DEPARTMENT OF ECONOMICS,
MANAGEMENT AND STATISTICS

UNIVERSITY OF MILAN – BICOCCA

DEMS WORKING PAPER SERIES

Further results on verification problems in extensive-form games

Nicola Gatti, Mario Gilli, Fabio Panozzo

No. 347 – July 2016

Dipartimento di Economia, Metodi Quantitativi e Strategie di Impresa
Università degli Studi di Milano - Bicocca
http://dems.unimib.it/

Further results on verification problems in extensive–form games

Nicola Gatti[a], Mario Gilli[b], Fabio Panozzo[a]

[a] *Dipartimento di Elettronica, Informazione e Bioningegneria, Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italy*

[b] *Department of Economics, Management, and Statistics, Università degli Studi di Milano Bicocca, Piazza dell'Ateneo Nuovo 1, 20126 Milano, Italy*

Abstract

The computational study of games is receiving increasing attention both in game theory and computer science. The challenge is distinguishing computationally tractable problems (also said *easy*), admitting polynomial–time algorithms, from the intractable ones (also said *hard*). In this paper, we focus on extensive form games, as the computational problems defined on such games are largely unexplored. We study the problem (aka *verification problem*) of certifying that a solution given in input is an equilibrium according to different refinements for extensive form games as the input change. We show that, when the input is a realization plan strategy profile (i.e., strategies for the sequence form representation), deciding whether the input is a Subgame Perfect Equilibrium or is a part of a Sequential Equilibrium is NP-hard even in two-player games (we conjecture the same holds also for Quasi Perfect Equilibrium). This means that there is no polynomial–time algorithm unless P = NP, but it is commonly believed that P ≠ NP. Subsequently, we show that in two–player games, when the input is a behavioral strategy profile, there is a polynomial–time algorithm deciding whether the input is a Quasi–Perfect Equilibrium, and a simple variation of the algorithm decides whether the input is part of some Sequential Equilibrium (in games with three or more players, the problem is known to be NP–hard for both Quasi–Perfect Equilibrium and Sequential Equilibrium). Finally, we show that, when the input is an assessment, there is a polynomial–time algorithm deciding whether the input is a Sequential Equilibrium regardless the number of players.

Key words: Efficient algorithms, extensive–form refinements

1. Introduction

The computational analysis of strategic interaction is receiving increasing attention both in computer science and in game theory, as witnessed by Papadimitriou (2015). The reasons underlying this approach towards game theory are various, however two are particularly relevant for this paper. First, the search for autonomous software agents able to act optimally, with a specific focus on formal methods to theoretically guarantee behavior optimality. Well–known successful applications based on these methods are, e.g., in physical security by Basilico et al. (2010) and Korzhyk et al. (2011), in poker by Gilpin et al. (2007) and billiard games by Archibald and Shoham (2009), and in economic transactions by Sandholm (2007) and Jordan et al. (2010). Second, the theory of computational complexity is well suited to study the inherent complexity of calculating or of verifying a strategic equilibrium. While standard game theory provides mathematical tools to model strategic interaction situations and characterize the appropriate solution concepts, however, it does not provide computational tools to find solutions. This problem, commonly called equilibrium computation, is instead central in computer science, whose aim includes assessing the complexity of finding an exact or approximate solution, designing exact or approximate algorithms, and evaluating the application of the algorithms in practical settings, see, e.g., Shoham and Leyton-Brown (2008). In particular, for anyone to play according to a specific equilibrium concept, it must be *verifiable* whether a strategy profile is an equilibrium of the game. The study of the verification problem is important, because it is necessary to certify that a software agent is playing or not an optimal strategy. In the case where the verification problem is computationally intractable[1], we can neither confirm or deny that the behavior of a player is optimal, making the adoption of autonomous players critical and pushing one to resort to new approximate solution concepts whose verification problem is tractable. Some computational results are known for Nash Equilibria (NE), in particular finding an NE of a given finite strategic–form game is hard, while to verify whether a given strategy profile is NE is not. However, while the verification of an NE is easy, few results are known about the verification problem of equilibrium refinements for extensive–form games and they are mostly negative as shown by Hansen et al. (2010).

The main contribution of this paper is to provide new positive computational results on solution concepts for extensive form games.[2] We focus on the problem of verifying whether a solution given in input is a Quasi Perfect Equilibrium (QPE) or a Sequential Equilibrium (SE), and we investigate how the specification of the input, according to the different representations of extensive–form games, affects the hardness of the problem. More precisely, we focus on the extensive–form by Kreps and Wilson (1982) and the sequence form by von Stengel (1996) representations, while we do not consider the normal form since its size is exponential in the size of the game tree, and therefore no efficient algorithm can exist. Actually, when only the behavioral strategy profile is given in input, it is known that the verification problem is hard with three or more players for SE and QPE by Hansen et al. (2010), but no result is known with two players. Furthermore,

[1]In computer science, a problem is said tractable or easy if there is a polynomial–time algorithm that solves it and intractable otherwise. Furthermore, an algorithm is said efficient if it requires polynomial time.

[2]A survey on Nash equilibrium refinements for extensive–form games can be found in Kohlberg and Mertens (1986). See also van Damme (2002) and Hillas and Kohlberg (2002).

when an assessment is given in input, to verify whether it is an SE, only an algorithm that is exponential in the worst case is known, see Kohlberg and Reny (1997), but no complexity results are available. This seems to suggest that the verification properties of NE for strategic form games are not shared by refinements based on the extensive form. Contrary to this view, in this paper,[3] we show that some problems are easy when the input is specified in behavioral strategies. More precisely, we prove:

- when the input is a realization plan strategy profile (i.e., strategies for the sequence form), then it is unlikely that there is a polynomial–time algorithm able to decide whether the input is part of some Subgame Perfect Equilibrium (SPE) even for a two–player game (technically speaking, we show that such a problem is NP–hard); the result extends to SE, while we conjecture that it holds also for QPE;
- when the input is a behavioral strategy profile, then there is an efficient algorithm certifying whether the strategy profile is a QPE for two–player games;
- when the input is a behavioral strategy profile, then there is an efficient algorithm certifying whether the strategy profile is part of some SE for two–player games; furthermore, in the case it is part of some SE, there is an efficient algorithm finding the players' consistent beliefs from the strategies;
- when the input is an assessment, then there is an efficient algorithm certifying whether the assessment is an SE with an arbitrary number of players.

We believe these results are not only important in themselves, they also show the importance of the input representation, in particular whether we consider behavioral strategies or realization plans, and once more the crucial relevance of beliefs to verify extensive form refinements.

The rest of the paper is structured as follows. Next section provides a quick review of the concepts used in the paper and of the main results proved by the algorithmic approach to game theory. In Section 3, we review the main concepts of extensive form and sequence form games, and the equilibrium refinements used, to make the paper self-contained. In Section 4, we show that, when the input is specified in terms of strategies defined over the sequence form, the verification problem for SPE and SE is NP–hard. In Section 5, we use the sequence form to study the verification problem for QPE and SE with two players when the input is a behavioral strategy profile. In Section 6, we study the verification problem for SE when the input is an assessment. Section 7 concludes the paper. In Appendix, we report part of the proofs of section 6.

[3] A preliminary version of some of the results presented in this paper appears in Gatti and Panozzo (2012).

2. Brief Review and Related Work[4]

This section provides a quick review of the most basic concept of computational theory just to make the paper self–contained and to explain its contribution. The basic game theoretic model instead is presented in Section 3.

Computational problems are situations where we are given some input data, e.g. games or strategy profiles, and we want to return a solution meeting certain criteria and we ask for an algorithm to provide the answer, see Arora and Barak (2009). An algorithm is an unambiguous sequence of elementary steps that applied on any input of the problem eventually stops with the correct solution of the problem. We consider a computational problem solved satisfactory when there exists a good algorithm, but what are good algorithms for solving a problem? We want the algorithm to always return a correct solution, but we are especially interested in how fast the algorithm returns a solution *vis a vis* the size of the input. An algorithm is considered computational efficient if its running time is at most a polynomial function of the size of the input. Of course, the same computational problem may admit both efficient and inefficient algorithms, however the theory of computational complexity aims to analyze the inherent complexity of the problem itself: how fast is the fastest correct algorithm for a given problem? P denotes the class of decision problems (problems that require a "yes" or "no" answer) that admit at least one efficient (polynomial–time) algorithm. Clearly to prove that a decision problem is in P is enough to explicitly provide an algorithm proving a bound on its running time, as we do in this paper referring to the class of linear mathematical programming problems. On the other hand, it is extremely rare to prove that a problem is not in P, because it would require to refer to any possible algorithms. Thus, usually to prove that a problem is hard, computer scientists use an indirect way proving results of the form: "If this problem can be solved efficiently, then so can every member of the class X" and the problem is said to be X–hard and X–complete if, additionally, the problem has also been shown to lie in X. The strength of such a hardness result depends on the class used. The class for which problems are most often shown to be hard is NP, i.e. the class of all decision problems such that, if the answer to a problem input is "yes", then there exists an algorithm that, given a "yes" certificate, can be used to check soundness of the certificate in polynomial time. Hence NP contains P, and it is generally considered unlikely that $\mathsf{P} = \mathsf{NP}$, actually complexity theory is based on this conjecture. Most decision problems of interest turn out to be either in P or NP–hard. According to the previous definition, NP–complete problems are the hardest among the NP problems. There are other classes of problems different from NP. For instance, while problems in NP usually ask whether a solution exists, their functional versions, whose class is denoted by FNP, ask not only if it exists but what its value is if it does. The subclass TFNP of FNP contains all the total functional problems (i.e., problems where a solution is guaranteed to exist), see Meggido and Pa-

[4] This review section is based on Papadimitriou (2015) and Conitzer and Sandholm (2008).

padimitriou (1989). A complexity class that received a lot of attention in game theory is PPAD and has been introduced in Papadimitriou (1994). It is a subclass of TFNP based on total functions that can be shown to be total by a parity argument. Examples of PPAD–complete problems include, among the others, finding NE, fixed points in Brouwer functions, and Arrow–Debreu equilibria in markets.

The problem of verifying whether a given solution is an equilibrium according to some solution concept is a decision problem in which the input is composed of a game and a solution. The crucial question is therefore whether a verification problem belongs to P or to NP–hard. Tab. 1 summarizes the main computational complexity results on the problem of verifying whether a given solution is an equilibrium according to some NE refinement solution concepts for extensive form games.

Solution concept	Input	Number of players	
		2	≥ 3
NE	realization–plan strategies	P	P
SPE	behavioral strategies	P	P
	realization–plan strategies	NP–hard (*)	NP–hard (*)
SE	assessment	P (*)	P (*)
	behavioral strategies	P (*)	NP–hard (Hansen et al. (2010))
	realization–plan strategies	NP–hard (*)	NP–hard (Hansen et al. (2010))
NFPE	behavioral strategies	P	NP–hard (Hansen et al. (2010))
QPE	behavioral strategies	P (*)	NP–hard (Hansen et al. (2010))
EFPE	behavioral strategies		NP–hard (Hansen et al. (2010))
–PE	behavioral strategies and perturbation	P	P
NFPrE	behavioral strategies		NP–hard (Hansen et al. (2010))
EFPrE	behavioral strategies		NP–hard (Hansen et al. (2010))
–PrE	behavioral strategies and perturbation	P	P

Table 1: Computational complexity results for equilibrium verification; the results with '(*)' are originally provided in the present paper.

Nash Equilibria and Subgame Perfect Equilibria. The verification problem for NE is known to be easy, even when the input is given in sequence form, just requiring the verification that a finite number of constraints are satisfied. The verification problem for SPE is easy when the input is completely specified with behavioral strategies, while in the present paper we show that the problem is NP–hard when the input is given in sequence form.

Perfection based equilibria. The verification problem for Normal Form Perfect Equilibrium (NFPE) is in P with two players and can be accomplished by checking whether or not the actions played with strictly positive probability are weakly dominated. In the former case, the solution is not a NFPE. With more than two players it is shown to be NP–hard in Hansen et al. (2010). The verification problem for Quasi Perfect Equilibria (QPE) is NP–hard with three or more players, see Hansen et al. (2010). In the present paper, we show instead that with two players there is an efficient algorithm. The verification problem for Extensive Form Perfect Equilibria (EFPE) is open with two players. With three or more players the problem is shown to be NP–hard in Hansen et al. (2010). The problem of verifying whether a strategy profile is a Perfect Equilibrium for a given perturbation is easy for all the Perfect Equilibrium solution concepts (i.e., NFPE, QPE, EFPE). Indeed, it requires one only to check whether a finite number of constraints are satisfied.

Sequential Equilibria. The only known result on the verification for SE when the input is specified as assessment can be found in Kohlberg and Reny (1997). The authors provided a finite–step algorithm to verify whether an assessment is an SE, but, as they state it, the algorithm is exponential in the worst case. In the present paper, we show that there is an efficient algorithm for an arbitrary number of players. Furthermore, it is known that, when the input is partially specified, the verification problem for an SE can be harder. More precisely, Hansen et al. (2010) show that with three or more players, verifying whether there is an SE with a given strategy is NP–hard. In the present paper, we show instead that with two players there is an efficient algorithm for the verification problem. Furthermore, we show that when the input is specified in realization plan strategies the verification problem is NP–hard.

Properness based equilibria. The computational complexity of verifying a Normal Form Proper Equilibrium (NFPrE) with two players is open and the only available algorithm can be found Belhaiza et al. (2012) that solves a number of mixed quadratic programs, while with three or more players the problem is shown to be NP–hard in Hansen et al. (2010). The recent result presented in Sørensen (2012), showing that computing a NFPrE with two players is PPAD–complete and can be performed by using Lemke's algorithm after a specific transformation, could represent an interesting tool to assess the complexity of verifying a NFPrE with two players. Also the problem of verifying an Extensive Form Proper Equilibrium (EFPrE) with two players is open. The problem of verifying whether a strategy profile is a Proper Equilibrium for a given perturbation is easy for all the proper equilibrium solution concepts (i.e., NFPrE, EFPrE). Indeed, it requires one only to check whether a finite number of constraints are satisfied. Note that recent results show that searching for a QPE with two players given a specific perturbation is PPAD–complete, see Miltersen and Sørensen (2010), and therefore the corresponding verification problem is in P. Nevertheless, the problem of searching for a QPE without any given perturbation is open and therefore we have no insight on the complexity of the corresponding verification problem.

3. Extensive and sequence form games, and solution concepts

Within the non-cooperative approach, most game theoretic models fall into one of three general mathematical formulations or some natural extension of them. These formulations are called the normal form, the extensive form and the sequence form, each based on a specific representation of the space of the players' strategies. Extensive form games provide a richer representation than normal form games, and extensive form equilibrium refinements exploit the sequential structure of decision making being described explicitly. In the following subsections we review notations and definitions first for extensive form then for sequence form games. Finally, we review the equilibrium concepts we use in this paper, Quasi Perfect and Sequential Equilibria.

3.1. Extensive form game definition

The notation and terminology in this section is due to Kreps and Wilson (1982) and to McKelvey and McLennan (1996). The reader should refer to these papers if the current presentation seems excessively terse.

The collection

$$\{T, \prec; A, \alpha; H; I, \iota\}$$

defines a finite extensive form. These objects have the following characteristics:

1. The set of nodes is T, a finite set, strictly partially ordered by a relation $\prec$ that represents precedence. The pair $(T, \prec)$ must be an arborescence: if $P(t) = \{x \in T | x \prec t\}$ is the set of predecessors of t, $P(t)$ is completely ordered by $\prec$ for all t, so each node is reached in only one way. The set of initial nodes is $W = \{w \in T | P(w) = \varnothing\}$. For non–initial nodes t we define the immediate predecessor of t to be $p_1(t) = \max P(t)$, and, proceeding inductively, we define the *nth* predecessor of t to be $p_n(t) = p_1(p_{n-1}(t))$ for those nodes with $p_{n-1}(t) \notin W$. We adopt the convention that $p_0(t) = t$ for all $t \in T$. The number of predecessors of t is the integer $l(t)$ such that $p_{l(t)}(t) \in W$. The set of immediate successors of t is $S(t) = p_j^{-1}(t) = \{t' \in T | p_1(t') = t\}$, the set of terminal nodes is $Z = \{z \in T | S(z) = \varnothing\}$, and the set of non–terminal nodes is $X = T \setminus Z$. For each $x \in X$ the set of terminal successors of x is $Z(x) = \{z \in Z | x \prec Z\}$.
2. The set of actions is A, a finite set, and the function $\alpha : T \setminus W \to A$ labels each non–initial node with the last action taken to reach it.
3. The set of information sets is H, a partition of X, the idea being that the player choosing an action at $h \in H$ does not know which node in h has occurred. For non–terminal x and t, $t' \in S(x)$ we require that $\alpha(t) \neq \alpha(t')$, and for $x' \in H(x)$ we require that $\alpha(S(x')) = \alpha(S(x))$, i.e., the set of available actions is the same at all nodes in $H(x)$. Thus we can write $A(h)$ to denote $\alpha(S(x))$, $x \in h$, the set of actions available at h. For notational convenience we assume that $\{A(h) | h \in H\}$ is a partition of A.
4. The set of players is I, a finite set, and the function $\iota : H \to I$ indicates the player responsible for choosing an action at each information set. For each player i let $H^i = \iota^{-1}(i)$, and let $A^i = \cup_{h \in H^i} A(h)$. By defining $\iota(x) = \iota(H(x))$ one can regard ι as a function on X.

To obtain an extensive form game we add to the extensive form a specification of the players' utilities assigned to the terminal nodes and the probabilities assigned to the initial nodes.

5. For each player i, the payoff function $u^i : Z \to \mathbb{R}$ assigns a real–valued von Neumann–Morgenstern utility to each outcome. We denote a specification of the payoffs by $\boldsymbol{u} = (u^i(z)) \in \mathbb{R}^{\#(I) \times \#(Z)}$, where $\#(\cdot)$ denotes the number of elements in a set $\cdot$.

