\Delta}^2$ as follows. Each vertex $v$ of $\bar{\Delta}^2$ is assigned a different label and it is connected to all the vertices of the subsimplices of $\Delta^2$ of a given facet such that one of the two labels appearing in the facet is the label of $v$. Two different vertices $v, v'$ of $\bar{\Delta}^2$ are connected to the vertices of different facets of $\bar{\Delta}^2$ except for the vertices of the simplex $\Delta^2$ that belong to different facets. Furthermore, the three vertices of $\bar{\Delta}^2$ are connected with each other. Thus, there is a fictitious simplex composed of the whole space outer than $\bar{\Delta}^2$ that is panchromatic. This simplex is the starting point of the Scarf's algorithm.
The algorithm follows a Sperner's walk leaving this simplex (notice that three different paths can follow). Each path will end in a panchromatic subsimplex.
We report below the construction needed for the application of the Scarf's algorithm and a path generated by an execution of the algorithm, specifically when the first traversed segment has colors blue and green.



**Exercise 10.0.6** (Solving 2–dimension Sperner). *Given a 2–dimension simplex with a simplicial subdivision and a Sperner's labeling, find a panchromatic subsimplex by means of Scarf's algorithm (three examples of exercises follow). Do that for all three possible initializations of the Scarf's algorithm.*



## 2.5 EC 2.11

**Exercise 1** (END–OF–LINE). *Provide the definition of the END–OF–LINE problem.*

(END-OF-LINE problem). END-OF-LINE is a problem characterized as follows:

- a graph composed of a potentially exponential number $O(2^n)$ of vertices labeled by a sequence of n bits;
- a function $s : \{0,1\}^n \to \{0,1\} \cup \{N\}$ polynomially computable that, given a vertex, returns the unique successor, if this exists, and N otherwise;
- a function $p : \{0,1\}^n \to \{0,1\} \cup \{N\}$ polynomially computable that, given a vertex, returns the unique successor, if this exists, and N otherwise;
- a node without predecessor;
- the question is the finding of a leaf vertex (i.e., a vertex without either a predecessor or a successor).

**Exercise 2** (PPAD class). *Provide the definition of PPAD computational class and its relationship with TFNP, FNP, and FNP–completeness.*

(PPAD complexity class). The PPAD complexity class is composed of all the problems P such that there is a polynomial-time reduction from P to END-OF-LINE problem, i.e., each instance of P can be formulated as an instance of END-OF-LINE whose size is bounded by a polynomial in the size of P.
In other words, a searching problem is in PPAD whether it can be solved by means of a path-following algorithm that starts from a source and, following a path, achieves a sink corresponding to a solution of the problem itself. We focus on the problems that are the hardest ones in the PPAD class.

(PPAD completeness). The set of PPAD-complete problems is a subset of PPAD problems containing all the problems P such that there is a polynomial-time reduction from END-OF-LINE to P.
Easily, a problem is PPAD-complete when it can be reduced from the END-OF-LINE problem. Furthermore, it is commonly believed the following.

(PPAD completeness and FP). It is commonly conjectured that the class of PPAD-complete problems is not in FP.
Therefore, the common conjecture is that FP $\subset$ PPAD $\subset$ FNP. A number of fixed-point problems are PPAD complete. We cite a few of these problems.
Manca relazione con TFNP e FNP-completeness.

**Exercise 3** (Nash complexity). *Describe the computational complexity of finding a Nash equilibrium.*

(PPAD-completeness of Nash equilibrium finding in 2-player games). Finding a Nash equilibrium in 2-player games is PPAD complete.
(PPAD completeness of $\epsilon$-Nash equilibrium finding in games with 3 or more players). Finding an $\epsilon$-Nash equilibrium in games with 3 or more players is PPAD complete.
(Pure-strategy Nash equilibrium complexity). Once n is fixed, computing a pure-strategy Nash equilibrium is in P class.

**Exercise 4** (Sperner complexity). *Describe the computational complexity of finding a panchromatic subsimplex in k–dimension Sperner's problem.*

(k-dimensional Sperner's Lemma PPAD completeness). Given $k \geqslant 2$, the problem of finding a panchromatic subsimplex of Sperner's Lemma is PPAD complete.
This leads us to the following observation. That is, in the worst case, the Scarf's algorithm is the best algorithm to find a panchromatic subsimplex.
(Scarf's algorithm and Sperner's Lemma complexity). No algorithm more efficient than the Scarf's algorithm exists to find a panchromatic subsimplex of the Sperner's Lemma unless FP = PPAD.

## 2.6 EC 2.12

**Exercise 12.0.1** (Nash equilibrium). *Provide the definition of Nash equilibrium.*

(Nash equilibrium finding). The problem of finding a Nash equilibrium can be formulated as a mathematical programming problem

$$\begin{aligned}
&\sigma_i(a_i)\left(v_i - \sum_{\mathbf{a}_{-i}\in A_{-i}}\left[U_i(a_i,\mathbf{a}_{-i})\prod_{j\in N\setminus\{i\}}\sigma_j(a_j)\right]\right) && = 0 \quad \forall i\in N, a_i\in A_i\\
&v_i - \sum_{\mathbf{a}_{-i}\in A_{-i}}\left[U_i(a_i,\mathbf{a}_{-i})\prod_{j\in N\setminus\{i\}}\sigma_j(a_j)\right] && \geqslant 0 \quad \forall i\in N, a_i\in A_i\\
&\sum_{a_i\in A_i}\sigma_i(a_i) && = 1 \quad \forall i\in N\\
&\sigma_i(a_i) && \geqslant 0 \quad \forall i\in N, a_i\in A_i
\end{aligned}$$

whose nature is Non-Linear Complementarity Constraint (NLCP).
In the special case of 2 players, we have that the program reduced to a Linear Complementarity Program (LCP) as follows.

(Nash equilibrium finding with 2 players (LCP)). The problem of finding a Nash equilibrium with 2 players can be formulated as a mathematical programming problem

$$\begin{aligned}
&\sigma_i(a_i)\left(v_i - \sum_{a_{-i}\in A_{-i}}\left[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\right]\right) && = 0 \quad \forall i\in N, a_i\in A_i\\
&v_i - \sum_{a_{-i}\in A_{-i}}\left[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\right] && \geqslant 0 \quad \forall i\in N, a_i\in A_i\\
&\sum_{a_i\in A_i}\sigma_i(a_i) && = 1 \quad \forall i\in N\\
&\sigma_i(a_i) && \geqslant 0 \quad \forall i\in N, a_i\in A_i
\end{aligned}$$

whose nature is Linear Complementarity (LCP).
Let us remark that the program keeps to be nonlinear, variables $v_i and \sigma_i$ being multiplied together. However, an LCP have better properties than a generic NLCP. In particular, the solutions of an LCP have rational values, while the solution sof NLCP may have irrational values.

**Exercise 12.0.2** (Nash equilibrium example). *Given a 2–player normal–form game with 2 actions per player, find all the Nash equilibria.*

TODO

**Exercise 12.0.3** (Mathematical programming formulation for normal–form games). *Given an 2–player normal–form game, provide the MILP/MINLP program for finding a Nash equilibrium and show its derivation.*

(Nash equilibrium finding with 2 players (MILP)). The problem of finding a Nash equilibrium with 2 players can be formulated as a mathematical programming problem

$$\begin{aligned}
&\sigma_i(a_i)\left(v_i - \sum_{a_{-i}\in A_{-i}}\Big[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\Big]\right) && = 0 \quad \forall i \in N, a_i \in A_i \\
&v_i - \sum_{a_{-i}\in A_{-i}}\Big[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\Big] && \geqslant 0 \quad \forall i \in N, a_i \in A_i \\
&\sum_{a_i\in A_i} \sigma_i(a_i) && = 1 \quad \forall i \in N \\
&\sigma_i(a_i) && \geqslant 0 \quad \forall i \in N, a_i \in A_i
\end{aligned}$$

whose nature is Mixed-Integer Linear (MILP); the constant $M_i$ can be set as $M_i = \max_a\{U_i(\mathbf{a})\} - \min_a\{U_i(\mathbf{a})\}$

*Proof.* We just need to show that this reformulation is equivalent to the previous LCP one. The novelty resides in the adoption of binary variables $b_i$. The rationale is that an action $a_i$ can be played with strictly positive probability only if $b_i\,(a_i)$ is equal to 1. This latter condition holds only the expected utility given by playing action $a_i$ equals $v_i$. Indeed, if $b_i\,(a_i) = 1$, we have that the term $M_i\,(1 - b_i\,(a_i)) = 0$ and therefore $v_i = \sum_{a_{-i}\in A_{-i}} [U_i\,(a_i, a_{-i})\,\sigma_{-i}\,(a_{-i})]$. If instead $b_i\,(a_i) = 0$, the expected utility given by action $a_i$ may be strictly smaller than $v_i$, since we that $v_i - \sum_{a_{-i}\in A_{-i}} [U_i\,(a_i, a_{-i})\,\sigma_{-i}\,(a_{-i})] \leqslant M_i$ that is always true by construction of $M_i$. In other words, the adoption of a binary $b_i\,(a_i)$ allows one to enable or disable a set of constraints.

**Exercise 12.0.4** (Solving a normal–form game). *Given a normal–form game with 2 players, find a Nash equilibrium by means of AMPL + GUROBI.*

TODO

## 2.7 EC 2.13

**Exercise 13.0.1** (Lemke–Howson's labeling). *Provide the definition of the Lemke–Howson's labeling.*

The Lemke-Howson algorithm is a path-following algorithm working in 2-player normal-form games, returning an exact Nash equilibrium. Differently from the Scarf's algorithm, the Lemke-Howson does not discretize the simplex space. Nevertheless, the Lemke-Howson algorithm is a combinatorial algorithm moving over a finite set of points. Initially, we introduce the set of labels.

(Labels). The set of labels $L$ is defined as $L = \bigcup_{i\in N} A_i$. That is, we have one label for each action of each player.
(Solution labeling). A solution labeling $l$ is a function $l : \times_{i\in N}\Delta(A_i) \to \wp(L)$.

(Lemke-Howson's labeling). The Lemke-Howson's labeling $l$ is defined as follows:

$$a_i \in l(\boldsymbol{\sigma}) \Longleftrightarrow \left( \left(\sigma_i\left(a_i\right) = 0\right) \vee \left( v_i = \sum_{a_{-i} \in A_{-i}} U_i\left(a_i, a_{-i}\right) \sigma_{-i}\left(a_{-i}\right) \right) \right)$$

**Exercise 13.0.2** (Lemke–Howson's algorithm definition). *Provide the sketch of the Lemke–Howson's algorithm.*

(Lemke-Howson (rationale)). The algorithm is structured as follows.