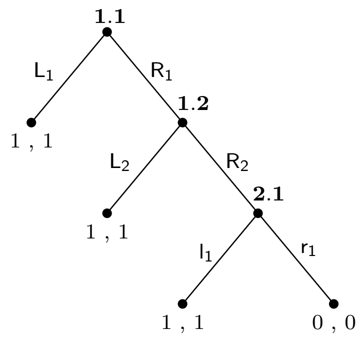

Figure 1: Example of two–player perfect–information extensive–form game (we denote by $i.j$ the j–th information set of player i, while $\mathsf{L}_1, \mathsf{R}_1, \mathsf{L}_2, \mathsf{R}_2$ are actions available to player 1 and $\mathsf{l}_1, \mathsf{r}_1$ are actions available to player 2).

6. Player i's initial assessment ρ is a probability measure on the set W of initial nodes with $\rho(w) > 0$ for all $w \in W$.

We report an example of extensive form game in Fig. 1.

A pure strategy for player i is an assignment $s^i : H^i \to A$ such that $s^i(h) \in A(h)$. This specifies what action player i will take each time it is her turn to choose, based on the information that he possesses. The set of player i's pure strategies is denoted by S^i. One defines a mixed strategy σ^i for player i as a probability distribution over the set of her pure strategies: $\sigma^i \in \Sigma^i = \Delta(S^i)$ where $\Delta(\cdot)$ denotes the set of all probability measure on the set $\cdot$. A behavioral strategy for player i is a function $\pi^i : A^i \to [0,1]$ satisfying the requirement that $\sum_{a \in A(h)} \pi^i(a) = 1$ for all $h \in H^i$, i.e., π^i is a collection of probability measures on the sets of available actions at the information sets controlled by i. The set of behavioral strategies for i is Π^i, and $\Pi = \times_{i \in I} \Pi^i$ is the set of behavioral strategy profiles. Since $\{A^i | i \in I\}$ is a partition of A, a behavioral strategy profile $\boldsymbol{\pi} \in \Pi$ can be regarded as a function from A to $[0,1]$ satisfying $\sum_{a \in A(h)} \pi^i(a) = 1$ for all $h \in H$, and π^i is said to be strictly positive if $\pi^i(a) > 0$ for all $a \in A^i$. Let $\Pi^{i,0}$ be the set of i strictly positive behavioral strategies. Given a behavioral strategy profile $\boldsymbol{\pi}$, the probability that node t occurs is $\mathbb{P}^{\boldsymbol{\pi}}(t) = \rho\left(p_{l(t)}(t)\right) \prod_{l=0}^{l(t-1)} \pi\left(\alpha\left(p_l(t)\right)\right)$; for each $h \in H$ we can also define "conditional" probability $\mathbb{P}^{\boldsymbol{\pi}}(\cdot|h)$ over Z:

$$\mathbb{P}^{\boldsymbol{\pi}}(z|h) \in \begin{cases} \{0\} & \text{if } z \notin Z(h) \\ \left\{ \frac{\mathbb{P}^{\boldsymbol{\pi}}(p_n(z))}{\sum_{x' \in h} \mathbb{P}^{\boldsymbol{\pi}}(x')} \prod_{l=0}^{n-1} \pi\left(\alpha\left(p_l(l)\right)\right) \right\} & \text{if } \sum_{x' \in H(x)} \mathbb{P}^{\boldsymbol{\pi}}(x') > 0, z \in Z(h) \ \& \ p_n(z) \in h \\ [0,1] & \text{otherwise} \end{cases}.$$

When convenient, we will use $\mathbb{P}^{\boldsymbol{\pi}}(h'|h)$ as a shorthand for $\mathbb{P}^{\boldsymbol{\pi}}(Z(h')|h)$. Then, given a behavioral strategy profile $\boldsymbol{\pi}$ the expected utility for $\iota(h)$, conditional on h being reached, is now defined, naturally, to be

$$\mathbb{E}^{\boldsymbol{\pi}}\left[u^{\iota(h)}(z)\,|h\right] = \sum_{z\in Z} u^{\iota(h)}(z)\,\mathbb{P}^{\boldsymbol{\pi}}(z|h)\,.$$

We focus on games with perfect recall, see Fudenberg and Tirole (1991), where each player recalls all her own previous actions and all her previous information. Perfect recall places non–trivial constraints over the structure the extensive form. We omit the formal definition description here,[5] not being necessary for our work. A useful implication of perfect recall is that it implies that there is no path between nodes of the same information set. The most important implication of perfect recall is Kuhn Theorem,[6] which states that under perfect recall, mixed and behavioral strategies are outcome equivalent, i.e. that for any mixed strategy profile there exists a behavioral strategy profile that generates the same probability distribution on final nodes.[7] Therefore, in extensive form games with perfect recall, behavioral strategies are commonly used.

Table 2 summarizes the notation on extensive form games introduced so far.

3.2. Sequence form

The terminology in this section is due to von Stengel (1996), the notation is however adapted to the previous section. The reader should refer to von Stengel (1996) if the current presentation seems excessively terse.

The extensive form with nodes, information sets, moves, chance probabilities and payoffs gives a complete picture of the strategic situation that is modeled, however strategic behavior as usually modeled might be huge in size and difficult to analyze, hence quite inefficient. In strategic form, strategies are fully specified as functions from each possible information set to actions, while in reduced strategic form, pure strategies are only partially specified, by omitting actions at information sets that cannot be reached because of previous own choices. The sequence form, introduced in von Stengel (1996), goes further because strategies are replaced by sequences that specifies a player's choice only along a path of the tree. More precisely, a $i's$ sequence is a set of $i's$ moves on the unique path from the root to a $i's$ node. Hence a sequence form is a compact and computational efficient representation for extensive form contexts. The starting point to introduce this new model is exactly the fact that in practice to describe a behavioral strategy in the original game tree it is required an enormous increase in the number of necessary parameters.[8] By definition, a pure strategy specifies a move for any information set of the player, so the number of pure strategies is often exponential in the size of the extensive game.

---

[5]A formal definition is also in Kreps and Wilson (1982).

[6] Kuhn (1953)

[7]The *vice versa* is trivial and does not require perfect recall.

[8]Remember we are assuming perfect recall so that according to Kuhn Theorem mixed and behavioral strategy are outcome equivalent.

Object	**Definition**	**Description**
T		set of nodes
$\prec$		"precedes"
$P(t)$	$\{x \in T \vert x \prec t\}$	predecessors of t
W	$\{w \in T \vert P(w) = \varnothing\}$	initial nodes
$p_1(t)$	$\max P(t)$	immediate predecessor of t
$p_n(t)$	$p_1(p_{n-1}(t))$	nth predecessor of t
$l(t)$	$p_{l(t)}(t) \in W$	number of predecessor of t
$S(t)$	$p_1^{-1}(t)$	immediate successor of t
Z	$\{z \in T \vert S(z) = \varnothing\}$	terminal nodes or outcomes
X	$T \setminus Z$	non–terminal nodes
A		actions
$\alpha(t)$		actions preceding t
H		information sets h
$H(x)$		information set of x
$A(h)$	$\alpha(S(x)), x \in h$	actions available at $h \in H$
I		players
$\iota(h)$		player in control at h
H^i	$\iota^{-1}(i)$	information sets of i
A^i	$\cup_{h \in H^i} A(h)$	player i's actions
u^i	$u^i : Z \to \mathbb{R}$	player i's utility function
$s^i \in S^i$	$s^i : H^i \to A$ s.t. $s^i(h) \in A(h)$	player i's pure strategy
σ^i	$\sigma^i \in \Delta\left(S^i\right)$	player i's mixed strategy
$\pi^i \in \Pi^i$	$\pi^i : A^i \to [0,1]$ s. t. $\sum_{a \in A(h)} \pi^i(a) = 1,\ \forall h \in H^i$	player i's behavioral strategy
$Z(h)$	$\forall x \in h \ \ \{z \in Z \vert x \prec z\}$	terminal successors of h
$h \preceq k$	$Z(h) \subseteq Z(k),\ h, k \in H$	h precedes k
$\mathbb{P}^\pi(t)$	$\rho\left(p_{l(t)}(t)\right) \prod_{l=0}^{l(t-1)} \pi(\alpha(p_l(t)))$	probability that node $t \in T$ occurs
$\mathbb{P}^\pi(z \vert h)$	$\begin{cases} 0 & z \notin Z(h) \\ \frac{\mathbb{P}^\pi(p_n(z))}{\sum_{x' \in h} \mathbb{P}^\pi(x')} \prod_{l=0}^{n-1} \pi(\alpha(p_l(l))) & \sum_{x' \in H(x)} \mathbb{P}^\pi(x') > 0, z \in Z(h), p_n(z) \in h \\ [0,1] & \text{otherwise} \end{cases}$	conditional probability over Z given h
$\mathbb{E}^\pi\left[u^{\iota(h)}(z) \vert h\right]$	$\sum_{z \in Z} u^{\iota(h)}(z) \mathbb{P}^\pi(z \vert h)$	$\iota(h)$ expected utility, conditional on h

Table 2: Notation on extensive form games.

Formally, the set of pure strategy is $S^i = \times_{h\in H^i} A(h)$ so that $\#(S^i) = \times_{h\in H^i}\#(A(h))$, i.e. the number of pure strategies is exponential in the number of information sets. This implies that a player's expected payoff when a behavioral strategy profile $\boldsymbol{\pi}$ is played is $U^i(\boldsymbol{\pi}) := \sum_{z\in Z} u^i(z)\,\rho\big(p_{l(z)}(z)\big)\prod_{l=0}^{l(z-1)} \pi(\alpha(p_l(z)))$, i.e. it involves products of probabilities, hence the resulting polynomials for computing equilibria is theoretically and practically difficult, while the expected payoff using the sequence form is multilinear.

The sequence form is a matrix scheme similar to normal form games where strategies are described as sequences of consecutive moves. Rather than planning a move for each information set, a sequence is a succession of actions that can lead to terminal nodes or not. Formally

Definition 3.1. *For each node $t\in T$, a* sequence *$q(t)\in Q^i$ of player $i=\iota(t)$ is a set of consecutive actions $a\in A^i$ on the path from the root w to t.*

Note that a sequence is defined as a set. Let $Q = \times_{i\in I} Q^i$ be the set of all sequences. A sequence can be terminal (e.g., $q = \mathsf{L}_1$ in Fig. 1), if, combined with some sequence of the opponents, it leads to a terminal node, or non–terminal (e.g., $q = \mathsf{R}_1$ in Fig. 1), if it cannot lead to any terminal node for each opponents' sequence. Each player has a fictitious initial sequence, denoted by $\varnothing$ and called empty sequence. Furthermore, note that by perfect recall, every node in an information set h defines the same sequence of actions for that player. Then, the following definition is well posed.

Definition 3.2. *Given a sequence $q(h)\in Q^i$ leading to some information set $h\in H$, we say that sequence q'* extends *$q(h)$ (and we denote it by $q' = q(h)a$) if the last action of q' is some action $a\in A(h)$.*

Thus, the set Q^i of sequences of player i can be represented as

$$Q^i = \{\varnothing\}\cup\big\{q(h)a | h\in H^i, a\in A(h)\big\},$$

hence $\#(Q^i) = 1 + \sum_{h\in H^i}\#(A(h))$, i.e. the size of the set of sequences is linear in the size of the game tree. Since in this paper we frequently exploit the correspondence between actions $a\in A$ and sequences $q\in Q$, we denote by $a(q)$ the last action of sequence q. Moreover, in this model, behavioral strategies are replaced by realization probabilities of sequences according to the following definition:

Definition 3.3. *A function $r^i : Q^i \to \mathbb{R}^+$ is a* realization plan *for player $i\in I$ if and only if it satisfies the following linear restrictions:*

$$\begin{aligned} r^i(\varnothing) &= 1 \\ -r^i(q(h)) + \textstyle\sum_{a\in A(h)} r^i(q(h)a) &= 0 \quad \text{for} \quad h\in H^i. \end{aligned}$$

We denote a realization plan profile by $\mathbf{r} = [\mathbf{r}^1, \ldots, \mathbf{r}^{\#(I)}]$, then the constraints on realization plans can be conveniently described as $F^i \cdot \mathbf{r}^i = \mathbf{f}^i$, where F^i is an opportune matrix with entries in $\{-1, 0, 1\}$ and $\mathbf{f}^i$ is an opportune vector with '1' in the first entry and '0' in all the other entries.

Then, we define the payoff for the sequence model.

Definition 3.4. *The* payoff function *associated to a sequence profile is defined as follows*

$$\forall \mathbf{q} \in \times_{i\in I} Q^i \quad \mathbf{v}(\mathbf{q}) = \begin{cases} \mathbf{u}(z) & \text{if } \mathbf{q} \text{ leads to } z \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

Note that the payoff function associated to a sequence profile is well defined because with perfect recall in a game tree to each node, including final nodes, is associated a unique path $\mathbf{q} \in \times_{i\in I} Q^i$. The expected payoff vector $\mathbf{v}(\mathbf{r}) \in \mathbb{R}^{\#(I)}$ is defined as $\mathbf{v}(\mathbf{r}) = \sum_{\mathbf{q}\in\mathbf{Q}} \mathbf{v}(\mathbf{q}) \prod_{i\in I} r^i(q^i)$ or, in a compact way, as $\mathbf{v}(\mathbf{r}) = (\mathbf{r}^i)^{\intercal} V^i \mathbf{r}^{-i}$, where player i's payoff is represented by a sparse multi–dimensional matrix V^i, specifying the value associated to every combination of players terminal sequences leading to an outcome, and $\mathbf{r}^j$ is the vector of j's realization plans. Since there are at most as many sequences as nodes in the game tree, the number of the sequences is linear in the size of the game tree, while the number of pure strategies can be exponential. Exactly, excluded the empty sequence, we have one sequence $q \in Q^i$ per action $a \in A^i$. So the size of the payoff matrix is also linear if it represented sparsely, while the payoff matrix of a normal form is usually full.

Finally, we are able to define the sequence form.

Definition 3.5. *The* sequence form *of an extensive game is given by the sets of sequences Q^i, the payoff functions $v^i : \times_{i\in I} Q^i \to \mathbb{R}$ and the constraints $F^i \cdot \mathbf{r}^i = \mathbf{f}^i$, i.e. the collection*

$$\left\{Q^i; v^i; F^i \cdot \mathbf{r}^i = \mathbf{f}^i | i \in I\right\}.$$

Thus, the sequence form is an abstraction like the normal form, however it has the advantage of a size linear in the size of the game tree, and the disadvantage of a less intuitive selection of sequences by realization plans, finitely described by the matrix constraints. In the next section, we will review how from the realization probabilities for the sequences one can reconstruct a behavior strategy and *vice versa*.

Table 3 summarizes the notation on sequence form introduced so far. To conclude, we provide an example of sequence form representation.

Example 3.6. *Consider the game depicted in Fig. 1, the sequence form utility bimatrix*

Object	Definition	Description
$q(t)$		set of consecutive actions $a \in A^i$ on the path from the root w to t
$q' = q(h)a$	$\{x \in T \mid P(x) = \varnothing\}$	extension of $q(h) \in Q^i$ with $a \in A(h)$
Q^i	$\{\varnothing\} \cup \{q(h)a \mid h \in H^i, a \in A(h)\}$	set of sequences of player i
$r^i : Q^i \to \mathbb{R}^+$		realization plan of player i
$F^i \cdot \mathbf{r}^i = \mathbf{f}^i$	$\begin{cases} r^i(\varnothing) = 1 \\ -r^i(q(h)) + \sum_{a \in A(h)} r^i(q(h)a) = 0,\ h \in H^i \end{cases}$	linear restriction on realization plans
$v^i : \times_{i \in I} Q^i \to \mathbb{R}$	$\mathbf{v}(\mathbf{q}) = \begin{cases} \mathbf{u}(z) & \text{if } \mathbf{q} \text{ leads to } z \\ \mathbf{0} & \text{otherwise.} \end{cases}$	player i's payoff function

Table 3: Notation on sequence form games.

and an example of strategies are:

player 2

player 1		$\varnothing$	l_1	r_1
	$\varnothing$			
	L_1	1, 1		
	R_1			
	$\mathsf{R}_1\mathsf{L}_2$	1, 1		
	$\mathsf{R}_1\mathsf{R}_2$		1, 1	0, 0

$$\mathbf{r}^1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{r}^2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

the constraints $F^1 \cdot \mathbf{r}^1 = \mathbf{f}^1$ *and* $F^2 \cdot \mathbf{r}^2 = \mathbf{f}^2$ *are:*

$$F^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \end{bmatrix}, \mathbf{f}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, F^2 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \end{bmatrix}, \mathbf{f}^2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

3.3. Relationships between the representations

We briefly describe the relationships among behavioral strategies and realization plans. Given a sequence form strategy, a behavioral strategy can be derived as follows[9]:

$$\forall a \in A(h) \quad \pi^i(a) = \begin{cases} \frac{r^i(q(h)a)}{r^i(q(h))} & \text{if } r^i(q(h)) > 0 \\ \pi^i(a) \geq 0 \text{ s.t. } \sum_{a \in A(h)} \pi^i(a) & \text{if } r^i(q(h)) = 0 \end{cases}$$

vice versa, given a behavioral strategy, a sequence q is played with probability

$$r^i(q) = \prod_{a \in q} \pi^i(a).$$

This construction is similar to the Kuhn Theorem stating that in a game with perfect recall, any mixed strategy can be replaced by an outcome equivalent behavior strategy. Then, more than one behavior strategy π^i may define the same realization plan r^i: this

[9]See Proposition 3.4 in von Stengel (1996).

is the case if the information set $h \in H^i$ cannot be reached when π^i is played, that is, if $r^i\left(q\left(h\right)\right) = 0$. In particular, more than one pure strategy may define the same realization plan, even if there is an intuitive correspondence between realization plans with integral values zero or one and pure strategies in the reduced normal form of the extensive game, as in the reduced normal form, any two pure strategies that differ only in choices at irrelevant information sets are identified.

A player can play the game optimally by appropriately choosing the realization probabilities for her sequences, and the expected payoff is linear in these variables. This is their key advantage over behavior strategies: the latter are also small in number and can be characterized by linear equations (as any probabilities), but the expected payoff usually involves products of behavior strategies. Therefore, using the resulting polynomials for computing equilibria is theoretically and practically much more difficult than using sequence form and realization plans. On the other hand, mixed and behavioral strategies have different degrees of expressiveness w.r.t. realization plan , e.g., sequence form strategies, differently from behavioral and normal form strategies, do not specify the actions a player would play at the information sets that are reached with a probability of zero, and mixed strategies, differently from behavioral strategies, correlate the actions a single player would play at every information set. As we show below, these differences in expressiveness play an important role in the definition of the solution concepts and in their verification.

3.4. Equilibrium refinements for extensive form games

It is well known that the concept of NE is not satisfactory with extensive form games, allowing players to play non–plausible actions. The reason for such apparent implausibility of some actions of an NE $\widehat{\pi}$ is that these actions are implemented if and only there is a deviation from the equilibrium path, i.e. these actions are implemented in information sets $H\left(x\right)$ such that $\sum_{x' \in H(x)} \mathbb{P}^{\widehat{\pi}}\left(x'\right) = 0$, hence the *ex ante* maximization of an NE does not restrict players' behavior at these information sets. The common approach to these problem, pioneered by Selten (1975) and that leads to a huge part of the refinement literature, is to assume trembles over behavioral strategies so that there are no are $H\left(x\right)$ such that $\sum_{x' \in H(x)} \mathbb{P}^{\widehat{\pi}}\left(x'\right) = 0$, i.e. the attempt is to eliminate unsatisfactory equilibria assuming that players make mistakes with small vanishing probability, restricting attention to the limits of the corresponding equilibria. Trembles are usually captured by perturbations. Call $\varepsilon^i(a,h) > 0$ the perturbation (in terms of probability) over action $a \in A\left(h\right)$ at information set $h \in H^i$, such that $\lim\ \varepsilon^i(a,h) = 0$. We denote by $\mathbf{E}^i$ the perturbation over all the player i's actions. A perturbed behavioral strategy profile $\boldsymbol{\pi}(\varepsilon)$ of $\boldsymbol{\pi}$ is a vector of behavioral strategies where $\pi^i(a|h,\varepsilon) \geq \varepsilon^i(a,h)$ for all $h \in H^i$, $a \in A\left(h\right)$ and $\lim\limits_{n\to\infty} \boldsymbol{\pi}(\varepsilon) = \boldsymbol{\pi}$. Selten assumes each of these probabilities $\pi^i(h,\varepsilon)$ and $\varepsilon^i(h)$ to be independent of each other and also to be independent of the corresponding probabilities of the other players. Thus, if a player i intends to play the behavior strategy π^i, she will actually play the behavior strategy $\pi^i(\varepsilon)$ such that for any $h \in H^i$ and $a \in A\left(h\right)$,

$\pi^i(a|h,\varepsilon) = (1-\varepsilon(a,h))\,\pi^i(a|h) + \varepsilon(a,h)\,\pi^{i,0}(a|h)$, with $\pi^{i,0} \in \Pi^{i,0}$. Obviously, given these mistakes all information sets are reached with positive probability. Now, we can provide the definition of Perfect and of Quasi Perfect Equilibrium.

Definition 3.7. *A strategy profile* $\boldsymbol{\pi}(\varepsilon)$ *is an* ε–Perfect Equilibrium *if and only if*

1. $\boldsymbol{\pi}(\varepsilon) \in \Pi^0$, *i.e. each action is taken with strictly positive probability,*
2. *if* $\mathbb{E}^{\left(a,\boldsymbol{\pi}(\varepsilon)^{-\iota(h)}\right)}\left[u^{\iota(h)}(z)\,|h\right] < \mathbb{E}^{\left(a',\boldsymbol{\pi}(\varepsilon)^{-\iota(h)}\right)}\left[u^{\iota(h)}(z)\,|h\right]$ *then* $\pi^{\iota(h)}(a|h,\varepsilon) \leq \varepsilon(a,h)$ *for all* $h \in H$, $a \in A(h)$, $\iota(h) \in I$.

Definition 3.8. *A strategy profile* $\boldsymbol{\pi}$ *is a* Perfect Equilibrium *if and only if there exists a sequence* $\{\varepsilon(a,h)\}_{n\in\mathbb{N}}$ *such that* $\boldsymbol{\pi}(\varepsilon)$ *is an* ε*–Perfect Equilibrium and* $\boldsymbol{\pi}(\varepsilon) \underset{n\to\infty}{\to} \boldsymbol{\pi}$.

The underlying idea is that players try to maximize whenever they have to move, but each time they make a mistake with vanishing probability. Note that for $\boldsymbol{\pi}$ to be a Perfect Equilibrium (PE), it is sufficient that $\boldsymbol{\pi}$ can be rationalized by some sequence of vanishing trembles, it is not necessary that $\boldsymbol{\pi}$ be robust against all possible trembles. Motivated by the consideration that a player may be more concerned with mistakes of others than with her own, van Damme (1984) introduces the concept of a Quasi Perfect Equilibrium (QPE). Here each player follows a strategy that at each node specifies an action that is optimal against mistakes of other players, keeping the player's own strategy fixed throughout the game. Mertens (1992) has argued that this concept of QPE is to be preferred to PE. In Lemma 1 of van Damme (1984), the author provides an alternative equivalent definition of PE that facilitates comparison with the notion of QPE.

Definition 3.9. *A strategy profile* $\boldsymbol{\pi}$ *is a* Perfect Equilibrium *if and only if there exists a sequence* $\{\boldsymbol{\pi}_n \in \Pi^0\}_{n\in\mathbb{N}} \underset{n\to\infty}{\to} \boldsymbol{\pi}$ *such that* $\pi^{\iota(h)}(h) \in \arg\max \mathbb{E}^{\left(\widehat{\pi}^{\iota(h)},\boldsymbol{\pi}_n^{-\iota(h)}\right)}\left[u^{\iota(h)}(z)\,|h\right]$ *for all* $n \in \mathbb{N}$, $h \in H$, $\iota(h) \in I$.

A QPE is defined as a behavior strategy profile that prescribes at every information set a choice that is optimal against mistakes of the other players. Before the formal definition, we need a further notation: $\left(\pi^i(h), \widehat{\pi}^i(k)\right) = \begin{cases} \widehat{\pi}^i & k \geq h \\ \pi^i & \text{otherwise} \end{cases}$ so that $\left(\pi^i(h), \widehat{\pi}^i(k), \pi^{-i}\right)$ is a well defined strategy profile.

Definition 3.10. *A strategy profile* $\boldsymbol{\pi}$ *is a* Quasi Perfect Equilibrium *if and only if there exists a sequence* $\{\boldsymbol{\pi}_n \in \Pi^0\}_{n\in\mathbb{N}} \underset{n\to\infty}{\to} \boldsymbol{\pi}$ *such that for all* $n \in \mathbb{N}, h, k \in H, \iota(h) \in I$:

$$\left(\pi^{\iota(h)}(h), \pi^{\iota(h)}(k)\right) \in \arg\max \mathbb{E}^{\left(\left(\pi^{\iota(h)}(h),\widehat{\pi}^{\iota(h)}(k)\right),\boldsymbol{\pi}_n^{-\iota(h)}\right)}\left[u^{\iota(h)}(z)\,|h\right].$$

Basically, a strategy profile is perfect when it is optimal even when there are perturbations. The common interpretation of the perturbations is based on the idea that players do not perfectly play a strategy, but they can tremble with a very small probability. In a QPE every player takes into account the opponents' trembles, but not own. Every game admits at least one QPE and for every combination of $\mathbf{E}^i$ there is a potentially different QPE.

Kreps and Wilson (1982) propose to eliminate irrational behavior at unreached information sets in a somewhat different way than Selten does. They propose to extend the applicability of rationality at out–of–equilibrium information sets by explicitly specifying beliefs (i.e. conditional probabilities) at each information set so that posterior expected payoffs can always be computed. Of course, players' beliefs should be consistent with the strategies actually played (i.e. beliefs should be computed from Bayes' rule whenever possible) and they should respect the structure of the game. Kreps and Wilson ensure that these conditions are satisfied by deriving the beliefs from a sequence of completely mixed strategies that converges to the strategy profile in question.

Consider the following basic definitions required for defining our equilibrium refinements for extensive form games.

Definition 3.11. *A* system of beliefs *is defined as a function* $\mu : X \to [0,1]$ *such that* $\sum_{x \in h} \mu(x) = 1$ *for all* $h \in H$*.*

Interpret $\mu(x)$ as the conditional probability assigned by $\iota(h)$ to $x \in h$ if h is reached.

Definition 3.12. *An* assessment *is a pair* $(\mu, \boldsymbol{\pi})$ *consisting of a behavioral strategy profile* $\boldsymbol{\pi}$ *and a system of beliefs* μ*.*

Given an assessment $(\mu, \boldsymbol{\pi})$, for each $h \in H$ we can define "conditional" probability $\mathbb{P}^{\mu,\boldsymbol{\pi}}(\cdot|h)$ over Z in the obvious fashion:

$$\mathbb{P}^{\mu,\boldsymbol{\pi}}(z|h) = \begin{cases} 0 & \text{if } z \notin Z(h) \\ \mu(p_n(z)) \prod_{l=0}^{n-1} \pi(\alpha(p_l(l))) & \text{if } z \in Z(h) \wedge p_n(z) \in h \end{cases}.$$

When convenient, we will use $\mathbb{P}^{\mu,\boldsymbol{\pi}}(h'|h)$ as a shorthand for $\mathbb{P}^{\mu,\boldsymbol{\pi}}(Z(h')|h)$.

Definition 3.13. *The* system of beliefs generated by $\boldsymbol{\pi}$ *is the function*

$$\mu(x) = \begin{cases} \frac{\mathbb{P}^{\boldsymbol{\pi}}(x)}{\sum_{x' \in H(x)} \mathbb{P}^{\boldsymbol{\pi}}(x')} & \text{if} \sum_{x' \in H(x)} \mathbb{P}^{\boldsymbol{\pi}}(x') > 0 \\ \mu \in [0,1] & \text{otherwise} \end{cases}.$$

Note that $\sum_{x \in h} \mu(x) = 1$ for all $h \in H$, and in general beliefs are conditional probabilities on the elements $x \in H(x)$ given the information set $H(x)$, calculated using Bayes rule and a behavioral strategy profile $\boldsymbol{\pi}$. Moreover, if $\boldsymbol{\pi}$ is strictly positive then $\mathbb{P}^{\boldsymbol{\pi}}(t) > 0$ for all t, since ρ is also strictly positive, so the system of beliefs generated by $\boldsymbol{\pi}$ is uniquely characterized by the function

$$\mu : X \to [0,1] \text{ such that } \mu(x) = \frac{\mathbb{P}^{\boldsymbol{\pi}}(x)}{\sum_{x' \in H(x)} \mathbb{P}^{\boldsymbol{\pi}}(x')}.$$

Definition 3.14. *The set of* consistent assessments *is* $\Psi = cl(\Psi^0)$, *where* Ψ^0 *is the set of pairs* $(\mu, \boldsymbol{\pi})$ *where* $\boldsymbol{\pi} \in \Pi^0$ *and* μ *is generated by* $\boldsymbol{\pi}$.

Therefore, $(\mu, \boldsymbol{\pi})$ is a consistent assessment if and only if there is a sequence $\{(\mu_n, \boldsymbol{\pi}_n)\}$ converging to $(\mu, \boldsymbol{\pi})$, where $\boldsymbol{\pi}_n \in \Pi^0$ and μ_n is generated by $\boldsymbol{\pi}_n$ for each n. The definition of consistency then resorts to perturbed strategies.

Given an assessment $(\mu, \boldsymbol{\pi})$ the expected utility for $\iota(h)$, conditional on h being reached, is now defined, naturally, to be

$$\mathbb{E}^{\mu,\boldsymbol{\pi}}\left[u^{\iota(h)}(z)\,|h\right] = \sum_{z \in Z} u^{\iota(h)}(z)\,\mathbb{P}^{\mu,\boldsymbol{\pi}}(z|h).$$

Definition 3.15. *The set of* sequentially rational assessments *is the set of pairs* $(\mu, \boldsymbol{\pi})$ *such that for all* h *and all* $\overline{\boldsymbol{\pi}}$ *such that* $\overline{\pi}^j = \pi^j$ *for* $j \neq \iota(h)$,

$$\mathbb{E}^{\mu,\boldsymbol{\pi}}\left[u^{\iota(h)}(z)\,|h\right] \geq \mathbb{E}^{\mu,\overline{\boldsymbol{\pi}}}\left[u^{\iota(h)}(z)\,|h\right].$$

In words, taking the beliefs as fixed, no player prefers at any point to change her part of the behavioral strategy profile π. Now, we are able to define a Sequential Equilibrium (SE).

Definition 3.16. *A* Sequential Equilibrium *is an assessment* $(\mu, \boldsymbol{\pi})$ *that is*

1. *sequentially rational,*
2. *consistent.*

A SE, roughly speaking, is a consistent sequentially rational assessment $(\mu, \boldsymbol{\pi})$; the set of SE is denoted by $\Phi \subseteq \Psi \times \Pi$. Consistency of $(\mu, \boldsymbol{\pi})$ requires that there exists a fully mixed perturbed strategy profile $\boldsymbol{\pi}(\varepsilon)$ of $\boldsymbol{\pi}$ such that, letting $\mu(\varepsilon)$ to be the beliefs derived from $\boldsymbol{\pi}(\varepsilon)$, then $\lim_{\varepsilon \to 0} \mu(\varepsilon) = \mu$.

Hence, the difference between PE and SE is that the former concept requires *ex post* optimality approaching the limit, while the latter requires this only at the limit. Roughly speaking, perfectness amounts to sequentiality plus the request that the prescribed actions are not locally dominated. Hence, if $\boldsymbol{\pi}$ is perfect, then there exists some μ such that $(\mu, \boldsymbol{\pi})$ is an SE, but the converse does not hold. The difference between the concepts is only marginal: for almost all games the concepts yield the same outcomes. The main innovation of the concept of SE is the explicit incorporation of the system of beliefs sustaining the strategies as part of the definition of equilibrium. In this, it provides a language for discussing the relative plausibility of various systems of beliefs and the associated equilibria sustained by them. Making explicit the construction of beliefs off the equilibrium path enables discussion of which beliefs are "plausible" and which are not. And such comparisons can often help one to choose among sequential/perfect equilibria. Also, the definition of QPE is largely the same as the definition of PE. It differs only in that, instead of the limit strategy $\boldsymbol{\pi}$ being optimal at each information set against behavior given by $\boldsymbol{\pi}_n$ at all other information sets, it is required that $\boldsymbol{\pi}$ be optimal at all information sets against behavior at other information sets given by $\boldsymbol{\pi}$ for information sets that are owned by the same player who owns the information set in question, and by $\boldsymbol{\pi}_n$ for other information sets. That is, the player does not take account of his own "mistakes", except to the extent that they may make one of his information sets reached that otherwise would not be. This change in the definition leads to some attractive properties. Like PE, QPE are SE strategies. But, unlike PE, QPE are always Normal Form Perfect, and thus admissible. Mertens (1995) argues that QPE is precisely the right mixture of admissibility and backward induction.

4. Verification for Subgame Perfect Equilibria when a realization plan strategy profile is given

We focus on the verification problem for Subgame Perfect Equilibria (SPE) when the input is partially specified, and, precisely, it is given in sequence form. Initially, we introduce the following corollary that we use below to prove our result.

Definition 4.1 (NE–MIN–PAYOFF). *The NE–MIN–PAYOFF is defined as:*

- *INPUT: a normal form game* Γ*, a player* i*, a real value* k*;*
- *OUTPUT: YES if there is at least a Nash Equilibrium of* Γ *in which player* i*'s expected utility is smaller than or equal to* k*, NO otherwise.*

Lemma 4.2. *NE–MIN–PAYOFF problem is* NP*–hard even in symmetric two–player games.*

Proof. This problem is a variation of the problem of deciding whether there is a Nash Equilibrium (NE) providing a player with a utility larger than or equal to a value given as input. This last problem is known to be NP–hard in Conitzer and Sandholm (2008) and Gilboa and Zemel (1989). The proof of the corollary is a simple variation of the proof used in Conitzer and Sandholm (2008), in which the authors reduce from SAT problem. More precisely, in Conitzer and Sandholm (2008), given any SAT instance, the authors provide a symmetric two–player game instance with the following properties:

- there is always a pure–strategy NE in which both players play their last action, denoted by f, providing a payoff of ρ to both players for any $\rho > 0$;
- there are mixed–strategy NE in which both players randomize over some subset of actions not containing action f, providing a payoff of $n-1$ to both players, if and only if the SAT instance admits a solution.

In order to prove the corollary above, it is sufficient to set $\rho = n$. Indeed, in this way, there is an NE with a utility smaller than or equal to $n-1$ if and only if the SAT instance admits a solution. □

Now, we are able to provide the main result of this section.