- Partition the simplex of player $i$ in terms of best response of player $-i$. Each strategy profile $\sigma$ corresponds to a pair of points, one in the simplex of player 1 and one in the simplex of player 2. Each strategy profile $\sigma$ is associated with a number of labels according to the Lemke-Howson labeling.
- Focus on the nodes obtained by the intersections between the best response conditions and simplex border. Each node has three different labels (in degenerate games the labels may be more than three). The algorithm moves only between nodes.
- Focus on completely labeled solutions and almost completely labeled solutions. An almost completely labeled solution $\sigma$ presents a label, say $\mathsf{a}$, twice. One is given in the simplex of player 1, while the other is given by the simplex of player 2. Therefore, given an almost completely labeled solution $\sigma$, there are two ways to remove the label that appears twice: moving in the simplex of player 1 and moving in the simplex of player 2. This means that moving along almost completely labeled solutions the algorithm moves along paths. These paths can be either cycles or non-cycles in which the starting solution and the ending solution are completely labeled solutions.
- Start from an artificial completely labeled solution: $\sigma_i(a_i) = 0$ for every $i \in N$ and for every $a_i \in A_i$. And move along almost completely labeled solutions until find a completely labeled solution.

**Exercise 13.0.3** (Lemke–Howson's algorithm application). *Given a 2–player game with 3 actions per player and the partitioning of the simplices based on the opponent's best responses, provide the labels associated with each area and apply the Lemke–Howson algorithm with three different initializations, corresponding to the three actions of player 1 (two examples of exercises follow).*

Consider the following 2-player game:

| 1 \ 2 | $\mathsf{a}_4$ | $\mathsf{a}_5$ | $\mathsf{a}_6$ |
|---|---|---|---|
| $\mathsf{a}_1$ | $2,3$ | $2,0$ | $0,0$ |
| $\mathsf{a}_2$ | $0,0$ | $3,3$ | $2,0$ |
| $\mathsf{a}_3$ | $3,0$ | $0,0$ | $1,3$ |

and the following strategy profile:

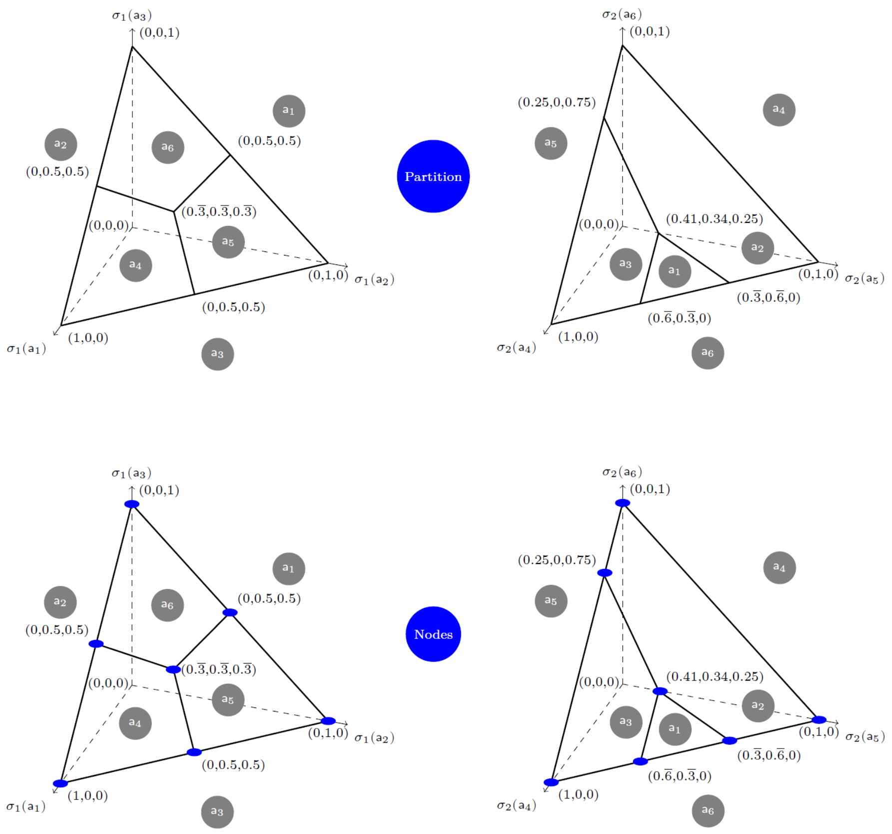
$$\sigma_1(a_1) = \begin{cases} 0.5 & \mathrm{a}_1 \\ 0.5 & \mathrm{a}_2 \\ 0.0 & \mathrm{a}_3 \end{cases} \qquad \sigma_2(a_2) = \begin{cases} 0.5 & \mathrm{a}_4 \\ 0.0 & \mathrm{a}_5 \\ 0.5 & \mathrm{a}_6 \end{cases}$$

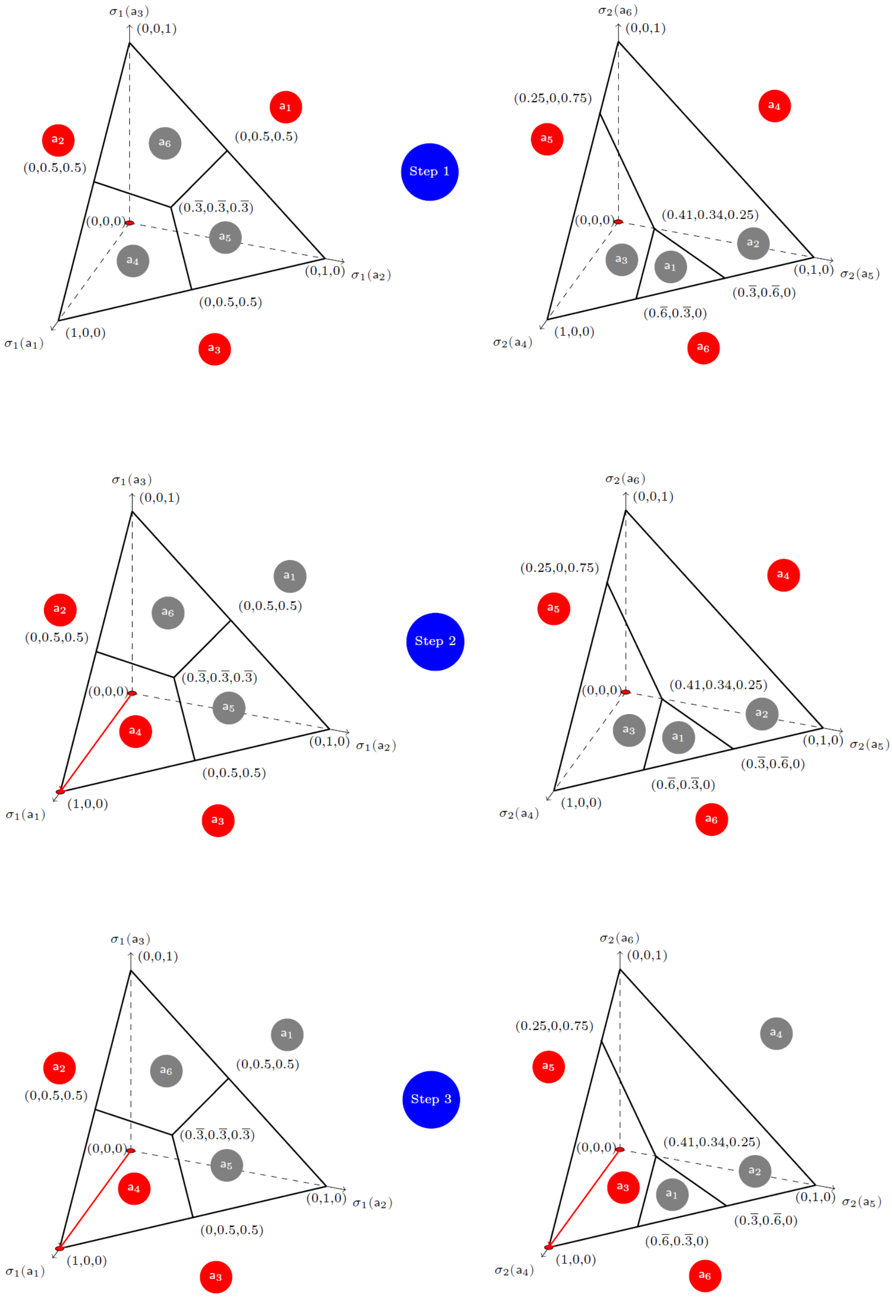
We have that $l(\sigma) = \{a_3, a_4, a_5\}$. Indeed, $a_3$ is the only best response of player 1 and therefore the labels of $a_1$ and $a_2$ that are played with strictly positive probability cannot be in $l(\sigma)$, while both $a_4$ and $a_5$ are best response of player 2 and therefore the label of $a_6$ that is played with strictly positive probability cannot be in $l(\sigma)$.

(Lemke-Howson). Consider the following 2-player game:

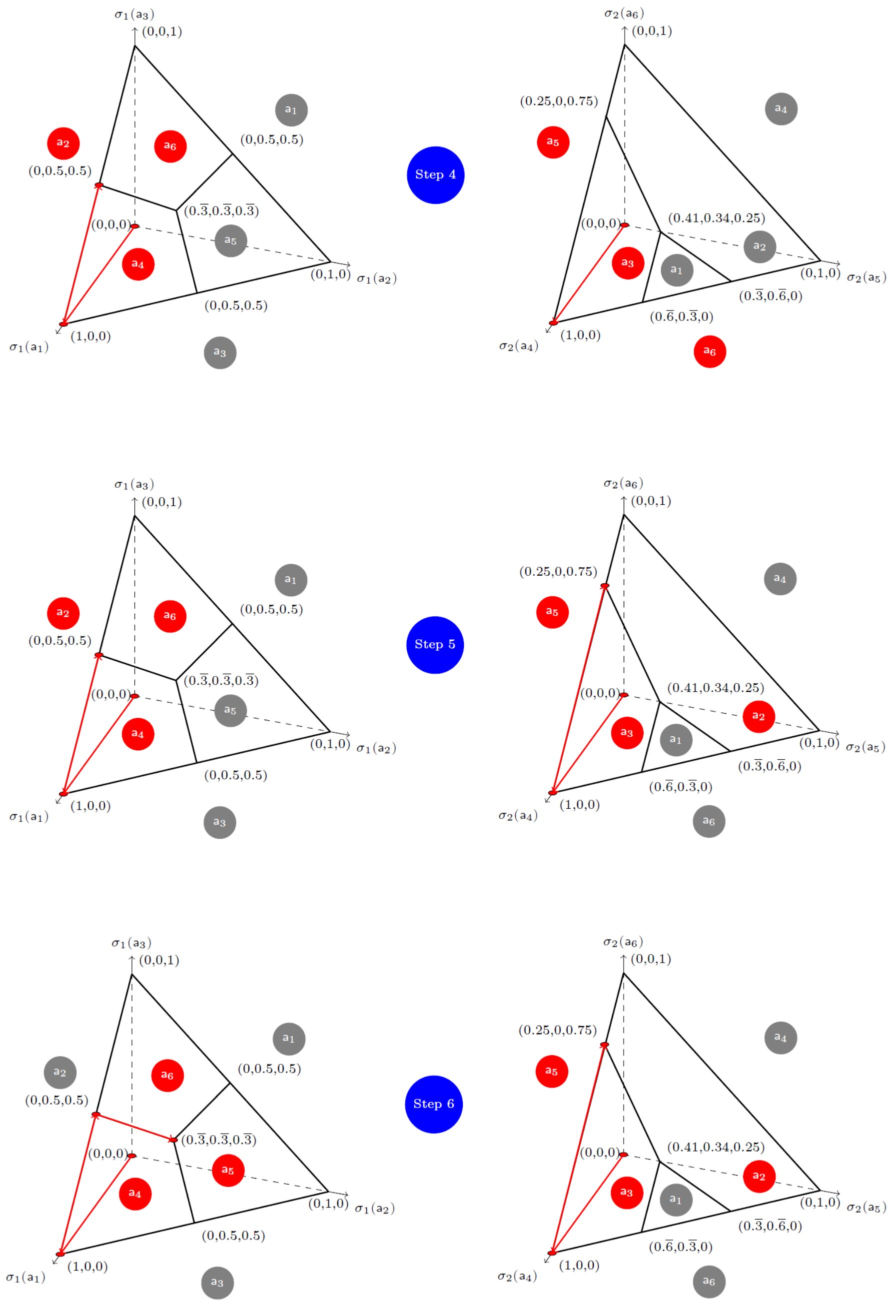
| 1 \ 2 | $\mathrm{a}_4$ | $\mathrm{a}_5$ | $\mathrm{a}_6$ |
|---|---|---|---|
| $\mathrm{a}_1$ | $2,3$ | $2,0$ | $0,0$ |
| $\mathrm{a}_2$ | $0,0$ | $3,3$ | $2,0$ |
| $\mathrm{a}_3$ | $3,0$ | $0,0$ | $1,3$ |

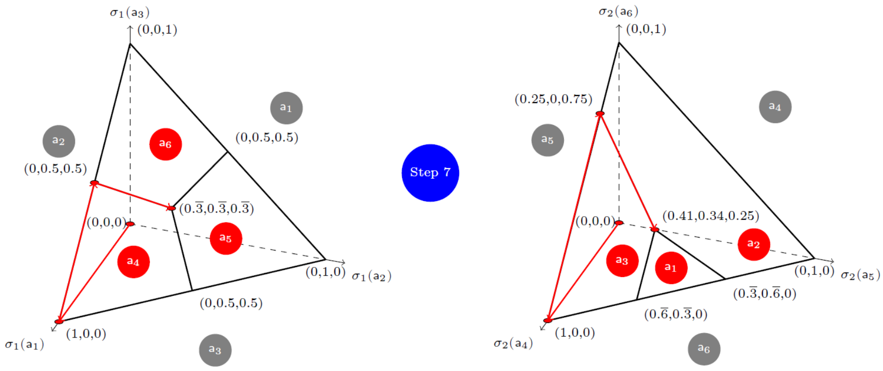
We report graphically the functioning of the algorithm.





$\sigma_1(a_3)$
(0,0,1)
$a_1$
$a_2$
$a_6$
(0,0.5,0.5)
(0,0.5,0.5)
$(0.\overline{3},0.\overline{3},0.\overline{3})$
(0,0,0)
$a_5$
$a_4$
(0,1,0)
$\sigma_1(a_2)$
(0,0.5,0.5)
$\sigma_1(a_1)$
(1,0,0)
$a_3$
Step 1
$\sigma_2(a_6)$
(0,0,1)
(0.25,0,0.75)
$a_4$
$a_5$
(0.41,0.34,0.25)
(0,0,0)
$a_2$
$a_3$
$a_1$
(0,1,0)
$\sigma_2(a_5)$
$(0.\overline{3},0.\overline{6},0)$
$(0.\overline{6},0.\overline{3},0)$
$\sigma_2(a_4)$
(1,0,0)
$a_6$
Step 2
Step 3


Step 4
Step 5
Step 6
$\sigma_1(a_3)$
(0,0,1)
$a_1$
$a_2$
$a_6$
(0,0.5,0.5)
(0,0,0)
($0.\overline{3}$,$0.\overline{3}$,$0.\overline{3}$)
$a_5$
$a_4$
(0,1,0)
$\sigma_1(a_2)$
$\sigma_1(a_1)$
(1,0,0)
$a_3$
$\sigma_2(a_6)$
(0.25,0,0.75)
$a_4$
$a_5$
(0.41,0.34,0.25)
$a_2$
$a_3$
$a_1$
$\sigma_2(a_5)$
($0.\overline{3}$,$0.\overline{6}$,0)
($0.\overline{6}$,$0.\overline{3}$,0)
$\sigma_2(a_4)$
$a_6$



$\sigma_1(a_3)$
(0,0,1)
$a_1$
$a_2$
$a_6$
(0,0.5,0.5)
(0,0.5,0.5)
$(0.\overline{3},0.\overline{3},0.\overline{3})$
(0,0,0)
$a_5$
$a_4$
(0,1,0)
$\sigma_1(a_2)$
(0,0.5,0.5)
(1,0,0)
$\sigma_1(a_1)$
$a_3$
Step 7
$\sigma_2(a_6)$
(0,0,1)
(0.25,0,0.75)
$a_4$
$a_5$
(0,0,0)
(0.41,0.34,0.25)
$a_2$
$a_3$
$a_1$
(0,1,0)
$\sigma_2(a_5)$
$(0.\overline{3},0.\overline{6},0)$
$(0.\overline{6},0.\overline{3},0)$
(1,0,0)
$\sigma_2(a_4)$
$a_6$

# Chapter 3

# EC3

## 3.1 EC 3.14

**Exercise 14.0.1** (Market model definition). *Provide the definition of the market model used by Arrow and Debreu.*

Consider a market model as follows:

- $N = \{1, \ldots, n\}$ is the set of players;
- $K = \{1, \ldots, k\}$ is the set of perfectly divisible commodities;
- $x_i \in \mathbb{R}_+^k$ with $i \in N$ is a vector of $k$ elements, where $x_{i,j}$, denoting the element in the $j$-th position of the vector, represents the amount of commodity $j$ available to player $i$;
- $e_i \in \mathbb{R}_+^k$ with $i \in N$ is the initial endowment of player $i$ and $e_{i,j}$ is the initial endowment for commodity $j$;
- $u_i : \mathbb{R}_+^k \to \mathbb{R}_+$ is the utility function of player $i$ that is continuous and concave;
- $p \in \mathbb{R}_+^k$ is a vector of $k$ elements, where $p_j$, denoting the element in the $j$-th position of the vector, represents the price of commodity $j$.

Each player maximizes her utility function under budget constraints by buying/selling commodities. Formally, we have:

$$\begin{array}{cl} \max_{x_i} & u_i(x_i) \\ \text{s.t.} & p \cdot x_i \leqslant p \cdot e_i \end{array}$$

where $p \cdot e_i$ represents the budget available to player $i$, while the argument of the above optimization problem, denoted by $x_i^*$, is the best amount of commodity for player $i$ given the market prices and the budget constraints. Notice that the above optimization problem does not take into account any constraint related to the amount of commodities available in the market. Thus, in principle, the above optimization problem may return a $x_i^*$ that is not implementable in practice, requiring an excessive amount of commodities larger than that one actually available in the market.

**Exercise 14.0.2** (Market clearing). *Provide the definition of market clearing.*

The market clears when the total amount of commodities $x = \sum_{i \in N} x_i$ equals to total amount of initial endowments $e = \sum_{i \in N} e_i$.
Notice that, when the price vector is arbitrary, there is no guarantee that the market clears. For instance, as already mentioned above, it may happen that the vector $x$ is larger than $e$ for some commodities. The market clearing property is of paramount importance and depends on the price vector. Indeed, when this property holds, each player in the market acts independently from the others. In other words, each player negotiates with a fictitious player corresponding to the market. This allows one to neglect the direct interaction between the players when searching for the market outcome.
Thus, the study of the conditions under which the market clears is crucial. This is stated by the Arrow-Debreu theorem.

**Exercise 14.0.3** (Arrow-Debreu equilibrium). *Provide the definition of the Arrow-Debreu equilibrium and argue the meaning of the result.*

There is always a price vector $p^*$ such that $\sum_{i \in N} x_i^* \leqslant e$ and $p^*$ is Pareto efficient for the players.
The problem of finding an Arrow-Debreu equilibrium is slightly different from the problem of finding a Nash equilibrium. On the one hand, each player behaves rationally, maximizing her utility function, as in a Nash equilibrium. In particular, the problem reduces to the problem of finding a Nash equilibrium when prices are fixed. On the other hand, prices are part of the problem. Thus the problem of finding an Arrow-Debreu equilibrium can be formulated as the problem of finding prices satisfying some conditions under the equilibrium constraints of the players. Interestingly, the problem of finding an Arrow-Debreu equilibrium presents the same computational properties of the problem of finding the Nash equilibrium, as stated below.
(Arrow-Debreu equilibrium complexity). The problem of finding a Arrow-Debreu market equilibrium is PPAD-complete even when all traders use additively separate, piecewise-linear and concave utility functions.

## 3.2 EC 3.15

**Exercise 15.0.1** (Leader-Follower equilibrium definition). *Provide the definition of the Leader-Follower equilibrium with 2 players.*

The problem of finding a Leader–Follower equilibrium is formulated as:

$$
\begin{array}{lllll}
\arg\max\limits_{s_l, s_f^*} & \sum\limits_{a_l \in A_l} \sum\limits_{a_f \in A_f} [U_l(a_l, a_f) s_l(a_l) s_f^*(a_f)] & & & \\
s.t. & \sum\limits_{a_l \in A_l} s_l(a_l) & = 1 & & \\
 & s_l(a_l) & \geqslant 0 & \forall a_l \in A_l & \\
 & s_f^* & \in \arg\max\limits_{s_l} & \sum\limits_{a_l \in A_l} \sum\limits_{a_l \in A_l} [U_f(a_f, a_l) s_f(a_f) s_l(a_l)] & \\
 & & s.t. & \sum\limits_{a_f \in A_f} s_f(a_f) & = 1 \\
 & & & s_f(a_f) & \geqslant 0 \ \forall a_f \in A_f
\end{array}
$$

**Exercise 15.0.2** (Leader-Follower equilibrium properties). *Describe the properties of the Leader-Follower equilibrium with 2 players.*

A Stackelberg equilibrium always exist when ties are broken in favour of the leader.
A Stackelberg equilibrium may not exist when ties are not broken in favour of the leader.
With 2 players (one leader and one follower), committing to a strategy never hurts the leader when ties are broken in its favour.
Equivalently, under those conditions, the best Nash for the leader is never strictly better than the Stackelberg equilibrium for the leader.