Theorem 4.3. *Given a realization plan strategy profile $(\mathbf{r}^1, \mathbf{r}^2)$, the problems of certifying that it is part of an SPE or part of an SE are* NP*–hard.*

Proof. We initially focus on the verification for SPE. We reduce from NE–MIN–PAYOFF. The reduction is based on the game tree depicted in Fig. 2 where the subgame starting with information set with label **1.2** and including information set with label **2.2** is an arbitrary general–sum normal form game Γ with two players. Consider the following sequence form strategies: $\mathbf{r}^1$ prescribes that sequence L_1 is played with a probability of one, and $\mathbf{r}^2$ prescribes that sequence l_1 is played with a probability of one. By definition of sequence form, for all the other sequences, profile $(\mathbf{r}^1, \mathbf{r}^2)$ prescribes a probability of zero. Profile $(\mathbf{r}^1, \mathbf{r}^2)$ constitutes a SPE if and only if the subgame starting at information set with label **1.2** admits an NE that provides player 2 with an expected utility smaller than or equal to 1 (notice that Γ could admit multiple NE, we are interested in one NE with a specific property). Since profile $(\mathbf{r}^1, \mathbf{r}^2)$ prescribes a probability of zero for all the sequences in Γ, it does not pose additional constraints over the problem of searching for an NE of Γ providing player 2 with no more than 1. Hence, our problem reduces from the problem of deciding whether there is an NE providing a player with a utility no larger than 1 that, as showed above, it is NP–hard.

Now, we focus on the verification for SE. Every SPE of the above extensive form game is also part of an SE in which the belief over each node x of the information set with label **2.2** is equal to the probability with which player 1 plays the action at information set with label **1.2** leading to x. On the other hand, we recall that every SE is also an SPE and therefore in the above extensive form game the sets of SPEs and SEs are the same set. From this observation, it trivially follows that certifying that a realization plan strategy profile $(\mathbf{r}^1, \mathbf{r}^2)$ is part of an SE is NP–hard. □

Let us observe that the proof of the above theorem does not apply directly to the verification for QPE. The reason is that the set of QPEs may be a strict subset of the set of SPEs. Furthermore, in the game instances considered in Conitzer and Sandholm (2008), and also in our variation introduced in Lemma 4.2, only the pure–strategy NE is perfect, while the mixed NEs are not - in these latter equilibria the actions in the

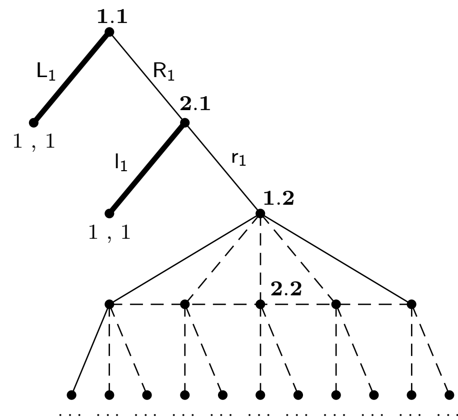

Figure 2: Game used in the proof of Theorem 4.3.

supports are weakly dominated, see Conitzer and Sandholm (2008). This pushes for considering new reductions when studying the verification problem for QPE. Nevertheless, we conjecture that the verification problem keeps to be NP–hard also for QPE and this is due to sequential rationality and to the fact that realization plan strategies partially specify the players' behavior on the game tree, leaving the behavior unspecified in a number of information sets. Therefore, having the input in behavioral strategies is a necessary condition to have tractable verification algorithms. In the next sections, we use inputs only in behavioral strategies.

Furthermore, we can state the following corollary, whose proof is straightforward given Theorem 4.3 (e.g., in Fig. 2 the path is composed only of L_1).

Corollary 4.4. *Given the strategies only on the equilibrium path, certifying that they are part of an SPE or part of an SE is a* NP*–hard problem.*

5. Verification with two players when the behavioral strategy profile is given

In the following sections, we show that there exists an efficient algorithm for the verification problem of Quasi Perfect Equilibrium (QPE) with two players and subsequently we show that a simple variation of the algorithm can be employed for the verification of Sequential Equilibrium (SE) with two players. Note that even if a QPE is a SE in strategies, the verification problem of a QPE cannot be trivially reduced to the verification problem of a SE. Indeed, differently from the concept of SE, the concept of QPE is defined only on strategies and not on beliefs and requires that the best response constraints hold even in presence of perturbations over the strategies.

5.1. Quasi Perfect Equilibrium verification in two–player games

In order to verify whether a strategy profile $\boldsymbol{\pi} = (\pi^1, \pi^2)$ is a QPE, we need to search for (if they exist):

- a fully mixed π_n^1 such that $\lim_{n\to\infty} \pi_n^1 = \pi^1$ and $(\pi^2(h), \pi^2(k)) \in \arg\max \mathbb{E}^{((\pi^2(h),\widehat{\pi}^2(k)),\pi_n^1)}[u^2(z)|h]$ for all $h \in H^2$,
- a fully mixed π_n^2 such that $\lim_{n\to\infty} \pi_n^2 = \pi^2$ and $(\pi^1(h), \pi^1(k)) \in \arg\max \mathbb{E}^{((\pi^1(h),\widehat{\pi}^1(k)),\pi_n^2)}[u^1(z)|h]$ for all $h \in H^1$.

Instead of searching directly for perturbed behavioral strategies π_n^i whose formulation of the best–response constraints is highly non–linear, we can work with perturbed sequence form strategies $\mathbf{r}_n^i$—similarly to Miltersen and Sørensen (2010)—whose formulation of the best–response constraints is instead linear. More precisely, our goal is to search for a pair of fully mixed $\mathbf{r}_n^i$ with the property that the fully mixed π_n^i derived from $\mathbf{r}_n^i$ converges to the input π^i as n goes to infinity.

Furthermore, although in principle the perturbation of $\mathbf{r}_n^i$ can be any function, we can safely limit our search to polynomial symbolic perturbations in $\varepsilon \in (0,1)$ and deal with them lexicographically, see Blum et al. (1991) and Govindan and Klumpp (2003). We provide the details.

Definition 5.1 (Polynomially perturbed sequence–form strategy). *A sequence form strategy $\mathbf{r}^i(\varepsilon)$ with polynomial symbolic perturbation in ε is defined as $\mathbf{r}^i(\varepsilon) = \sum_k \mathbf{r}_k^i(\varepsilon) \cdot \varepsilon_k$ where $\mathbf{r}_k^i(\varepsilon)$ is the vector of coefficients of ε_k.*

A sequence form strategy $\mathbf{r}^i(\varepsilon)$ with symbolic polynomial perturbation can be conveniently represented as a matrix whose column vectors are the coefficients $\mathbf{r}_k^i(\varepsilon)$ from $k = 0$ on as:

$$\mathbf{r}^i(\varepsilon) = \begin{bmatrix} \mathbf{r}_0^i(\varepsilon) & \mathbf{r}_1^i(\varepsilon) & \mathbf{r}_2^i(\varepsilon) & \dots \end{bmatrix}$$

The condition $\mathbf{r}^i(\varepsilon) \geq \mathbf{0}$ as ε goes to zero can be expressed by resorting to the relation $\geq_{\text{lex}}$, i.e. 'lexicographically larger than or equal to', defined as follows.

Definition 5.2 ($\geq_{\text{lex}}$). *Given a vector $\mathbf{y}$, we have $\mathbf{y} \geq_{\text{lex}} \mathbf{0}$ if the first (in lexicographic order) non–zero element of $\mathbf{y}$ is positive.*

We can state the condition for the positiveness of a strategy and the condition to have a fully mixed strategy as discussed in Miltersen and Sørensen (2010).

Definition 5.3 (Perturbed strategy positiveness). *A sequence form strategy $\mathbf{r}^i(\varepsilon)$ with symbolic polynomial perturbation is positive as ε goes to zero when $\mathbf{r}^i(\varepsilon) \geq_{\text{lex}} \mathbf{0}$.*

Definition 5.4 (Fully mixed strategy). *A sequence form strategy $\mathbf{r}^i(\varepsilon)$ with symbolic polynomial perturbation is fully mixed as ε goes to zero when $r^i(q,\varepsilon) >_{\text{lex}} 0$ for every $q \in Q$.*

Now we can formulate the conditions such that a given $\boldsymbol{\pi} = (\pi^1, \pi^2)$ is a QPE in terms of fully mixed $\mathbf{r}^1(\varepsilon)$ and $\mathbf{r}^2(\varepsilon)$. Initially, we provide the following (non–linear) mathematical program where $\mathbf{w}^i$ and $\mathbf{r}^i$ are variables, while π^i is a parameter given in input.

Program 5.5.

$$
\begin{aligned}
F^i \cdot \mathbf{r}^i(\varepsilon) &= \mathbf{f}^i && (1)\\
\mathbf{r}^i(\varepsilon) &>_{\text{lex}} \mathbf{0} && (2)\\
(F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i(\varepsilon) &\geq_{\text{lex}} \mathbf{0} && (3)\\
((F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i(\varepsilon))_{qa} &= 0 \qquad \forall qa \in Q^{-i} : \pi^{-i}(a) > 0 && (4)\\
\lim_{\varepsilon \to 0} \frac{r^i(qa, \varepsilon)}{r^i(q, \varepsilon)} &= \pi^i(a) \quad \forall a \in A^i, q \in Q^i : qa \in Q^i, r^i(q, \varepsilon) >_{\text{lex}} 0 && (5)
\end{aligned}
$$

Proposition 5.6 (QPE conditions in sequence form). *Given strategy profile* $\boldsymbol{\pi} = (\pi^1, \pi^2)$ *in input,* (π^1, π^2) *is a QPE if and only if, for every* $i \in I$*, the Program 5.5 admits a solution.*

Proof. The proof follows from the definitions of sequence form and QPE. However, we provide the details because we exploit them in the subsequent propositions. The above program captures the definition of QPE with two players since:

- $\mathbf{r}^i(\varepsilon)$ is a well–defined perturbed sequence form strategy due to constraints (1)—these constraints apply the definition of sequence form (see Section 3.2);
- $\mathbf{r}^i(\varepsilon)$ is fully mixed due to constraints (2)—these constraints apply the definition of strict lexico positiveness (see Proposition 5.4);
- π^{-i} is a best response to $\mathbf{r}^i(\varepsilon)$ even for $\varepsilon > 0$ due to constraints (3) and constraints (4)—more precisely, constraints (3) are the dual best–response constraints derived as in Miltersen and Sørensen (2010), forcing the value associated with an information set to be at least the value given by the best action player $-i$ can play at such information set, and constraints (3) force that sequences q are the best sequences player $-i$ can play;
- $\mathbf{r}^i(\varepsilon)$ and π^i are the same strategy as ε goes to zero due to constraints (5) and constraints (2)-indeed, constraints (5) force the two strategies to be same as $\varepsilon \to 0$ for every strictly positive $r^i(q)$ and constraints (2) force all the $r^i(q)$ to be strictly positive.

Hence, if Program 5.5 admits a solution, then there exists a well–defined fully mixed $\mathbf{r}^i(\varepsilon)$ from which we can derive a fully mixed π^i_n such that $\lim_{n\to\infty} \pi^i_n = \pi^i$ and π^{-i} is best response to π^i_n. Thus, if Program 5.5 admits a solution for all $i \in I$, (π^1, π^2) is a QPE. The "only if" is trivial and follows from the fact that, if there is perturbed behavioral strategy π^i_n with $\lim_{n\to} \pi^i_n$ such that π^i_n is best response to π^{-i}, then we can always derive a realization–equivalent realization plan r^i_n that is a polynomial in ε and that satisfies the constraints of the above program. This completes the proof. □

We introduce a sufficient condition to have equivalence between perturbed strategies, whose proof is omitted being straightforward-we point the interested reader to van Damme (1991)—; we use this condition in the following.

Proposition 5.7 (Equivalent perturbed strategies). *Given two perturbed strategies* $\mathbf{r}^i(\varepsilon)$ *and* $\widehat{\mathbf{r}}^i(\varepsilon)$*, a sufficient condition such that* $\mathbf{r}^i(\varepsilon)$ *and* $\widehat{\mathbf{r}}^i(\varepsilon)$ *are equivalent is: for all* k *it holds that* $r^i_k(q) = \alpha \cdot \widehat{r}^i_k(q)$ *with* $\alpha > 0$ *for every* $q \in Q$.

Obviously, given two equivalent perturbed strategies $\mathbf{r}^i(\varepsilon)$ and $\widehat{\mathbf{r}}^i(\varepsilon)$, if $\mathbf{r}^i(\varepsilon)$ satisfies Program 5.5, then also $\widehat{\mathbf{r}}^i(\varepsilon)$ does and *vice versa*.

We can show that there exists an efficient algorithm for Program 5.5.

Proposition 5.8. *Given* (π^1, π^2) *in input, there is an efficient algorithm solving Program 5.5.*

Proof. We show that Algorithm 1 solves Program 5.5 in polynomial time. The basic idea of the algorithm is the following:

- we initialize $r^i_k(q,\varepsilon) = 0$ for every q and $k \in \{0,\ldots,|Q^i|\}$,
- the algorithm is iterative,
- at each iteration k the algorithm attempts to increase the number of strictly lexico positive strategies w.r.t. iteration $k-1$,
- at each iteration k the values of $\mathbf{r}^i_k(\varepsilon)$ satisfy constraints (1), (3), (4), (5) and satisfy $\mathbf{r}^i(\varepsilon) \geq_{\text{lex}} 0$.

Thus, if at some iteration a fully mixed $\mathbf{r}^i(\varepsilon)$ is found, such a $\mathbf{r}^i(\varepsilon)$ satisfies constraints of Program 5.5.

Iteration $k = 0$. We derive $\mathbf{r}^i_k(\varepsilon)$ from π^i as $r^i_k(q,\varepsilon) = \prod_{a\in q} \pi^i(a)$ and we verify whether the following constraints hold:

$$(F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}_0(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i_0(\varepsilon) \geq \mathbf{0} \tag{6}$$

$$((F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}_0(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i_0(\varepsilon))_{qa} = 0 \qquad \forall q \in Q^{-i} : \pi^{-i}(a) > 0 \tag{7}$$

where constraints (6) and (7) correspond to constraints (3) and (4), respectively, in absence of perturbations. If the above constraints are satisfied by $\mathbf{r}^i_0(\varepsilon)$, then π^{-i} is a best response to π^i and the algorithm goes to the next iteration. Furthermore, if the above constraints are satisfied by $\mathbf{r}^i_0(\varepsilon)$, then $\mathbf{r}^i_0(\varepsilon)$ satisfies constraints (1), (3), (4), (5) and satisfy $\mathbf{r}^i_0(\varepsilon) \geq_{\text{lex}} \mathbf{0}$. Otherwise, the algorithm terminates returning non–existence, π not being neither an NE.

Iteration $k \geq 1$. From $k = 1$ on, the algorithm solves the Program 5.9 until $\mathbf{r}^i(\varepsilon)$ is not strictly lexico positive.

Algorithm 1 findPerturbedStrategy($\boldsymbol{\pi}^1, \boldsymbol{\pi}^2, i$)

$r^i_k(q,\varepsilon) = 0$ for every $q \in Q^i$ and $k \in \{0, \ldots, |Q^i|\}$
$r^i_0(q,\varepsilon) = \prod_{a \in q} \pi^i(a)$ for every $q \in Q^i$
verify constraints (6)–(7) for player i
if constraints are not satisfied **then**
  return non–existence
while $\mathbf{r}^i(\varepsilon)$ is not strictly lexico positive **do**
  $k = k + 1$
  solve program (8)–(14) for player i
  if objective function is 0 **then**
    return non–existence
return $\mathbf{r}^i(\varepsilon)$

Program 5.9.

$$\max \quad \sum_{\forall k' < k, r^i_{k'}(q,\varepsilon) = 0} r^i_k(q,\varepsilon) \tag{8}$$

$$F^i \cdot \mathbf{r}^i_k(\varepsilon) = \mathbf{0} \tag{9}$$

$$r^i_k(q,\varepsilon) \geq 0 \qquad \forall q \in Q^i : r^i_{k'}(q,\varepsilon) = 0, \forall k' < k \tag{10}$$

$$r^i_k(q,\varepsilon) \leq 1 \qquad \forall q \in Q^i \tag{11}$$

$$r^i_k(q(h)a,\varepsilon) = \pi^i(a) \cdot g(h) \qquad \forall h \in H^i, q(h) \in Q^i : a \in A(h), \forall a' \in A(h), r^i_{k'}(q(h)a',\varepsilon) = 0, \forall k' < k \tag{12}$$

$$g(h) \geq 0 \qquad \forall h \in H^i \tag{13}$$

$$((F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}_k(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i_k(\varepsilon))_q \geq 0 \qquad \forall q \in Q^{-i} : \forall k' < k, ((F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}_{k'}(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i_{k'}(\varepsilon))_q = 0 \tag{14}$$

$$((F^{-i})^{\mathsf{T}} \cdot \mathbf{w}^{-i}_k(\varepsilon) - (V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i_k(\varepsilon))_{qa} = 0 \qquad \forall qa \in Q^{-i} : \pi^{-i}(a) > 0 \tag{15}$$

In Program 5.9:

- constraints (9) assure the perturbation of degree k to be well defined according to the definition of sequence form, forcing $\mathbf{r}^i(\varepsilon)$ to satisfy constraints (1);
- constraints (10) assure that $\mathbf{r}^i(\varepsilon) \geq_{\text{lex}} \mathbf{0}$, forcing $r^i_k(q,\varepsilon) \geq 0$ for all the strategies $r^i(q,\varepsilon)$ that are not strictly lexico positive yet (i.e. for $k' < k$)—instead, the constraints $r^i_k(q,\varepsilon) \geq 0$ are not applied to all the strategies that are already (i.e. for $k' < k$) strictly lexico positive;

- constraints (11) pose - without loss of generality as shown by Proposition 5.7—an upper bound of 1 over the coefficients of ε_k;

- constraints (12) and (13) force $\mathbf{r}^i(\varepsilon)$ to satisfy constraints (5); this is accomplished forcing that if the strategy $r^i(qa,\varepsilon)$ becomes strictly lexico positive then all the strategies $r^i(qa',\varepsilon)$ where a and a' are available at the same information set h must assume values that are consistent to π^i. More precisely, if the strategy $r^i(qa,\varepsilon)$ becomes strictly lexico positive, it means that the auxiliary variable $g(h)$ with $a \in A(h)$ is strictly positive. Then, the sum of all the $r^i_k(qa,\varepsilon)$ with $a \in A(h)$ is equal to $g(h)$ (since $\sum_{a\in A(h)} \pi^i(a) = 1$), that in turn is equal, by definition of sequence form, to $r^i_k(q,\varepsilon)$ and therefore $\lim_{\varepsilon\to 0} \frac{r^i_k(qa,\varepsilon)}{r^i_k(q,\varepsilon)} = \pi^i(a)$;

- constraints (14) and (15) force $\mathbf{r}^i(\varepsilon)$ to satisfy constraints (3) and (4), respectively, and therefore that sequences $qa \in Q^{-i}$ with $\pi^{-i}(qa) > 0$ are the best sequences player $-i$ can play at each information set; constraints (14) are applied only to sequences q that provide the maximum utility for all the previous $k' < k$, because for the other sequences q the values $w(h)^{-i}$ are already (i.e. for $k' < k$) lexicographically strictly larger than the expected utility provided by q and therefore, for these sequences, constraints (3) are satisfied independently of the perturbation of degree k;

- objective function (8) aims at maximizing the sum of the coefficients $r^i_k(q,\varepsilon)$ such that $r^i_{k'}(q,\varepsilon) = 0$ for all $k' < k$. In this way, if it is possible to make some $r^i(q,\varepsilon)$ strictly lexico positive, it will be done.