**Exercise 15.0.3** (Leader-Follower equilibrium example). *Given a 2-player normal-form game with 2 actions per player, find a Leader-Follower equilibrium.*

Consider the following 2-player game with 2 actions per player:

| | | 2 | |
|---|---|---|---|
| | | $\mathsf{a}_3$ | $\mathsf{a}_4$ |
| 1 | $\mathsf{a}_1$ | $4, 3$ | $3, 1$ |
| | $\mathsf{a}_2$ | $5, 0$ | $2, 4$ |

Under the constraint that the follower plays $\mathsf{a}_3$, the best strategy of the leader and the corresponding value is:

$$
s_l = \begin{cases} \frac{2}{3} & \mathsf{a}_1 \\ \frac{1}{3} & \mathsf{a}_2 \end{cases} \qquad v_l = \frac{13}{3}
$$

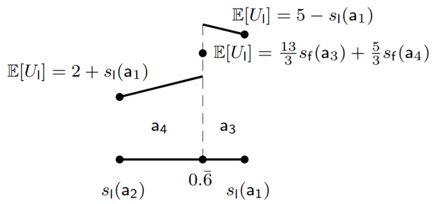
while under the constraint that the follower plays $\mathsf{a}_4$, the best strategy of the leader and the corresponding value is:

$$
s_l = \begin{cases} \frac{2}{3} & \mathsf{a}_1 \\ \frac{1}{3} & \mathsf{a}_2 \end{cases} \qquad v_l = \frac{8}{3}
$$

Under the assumption that the follower breaks ties in favor of the leader, the equilibrium is:

$$
s_l = \begin{cases} \frac{2}{3} & \mathsf{a}_1 \\ \frac{1}{3} & \mathsf{a}_2 \end{cases} \qquad s_f = \begin{cases} 1 & \mathsf{a}_1 \\ 0 & \mathsf{a}_2 \end{cases}
$$

We show that no equilibrium exists if the follower breaks ties in a different way. For any fully mixed $s_f$ played when $s_l = (\frac{2}{3}, \frac{1}{3})$, the utility function of the leader has a superior at $s_l = (\frac{2}{3}, \frac{1}{3})$ but the maximum does not exist. Indeed, the best strategy of the leader is $s_l = (\frac{2}{3} + \epsilon, \frac{1}{3}) + \epsilon$ with $\epsilon \to 0$, but $\epsilon$ cannot be 0. In other words, the limit does not exist. The expected utility of the leader is depicted in the figure below.




**Exercise 15.0.4** (Leader-Follower equilibrium formulation). *Provide the mathematical programming formulation to find a Leader-Follower equilibrium with 2 players.*

The Leader–Follower equilibrium can be found by solving the following mathematical programming problem:

$$
\begin{array}{llllll}
\max\limits_{\bar{a}_f} & \max\limits_{s_l} & \sum\limits_{a_l \in A_l} [U_l(a_l, \bar{a}_f) s_l(a_l)] & & & \\
 & s.t. & \sum\limits_{a_l \in A_l} s_l(a_l) & & = 1 & \\
 & & s_l(a_l) & & \geqslant 0 & \forall a_l \in A_l \\
 & & \sum\limits_{a_l \in A_l} [U_f(\bar{a}_f, a_l) s_l(a_l)] - \sum\limits_{a_l \in A_l} [U_f(a_f, a_l) s_l(a_l)] & & \geqslant 0 & \forall a_f \in A_f
\end{array}
$$

whose nature is Multi–Linear Program (Multi–LP).

**Exercise 15.0.5** (Finding a Leader-Follower equilibrium). *Given a 2-player normal-form game, find a Leader-Follower equilibrium by means of AMPL + GUROBI.*

TODO

**Exercise 15.0.6** (Leader-Follower equilibrium complexity). *What is the computational complexity of finding a Leader-Follower equilibrium?*

The problem of finding a Leader–Follower equilibrium is in FP class.

**Exercise 15.0.7** (Leader-Follower equilibrium vs. Nash equilibrium). *Describe the relationships between Leader-Follower equilibrium and Nash equilibrium.*

The value of the leader in a Leader-Follower equilibrium is never strictly smaller than the value of the leader in the Nash equilibrium that is best one for the leader.
*Proof.* In the worst case for the leader, in the Leader-Follower equilibrium the leader plays her strategy of the Nash equilibrium that is the best for her, while the follower plays in pure strategies. Since the follower breaks ties in favor of the leader, the value of the leader is always (except for degenerate cases) larger

than the value she would receive in the Nash.
(Leader-Follower vs. Nash strategy). One could question whether in a Leader-Follower equilibrium the leader always plays the strategy she would play in a Nash equilibrium and the follower plays in pure strategy the best response that is the best for the leader. If this property were true, we could find a Nash equilibrium in polynomial time by, at first, finding the Leader-Follower equilibrium (in polynomial time), and then, given the leader's strategy, by finding the strategy that the follower would play in the Nash equilibrium (it can be easily showed that also this step can be done in polynomial time). This would show that $\mathsf{P} = \mathsf{PPAD}$. Instead, it can be showed that in a Leader-Follower equilibrium the leader could play actions that she would never play in any Nash equilibrium as remarked below.
(Leader-Follower vs. Nash strategy). In a Leader-Follower equilibrium the leader could play actions that she would never play in any Nash equilibrium and the value of the leader in a Leader-Follower equilibrium may be arbitrarily larger than the value in the best for her Nash equilibrium.

**Exercise 15.0.8** (Leader-Follower equilibrium formulation with multi-type follower). *Provide the mathematical programming formulation to find a Leader-Follower equilibrium with 2 players when the follower can be of multiple types.*

The Leader-Follower equilibrium,when the follower can be of different types, can be found by solving the following mathematical programming problem (where $k = |\Theta_f|$):

$$\begin{array}{llll}
\max\limits_{\overline{a}_{\theta_{f.1}},\dots,\overline{a}_{\theta_{f.k}}} \quad \max\limits_{s_l} & \sum\limits_{\theta_f \in \Theta_f} \Omega(\theta_f) \sum\limits_{a_l \in A_l} [U_l(a_l, \overline{a}_{\theta_f}) s_l(a_l)] & & \\
\quad s.t. & \sum\limits_{a_l \in A_l} s_l(a_l) & = 1 & \\
& s_l(a_l) & \geqslant 0 & \forall a_l \in A_l \\
& \sum\limits_{a_l \in A_l} [U_f(\overline{a}_f, a_l, \theta_f) s_l(a_l)] - \sum\limits_{a_l \in A_l} [U_f(a_f, a_l, \theta_f) s_l(a_l)] & \geqslant 0 & \forall a_f \in A_f, \theta_f \in \Theta_f
\end{array}$$

whose nature is Multi–Linear Program (Multi–LP).

**Exercise 15.0.9** (Finding a Leader-Follower equilibrium). *Given a 2-player normal-form game in which the follower can be of multiple types, find a Leader-Follower equilibrium by means of AMPL + GUROBI.*

**Exercise 15.0.10** (Leader-Follower equilibrium complexity). *What is the computational complexity of finding a Leader-Follower equilibrium when the follower can be of multiple types?*

(Multi-LP complexity).The Multi-LP has exponential complexity, enumerating an exponential number of LPs $O(m^k)$, each with a linear number of constraints $O(m \cdot k)$.
(Bayesian Leader-Follower complexity with single-type leader). n nThe problem of finding a Leader-Follower equilibrium when the follower can be of different types is in $\mathsf{FNP}$-hard.

## 3.3 EC 3.16

**Exercise 16.0.1** (Congestion game definition). *Provide the definition of Congestion game.*

A congestion game is a tuple $(N, M, (A_i)_{i \in N}, (c_j)_{j \in M}$ where:

- $N = \{1, 2, \ldots, n\}$ is the set of players;
- $M = \{1, 2, \ldots, m\}$ is the set of resources;
- $A_i \subseteq \wp(M)$ is the set of action of player $i$, where each action $a$ is a subset of the set of resources;
- $c_j : N \to \mathbb{R}$ is a function returning the cost related to resource $j$ when it is used by a given number of players.

**Exercise 16.0.2** (Pure-strategy Nash equilibrium in congestion games). *Prove that every finite congestion game admits at least a pure-strategy Nash equilibrium.*
(Pure-strategy Nash equilibrium existence). Every finite congestion game has a pure-strategy Nash equilibrium.
*Proof.* Let $\Phi : A \to \mathbb{R}$ be a function, called potential function, defined as:

$$\Phi(\mathbf{a}) = \sum_{j=1}^{m} \sum_{k=1}^{\mathsf{cong}_j(\mathbf{a})} c_j(k)$$

Initially, we show that if, given an action profile $\mathbf{a}$, a player, say $i$, changes its strategy, say from $a_i$ to $a_i'$, we have $\Delta\Phi = \Delta C_i$, where:

$$\Delta\Phi = \Phi\left(a_i', \mathbf{a}_{-i}\right) - \Phi(\mathbf{a})$$

$$\Delta C_i = C_i\left(a_i', \mathbf{a}_{-i}\right) - C_i(\mathbf{a})$$

That is, we can show that the difference in terms of costs incurred to player $i$ from switching from $a_i$ to $a_i'$ equals the difference of the potential function. The calculations are:

$$C_i\left(a_i', \mathbf{a}_{-i}\right) - C_i(\mathbf{a}) = \sum_{j \in a_i' \setminus a_i} c_j\left(\mathrm{cong}_j(\mathbf{a}) + 1\right) - \sum_{j \in a_i \setminus a_i'} c_j\left(\mathrm{cong}_j(\mathbf{a})\right) = \Phi_i\left(a_i', \mathbf{a}_{-i}\right) - \Phi_i(\mathbf{a})$$

Now we show that there is always at least a pure-strategy Nash equilibrium. Consider an algorithm in which at each step a single player changes her strategy by playing her best response (unless the current strategy is already a best response). At each step, the player that makes the move that reduces her cost by $\Delta C_i$ and consequently reduces the potential function. Since $\Phi$ can assume only finite values, the application of such an algorithm eventually returns a local minimum corresponding to an action profile in which no player can reduce further her cost by unilateral deviations. That is, a local minimum of $\Phi$ corresponds to a pure-strategy Nash equilibrium.

**Exercise 16.0.3** (Best-response paths). *Provide the definition of best-response paths and prove that in every finite congestion game all the best-response paths* *are finite.*

A best-response path is a sequence of action profiles such that every pair of consecutive action profiles $a$, $a'$ differ for the action of a single player that, in $a'$, plays her best response.

The proof follows a simple algorithm in which, given an action profile,

1. take a player that is not playing her best response
2. make her to play the best response
3. this leads to a switch in which the cost reduces by a finite value
4. repeat it, until a pure Nash is not reached
5. since at every switch the potential function reduces by a non-infinitesimal value and the potential function is lower bounded, the algorithm must terminate in finite time

**Exercise 16.0.4** (Finding best-response paths). *Given a congestion game, find a best-response path.*

TODO

**Exercise 16.0.5** (Potential functions). *Provide the definitions of potential functions and discuss the relationships between potential games and congestion games.*

(Exact potential function). A function $\Phi : A \to \mathbb{R}$ is an exact potential function for a game if, for every $a \in A$ and for every $i \in N$, $\Delta\Phi = \Delta C_i$.
(Weighted potential function). A function $\Phi : A \to \mathbb{R}$ is a weighted potential function for a game if, for every $a \in A$ and for every $i \in N$,$\Delta\Phi = \omega_i \Delta C_i$ for some positive $\omega_i$.
(Ordinal potential function). A function $\Phi : A \to \mathbb{R}$ is an ordinal potential function for a game if, for every $a \in A$ and for every $i \in N$, $(\Delta C_i < 0) \Rightarrow (\Delta\Phi < 0)$.

**Exercise 16.0.6** (Characterization of games admitting an exact potential function). *Discuss when a normal-form game admits an exact potential function and prove that.*

A game $(N, A, U)$ admits an exact potential function if and only if there are utility functions $\{U_i^c\}_{i \in N}$ and $\{U_i^d\}_{i \in N}$ such that:

- for every player $i \in N$ utility function $U_i(\mathbf{a})$ satisfies the property $U_i(\mathbf{a}) = U_i^c(\mathbf{a}) + U_i^d(\mathbf{a})$;
- the game $(N, A, U^c)$, $(U^c = \{U_1^c, \dots, U_n^c\}$, is a coordination game;
- the game $(N, A, U^d)$, where $(U^d = \{U_1^d, \dots, U_n^d\}$, is a dummy game.

Proof.(If) If each $U_i(\mathbf{a})$ can be written as $U_i(\mathbf{a}) = U_i^c(\mathbf{a}) + U_i^d(\mathbf{a})$, then function $\Phi(\mathbf{a}) = U_i^c(\mathbf{a})$ is a potential function. Indeed, $\Phi(a_i, \mathbf{a}_{-i} - \Phi(a_i', \mathbf{a}_{-i})$ equals the

difference in utility for player $i$ from strategy profile $(a_i', \mathbf{a}_{-i})$ to $(a_i, \mathbf{a}_{-i})$.
(Only if) Assume that a game admits a potential function $\Phi$. Set $U_i^c(\mathbf{a}) = \Phi(\mathbf{a})$ and $U_i^d(\mathbf{a}) = U_i^(\mathbf{a}) - \Phi(\mathbf{a})$. We have:

- the game with $U_i^c(\mathbf{a})$ is a coordination game since all the players have the same utility;
- the game with $U_i^c(\mathbf{d})$ is a dummy game since, for every $a_i, a_i' \in A_i$

$$U_i(a_i, \mathbf{a}_{-i}) - U_i(a_i', \mathbf{a}_{-i}) = \Phi(a_i, \mathbf{a}_{-i}) - \Phi(a_i', \mathbf{a}_{-i}) \Leftrightarrow$$
$$\Leftrightarrow \underbrace{U_i(a_i, \mathbf{a}_{-i}) - \Phi(a_i, \mathbf{a}_{-i})}_{U_i^d(a_i, \mathbf{a}_{-i})} = \underbrace{U_i(a_i', \mathbf{a}_{-i}) - \Phi(a_i', \mathbf{a}_{-i})}_{U_i^d\left(a_i', \mathbf{a}_{-i}\right)}$$

**Exercise 16.0.7** (Inefficiency bounds). *Provide the definition of Price of Anarchy and Price of Stability.*

(Socially best solution). The socially best solution $\mathbf{a}^{SB}$ is the action profile minimizing the social cost, i.e.

$$\mathbf{a}^{SB} \in \arg\min_{\mathbf{a}\in A} \sum_{i\in N} C_i(\mathbf{a})$$

(Socially best Nash equilibrium). The socially best Nash equilibrium $\mathbf{a}^{SBN}$ is the Nash equilibrium minimizing the social cost, i.e.

$$\mathbf{a}^{SBN} \in \arg \min_{\mathbf{a}\in A:\mathbf{a} \text{ is a Nash equilibrium}} \sum_{i\in N} C_i(\mathbf{a})$$

(Socially worst Nash equilibrium). The socially worst Nash equilibrium $\mathbf{a}^{SWN}$ is the Nash equilibrium maximizing the social cost, i.e.

$$\mathbf{a}^{SWN} \in \arg \max_{\mathbf{a}\in A:\mathbf{a} \text{ is a Nash equilibrium}} \sum_{i\in N} C_i(\mathbf{a})$$

(Price of Stability). Price of Stability (PoS) provides the inefficiency of the socially best Nash equilibrium w.r.t. the best social solution as:

$$\text{PoS} = \frac{\sum_{i\in N} C_i\left(\mathbf{a}^{SBN}\right)}{\sum_{i\in N} C_i\left(\mathbf{a}^{SB}\right)}$$

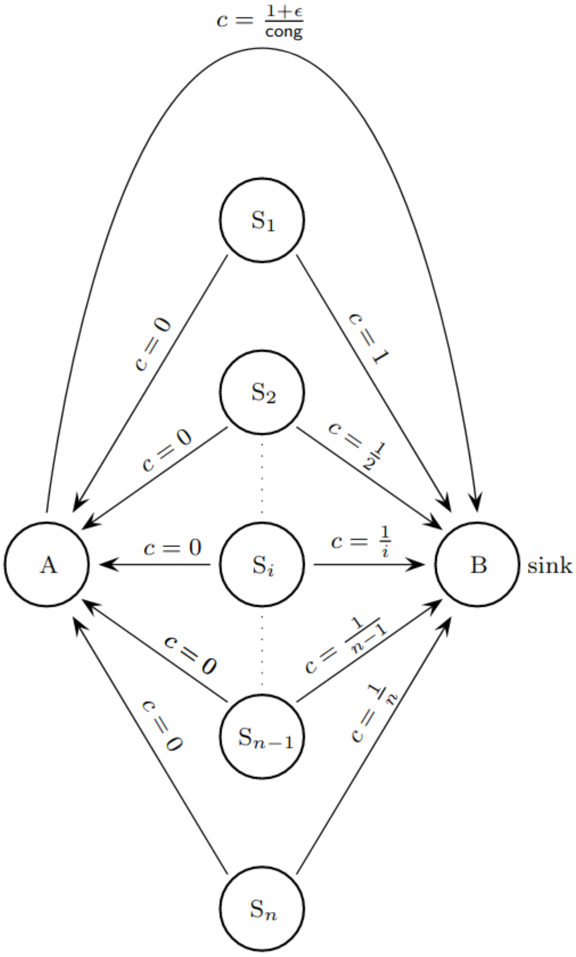
that is always larger than or equal to 1.
(Price of Anarchy). Price of Anarchy (PoA) provides the inefficiency of the socially worst Nash equilibrium w.r.t. the best social solution as:

$$\text{PoA} = \frac{\sum_{i\in N} C_i\left(\mathbf{a}^{SWN}\right)}{\sum_{i\in N} C_i\left(\mathbf{a}^{SB}\right)}$$

that is always larger than or equal to 1.

**Exercise 16.0.8** (PoA and PoS unboundness). *Provide an example in which PoA and PoS are unbounded.*

Consider the routing game with the following network:




There are $n$ players, each player $i$ starting from node $S_i$. There is only one sink that is node $B$. All the players using an edge (corresponding to a resource) share uniformly the cost. Notice that all the edges can be used by only one player, except the edge connecting node $A$ to node $B$ that, in principle, can be used by all the players.Let $k$ to be the users using such an edge, the cost of each user is $(1+\epsilon)/k$. The PoS of the above game in unbounded, asymptotically depending logarithmically on $n$. The socially best solution is when all the players move to $A$ and then to $B$, with a social cost of $1+\epsilon$. Each player has a cost of $(1+\epsilon)/n$. This is not a Nash equilibrium, since player $n$ strictly prefers to go directly to $B$, with a cost of $1/n$. Once player $n$ changed her action, player $n-1$ does the same. Indeed, $(1+\epsilon)/(n-1)$ is strictly larger than $1/(n-1)$. The same happens for all the other players. The only possible Nash equilibrium is when each player moves directly to $B$, with a social cost of $\sum_{i=1}^{n} \frac{1}{i}$. Therefore, we have that $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\ln(n))$ goes to infinity.

**Exercise 16.0.9** (Upper bound over best-response paths length). *Provide an upper bound over the length of best-response paths in singleton congestion games and prove it.*

(Singleton congestion game).A congestion game is called singleton if, for every player $i$ all the actions $a_i \in A_i$ contain a single resource.
(Upper bound over the best-response paths). In a singleton congestion game with $n$ players and $m$ resources, best-response paths have length $O(n?2m)$.
*Proof.* We know that any singleton congestion game with $n$ players and $m$ resources is equivalent to a game with $n$ players and $m$ resources in which the maximum cost is $nm$. It can be easily showed that $Phi$ is upper bounded by $n^2m$. Indeed, we have:

$$\Phi(\mathbf{a}) = \sum_{j=1}^{m} \sum_{k=1}^{\operatorname{cong}_j(\mathbf{a})} c_j(k) = \sum_{j=1}^{m} \sum_{k=1}^{\operatorname{cong}_j(\mathbf{a})} nm = n^2 m$$

Furthermore, along best-response paths $\Delta\Phi \leqslant 1$ since each pair of costs differs for at least 1. Since $\Phi$ strictly reduces along best-response paths, no best-response path can be longer than $n^2m$.

## 3.4 EC 3.17

**Exercise 17.0.1** (Social choice functions). *Provide the definition of social choice function. Provide one example.*

A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ assigns an outcome $x$ to each possible profile of players' types.

**Exercise 17.0.2** (Efficiency). *Provide the definitions of ex ante, ex interim, ex post efficiency for social choice functions. Answer to the following questions.*

- *Is the social choice function implemented by the Second-Price auction efficient in ex post?*
- *Provide an example of social choice function that is efficient in ex post, but not in ex interim.*
- *Provide an example of social choice function that is efficient in ex interim, but not in ex post.*

(Ex post efficiency). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is ex post efficient if there is no other social choice function $f'$ such that:

$$\begin{array}{ll} \forall i \in N, \forall \theta \in \Theta : & U_i\left(f'(\theta), \theta_i\right) \geqslant U_i\left(f(\theta), \theta_i\right) \quad \text{and} \\ \exists i \in N, \exists \theta \in \Theta : & U_i\left(f'(\theta), \theta_i\right) > U_i\left(f(\theta), \theta_i\right) \end{array}$$

(Ex interim efficiency). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is ex interim efficient if there is no other social choice function $f'$ such that:

$$\begin{array}{ll} \forall i \in N, \forall \theta_i \in \Theta_i : & \mathbb{E}_{\theta_{-i}}\left[U_i\left(f'(\theta), \theta_i\right)\right] \geqslant \mathbb{E}_{\theta_{-i}}\left[U_i\left(f(\theta), \theta_i\right)\right] \text{ and} \\ \exists i \in N, \exists \theta_i \in \Theta_i : & \mathbb{E}_{\theta_{-i}}\left[U_i\left(f'(\theta), \theta_i\right)\right] > \mathbb{E}_{\theta_{-i}}\left[U_i\left(f(\theta), \theta_i\right)\right] \end{array}$$

(Ex ante efficiency). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is ex ante efficient if $f$ there no other social choice function $f'$ such that:

$$\begin{array}{ll} \forall i \in N : & \mathbb{E}_\theta \left[U_i \left(f'(\theta), \theta_i\right)\right] \geqslant \mathbb{E}_\theta \left[U_i \left(f(\theta), \theta_i\right)\right] \text{ and} \\ \exists i \in N : & \mathbb{E}_\theta \left[U_i \left(f'(\theta), \theta_i\right)\right] > \mathbb{E}_\theta \left[U_i \left(f(\theta), \theta_i\right)\right] \end{array}$$

(Ex post efficiency).Consider the above social choice functions:

- Continuous voting with average allocation: it is ex post efficient. Indeed, moving away from the outcome chosen by $f$, the utility of at least one player decreases.
- Continuous voting with median allocation: it is ex post efficient. Indeed, moving away from the outcome chosen by $f$, the utility of at least one player decreases.
- Auction without payments: it is ex post efficient. Indeed, each outcome is Pareto efficient.
- First-price auction: it is not ex post efficient. Indeed $f'$ defined in the auction without payments Pareto dominates $f$ defined in first-price auction.
- Second-price auction: it is not ex post efficient. Indeed $f'$ defined in the auction without payments Pareto dominates $f$ defined in second-price auction.