Notice that the solution $r^i_k(q,\varepsilon) = 0$ for every q is always a feasible solution. If at the optimal solution of Program 5.9 the objective function has a value of zero, then the algorithm terminates returning non–existence. The algorithm goes to the next iteration otherwise.

We study the properties of the algorithm through three lemmas. The proof of the first lemma is omitted being straightforward.

Lemma 5.10. *If Algorithm 1 terminates returning a solution $\mathbf{r}^i(\varepsilon)$, then $\mathbf{r}^i(\varepsilon)$ is strictly lexico positive.*

Then, we state the following lemma.

Lemma 5.11. *If Algorithm 1 terminates with* non–existence*, then there is no strictly lexico–positive strategy $\mathbf{r}^i(\varepsilon)$ such that π^{-i} is best response to $\mathbf{r}^i(\varepsilon)$.*

Proof. If the algorithm terminates at $k = 0$, the proof is straightforward. We prove the lemma for the case in which the algorithm terminates at $k \geq 1$ into two steps:

- Step 1: we prove that, given $\mathbf{r}^i(\varepsilon)$ returned at iteration $k-1$, if the algorithm terminates at iteration k returning non–existence, then there is no perturbation with degrees $k' \geq k$ such that, once applied to $\mathbf{r}^i(\varepsilon)$, $\mathbf{r}^i(\varepsilon)$ satisfies constraints of Program 5.5. If the objective function is zero, then it is not possible to introduce any perturbation in $\mathbf{r}^i(\varepsilon)$ of degree k without violating the best–response constraints (14)–(15) of player $-i$ and/or constraints (12) and (13). Therefore $\mathbf{r}^i(\varepsilon)$ cannot be made strictly lexico positive unless violating constraints (3)–(5).

- Step 2: we prove that, given $\mathbf{r}^i(\varepsilon)$ returned at iteration $k-1$, if the algorithm terminates at iteration k returning non–existence, then there is no other perturbed strategy satisfying constraints of Program 5.5. By construction, any perturbed strategy fulfilling Program 5.5 satisfies constraints (6)–(7) at $k=0$ and constraints (9)–(14) at $k \geq 1$. Therefore, each such a perturbed strategy can be found by Algorithm 1 with an opportune objective function in place of objective function (8). However, constraints (9)–(15) strictly relax from an iteration to the subsequent one. More precisely, constraints (10), (12), and (14) relax, while the others keep to be the same. Therefore, all the possible paths that Algorithm 1 can follow with any possible objective function lead to the same result in terms of existence or non–existence of a strategy $\mathbf{r}^i(\varepsilon)$ fulfilling constraints of Program 5.5. We notice that the perturbed strategies found by the algorithm could be different if different paths are followed. Among all the objective functions, objective function (8) minimizes the number of iterations, maximizing at each iteration the number of strictly lexico–positive strategies. Therefore, the algorithm always terminates correctly. □

Lemma 5.12. *Algorithm 1 runs in polynomial time in the size of the game.*

Proof. We prove the lemma by showing that the number of iterations of the algorithm is in the worst case linear in the size of the game. At each iteration k, either some $r^i_k(q, \varepsilon)$ that is zero for every $k' < k$ becomes strictly positive or the algorithm stops. In the worst case, only one sequence becomes strictly lexico positive at each iteration and therefore the number of iterations is equal to the number of sequences $|Q^i|$. Thus, since linear mathematical programming runs in polynomial time, the lemma is proved. □

Hence, Lemma 5.10 and Lemma 5.11 prove that Algorithm 1 solves Program 5.5, and Lemma 5.12 shows that the algorithm requires polynomial time. This completes the proof of the proposition.

We are now in the position to prove the following theorem.

Theorem 5.13. *There exists an efficient algorithm certifying that a behavioral strategy profile $\boldsymbol{\pi} = (\pi^1, \pi^2)$ given in input is a QPE with two–player games.*

Proof. By Proposition 5.6, the problem of certifying that a behavioral strategy profile $\boldsymbol{\pi}$ is a QPE with two–player games can be formulated as a pair of mathematical programs whose resolution can be achieved, by Proposition 5.8, in polynomial time. □

We provide two examples to which we apply Algorithm 1.

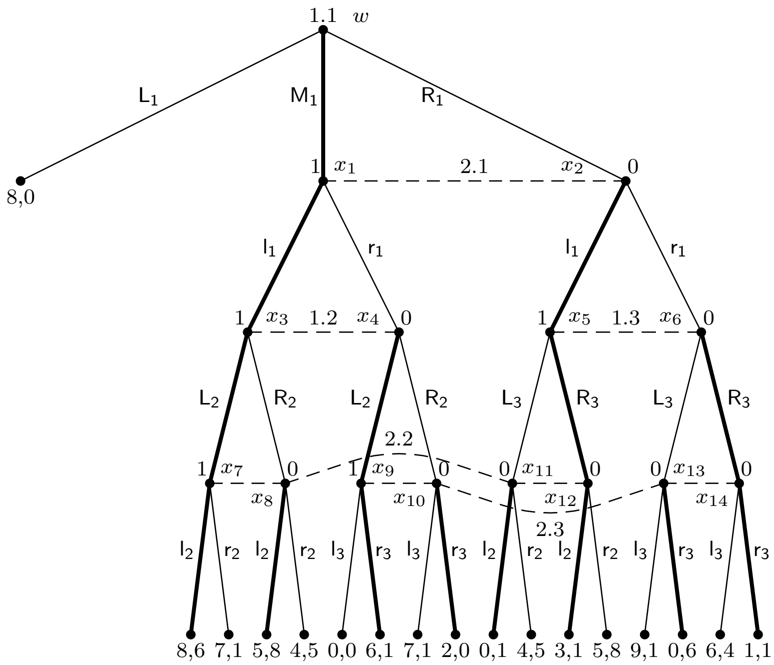

Figure 3: Behavioral strategies $\boldsymbol{\pi}$ are represented by using bold lines to denote actions played with positive probability and $\boldsymbol{\mu}$ is represented reporting the beliefs close to the nodes of each information set. The strategy profile that is not a QPE, but the assessment $(\boldsymbol{\mu}, \boldsymbol{\pi})$ where $\boldsymbol{\pi}$ is sequentially rational and $\boldsymbol{\mu}$ is consistent.

Example 5.14. *Consider the game depicted in Fig. 3, where* $\boldsymbol{\pi} = (\pi^1(\mathsf{M}_1) = 1, \pi^1(\mathsf{L}_2) = 1, \pi^1(\mathsf{R}_3) = 1, \pi^2(\mathsf{l}_1) = 1, \pi^2(\mathsf{l}_2) = 1, \pi^2(\mathsf{r}_3) = 1)$. *The strategy profile is not a QPE. We show how the Algorithm 1 works. Initially, we report the utility bimatrix in sequence form:*

		player 2						
		$\varnothing$	l_1	r_1	$\mathsf{l}_1\mathsf{l}_2$	$\mathsf{l}_1\mathsf{r}_2$	$\mathsf{r}_1\mathsf{l}_3$	$\mathsf{r}_1\mathsf{r}_3$
	$\varnothing$							
	L_1	$8, 0$						
	M_1							
	R_1							
player 1	$\mathsf{M}_1\mathsf{L}_2$				$8, 6$	$7, 1$	$0, 0$	$6, 1$
	$\mathsf{M}_1\mathsf{R}_2$				$5, 8$	$4, 5$	$7, 1$	$2, 0$
	$\mathsf{R}_1\mathsf{L}_3$				$0, 1$	$4, 5$	$9, 1$	$0, 6$
	$\mathsf{R}_1\mathsf{R}_3$				$3, 1$	$5, 8$	$6, 4$	$1, 1$

For the sake of presentation, we omit matrices F_1 *and* F_2. *We apply Algorithm 1 to find* $\mathbf{r}^1(\varepsilon)$:

- iteration 0: *the strategy prescribes:* $r^1_0(\mathsf{M}_1, \varepsilon) = r^1_0(\mathsf{M}_1\mathsf{L}_2, \varepsilon) = 1$, *while the other sequences are played with a probability of zero; the strategy satisfies constraints (6)–(7);*

- iteration 1*: in constraints (12) the only sequence with a probability of zero is* L_3*; the maximization is over all the sequences except* M_1 *and* $\mathsf{M}_1\mathsf{L}_2$*; the resulting strategy prescribes:* $r_1^1(\mathsf{L}_1,\varepsilon) = r_1^1(\mathsf{R}_1,\varepsilon) = r_1^1(\mathsf{M}_1\mathsf{R}_2,\varepsilon) = r_1^1(\mathsf{R}_1\mathsf{R}_3,\varepsilon) = 1$ *and* $r_1^1(\mathsf{M}_1,\varepsilon) = -2$ *and* $r_1^1(\mathsf{M}_1\mathsf{L}_2,\varepsilon) = -3$*, while the other sequences are played with a probability of zero;*

- iteration 2*: no sequence appears in constraints (12); the maximization is only over* L_3*; the resulting strategy prescribes:* $r_2^1(\mathsf{L}_3,\varepsilon) = 1$ *and* $r_2^1(\mathsf{R}_3,\varepsilon) = -1$*.*

Summarily the resulting $\mathbf{r}^1(\varepsilon)$ *is (∅ is omitted for simplicity):*

	L_1	M_1	R_1	$\mathsf{M}_1\mathsf{L}_2$	$\mathsf{M}_1\mathsf{R}_2$	$\mathsf{R}_1\mathsf{L}_3$	$\mathsf{R}_1\mathsf{R}_3$
ε^0	0	1	0	1	0	0	0
ε^1	1	−2	1	−3	1	0	1
ε^2	0	0	0	0	0	1	−1

Therefore, there exists a fully mixed strategy $\lim_{\varepsilon\to 0}\pi^1(\varepsilon) \to \pi^1$ *such that* π^2 *is best response to* $\pi^1(\varepsilon)$*.*

Now, we apply Algorithm 1 to $\mathbf{r}^2(\varepsilon)$*. In this case, the algorithm stops with the resulting strategy (∅ is omitted for simplicity):*

	l_1	r_1	$\mathsf{l}_1\mathsf{l}_2$	$\mathsf{l}_1\mathsf{r}_2$	$\mathsf{r}_1\mathsf{l}_3$	$\mathsf{r}_1\mathsf{r}_3$
ε^0	1	0	1	0	0	0
ε^1	0	0	0	0	0	0

At iteration 1, the algorithm stops because the objective function is zero. The reason is that any possible perturbation over $\mathbf{r}^2(\varepsilon)$ *would make the probability with which player 2 plays* $\mathsf{l}_1\mathsf{l}_2$ *strictly smaller than one and therefore the expected utility of player 1 from playing* M_1 *would be strictly smaller than 8. Thus, player 1 would play* L_1 *gaining exactly 8. In practice, the algorithm cannot assign a strictly positive perturbation over* $\mathsf{l}_1\mathsf{r}_2$ *without violating the constraints of best response of player 1. As a result, there is no fully mixed* $\pi^2(\varepsilon)$ *such that* π^1 *is a best response and, therefore,* $\boldsymbol{\pi} = (\pi^1, \pi^2)$ *is not a QPE.*

Example 5.15. *Consider the game depicted in Fig. 4, where* $\boldsymbol{\pi} = (\pi^1(\mathsf{L}_1) = 1, \pi^1(\mathsf{L}_2) = 1, \pi^1(\mathsf{R}_3) = 1, \pi^2(\mathsf{l}_1) = 1, \pi^2(\mathsf{l}_2) = 1, \pi^2(\mathsf{r}_3) = 1)$*. The strategy profile is a QPE. From the application of Algorithm 1, we show that we obtain the following fully mixed* $\mathbf{r}^1(\varepsilon)$ *(∅ is omitted for simplicity):*

	L_1	M_1	R_1	$\mathsf{M}_1\mathsf{L}_2$	$\mathsf{M}_1\mathsf{R}_2$	$\mathsf{R}_1\mathsf{L}_3$	$\mathsf{R}_1\mathsf{R}_3$
ε^0	1	0	0	0	0	0	0
ε^1	$-\frac{5}{4}$	1	$\frac{1}{4}$	1	0	0	$\frac{1}{4}$
ε^2	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	1	−1

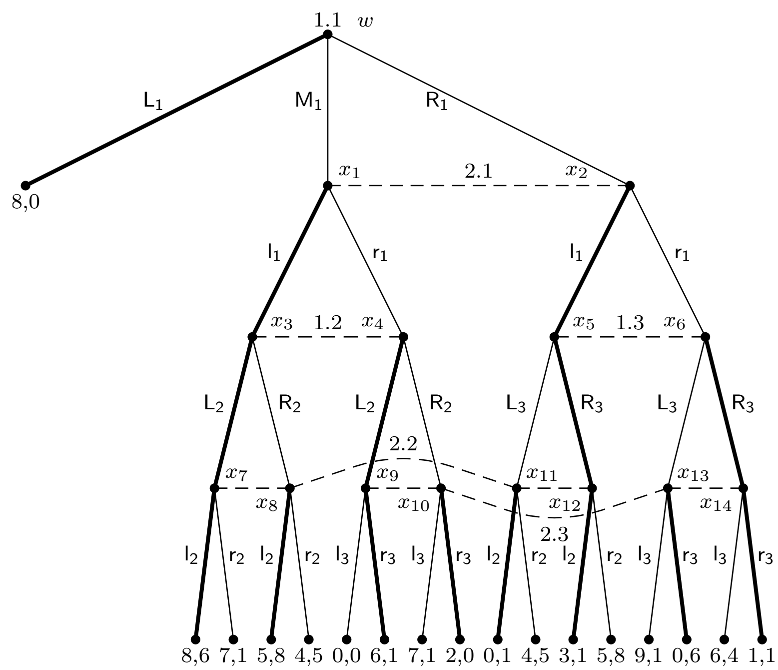

Figure 4: Example of strategy profile $\boldsymbol{\pi}$ expressed in behavioral strategies that is a QPE ($\boldsymbol{\pi}$ is represented by using bold lines to denote actions played with positive probability).

and the following fully mixed $\mathbf{r}^2(\varepsilon)$ *($\varnothing$ is omitted for simplicity):*

	l_1	r_1	$\mathsf{l}_1\mathsf{l}_2$	$\mathsf{l}_1\mathsf{r}_2$	$\mathsf{r}_1\mathsf{l}_3$	$\mathsf{r}_1\mathsf{r}_3$
ε^0	1	0	1	0	0	0
ε^1	−1	1	−2	1	0	1
ε^2	0	0	0	0	1	−1

therefore $\boldsymbol{\pi}$ *is a QPE.*

We briefly discuss why Algorithm 1 cannot be trivially extended to games with three or more players with the aim to identify the reasons why with three or more players the verification problem is hard, as shown in Hansen et al. (2010), while with two players it is easy. Program 5.5 can be easily extended to the case with three players by substituting constraints (3) with, e.g.:

$$(F^2)^\intercal \cdot \mathbf{w}^2(\varepsilon) - (V^2)^\intercal \cdot \mathbf{r}^1(\varepsilon) \cdot \mathbf{r}^3(\varepsilon) \geq_{\text{lex}} \mathbf{0}$$

and constraints (4) with, e.g.:

$$((F^2)^\intercal \cdot \mathbf{w}^2(\varepsilon) - (V^2)^\intercal \cdot \mathbf{r}^1(\varepsilon) \cdot \mathbf{r}^3(\varepsilon))_{qa} = 0 \qquad \forall q \in Q^2 : \pi^2(a) > 0$$

These constraints make the program much harder due to the following two reasons:

- given that the constraints depend both on $\mathbf{r}^1(\varepsilon)$ and $\mathbf{r}^3(\varepsilon)$, it is not possible to separate the three programs (the one in which the variables are $\mathbf{r}^1(\varepsilon)$ and $\mathbf{r}^3(\varepsilon)$ from the other two in which, in the first, the variables are $\mathbf{r}^1(\varepsilon)$ and $\mathbf{r}^2(\varepsilon)$ and, in the second, they are $\mathbf{r}^2(\varepsilon)$ and $\mathbf{r}^3(\varepsilon)$), as instead it is possible for the case with two players;
- each single program presents the product of two strategies.

5.2. Sequential Equilibrium verification in two–player games

Now, we show how a simple variation of Algorithm 4 can be applied to the verification problem of SE with two players.

Initially, we define the binary relation $\geq_{\text{lex–weak}}$ as follows.

Definition 5.16 ($\geq_{\text{lex–weak}}$)**.** *Given vectors* $\mathbf{y}_1, \mathbf{y}_2$*, we have* $\mathbf{y}_1 \geq_{\text{lex–weak}} \mathbf{y}_2$ *if and only if* $\mathbf{y}_1 \geq_{\text{lex}} \mathbf{y}_2$ *or, let*

- k_1 *the minimum* k *such that* $y_{1,k} \neq 0$ *and*
- k_2 *the minimum* k *such that* $y_{2,k} \neq 0$,

$k_1 = k_2 = k$ *and* $y_{1,k} = y_{2,k}$.

Notice that the above relation may be true even when $\mathbf{y}_2 \geq_{\text{lex}} \mathbf{y}_1$. On the basis of the above relation, we can formulate the verification problem for SE with two players when only strategies are provided as a non–linear mathematical program. We denote by $q \to h'$ that there exists some sequence of the opponent such that, combined with sequence q, leads to information set h'. Initially, we provide the following:

Program 5.17.

$$\text{Constraints (1), (2), (4), (5)}$$

$$w^{-i}(h,\varepsilon) \geq_{\text{lex–weak}} \sum_{h':q\in\{q(h')\}} w^{-i}(h',\varepsilon) + ((V^{-i})^{\mathsf{T}} \cdot \mathbf{r}^i(\varepsilon))_{qa} \qquad \forall h \in H^{-i}, qa \in Q^{-i} : a \in A(h) \qquad (16)$$

Proposition 5.18. *Strategy profile* $\boldsymbol{\pi} = (\pi^1, \pi^2)$ *given in input is part of an SE if and only if Program 5.17 admits solution for every* $i \in I$.

Proof. The above program differs from Program 5.5 presenting constraints (16) in place of constraints (3). The differences between these two groups of constraints lay only in the use of $\geq_{\text{lex-weak}}$ in constraints (16) in place of $\geq_{\text{lex}}$ used in constraints (3). In addition, in constraints (16) we explicitly detail all the terms instead of using the matrix–like representation used in constraints (3) to specify the terms on the left hand of $\geq_{\text{lex-weak}}$ and those on the right hand. By definition of $\geq_{\text{lex-weak}}$, the above program is satisfied by all the solutions of Program 5.5. Furthermore, the above program, thanks to $\geq_{\text{lex-weak}}$, can be satisfied by additional solutions in which the value $v_{-i,h}$ is equal to the expected utility provided by the best sequences when perturbations are not present, but it may be smaller when perturbations are present. This is exactly the condition required by SE definition. □

We focus on the complexity of solving the above mathematical program.

Proposition 5.19. *Given (π^1, π^2) in input, there exists an efficient algorithm finding a solution of Program 5.17.*

Proof. Program 5.17 can be solved with a simple variation of Algorithm 1. More precisely, constraints (14) must be substituted with the following constraints:

$$w_k^{-i}(h,\varepsilon) \geq \sum_{h':q\in\{q(h')\}} w_k^{-i}(h',\varepsilon) + \left((V^{-i})^{\mathsf{T}} \cdot \mathbf{r}_k^i(\varepsilon)\right)_q \qquad \begin{array}{r} \forall h \in H^{-i}, q \in Q^{-i} : a(q) \in A(h), \\ w_{k'}^{-i}(h,\varepsilon) = \displaystyle\sum_{h':q\in\{q(h')\}} w_{k'}^{-i}(h',\varepsilon) + \\ \left((V^{-i})^{\mathsf{T}} \cdot \mathbf{r}_{k'}^i(\varepsilon)\right)_q = 0, \forall k' < k \end{array}$$

The above constraints relax constraints (14), requiring $\geq_{\text{lex-weak}}$ in place of $\geq_{\text{lex}}$. Therefore, a solution can be found in polynomial time. □

Finally, we can prove the following theorem.

Theorem 5.20. *There exists an efficient algorithm certifying that a behavioral strategy profile $\boldsymbol{\pi} = (\pi^1, \pi^2)$ given in input is part of an SE in a two–players game.*

Proof. The problem of certifying that a behavioral strategy profile $\boldsymbol{\pi} = (\pi^1, \pi^2)$ is an SE in a two–players game can be formulated, by Proposition 5.18, as a pair of Program 5.17 and, by Proposition 5.19, these programs can be solved in polynomial time. □

We can show how the above result can be adopted to derive consistent (in the sense of Kreps and Wilson) beliefs when only the behavioral strategies are given. First we introduce an efficient algorithm that given a perturbed strategy profile computes a profile of beliefs that is consistent w.r.t. the strategy profile.

Lemma 5.21. *There exists an efficient algorithm that given a perturbed strategy profile computes a profile of beliefs that is consistent w.r.t. the strategy profile.*

Proof. Algorithm 2 computes a profile of beliefs that is consistent w.r.t. the input strategy profile. The profile of beliefs is consistent because in Step 2 we use the Bayes rule to compute it. The algorithm is efficient because it requires a number of iterations linear in the size of game and the computation of Step 3 is linear in the size of game by Proposition A.6. □

Algorithm 2 computeBeliefs($\mathbf{r}(\varepsilon)$)

1: **for all** $i \in \{1,2\}, h \in H^i, x \in h$ **do**

2: $\quad \mu(x,\varepsilon) = \dfrac{r^{-i}(q(x))}{\sum\limits_{x' \in h} r^{-i}(q(x'))}$

3: $\lim\limits_{\varepsilon \to 0} \boldsymbol{\mu}(\varepsilon) = \boldsymbol{\mu}$

4: **return** $\boldsymbol{\mu}$

Proposition 5.22. *Given a strategy profile $\boldsymbol{\pi} = (\pi^1, \pi^2)$ that is part of an SE, there exists an efficient algorithm finding a profile of beliefs (μ^1, μ^2) such that $(\mu, \boldsymbol{\pi})$ is an SE.*

Proof. Given $\boldsymbol{\pi}$, by Algorithm 1 we can efficiently verify whether $\boldsymbol{\pi}$ is part of a SE and we can efficiently compute a perturbed strategy profile in sequence form $\mathbf{r}(\varepsilon)$. Given $\mathbf{r}(\varepsilon)$, by Algorithm 2 we can efficiently compute a profile of beliefs that is consistent w.r.t. the strategy profile. Thus, given a strategy profile $\boldsymbol{\pi}$, we can efficiently compute an assessment $(\mu, \boldsymbol{\pi})$ that is a SE. □

Example 5.23. *Consider the game depicted in Fig. 4, where $\boldsymbol{\pi} = (\pi^1(\mathsf{L}_1) = 1, \pi^1(\mathsf{L}_2) = 1, \pi^1(\mathsf{R}_3) = 1, \pi^2(\mathsf{l}_1) = 1, \pi^2(\mathsf{l}_2) = 1, \pi^2(\mathsf{r}_3) = 1)$. We compute the profile of beliefs such that $(\mu, \boldsymbol{\pi})$ is a SE:*

$$\begin{array}{llll} \mu(x_3) = 1, & \mu(x_4) = 0, & & \\ \mu(x_5) = 1, & \mu(x_6) = 0. & & \\ \mu(x_1) = 0.8, & \mu(x_2) = 0.2, & & \\ \mu(x_7) = 0.8, & \mu(x_8) = 0, & \mu(x_{11}) = 0, & \mu(x_{12}) = 0.2, \\ \mu(x_9) = 0.8, & \mu(x_{10}) = 0, & \mu(x_{13}) = 0, & \mu(x_{14}) = 0.2. \end{array}$$

6. Verification with n players when the assessment is given

In this section we focus on the problem of deciding whether or not an assessment $(\mu, \boldsymbol{\pi})$ is an SE when the number of players is arbitrary. As summarized in Section 3, an assessment $(\mu, \boldsymbol{\pi})$ is an SE if and only if $\boldsymbol{\pi}$ is sequentially rational w.r.t. μ, and μ consistent w.r.t. $\boldsymbol{\pi}$. Hence, our proof is in two steps, first we show that given an assessment $(\mu, \boldsymbol{\pi})$, there is an efficient algorithm certifying that $\boldsymbol{\pi}$ is sequentially rational given μ, then that μ

is consistent given $\boldsymbol{\pi}$. The first step is simple since it is just an algorithmic reformulation of backward induction w.r.t. μ, while the second step is more involved because certificating that μ is consistent requires to search for a perturbed behavioral strategy profile $\boldsymbol{\pi}(\varepsilon)$ to derive $\mu(\varepsilon)$ by Bayes rule, then showing that $\lim_{\varepsilon \to 0} \mu(\varepsilon) = \mu$.

First we provide an efficient algorithm to verify sequential rationality given an assessment.

Proposition 6.1. *Given $(\mu, \boldsymbol{\pi})$, there exists an efficient algorithm certifying that $\boldsymbol{\pi}$ is sequentially rational w.r.t. μ.*

Proof. We say that a decision node $x \in T$ is *penultimate* if $S(x) \in Z$ (i.e., any immediate successor of x is a terminal node). Sequential rationality of $\boldsymbol{\pi}$ can be verified by means of a variation of the backward induction algorithm, see Fudenberg and Tirole (1991), incorporating μ as described in Algorithm 3. Basically, Algorithm 3 verifies, at each information set h such that all the decision nodes $x \in h$ are penultimate, that $\boldsymbol{\pi}^{\iota(h)}$ is optimal in expectation w.r.t. beliefs μ. Then, the game is reduced considering any penultimate x as a final node and assigning such a node with the corresponding expected utility.

Algorithm 3 requires a number of maximizations that is linear in the size of the game, and each single maximization is over a number of elements (i.e., actions) that is linear in the size of the game. This completes the proof of the proposition. □

Algorithm 3 verifySequentialRationality($\mu, \boldsymbol{\pi}$)

for all information sets h such that all the nodes $x \in h$ are penultimate **do**
  if $\iota(h)$ is the nature **then**
    eliminate all the nodes following $x \in h$ from the game tree and consider x as terminal nodes with utility $u^i(x) = \mathbb{E}^{\mu,\boldsymbol{\pi}}[u(z)|h]$ for $z \in S(x)$ for each player i
  if $\iota(h)$ is not the nature **then**
    if there is some action a' such that $\pi^{\iota(h)}(a') > 0$ and $a' \notin \arg\max_{a \in A(h)} \mathbb{E}^{\mu,\boldsymbol{\pi}}[u(z)|h]$ **then**
      return non–sequentially rational
    for all $x' \in h$ **do**
      eliminate all the immediate successors of x' from the game tree and consider x' as terminal nodes with utility $u^i(x) = \mathbb{E}^{\mu,\boldsymbol{\pi}}[u(z)|h]$ for $z \in S(x)$ for each player i
return sequentially rational

Now, we focus on the consistency of the beliefs. Verifying consistency of μ w.r.t. $\boldsymbol{\pi}$ requires to find a fully mixed perturbed strategy profile $\boldsymbol{\pi}(\varepsilon)$ with the property that the beliefs $\mu(\varepsilon)$ derived from $\boldsymbol{\pi}(\varepsilon)$ by Bayes rule converge to μ as ε goes to zero. Streufert (2006, 2007) shows that the problem of searching for such a $\boldsymbol{\pi}(\varepsilon)$ can be formulated as the problem of searching for a b–labeling.[10] In words, a b–labeling assigns zero to the labels of actions played with strictly positive probability and requires that, if and only if the belief over a node is strictly positive, then the sum of the labels of the actions leading to such node from the root node is the smallest among the sums of the labels of the actions leading to all the other nodes of the same information set. It is worth remarking that an assessment $(\mu, \boldsymbol{\pi})$ may admit multiple b–labelings. In this case, all the b–labelings lead to the same result: if, with a given b–labeling, $\mu(\varepsilon)$ derived from $\boldsymbol{\pi}(\varepsilon)$ by Bayes rule converges to μ as $\varepsilon \to 0$, then the same happens with all the other b–labelings. Therefore, it is sufficient to search for a (generic) b–labeling. In the Appendix, after reporting the result proved by Kreps and Wilson as refined by Streufert, we focus on the problem of searching for a b–labeling. Initially, we show that the problem of searching for a (specific, i.e., minimizing the sum of the labels) b–labeling can be formulated as an integer linear mathematical program. Although solving an integer linear mathematical program is NP–hard in the worst case, exploiting the property of total unimodularity provided in Ghouila-Houri (1962), we are able to show that our program can be solved exactly in polynomial time by means of linear programming. Thus, we are in the position to prove what follows.

Proposition 6.2. *Given $(\mu, \boldsymbol{\pi})$, there exists an efficient algorithm verifying that μ is consistent w.r.t. $\boldsymbol{\pi}$.*

Proof. Consistency can be verified by means of Algorithm 4. By Proposition A.8, a b–labeling can be found by solving the linear integer Program A.7. Since the matrix of constraints M of this program is totally unimodular, as shown by Proposition A.10, and the vector of constants $\mathbf{b}$ is integer, all the vertices of the polytope $\{\mathbf{z} | M\mathbf{z} = \mathbf{b}\}$ with $\mathbf{z}$ real–value variables are integers, see Ghouila-Houri (1962). Thus, we can find a solution of Program A.7 by searching for a basic solution of its continuous relaxation by means of linear mathematical programming techniques. Since a linear mathematical program can be solved in polynomial time, Program A.7 can be solved in polynomial time. If the program does not admit any b–labeling, then, by Theorem A.5, μ is not consistent. If the program admits a b–labeling $\boldsymbol{\lambda}$, then, by Theorem A.5, μ is consistent if and only if $\overline{\boldsymbol{\pi}}(\varepsilon)$, as defined in Theorem A.5, is such that $\mu(\varepsilon)$, derived from $\overline{\boldsymbol{\pi}}(\varepsilon)$ by Bayes rule, converges to μ as ε goes to zero. Verifying such a convergence requires polynomial time in the size of the game, as shown by Proposition A.6. This completes the proof of the proposition. □

We can leverage on the above results to prove the following theorem.

Theorem 6.3. *There exists an efficient algorithm certifying that an assessment $(\mu, \boldsymbol{\pi})$ given in input is an SE.*

[10] The original result is in Kreps and Wilson (1982), then corrected by Streufert (2006, 2007).

Algorithm 4 verifyConsistency$(\mu, \boldsymbol{\pi})$

1: solve the continuous relaxation of Program A.7 by linear mathematical programming
2: **if** the program does not admit any solution λ **then**
3:     **return** non–consistent
4: derive $\overline{\boldsymbol{\pi}}(\varepsilon)$ from λ
5: derive $\mu(\varepsilon)$ from $\overline{\boldsymbol{\pi}}(\varepsilon)$ by Bayes rule
6: **if** $\lim_{\varepsilon \to 0} \mu(\varepsilon) \neq \mu$ **then**
7:     **return** non–consistent
8: **return** consistent

Proof. Algorithm 5 certificates in polynomial time that $(\mu, \boldsymbol{\pi})$ is an SE. Indeed, by Proposition 6.1, we can certify whether $\boldsymbol{\pi}$ is sequentially rational w.r.t. μ in polynomial time and, by Proposition 6.2, we can certify whether μ is consistent w.r.t. $\boldsymbol{\pi}$ in polynomial time. This concludes the proof of the theorem. □

Algorithm 5 SEcertifying$(\mu, \boldsymbol{\pi})$

1: **if** verifySequentialRationality$(\mu, \boldsymbol{\pi})$ returns non–sequentially rational or verifyConsistency$(\mu, \boldsymbol{\pi})$ returns non–consistent **then**
2:     **return** non–SE
3: **return** SE

The above theorem shows that there is an efficient algorithm for the verification problem of SE with any number of players, and therefore such a problem is easier than the verification problems for QPE, EFPE, and NFPrE that are instead NP–hard with three or more players as shown in Hansen et al. (2010).

We provide two examples to which we apply Algorithm 5.