**Exercise 17.0.3** (Individual rationality). *Provide the definitions of ex ante, ex interim, ex post individual rationality for social choice functions. Answer to the following questions.*

- *Is the social choice function implemented by the Second-Price auction individually rational in ex post?*
- *Provide an example of social choice function that is individually rational in ex interim, but not in ex post.*
- *Provide an example of social choice function that is individually rational in ex ante, but not in ex interim.*
- *Provide an example of social choice function that is individually rational in ex interim, but not in ex ante.*

(Ex post individually rationality). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is ex post individually rational if:

$$\forall i \in N, \forall \theta \in \Theta : \quad U_i \left(f(\theta), \theta_i\right) \geqslant \bar{U}_i \left(\theta_i\right)$$

where $\bar{U}_i(\theta_i)$ is the utility of player $i$ when her type is $\theta_i$ and she does not participate to the social choice.
(Ex interim individually rationality). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is ex interim individually rational if:

$$\forall i \in N, \forall \theta_i \in \Theta_i : \quad \mathbb{E}_{\theta_{-i}} \left[U_i \left(f(\theta), \theta_i\right)\right] \geqslant \bar{U}_i \left(\theta_i\right)$$

where $\bar{U}_i(\theta_i)$ is the utility of player $i$ when her type is $\theta_i$ and she does not participate to the social choice.
(Ex ante individually rationality). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is ex ante individually rational if:

$$\forall i \in N : \quad \mathbb{E}_\theta \left[U_i\left(f(\theta), \theta_i\right)\right] \geqslant \mathbb{E}_\theta \left[\bar{U}_i\left(\theta_i\right)\right]$$

where $\bar{U}_i(\theta_i)$ is the utility of player $i$ when her type is $\theta_i$ and she does not participate to the social choice.
(Ex post individually rationality). Consider the above social choice functions, under the assumption that $barU_i(\theta_i) = 0$ for every $i \in N$ and $\theta_i \in \Theta_i$:

- Continuous voting with average allocation: it is ex post individually rational. Indeed, for every outcome $x \in X$ the utility of each player is not negative.
- Continuous voting with median allocation: it is ex post individually rational. Indeed, for every outcome $x \in X$ the utility of each player is not negative.
- Auction without payments: it is ex post individually rational. Indeed, each outcome provides each player with a non-negative utility.
- First-price auction: it is ex post individually rational. Indeed, each outcome provides each player with a non-negative utility.
- Second-price auction: it is ex post individually rational. Indeed, each outcome provides each player with a non-negative utility.

**Exercise 17.0.4** (Dictatorship). *Provide the definition of dictatorship for social choice functions. Provide an example.*

A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is dictatorial if there is a player $i$ (said dictator) such that for every $\theta \in \Theta$ it holds:

$$U_i\left(f(\theta), \theta_i\right) \geqslant U_i\left(x, \theta_i\right) \quad \forall x \in X$$

## 3.5 EC 3.18

**Exercise 18.0.1** (Economic mechanism). *Provide the definition of economic mechanism.*

An economic mechanism is a tuple $(A_1, \ldots, A_n, X, g)$ where:

- $A_i$ is the set of actions of player $i$;
- $X$ is the set of outcomes;
- $g : A_1 \times \ldots \times A_n \to X$ is the outcome function.

**Exercise 18.0.2** (Implementation of a social choice function). *Provide the definition of implementation of a social choice function.*

An economic mechanism $(A_1, \ldots, A_n, X, g)$ implements a social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ if there is a pure-strategy equilibrium (according to some solution concept) strategy profile $(s_1^*, \ldots, s_n^*)$ of the Bayesian game induced by the economic mechanism such that:

$$g\left(s_1^*\left(\theta_1\right), \ldots, s_n^*\left(\theta_n\right)\right) = f\left(\theta_1, \ldots, \theta_n\right) \text{ for every } \left(\theta_1, \ldots, \theta_n\right) \in \Theta_1 \times \ldots \times \Theta_n$$

where $s_i^*\left(\theta_i\right)$ denotes the optimal strategy of player i when her type is $\theta_i$.

**Exercise 18.0.3** (Direct-revelation mechanism). *Provide the definitions of direct-revelation mechanism and indirect-revelation mechanism and provide an example for each one of them.*

(Direct (revelation) economic mechanism). Given a social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$, a direct (revelation) economic mechanism is a mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$
(Direct (revelation) economic mechanism). *A* direct mechanism is an economic mechanism in which the actions available to each player $i$ are given by the set of types of player $i$ (i.e., each player can only report a type from the set of the possible ones) and the outcome function is a social choice function.
(Indirect (revelation) economic mechanism). An economic mechanism that is not direct is said indirect.

**Exercise 18.0.4** (Incentive compatibility). *Provide the definition of incentive compatibility.*

A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$, is incentive compatible (or truthfully implementable) if the Bayesian game induced by the direct revelation economic mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$ has a pure equilibrium (according to some solution concept) $(s_1^*, \ldots, s_n^*)$ such that $s_i^*(\theta_i) = \theta_i$ for every player $i$ and type $\theta_i$. (And therefore the direct revelation economic mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$ implements $f$).
Incentive compatibility can be satisfied according to different solution concepts, among them:

- Dominant-strategy incentive compatibility (DSIC): if $(s_1^*, \ldots, s_n^*)$ where $s_i^*(\theta_i) = \theta_i$ for every player $i$ and type profile $\theta_i$ is a (weak) Dominant-Strategy equilibrium;
- Bayesian incentive compatibility (BNIC): if $(s_1^*, \ldots, s_n^*)$ where $s_i^*(\theta_i) = \theta_i$ for every player $i$ and type $\theta_i$ is a Bayes-Nash equilibrium.

**Exercise 18.0.5** (Checking incentive compatibility). *Given a social choice function, check whether it is incentive compatible.*

TODO

**Exercise 18.0.6** (Revelation principle for Dominant-Strategy equilibrium). *Provide the statement of the revelation principle and the proof.*

Given a social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$, if there is an economic mechanism $(A_1, \ldots, A_n, X, g)$ implementing $f$ in Dominant Strategy equilibrium, then $f$ is DSIC. Proof. If $(A_1, \ldots, A_n, X, g)$ implements $f$ in Dominant-Strategy equilibrium, then there is a strategy profile $(s_1^*, \ldots, s_n^*)$ such that:

$$g\left(s_1^*\left(\theta_1\right), \ldots, s_n^*\left(\theta_n\right)\right) = f\left(\theta_1, \ldots, \theta_n\right) \quad \forall \left(\theta_1, \ldots, \theta_n\right) \in \Theta$$

and

$$U_i\left(g\left(s_i^*\left(\theta_i\right), s_{-i}\left(\theta_{-i}\right)\right), \theta_i\right) \geqslant U_i\left(g\left(s_i\left(\theta_i\right), s_{-i}\left(\theta_{-i}\right)\right), \theta_i\right) \quad \forall i \in N, \theta_i \in \Theta_i, \theta_{-i} \in \Theta_{-i}, s_i, s_{-i}$$

The above condition is obviously satisfied for any restriction over $s_i$ and $s_{-i}$. Thus, we fix $s_i\left(\theta_i\right) = s_i^*\left(\hat{\theta}_i\right)$ (that is, we are restricting $s_i\left(\theta_i\right)$ to the set of the optimal strategies $s_i^*$ allowing player $i$ to select the strategy just by misreporting the type) and we fix $s_{-i}\left(\theta_{-i}\right) = s_{-i}^*\left(\theta_{-i}\right)$ (that is, we are restricting $s_{-i}\left(\theta_{-i}\right)$ to the set of the optimal strategies $s_{-i}^*\left(\theta_{-i}\right)$). We obtain:

$$U_i\left(g\left(s_i^*\left(\theta_i\right), s_{-i}^*\left(\theta_{-i}\right)\right), \theta_i\right) \geqslant U_i\left(g\left(s_i^*\left(\hat{\theta}_i\right), s_{-i}^*\left(\theta_{-i}\right)\right), \theta_i\right) \quad \forall i \in N, \theta_i \in \Theta_i, \theta_{-i} \in \Theta_{-i}, \hat{\theta}_i \in \Theta_i$$

Now we substitute $g\left(s_i^*\left(\theta_i\right), s_{-i}^*\left(\theta_{-i}\right)\right)$ with $f\left(\theta_i, \theta_{-i}\right)$ and $g\left(s_i^*\left(\hat{\theta}_i\right), s_{-i}^*\left(\theta_{-i}\right)\right)$ with $f\left(\hat{\theta}_i, \theta_{-i}\right)$, obtaining:

$$U_i\left(f\left(\theta_i, \theta_{-i}\right), \theta_i\right) \geqslant U_i\left(f\left(\hat{\theta}_i, \theta_{-i}\right), \theta_i\right) \quad \forall i \in N, \theta_i \in \Theta_i, \theta_{-i} \in \Theta_{-i}, \hat{\theta}_i \in \Theta_i$$

This condition exactly corresponds to the condition of incentive compatibility in dominant strategies.

**Exercise 18.0.7** (Gibbard-Satterthwaite impossibility theorem). *Provide the statement of the Gibbard-Satterthwaite impossibility theorem.*

Given a social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$, if

- $X$ is finite and $|X| \leqslant 3$,
- $f$ is onto (surjective),
- the utility functions of all the players cover the entire space of strict-preference utility functions, then the social choice function $f$ is DSIC if and only if it is dictatorial.

# Chapter 4

# EC4

## 4.1 EC 4.19

**Exercise 19.0.1** (Quasi-linear environment). *Provide the definition of quasi-linear environment and provide an example of social choice function in quasi-linear environment.*

A quasi-linear environment is characterized by:

- outcomes space: $X = \{k, p_1, \ldots, p_n) : k \in K, p_i \in \mathbb{R}\})$;
- utility functions: $U_i(x, \theta_i) = U_i((k, p_1, \ldots, p_n), \theta_i) = v_i(k, \theta_i) - p_i$, where $v_i : K \times \Theta_i \to \mathbb{R}$.

The set $K$ represents the set of allocations. The function $v_i(k, \theta_i)$ represents the valuation of allocation $k$ for type $\theta_i$ of player $i$. In principle, function $v_i$ can be any. The term $p_i$ represents the monetary payment of player $i$ to the mechanism. (The name "quasi-linear" is due to the form of the utility function, that is linear except for $v_i$ that can be any.)