Example 6.4. *Consider the game depicted in Fig. 5, where* $\boldsymbol{\pi} = (\pi^1(\mathsf{L}^1) = 1, \pi^1(\mathsf{L}_2) = 1, \pi^1(\mathsf{R}_3) = 1, \pi^2(\mathsf{l}_1) = 1, \pi^2(\mathsf{r}_2) = 1, \pi^2(\mathsf{l}_3) = 1)$*, while beliefs* μ *are reported in the figure aside the corresponding nodes. The assessment* $(\mu, \boldsymbol{\pi})$ *is not an SE and we show that the Algorithm 5 returns* non–SE*. Below we instantiate constraints of Program A.7.*

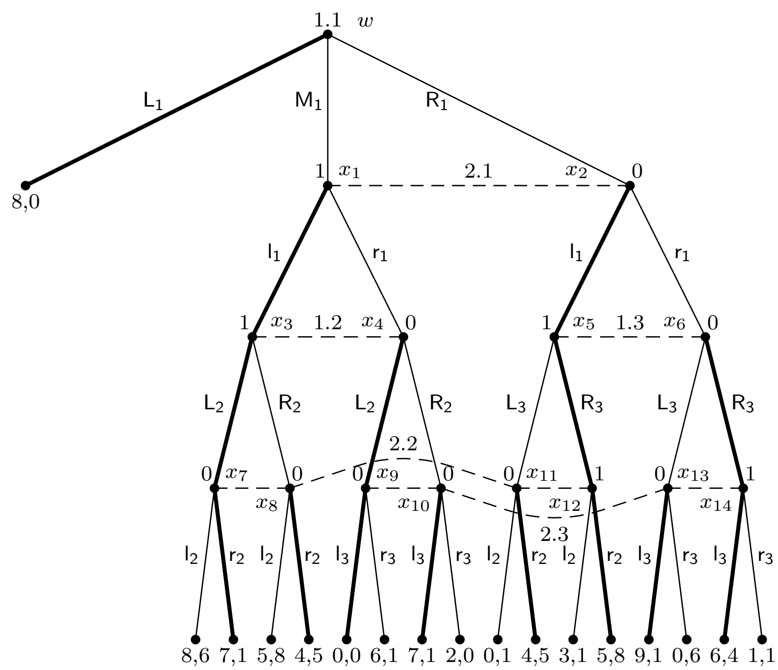

Figure 5: Example of assessment $(\mu, \boldsymbol{\pi})$ where $\boldsymbol{\pi}$ is sequentially rational, but μ is not consistent ($\boldsymbol{\pi}$ is represented by using bold lines to denote actions played with positive probability and μ is represented reporting the beliefs close to the nodes of each information set).

$$\lambda(\mathsf{L}_1 = \lambda(\mathsf{L}_2) = \lambda(\ \mathsf{R}_3) = \lambda(\mathsf{l}_1) = \lambda(\mathsf{r}_2) = \lambda(\mathsf{l}_3) = 0$$
$$\lambda(\mathsf{M}_1) - s(\mathsf{M}_1) = \lambda(\mathsf{R}_1) - s(\mathsf{R}_1) = \lambda(\mathsf{R}_2) - s(\mathsf{R}_2) = \lambda(\mathsf{R}_3) - s(\mathsf{R}_3) = 1$$
$$\lambda(\mathsf{r}_1) - s(\mathsf{r}_1) = \lambda(\mathsf{l}_2) - s(\mathsf{l}_2) = \lambda(\mathsf{r}_3) - s(\mathsf{r}_3) = 1$$
$$\gamma(w) = \gamma(w) + \lambda(\mathsf{M}_1) - \gamma(x_1) = \gamma(w) + \lambda(\mathsf{R}_1) - \gamma(x_2) = 0$$
$$\gamma(x_1) + \lambda(\mathsf{l}_1) - \gamma(x_3) = \gamma(w_1) + \lambda(\mathsf{r}_1) - \gamma(x_4) = \gamma(x_2) + \lambda(\mathsf{l}_2) - \gamma(x_5) = \gamma(x_2) + \lambda(\mathsf{r}_2) - \gamma(x_6) = 0$$
$$\gamma(x_3) + \lambda(\mathsf{L}_2) - \gamma(x_7) = \gamma(x_3) + \lambda(\mathsf{R}_2) - \gamma(x_8) = \gamma(x_4) + \lambda(\mathsf{L}_2) - \gamma(x_9) = \gamma(x_4) + \lambda(\mathsf{R}_2) - \gamma(x_{10}) = 0$$
$$\gamma(x_5) + \lambda(\mathsf{L}_3) - \gamma(x_{11}) = \gamma(x_5) + \lambda(\mathsf{R}_3) - \gamma(x_{12}) = \gamma(x_6) + \lambda(\mathsf{L}_3) - \gamma(x_{13}) = \gamma(x_6) + \lambda(\mathsf{R}_3) - \gamma(x_{14}) = 0$$
$$\gamma(x_3) - \nu(1.2) = \gamma(x_5) - \nu(1.3) = \gamma(x_1) - \nu(2.1) = \gamma(x_{12}) - \nu(2.2) = \gamma(x_{14}) - \nu(2.3) = 0$$
$$\gamma(x_4) - \nu(1.2) - t(x_4) = \gamma(x_6) - \nu(1.3) - t(x_6) = 1$$
$$\gamma(x_2) - \nu(2.1) - t(x_2) = \gamma(x_7) - \nu(2.2) - t(x_7) = \gamma(x_8) - \nu(2.2) - t(x_8) = \gamma(x_{11}) - \nu(2.2) - t(x_{11}) = 1$$
$$\gamma(x_9) - \nu(2.3) - t(x_9) = \gamma(x_{10}) - \nu(2.3) - t(x_{10}) = \gamma(x_{13}) - \nu(2.3) - t(x_{13}) = 1$$

Now we show that there is a contradiction:

(a) from $\gamma(w) + \lambda(\mathsf{M}_1) - \gamma(x_1) = 0$ *and* $\gamma(w) = 0$*, we have* $\gamma(x_1) = \lambda(\mathsf{M}_1)$*,*

(b) from $\gamma(w) + \lambda(\mathsf{R}_1) - \gamma(x_2) = 0$ *and* $\gamma(w) = 0$*, we have* $\gamma(x_2) = \lambda(\mathsf{R}_1)$*,*

(c) from $\gamma(x_1) - \nu(2.1) = 0$ *and* $\gamma(x_1) = \lambda(\mathsf{M}_1)$ *by (a), we have* $\nu(2.1) = \lambda(\mathsf{M}_1)$,

(d) from $\gamma(x_2) - \nu(2.1) \geq 1$, $\gamma(x_2) = \lambda(\mathsf{R}_1)$ *by (b) and* $\nu(2.1) = \lambda(\mathsf{M}_1)$ *by (c), we have* $\lambda(\mathsf{R}_1) \geq 1 + \lambda(\mathsf{M}_1)$,

(e) from $\lambda(\mathsf{L}_2) = \lambda(\mathsf{l}_1) = 0$, $\gamma(x_1) + \lambda(\mathsf{l}_1) - \gamma(x_3) = \gamma(x_3) + \lambda(\mathsf{L}_2) - \gamma(x_7) = 0$ *and* $\gamma(x_1) = \lambda(\mathsf{M}_1)$ *by (a), we have* $\gamma(x_7) = \lambda(\mathsf{M}_1)$,

(f) from $\lambda(\mathsf{R}_3) = \lambda(\mathsf{l}_1) = 0$, $\gamma(x_2) + \lambda(\mathsf{l}_2) - \gamma(x_5) = \gamma(x_5) + \lambda(\mathsf{R}_3) - \gamma(x_{12}) = 0$, $\gamma(x_{12}) - \nu(2.2) = 0$ *and* $\gamma(x_2) = \lambda(\mathsf{R}_1)$ *by (b), we have* $\lambda(\mathsf{R}_1) = \nu(2.2)$,

(g) from $\gamma(x_7) - \nu(2.2) \geq 1$, $\gamma(x_7) = \lambda(\mathsf{M}_1)$ *by (e) and* $\lambda(\mathsf{R}_1) = \nu(2.2)$ *by (f), so we have* $\lambda(\mathsf{M}_1) \geq 1 + \lambda(\mathsf{R}_1)$,

(h) we have that (d) and (g) are in contradiction.

Thus, the above set of constraints does not admit any feasible assignment and therefore the assessment given in input is not an SE.

Example 6.5. *Consider the game depicted in Fig. 3, where* $\boldsymbol{\pi} = (\pi^1(\mathsf{M}_1) = 1, \pi^1(\mathsf{L}_2) = 1, \pi^1(\mathsf{R}_3) = 1, \pi^2(\mathsf{l}_1) = 1, \pi^2(\mathsf{l}_2) = 1, \pi^2(\mathsf{r}_3) = 1)$*, while beliefs* μ *are reported in the figure aside the corresponding nodes. The assessment* $(\mu, \boldsymbol{\pi})$ *is an SE and we show that the Algorithm 5 returns* **SE**. *Below we instantiate constraints of Program A.7.*

$$\lambda(\mathsf{M}_1) = \lambda(\mathsf{L}_2) = \lambda(\ \mathsf{R}_3) = \lambda(\mathsf{l}_1) = \lambda(\mathsf{l}_2) = \lambda(\mathsf{r}_3) = 0$$
$$\lambda(\mathsf{L}_1) - s(\mathsf{L}_1) = \lambda(\mathsf{R}_1) - s(\mathsf{R}_1) = \lambda(\mathsf{R}_2) - s(\mathsf{R}_2) = \lambda(\mathsf{L}_3) - s(\mathsf{L}_3) = 1$$
$$\lambda(\mathsf{r}_1) - s(\mathsf{r}_1) = \lambda(\mathsf{r}_2) - s(\mathsf{r}_2) = \lambda(\mathsf{l}_3) - s(\mathsf{l}_3) = 1$$
$$\gamma(w) = \gamma(w) + \lambda(\mathsf{M}_1) - \gamma(x_1) = \gamma(w) + \lambda(\mathsf{R}_1) - \gamma(x_2) = 0$$
$$\gamma(x_1) + \lambda(\mathsf{l}_1) - \gamma(x_3) = \gamma(x_1) + \lambda(\mathsf{r}_1) - \gamma(x_4) = \gamma(x_2) + \lambda(\mathsf{l}_2) - \gamma(x_5) = \gamma(x_2) + \lambda(\mathsf{r}_2) - \gamma(x_6) = 0$$
$$\gamma(x_3) + \lambda(\mathsf{L}_2) - \gamma(x_7) = \gamma(x_3) + \lambda(\mathsf{R}_2) - \gamma(x_8) = \gamma(x_4) + \lambda(\mathsf{L}_2) - \gamma(x_9) = \gamma(x_4) + \lambda(\mathsf{R}_2) - \gamma(x_{10}) = 0$$
$$\gamma(x_5) + \lambda(\mathsf{L}_3) - \gamma(x_{11}) = \gamma(x_5) + \lambda(\mathsf{R}_3) - \gamma(x_{12}) = \gamma(x_6) + \lambda(\mathsf{L}_3) - \gamma(x_{13}) = \gamma(x_6) + \lambda(\mathsf{R}_3) - \gamma(x_{14}) = 0$$
$$\gamma(x_3) - \nu(1.2) = \gamma(x_5) - \nu(1.3) = \gamma(x_1) - \nu(2.1) = \gamma(x_7) - \nu(2.2) = \gamma(x_9) - \nu(2.3) = 0$$
$$\gamma(x_4) - \nu(1.2) - t(x_4) = \gamma(x_6) - \nu(1.3) - t(x_6) = 1$$
$$\gamma(x_2) - \nu(2.1) - t(x_2) = \gamma(x_8) - \nu(2.2) - t(x_8) = \gamma(x_{11}) - \nu(2.2) - t(x_{11}) = 1$$
$$\gamma(x_{12}) - \nu(2.2) - t(x_{12}) = \gamma(x_{14}) - \nu(2.3) - t(x_{14}) = \gamma(x_{10}) - \nu(2.3) - t(x_{10}) = \gamma(x_{13}) - \nu(2.3) - t(x_{13}) = 1$$

A b–labeling is: $\lambda(a) = 1$ *for all* $a \in A^i$ *with* $\pi^i(a) = 0$*. Therefore, the assessment given in input is an SE.*

7. Conclusions and future works

In this paper, we focus on the verification problem, i.e., certifying that a solution given in input is an equilibrium according to some solution concept, for Nash Equilibrium refinements for extensive-form games. While the verification problem for Nash Equilibrium is trivial, with Nash Equilibrium refinements for extensive form games the problem may be hard. The only results known so far show that the verification problem is NP–hard for a number of cases.

We provide the following contributions. We show that, when the input is a realization plan profile (i.e., strategies for sequence form representation), deciding whether the input is part of a Subgame Perfect Equilibrium or part of a Sequential Equilibrium is NP–hard even in two–player games. This means that there is no polynomial–time algorithm unless P = NP, but it is commonly believed that P ≠ NP. Then, we show that, when the input is a behavioral strategy profile, there is a polynomial–time algorithm deciding whether the input is a Quasi Perfect Equilibrium in two–player games. A simple variation of this algorithm decides whether the input is part of some Sequential Equilibrium. This result completes the complexity of verification for Quasi Perfect Equilibrium and Sequential Equilibrium since, with three or more players, the problem is known to NP–hard. Finally, we show that, when the input is an assessment, there is a polynomial–time algorithm to decide whether the input is a Sequential Equilibrium regardless the number of players.

We believe these results are not only important in themselves, they also show the role played by inputs in extensive-form refinements. In particular, they show that beliefs are important not only to discuss the plausibility of refinements, they are also crucial for verification problem.

The main verification problems left open concern Extensive Form Perfect Equilibria and Proper Equilibria with two players. We will investigate them in future work.

References

C. Archibald and Y. Shoham. Modeling billiards games. In *AAMAS*, pages 193–199, 2009.

S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 2009.

N. Basilico, N. Gatti, and F. Villa. Asynchronous multi–robot patrolling against intrusions in arbitrary topologies. In *AAAI*, 2010.

S. Belhaiza, C. Audet, and P. Hansen. On proper refinement of Nash equilibria for bimatrix games. *AUTOMATICA*, 2:297–303, 2012.

L. Blum, A. Brandenburger, and E. Dekel. Lexicographic probabilities and equilibrium refinements. *ECONOMETRICA*, 59(1):81–98, 1991.

R. Chandrasekaran. Total unimodularity of matrices. *SIAM J APPL MATH*, 17(6): 1032–1034, 1969.

V. Conitzer and T. Sandholm. New complexity results about nash equilibria. *GAME ECON BEHAV*, 63(2):621–641, 2008.

D. Fudenberg and J. Tirole. *Game Theory*. 1991.

N. Gatti and F. Panozzo. New results on the verification of nash refinements for extensive–form games. In *AAMAS*, pages 813–820, Valencia, Spain, June 4–8 2012.

A. Ghouila-Houri. Caracterisation des matrices totalement unimodulaires. *C. R. Acad. Sci. Paris*, 254:1192–1194, 1962.

I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *GAME ECON BEHAV*, 1(1):80–93, 1989.

A. Gilpin, T. Sandholm, and T. B. Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *AAAI*, pages 50–57, 2007.

S. Govindan and T. Klumpp. Perfect equilibrium and lexicographic beliefs. *INT J GAME THEORY*, 31(2):229–243, 2003.

K. A. Hansen, P. B. Miltersen, and T. B. Sørensen. The computational complexity of trembling hand perfection and other equilibrium refinments. In *SAGT*, pages 198–209, 2010.

J. Hillas and E. Kohlberg. Chapter 42 foundations of strategic equilibrium. volume 3 of *Handbook of Game Theory with Economic Applications*, pages 1597–1663. Elsevier, 2002.

P. R. Jordan, M. P. Wellman, and G. Balakrishnan. Strategy and mechanism lessons from the first ad auctions trading agent competition. In *ACM EC*, pages 287–296, 2010.

E. Kohlberg and J. F. Mertens. On the strategic stability of equilibria. *ECONOMETRICA*, 54:1003–1037, 1986.

E. Kohlberg and P. J. Reny. Independence on relative probability spaces and consistent assessments in game trees. *J ECON THEORY*, 75(2):280–313, 1997.

D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J ARITF INTELL RES*, 41:297–327, 2011.

D. R. Kreps and R. Wilson. Sequential equilibria. *ECONOMETRICA*, 50(4):863–894, 1982.

H. W. Kuhn. *Contributions to the Theory of Games*, chapter Extensive Games and the Problem of Information, pages 193–216. Princeton University Press, 1953.

R. D. McKelvey and A. McLennan. Computation of equilibria in finite games. In H. M. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1, chapter 02, pages 87–142. Elsevier, 1 edition, 1996.

N. Meggido and C. H. Papadimitriou. A note on total functions, existence theorems, and computational complexity. Technical Report RJ 7091, IBM Research Report, 1989.

J. F. Mertens. The small worlds axiom for stable equilibria. *GAME ECON BEHAV*, 4 (4):553–564, 1992.

J. F. Mertens. Two examples of strategic equilibrium. *GAME ECON BEHAV*, 8(2): 378—388, 1995.

P. B. Miltersen and T. B. Sørensen. Computing a quasi–perfect equilibrium of a two–player game. *ECON THEOR*, 42(1):175–192, 2010.

C. Papadimitriou. Chapter 14 - the complexity of computing equilibria. volume 4 of *Handbook of Game Theory with Economic Applications*, pages 779–810. Elsevier, 2015.

C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J COMP SYS SCI*, 48(3):498–532, 1994.

T. Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3):45–58, 2007.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. 2008.

T. B. Sørensen. Computing a proper equilibrium of a bimatrix game. In *ACM EC*, Valencia, Spain, June 4–8 2012.

P. A. Streufert. A Comment on "Sequential Equilibria". UWO Department of Economics Working Paper 20063, University of Western Ontario, Department of Economics, 2006.

P. A. Streufert. Characterizing Consistency by Monomials. UWO Department of Economics Working Paper, University of Western Ontario, Department of Economics, 2007.

E. van Damme. A relation between perfect equilibria in extensive form games and equilibria in normal form games. *INT J GAME THEORY*, 13(2):1–13, 1984.

E. van Damme. *Stability and Perfection of Nash Equilibria*. Springer, 1991.

E. van Damme. Chapter 41 strategic equilibrium. volume 3 of *Handbook of Game Theory with Economic Applications*, pages 1521–1596. Elsevier, 2002.

B. von Stengel. Efficient computation of behavior strategies. *GAME ECON BEHAV*, 14 (2):220–246, 1996.

L. A. Wolsey. *Integer Programming*. Wiley–Interscience, 1998.

A. Beliefs consistent verification when the assessment is given

This section is devoted to construct the verification algorithm for the consistency of the beliefs. Verifying consistency of μ w.r.t. $\boldsymbol{\pi}$ requires to find a fully mixed perturbed strategy profile $\boldsymbol{\pi}(\varepsilon)$ with the property that the beliefs $\mu(\varepsilon)$ derived from $\boldsymbol{\pi}(\varepsilon)$ by Bayes rule converge to μ as ε goes to zero. Kreps and Wilson (1982), as corrected and integrated by Streufert (2006, 2007), show that the problem of searching for such a $\boldsymbol{\pi}(\varepsilon)$ can be formulated as the problem of searching for a b–labeling. Initially, we introduce the following definitions taken from Kreps and Wilson (1982); Streufert (2006, 2007).

Definition A.1. *A* basis *for the extensive form $\{T, <; A, \alpha; H; I, \iota\}$ is an index set b consisting of decision nodes $x \in X$ and actions $a \in A$.*

Definition A.2. *A basis b is consistent if the set Ψ_b, defined as*

$$\Psi_b = \{(\mu, \boldsymbol{\pi}) \in \Psi | \mu(x) > 0 \Leftrightarrow x \in b \quad \wedge \quad \boldsymbol{\pi}(a) > 0 \Leftrightarrow a \in b\}$$

is such that $\Psi_b \neq \varnothing$.