**Exercise 19.0.2** (Properties in quasi-linear environment). *Provide the definition of the following properties in quasi-linear environment and provide an example of social choice function satisfying each property:*

- *weak and strict budget balance;*
- *allocation efficiency;*
- *maximality in the range.*

(Allocation efficiency). A social choice function $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$, where $k(\theta)$ is called allocation function, is allocatively efficient if $k(\theta)$ is defined as:

$$k(\theta) \in \arg\max_{k' \in K} \sum_{i \in N} v_i(k', \theta_i)$$

(Maximality in the range). A social choice function $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$ is maximal in its range $K' \subseteq K$ if $k(\theta)$ is allocatively efficient over $K'$.

((Weak) budget balance). A social choice function $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$ is weakly budget balanced if:

$$\sum_{i \in N} p_i(\theta) \geqslant 0 \quad \forall \theta \in \Theta$$

while it is strictly budget balanced if

$$\sum_{i \in N} p_i(\theta) = 0 \quad \forall \theta \in \Theta$$

(Auctions and (weak) budget balance).A social choice that is weakly budget balanced assures the auctioneer to be never (i.e., for every type profile $\theta \in \Theta$) in deficit and therefore the auctioneer has always a non-negative revenue.

**Exercise 19.0.3** (Dictatorship and quasi-linear environment). *Prove that no social choice function in quasi-linear environment can be dictatorial.*

No social choice function quasi-linear environment is dictatorial.
*Proof.* Assume by contradiction that there is a social choice function $f$ that is dictatorial in quasi-linear environment. This means that there is a dictator player $d$ such that:

$$U_d\left(f(\theta), \theta_d\right) \geqslant U_d\left(x, \theta_d\right) \quad \forall x \in X, \theta \in \Theta$$

where $U_d\left(f(\theta), \theta_d\right) = v_d\left(k(\theta), \theta_d\right) - p_d(\theta)$. However, it is sufficient to consider the outcome $\bar{x} = (k(\theta), p_1, \ldots, p_d - \epsilon, \ldots, p_n)$ where $\epsilon > 0$ to observe that $U_d\left(\bar{x}, \theta_d\right) \geqslant U_d\left(f(\theta), \theta_d\right)$ and therefore that we have a contradiction. $\square$

## 4.2 EC 4.20

**Exercise 20.0.1** (Groves mechanisms). *Provide the definition of Groves mechanisms.*

A direct revelation economic mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$ in which $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$ is a Groves mechanism if:

- $k(\theta) \in \arg\max_{k' \in K} \sum_{i \in N} v_i\left(k', \theta_i\right)$ where $K$ is the space of allocations,
- $p_i(\theta) = h_i\left(\theta_{-i}\right) - \sum_{j \in N \setminus \{i\}} v_j\left(k(\theta), \theta_j\right) \quad \forall i \in N$,

where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function on $\theta_{-i}$

**Exercise 20.0.2** (Groves Theorem). *Provide the statement of Groves theorem and provide a sketch of the proof.*

Any social choice function $f$ such that $(\Theta_1, \ldots, \Theta_n, X, f)$ is a Groves mechanism is DSIC.
*Proof.* Given the reported types of the all the players $\hat{\theta}$, the mechanism chooses the allocation as:

$$k(\hat{\theta}) \in \arg\max_{k' \in K} \sum_{i \in N} v_i\left(k', \hat{\theta}_i\right)$$

That is, the mechanism maximizes the social welfare given the types reported by the players. Focus on a generic player $i$, her utility is:

$$U_i\left(k\left(\hat{\theta}_i, \hat{\theta}_{-i}\right), \theta_i\right) = \underbrace{v_i\left(k\left(\hat{\theta}_i, \hat{\theta}_{-i}\right), \theta_i\right) + \sum_{j \in N \setminus \{i\}} v_j\left(k\left(\hat{\theta}_i, \hat{\theta}_{-i}\right), \hat{\theta}_j\right)}_{Z(\hat{\theta}_i)} - h_i\left(\hat{\theta}_{-i}\right)$$

where the term $Z(\hat{\theta}_i)$ can be controlled by player $i$ by misreporting her type, since by varying $\hat{\theta}_i$ the allocation chosen by the mechanism changes. However, it can be easily seen that the term $Z(\hat{\theta}_i)$ is maximizes when $\hat{\theta}_i = \theta_i$ independently of $\hat{\theta}_{-i}$.

**Exercise 20.0.3** (Application of Groves mechanisms). *Given a function $h_i(\theta_{-i})$ and a setting (composed of players, their types, valuation function, outcomes), apply the corresponding Groves mechanism. Additional question: prove that the given Groves mechanism is or not individually ration/weakly budget balanced in the given setting.*

TODO

**Exercise 20.0.4** (Groves mechanisms and individually rationality). *Is a Groves mechanism always individually rational (in ex post)? Provide, if there exist, two examples: in the first, a Groves mechanism is individually rational (in ex post), and, in the second, a Groves mechanism (potentially different from the previous one) is not individually rational (in ex post)?*

It is possible to design Groves mechanisms that are not individually rational. For example, in the case of single-item auction, it is sufficient to design $h_i(\hat{\theta}_{-i})$ as a large constant. This would make all the payments strictly positive and strictly larger than the players' valuations over the allocations. The definition of the appropriate $h_i(\hat{\theta}_i)$ allows the mechanism to be individually rational.

**Exercise 20.0.5** (Groves mechanisms and weak budget balance). *Is a Groves mechanism always weak budget balance (in ex post)? Provide, if there exist, two examples: in the first, a Groves mechanism is weak budget balance (in ex post), and, in the second, a Groves mechanism (potentially different from the previous one) is not weak budget balance (in ex post)?*

It is possible to design Groves' mechanisms that are not weakly budget balance. For example, in the case of single-item auction, it is sufficient to design $h_i(\hat{\theta}_{-i}) = 0$. This would make the payments of all the players except for the winner of the auction strictly negative and therefore the revenue of the mechanism would be strictly negative. The definition of the appropriate $h_i(\hat{\theta}_i)$ allows the mechanism to be weakly budget balanced.

**Exercise 20.0.6** (Weighted Groves mechanisms). *Provide the definition of weighted Groves mechanisms.*

A direct revelation economic mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$ in which $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$ is a weighted Groves mechanism if:

- $k(\theta_1, \ldots, \theta_n) \in \arg\max_{k' \in K'} \left(c_{k'} + \sum_{i \in N} w_i v_i(k', \theta_i)\right)$
- $p_i(\theta) = \frac{1}{w_i} h_i(\theta_{-i}) - \sum_{j \in N \setminus \{i\}} \frac{w_j}{w_i} v_j(k(\theta), \theta_j) - \frac{c_{k(\theta)}}{w_i} \quad \forall i \in N$

where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function on $\theta_{-i}$.

**Exercise 20.0.7** (Application of weighted Groves mechanisms). *Given a function $h_i(\theta_{-i})$ and a setting (com-posed of players, their types, valuation function, outcomes, and weights), apply the corresponding weighted Groves mechanism. Additional question: prove that the given weighted Groves mechanism is or not individually ration/weakly budget balanced in the given setting.*

TODO

**Exercise 20.0.8** (VCG mechanism). *Provide the definition of VCG mechanism.*

A direct revelation economic mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$ in which $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$ is a Clarke mechanism if:

- $k(\theta) \in \arg\max_{k' \in K} \sum_{i \in N} v_i(k', \theta_i)$,
- $p_i(\theta) = \max_{k' \in K_{-i}} \sum_{j \in N \setminus \{i\}} v_j(k', \theta_j) - \sum_{j \in N \setminus \{i\}} v_j(k(\theta), \theta_j) \quad \forall i \in N$,

where $K_{-i}$ is the set of allocations when player $i$ is not present.

**Exercise 20.0.9** (Application of VCG mechanisms). *Given a setting (composed of players, their types, valuation function, outcomes), apply the VCG mechanism. Additional question: prove that the VCG is or not individually ration/weakly budget balanced in the given setting.*

TODO

**Exercise 20.0.10** (VCG mechanism and individually rationality). *Is the VCG mechanism always individually rational (in ex post)? Provide, if there exist, two examples: in the first, the VCG mechanism is individually rational (in ex post), and, in the second, the VCG mechanism (potentially different from the previous one) is not individually rational (in ex post)?*

(Choice set monotonicity, no-negative externality and Clarke mechanism).The Clarke mechanism when:

- $\bar{U}_i(\theta_i) = 0$ for every $i \in N, \theta_i \in \Theta_i$;
- choice-set monotonicity is satisfied;
- no-negative externality is satisfied;

is individually rational in ex post.

**Exercise 20.0.11** (VCG mechanism and weak budget balance). *Is VCG mechanism always weak budget balance(in ex post)? Provide, if there exist, two examples: in the first, the VCG mechanism is weak budget balance(in ex post), and, in the second, the VCG mechanism (potentially different from the previous* *one) is not weak budget balance (in ex post)?*

(No-single-agent effect and Clarke mechanism).If no-single-agent effect property holds, the Clarke mechanism is weakly budget balanced and no payment is strictly negative.

## 4.3 EC 4.21

**Exercise 21.0.1** (Redistribution function). *Provide the definition of redistribution function.*

A redistribution function $r_i : \Theta_{-i} \to \mathbb{R}^+$ is a function returning the amount of monetary resources the mechanism redistributes to each player after it received the payments from the players. Thus, given an economic mechanism with payments $p_i(\theta)$, the monetary resources actually paid by player $i$ are $p_i(\theta) - r_i(\theta_{-i})$.

**Exercise 21.0.2** (Cavallo' redistribution). *Provide the definition of Cavallo's redistribution function.*

Let $p_i^{VCG}(\theta)$ be the payment used in the VCG mechanism, Cavallo's redistribution $r_i(\theta)$ is defined as:

$$r_i(\theta_{-i}) = \frac{1}{n} \min_{\bar{\theta}_i \in \Theta_i} \sum_{j \in N} p_j^{VCG}\left(\bar{\theta}_i, \theta_{-i}\right) =$$

$$= \frac{1}{n} \min_{\bar{\theta}_i \in \Theta_i} \left\{ \sum_{j \in N} \max_{k' \in K_{-j}} \sum_{l \in N \setminus \{j\}} v_l(k', \theta_l) - (n-1) \sum_{j \in N} v_j\left(k\left(\bar{\theta}_i, \theta_{-i}\right), \theta_j\right) \right\}$$

Basically, Cavallo's mechanism redistributes to each player $\frac{1}{n}$ of the total VCG payment that would result if minimizing over the space of types of the player (in the cases).
The VCG mechanism with Cavallo's redistribution is a Groves mechanism in which $h_i(\theta_{-i})$ is defined as:

$$h_i(\theta_{-i}) = \max_{k' \in K_i} \sum_{j \in N \setminus \{i\}} v_j(k', \theta_j) - r_i(\theta_{-i})$$

**Exercise 21.0.3** (VCG with Cavallo' redistribution properties). *Describe and prove the properties of the VCG mechanism with the Cavallo's redistribution function.*

When the VCG is individually rational and weak budget balanced, the VCG mechanism with Cavallo's redistribution satisfies the following properties: ex post individual rationality, weak budget balance, allocative efficiency, DSIC. Furthermore, among all the mechanisms satisfying the four properties is the one such that no mechanism yields greater payoffs to the players.
*Proof.* We consider each property singularly.
*Individual rationality.* Initially, we observe that, if the VCG is weakly budget balanced, then $\sum_{j \in N} p_j^{VCG}(\theta) \geqslant 0$ for every $\theta \in \bar{\Theta}$ and therefore, a fortiori,

$\min_{\bar{\theta}_i \in \Theta_i} \sum_{j \in N} p_j^{VCG}\left(\bar{\theta}_i, \theta_{-i}\right) \geqslant 0$ for every $\theta \in \bar{\Theta}$. This means that $r_i\left(\theta_{-i}\right) \geqslant 0$ for every $\theta_{-i} \in \bar{\Theta}_{-i}$. Thus, the total payments of the VCG mechanism with Cavallo's redistribution are smaller than the payments of the VCG mechanism. Therefore, if the VCG mechanism is individually rational, the same holds a fortiori for the VCG mechanism with Cavallo's redistribution.
*Weak budget balance.* By definition, we have $r_i\left(\theta_{-i}\right) = \frac{1}{n} \min_{\bar{\theta}_i \in \Theta_i} \sum_{j \in N} p_j^{VCG}\left(\bar{\theta}_i, \theta_{-i}\right) \leqslant \frac{1}{n} \sum_{j \in N} p_j^{VCG}(\theta)$. Therefore, it holds $\sum_{i \in N} r_i\left(\theta_{-i}\right) \leqslant \sum_{i \in N} \frac{1}{n} \sum_{j \in N} p_j^{VCG}(\theta) = \sum_{j \in N} p_j^{VCG}(\theta)$.
*Allocative efficiency.* It easily follows from the fact that the VCG mechanism with Cavallo's redistribution is a Groves mechanism.
*DSIC.* It easily follows from the fact that the VCG mechanism with Cavallo's redistribution is a Groves mechanism.

**Exercise 21.0.4** (Application of Cavallo' redistribution). *Given a setting and a mechanism, apply the Cavallo's redistribution function.*

TODO

**Exercise 21.0.5** (Strict-budget balanced mechanisms). *Provide a mechanism that is individually rational,weakly budget balanced, DSIC, and strict budget balanced. Can Groves mechanisms be strict budget balanced?*

TODO

## 4.4 EC 4.22

**Exercise 22.0.1** (Single-parameter linear environment). *Provide the definition of single-parameter linear environment.*

A single-parameter linear environment (a special subclass of quasi-linear environment) is characterized by:

- outcomes space: $X = \{(k, p_1, \ldots, p_n) : k \in K, p_i \in \mathbb{R}\}$;
- utility functions: $U_i\left(x, \theta_i\right) = U_i\left(\left(k, p_1, \ldots, p_n\right), \theta_i\right) = \theta_i \cdot \rho_i(k) - p_i,$ where $\rho_i : K \to \mathbb{R}$ and $\Theta_i \subset \mathbb{R}$

The set $K$ represents the set of allocations. The utility of each player $i$ is factorized w.r.t. type $\theta_i$ and a function $\rho_i(k)$ defined only on the allocation $k$ (and not on the type.) In principle, function $\rho_i$ can be any. (The name "linear" is due to the form of the utility function, that is linear in the type and in the payment.)
**Exercise 22.0.2** (Weak monotonicity). *Provide the definition of weak monotonicity for single-parameter linear environment.*

An allocation function $k(\theta)$ is weakly monotonic if:

$$\theta_i > \theta_i' \Rightarrow \rho_i\left(k\left(\theta_i, \theta_{-i}\right)\right) \geqslant \rho_i\left(k\left(\theta_i', \theta_{-i}\right)\right) \quad \forall i \in N, \theta_i, \theta_i' \in \Theta_i, \theta_{-i} \in \Theta_{-i}$$

**Exercise 22.0.3** (Myerson mechanisms). *Provide the definition of Myerson mechnaisms for single-parameter linear environment.*

A direct revelation economic mechanism $(\Theta_1, \ldots, \Theta_n, X, f)$ in which $f(\theta) = (k(\theta), p_1(\theta), \ldots, p_n(\theta))$ is a Myerson mechanism if: - $k(\theta)$ is an arbitrary weakly monotone allocation function,

$$\bullet p_i(\theta) = \theta_i \cdot \rho_i(k(\theta)) - \int_0^{\theta_i} \rho_i\left(k\left(\theta_i', \theta_{-i}\right)\right) \cdot d\theta_i' + h_i\left(\theta_{-i}\right)$$

where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function on $\theta_{-i}$.

**Exercise 22.0.4** (Myerson theorem). *Provide the statement of the Myerson theorem for dominant-strategy incentive compatibility in single-parameter linear environment and prove it.*

Any social choice function $f$ such that $(\Theta_1, \ldots, \Theta_n, X, f)$ is a Myerson mechanism is DSIC.

## 4.5 EC 4.23

**Exercise 23.0.1** (Knapsack auction). *Provide the definition of knapsack auction.*

A knapsack problem, say KNAPSACK, is defined as:

- $I = \{1, \ldots, n\}$ is a set of items;
- $\boldsymbol{S} = \{s_1, \ldots, s_n\}$ where $s_i \in \mathbb{N}^+$ is the size of item $i$
- $W = \{w_1, \ldots, w_n\}$ where $w_i \in \mathbb{N}^+$ is the value of item $i$
- $C$ e $\mathbb{N}^+$ is the capacity of the knapsack.

The goal of the knapsack problem is:

$$\begin{aligned} \arg\max_{I' \subseteq I} \sum_{i \in I} w_i \\ \text{s.t. } \sum_{i \in I} s_i \leqslant C \end{aligned}$$

**Exercise 23.0.2** (Knapsack auction approximation algorithm). *Describe a monotone algorithm to approximate the optimal allocation of the knapsack auction with a ratio of $\frac{1}{2}$ and prove the theoretical bound on the approximation ratio.*

The algorithm, say ApxKnapsack, develops in the following steps:

1. sort all the items in decreasing order in $\frac{w_i}{s_i}$ and then relabel the items such that item 1 is the first in the order and item $n$ is the last one;
2. repeatedly add the items to $I'$ according to the above order while the capacity constraint is not violated and call $|I'| = n'$ (this means that $\sum_{i \leqslant n'} s_i \leqslant C$ and, if $n' < n, C < \sum_{i \leqslant n'+1} s_i\Big)$

3. return $\max\left\{\sum_{i\leqslant m'} w_i, w_{n'+1}\right\}$

The complexity of the algorithm is $O\left(n\log_2(n)\right)$.

**Exercise 23.0.3** (Application of Knapsack auction). *Given a knapsack auction setting, find the optimal allocation returned by the VCG mechanism and find the allocation returned by the $\frac{1}{2}$ approximation monotone algorithm.*

TODO

## 4.6 EC 4.24

**Exercise 24.0.1** (Combinatorial auction). *Provide the definition of combinatorial auction.*

A combinatorial auction is defined as:

- $N = \{1,\ldots,n\}$ is the set of players;
- $I = \{1,\ldots,m\}$ is the set of items;
- $S = \wp(I)\backslash\varnothing$ is the set of possible bundles of items;
- $\theta_i = \{\theta_{i,s} : \theta_{i,s} \in \mathbb{R}^+, s \in S\}$ is the type of player $i$, composed of a parameter for every possible bundle $k = \{(s,i) : s \in S, i \in N$, and for every s there is at most an $i\}$ is an allocation specifying the set of allocated bundles and for each bundle the player who won it, while $K$ is the set of allocations;
- $v_i\left(k,\theta_i\right) = \sum_{s:(s,i)\in k}\theta_{i,s}$ is the valuation function of player $i$

The problem of finding the optimal allocation is called COMB − AUCTION.

**Exercise 24.0.2** (Combinatorial auction approximation algorithm). *Describe a monotone algorithm to approximate the optimal allocation of a combinatorial auction.*

The algorithm, say ApxCombAuction, develops in the following steps:

1. sort all the players on the basis of their unique non-zero $\theta_{i,s}$ in decreasing order in $\frac{\theta_{i,s}}{\sqrt{|s|}}$;
2. scan all the players according to the above order and for each player $i$ allocate the bundles such that $\theta_{i,s} > 0$ to player $i$ if $s$ does not contain any item appearing in some bundle previously allocated;
3. return the allocation found.

The complexity of the algorithm is $O(n\log_2(n))$.

**Exercise 24.0.3** (Application of combinatorial auction). *Given a setting of combinatorial auction, find the allocation returned by the* $\frac{1}{\sqrt{|I|}}$*-approximation monotone algorithm, where is the set of items. Additional question: find the Myerson payments when the approximation algorithm is used.*
TODO

## 4.7 EC 4.25

**Exercise 25.0.1** (Double auction). *Provide the formal model of double auction.*

**Exercise 25.0.2** (McAfee mechanism). *Describe the McAfee mechanism for double auctions and show that it is a particular Myerson mechanism.*

**Exercise 25.0.3** (Application of double auction). *Given a setting of double auction, apply the McAfee mechanism, returning the allocation chosen by the mechanism and the the payments to the players.*

## 4.8 EC 4.26

**Exercise 26.0.1** (Sponsored search auction). *Provide the formal model of sponsored search auction.*

**Exercise 26.0.2** (VCG mechanism). *Describe the VCG mechanism.*

**Exercise 26.0.3** (Application of the VCG mechanism). *Given a setting of sponsored search auctions, apply the VCG mechanism, returning the allocation chosen by the mechanism and the pay-per-click payments to the players.*

**Exercise 26.0.4** (Application of the VCG mechanism with estimated qualities). *Given a setting of sponsored search auctions with estimated qualities, apply the VCG mechanism, returning the allocation chosen by the mechanism and the pay-per-click payments to the players.*

## 4.9 EC 4.27

**Exercise 27.0.1** (Dominant strategy implementation). *Given:*

- *a set of outcomes,*
- *a set of players,*
- *a finite set of types per player,*
- *the valuation function of every player,*

*write in AMPL the MOD file and the DAT file to design in automatic fashion a mechanism that may be:*

- *truthful in dominant strategies,*
- *individually rational,*
- *weakly budget balanced,*
- *allocatively efficient,*

*and that might maximize/minimize the revenue of the auctioneer.*

**Exercise 27.0.2** (Bayes-Nash implementation). *Given:*

- *a set of outcomes,*
- *a set of players,*
- *a finite set of types per player,*
- *the valuation function of every player,*

*write in AMPL the MOD file and the DAT file to design in automatic fashion a mechanism that may be:*

- *truthful in Bayes-Nash equilibrium,*
- *individually rational,*
- *weakly budget balanced,*
- *allocatively efficient,*

*and that might maximize/minimize the revenue of the auctioneer.*