Definition A.3. *A labelling for the extensive form $\{T, <; A, \alpha; H; I, \iota\}$ is a function $\lambda : A \to \mathbb{N}$ that assigns a label (expressed as a nonnegative integer) to all the actions $a \in A$. For any given labelling λ, there is and associated function $J_\lambda : X \to \mathbb{N}$ defined as follows:*

$$J_\lambda(x) = \begin{cases} \sum_{k=0}^{l(x)-1} \lambda(\alpha(p_k(x))) | x \in H(x) & \text{if } x \in X \setminus W \\ 0 & \text{if } x \in W \end{cases}.$$

In words, λ labels the branches of the game tree with nonnegative integers (in a way that respects the informational constraints of the game) and J_λ gives for each node x the sum of the labels on branches from the beginning of the tree to x.

Definition A.4. *A labeling λ is a b–labelling if and only if*

$$\forall h, \exists a \in A(h) \qquad \lambda(a) = 0$$
$$a \in b \Longleftrightarrow \lambda(a) = 0$$
$$x \in b \Longleftrightarrow x \in \arg\min\{J_\lambda(x') \,|\, x' \in H(x)\}$$

In words, a b–labeling assigns zero to the labels of actions played with strictly positive probability and requires that, if and only if the belief over a node is strictly positive, then the sum of the labels of the actions leading to such node from the root node is the smallest among the sums of the labels of the actions leading to all the other nodes of the same information set.

Theorem A.5. [11] *Let*

$$\pi^i(a,\varepsilon) = \begin{cases} c(H(a),\varepsilon)\cdot\pi^i(a)\cdot\varepsilon^{\lambda(a)} & \text{if } \pi^i(a) > 0 \\ c(H(a),\varepsilon)\cdot\varepsilon^{\lambda(a)} & \text{otherwise} \end{cases}$$

where a is an action played by i at information set $H(a)$ from which the action a can be chosen, i.e. $H(a) = A^{-1}(a)$, and $c(H(a),\varepsilon)$ is the appropriate normalizing constant; then μ is consistent w.r.t. $\boldsymbol{\pi}$ if and only if a b–labeling λ exists.

It is worth remarking that an assessment $(\mu,\boldsymbol{\pi})$ may admit multiple b–labelings. In this case, all the b–labelings lead to the same result: if, with a given b–labeling, $\mu(\varepsilon)$ derived from $\boldsymbol{\pi}(\varepsilon)$ by Bayes rule converges to μ as $\varepsilon \to 0$, then the same happens with all the other b–labelings. Therefore, it is sufficient to search for a (generic) b–labeling.

Starting from the results of Kreps and Wilson (1982); Streufert (2006, 2007), we can show the following result.

Proposition A.6. [12] *Given an assessment $(\mu,\boldsymbol{\pi})$ and a b–labeling, there exists an efficient algorithm verifying that $\mu(\varepsilon)$, derived from $\boldsymbol{\pi}(\varepsilon)$ by Bayes rule, converges to μ as ε goes to zero.*

Proof. define $\mathsf{path}(w,x)$ as the sequence of pairs (x',a) with $a \in A(x')$ connecting the root node w of the game tree to decision node x. Beliefs can be derived from fully mixed strategies by Bayes rule as follows:

$$\mu(x,\varepsilon) = \frac{\prod_{(x',a)\in\mathsf{path}(w,x)} \pi^{\iota(x')}(a,\varepsilon)}{\sum_{x''\in H(x)} \prod_{(x',a)\in\mathsf{path}(w,x'')} \pi^{\iota(x')}(a,\varepsilon)}$$

[11]The statement of this theorem is a slight reformulation of Lemma A1 of Kreps and Wilson (1982) integrated by Theorem 2.1 of Streufert (2007).

[12]This result is clearly related to Theorem 2.1 on consistency and monomials of Streufert (2007).

The derivation of $\mu(x,\varepsilon)$ requires a number of operations that is linear in the size of the game because both $|\mathsf{path}(w,x)|$ and $|A(h)|$ are linear in the size of the game. We study the complexity of computing $\lim_{\varepsilon\to 0}\mu(x,\varepsilon)$. The normalization constants $c(H(a),\varepsilon)$ are:

$$c(H(a),\varepsilon) = \frac{1}{\sum_{a\in A(h):\pi^{\iota(h)}(a)>0} \pi^{\iota(h)}(a)\cdot\varepsilon^{\lambda(a)} + \sum_{a\in A(h):\pi^{\iota(h)}(a)=0} \varepsilon^{\lambda(a)}}$$

Then, let:

$$\begin{aligned}
d(H(a),\varepsilon) &= \frac{1}{c(H(a),\varepsilon)}\\
\psi_x &= \prod_{(x',a)\in\mathsf{path}(w,x)} \varepsilon^{\lambda(a)}\\
\phi_x &= \prod_{(x',a)\in\mathsf{path}(w,x):\pi^{\iota(x')}(a)>0} \pi^{\iota(x')}(a)\\
D_x &= \prod_{(x',a)\in\mathsf{path}(w,x),x'\in H(x)} d(h,\varepsilon)
\end{aligned}$$

Notice that $d(H(a),\varepsilon)$ is a polynomial in ε. We can write $\mu(x,\varepsilon)$ as

$$\mu(x,\varepsilon) = \frac{\psi_x\cdot\phi_x\cdot\dfrac{1}{D_x}}{\sum_{x''\in H(x)}\psi_{x''}\cdot\phi_{x''}\cdot\dfrac{1}{D_{x''}}} = \frac{\psi_x\cdot\phi_x\cdot\prod_{x''\in H(x),x''\neq x} D_{x''}}{\sum_{x''\in H(x)}\left(\psi_{x''}\cdot\phi_{x''}\cdot\prod_{x'''\in H(x),x'''\neq x''} D_{x'''}\right)}$$

In order to compute $\lim_{\varepsilon\to 0}\mu(x,\varepsilon)$, it is sufficient to isolate, for each $d(H(a),\varepsilon)$, the minimum degree of ε and its coefficient, discarding all the higher degrees of ε from $d(H(a),\varepsilon)$, and then calculate the multiplications and the sums to obtain the form $\frac{A_{\text{num}}\cdot\varepsilon^{B_{\text{num}}}}{A_{\text{den}}\cdot\varepsilon^{B_{\text{den}}}}$. This requires a number of operations that is linear in the size of the game. Notice that $B_{\text{num}} \geq B_{\text{den}}$. Indeed, the denominator of $\mu(x,\varepsilon)$ is given by the sum of some terms and the numerator appears among these terms. Therefore, the minimum degree of ε in the numerator cannot smaller than the minimum degree of ε in the denominator. Then, the calculation of the limit is customary:

- if $B_{\text{num}} > B_{\text{den}}$, then $\lim_{\varepsilon\to 0}\mu(x,\varepsilon) = 0$;
- if $B_{\text{num}} = B_{\text{den}}$, then $\lim_{\varepsilon\to 0}\mu(x,\varepsilon) = \frac{A_{\text{num}}}{A_{\text{den}}}$.

Hence, the proof of the proposition is complete. □

Taken together, Proposition A.6 and Theorem A.5 are a reformulation of Theorem 2.1 of Streufert (2007) focused to our aim, however it has a more general interesting content.

Broadly speaking, the results suggests that it is useful to represent each probability with a "monomial," defined in Streufert (2007) to be an algebraic expression of the form $c\varepsilon^{\lambda}$ in which c is a positive real number and λ is a nonnegative integer. Essentially, monomials with zero exponents express positive probabilities and monomials with positive exponents express different levels of zero probability (greater exponents express lesser zero probabilities). It is intuitive that the definition of consistency is satisfied if a monomial $c(a)\,\varepsilon^{\lambda(a)}$ can be assigned to each action a in such a way that

1. action a is played with probability $c(a)$ if the exponent $\lambda(a)$ is zero and is not played if $\lambda(a)$ is strictly positive, and,
2. the beliefs at each information set are found, first by calculating the product of the monomials along the path leading to each of the nodes in the information set, second by placing zero probability on every node whose product's exponent is less than that of another node, and finally by assigning positive probability over the remaining nodes in proportion to their products' coefficients.

The results in Kreps and Wilson (1982); Streufert (2006, 2007) and used here to our aim, formalize this intuition stating that consistency is equivalent to the existence of such monomials.

Now, we focus on the problem of searching for a b–labeling. Initially, we provide an integer linear mathematical program in Program A.7, where $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$ are auxiliary variables, while $\mathbf{s}$ and $\mathbf{t}$ are slack variables.

Program A.7.

$$\min \sum_{a\in A} \lambda(a) \tag{17}$$

$$\lambda(a) = 0 \qquad \forall a \in A : \pi^i(a) > 0, i \in I \tag{18}$$

$$\lambda(a) - s(a) = 1 \qquad \forall a \in A : \pi^i(a) = 0, i \in I \tag{19}$$

$$\gamma(w) = 0 \tag{20}$$

$$\gamma(x') + \lambda(a) - \gamma(x) = 0 \qquad \forall x, x' \in X, a \in A : x \in S(x'), a = \alpha(x) \tag{21}$$

$$\gamma(x) - \nu(h) = 0 \qquad \forall h \in H^i, x \in h : \mu(x) > 0, i \in I \tag{22}$$

$$\gamma(x) - \nu(h) - t(x) = 1 \qquad \forall h \in H^i, x \in h : \mu(x) = 0, i \in I \tag{23}$$

$$\lambda(a) \in \mathbb{N} \qquad \forall a \in A \tag{24}$$

$$s(a) \geq 0 \qquad \forall a \in A \tag{25}$$

$$t(x) \geq 0 \qquad \forall x \in X \tag{26}$$

Proposition A.8. *Given $(\mu, \boldsymbol{\pi})$, the problem of searching for a (specific, i.e., minimizing the sum of the labels) b–labeling can be formulated as Program A.7.*

Proof. We show that the above program captures the definition of b–labeling. Constraints (18) force labels of actions played with positive probability to be equal to '0'; constraints (19) force labels of actions played with zero probability to be larger than or equal to '1'; constraint (20) assigns '0' to auxiliary variable $\gamma(w)$ associated with root node w (auxiliary variable $\gamma(x)$ expresses the sum of the labels of all the actions leading to node x from the root node); constraints (21) assign the auxiliary variable $\gamma(x)$ associated with node x a value equal to the sum of the value $\gamma(x')$ of the parent node x' and the label of the action connecting x' to x; constraints (22) force the values of all the $\gamma(x)$s associated with the nodes xs with $\mu(x) > 0$ belonging to the same information set to be same (i.e., $H(x)$); constraints (23) force the other nodes (those with $\mu(x) = 0$) to have a value γ strictly larger than the minimum value of the information set; constraints (24)–(26) fix the domains of the variables (notice that, with these domains, all the variables have non–negative values). Objective function $\sum_{a \in A} \lambda(a)$ is lower bounded given that labels are non–negative. Therefore, the minimization (17) always returns a b–labeling if this exists. □

Although solving an arbitrary integer linear mathematical program may be NP–hard in the worst case, we can show that Program A.7 can be solved in polynomial time. A preliminary observation is that we can find a feasible (potentially non–minimizing the objective function) solution in polynomial time by the following procedure: we relax the integrity constraints, we solve the continuous relaxation by linear mathematical programming (that can be achieved in polynomial time) returning a rational solution if it exists, then, if the solution is fractional, we find the least common multiple of all the denominators (that can be found in polynomial time) and finally we multiply the solution by that number. In this way, the solution obtained is integer. Notice that such a procedure would require the adoption arbitrary–precision arithmetic during the resolution of the linear mathematical program, being necessary to represent exactly the denominators, and this would make the resolution process much more expensive. However, we can show that such a procedure can be dramatically simplified, since every basic solution of the continuous relaxation of Program A.7 is integer and therefore any basic solution of the continuous relaxation is also solution for the integer program. As a result, a solution can be found by means of only linear programming without resorting to arbitrary–precision arithmetic and without resorting to any algorithm to compute the least common multiple. To prove it we exploit the property of total unimodularity provided in Ghouila-Houri (1962).

Given that a matrix Ξ is totally unimodular if and only if the transpose Ξ^T is totally unimodular as shown by Chandrasekaran (1969), we can restate the definition of total unimodularity provided by Ghoulia–Houri as follows.

Definition A.9 (Total unimodularity). *Matrix Ξ is totally unimodular if and only if for every subset Ξ' of columns of Ξ it is possible to find a partition of columns $\{\Xi'_1, \Xi'_2\}$*

such that:

$$\forall k \left(\sum_{j:\xi_{kj} \in \Xi'_1} \xi_{kj} - \sum_{j:\xi_{kj} \in \Xi'_2} \xi_{kj} \right) \in \{-1, 0, 1\} \tag{27}$$

where ξ_{kj} a generic entry of matrix Ξ.

In order to resort to total unimodularity for Program A.7, we need to formulate such program in standard form and isolate the matrix of constraints. More precisely, constraints (18)–(26) can be expressed as $M \cdot \mathbf{y} = \mathbf{b}$ with $\mathbf{y} \geq 0$ and $\boldsymbol{\lambda} \in \mathbb{N}^{|A|}$, where:

$$M = \begin{bmatrix} C & 0 & 0 & 0 & 0 \\ C' & 0 & 0 & -I & 0 \\ D & E & 0 & 0 & 0 \\ 0 & G & K & 0 & 0 \\ 0 & G' & K' & 0 & -I \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \boldsymbol{\nu} \\ \mathbf{s} \\ \mathbf{t} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

such that:

- $C \cdot \boldsymbol{\lambda} = \mathbf{0}$ captures constraints (18),
- $C' \cdot \boldsymbol{\lambda} - I \cdot \mathbf{s} = \mathbf{1}$ captures constraints (19),
- $D \cdot \boldsymbol{\lambda} + E \cdot \boldsymbol{\gamma} = \mathbf{0}$ captures constraints (20) and (21),
- $G \cdot \boldsymbol{\gamma} + K \cdot \boldsymbol{\nu} = \mathbf{0}$ captures constraints (22), and
- $G' \cdot \boldsymbol{\gamma} + K' \cdot \boldsymbol{\nu} - I \cdot \mathbf{t} = \mathbf{1}$ captures constraints (23).

Since vector $\mathbf{b}$ is integer, if M is totally unimodular, then any basic solution of the continuous relaxation is integer, see Wolsey (1998). To show that M is totally unimodular we need two lemmas, that are provided within the proof of the propositions because they refers to concepts introduced during the proof.

Proposition A.10. *Matrix M is totally unimodular.*

Proof. Let us note that, usually, totally unimodular matrices have a network–flow interpretation which makes the proof of total unimodularity straightforward Schrijver (2003). In our case, we did not find any simple such interpretation and therefore we prove the proposition by analyzing the structure of matrix M.

Initially, we remark that all the entries of M belongs to $\{-1, 0, 1\}$ and that the submatrices of M have the following property:

- C, C', G, and G' have one '1' per row and zero or one '1' per column;
- D is composed of a row of '0's and identity matrix I;

- E has one '1' in the first row and one '1' and one '−1' in all the other rows;

- K and K' present one '−1' per row.

Let Λ_k the k–th block of rows of M (from the top to the bottom) and let Δ_j the j–th block of columns of M (from the left to the right), as shown below:

C	0	0	0	0	Λ_1
C'	0	0	$-I$	0	Λ_2
D	E	0	0	0	Λ_3
0	G	K	0	0	Λ_4
0	G'	K'	0	$-I$	Λ_5
Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	

Let M' a subset of columns of M and m_{kj} a generic element of M. We need to show that, for any M', we can find a partition of columns $\{M'_1, M'_2\}$ such that constraints (27) are satisfied.

At first, we notice that the total unimodularity of M is not conditioned by the blocks of columns Δ_4 and Δ_5.

Lemma A.11. *If matrix $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$ is totally unimodular, then M is totally unimodular.*

Proof. Assume that for every M' it is possible to find a partition of columns $\{M'_1, M'_2\}$ such that

$$\forall k \left(\sum_{j:m_{kj} \in M'_1 \cap \{\Delta_1 \cup \Delta_2 \cup \Delta_3\}} m_{kj} - \sum_{j:m_{kj} \in M'_2 \cap \{\Delta_1 \cup \Delta_2 \cup \Delta_3\}} m_{kj} \right) \in \{-1, 0, 1\} \tag{28}$$

In words: we are requiring that constraints (27) are satisfied under the additional constraint $m_{kj} \in M' \cap \{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$. Now, we prove the Lemma, providing an iterative procedure to assign each column of $M' \cap \{\Delta_4 \cup \Delta_5\}$ to M_1 or M_2. Initially, we observe that each column of $\{\Delta_4 \cup \Delta_5\}$ contains one '−1', while all the other entries are '0s', and that each row contains at most one '−1'. Therefore, each column can be assigned to M_1 or M_2 independently from the assignment of the others. Thus, take a column $\overline{r}$ of $M' \cap \{\Delta_4 \cup \Delta_5\}$ and call $\overline{k}$ the row in which there is '−1'. Assign column $\overline{k}$ to M'_1 or M'_2 as follows:

- if the left hand of constraints (27) for $k = \overline{k}$ is equal to '−1', then assign column $\overline{r}$ to M'_2, in this way $\sum_{j:m_{\overline{k}j} \in M'_1} m_{\overline{k}j} - \sum_{j:m_{\overline{k}j} \in M'_2} m_{\overline{k}j} = 0$,

- if the left hand of constraints (27) for $k = \overline{k}$ is equal to '1', then assign column $\overline{r}$ to M'_1, in this way $\sum_{j:m_{\overline{k}j} \in M'_1} m_{\overline{k}j} - \sum_{j:m_{\overline{k}j} \in M'_2} m_{\overline{k}j} = 0$,

- if the left hand of constraints (27) for $k = \overline{k}$ is equal to '0', then assign column $\overline{r}$ to M_1' or M_2' indifferently, in this way $\sum_{j:m_{\overline{k}j} \in M_1} m_{\overline{k}j} - \sum_{j:m_{\overline{k}j} \in M_2'} m_{\overline{k}j} \in \{-1, 1\}$.

Since all the other rows k of $\overline{r}$ contain '0s', $\sum_{j:m_{kj} \in M_1'} m_{kj} - \sum_{j:m_{kj} \in M_2'} m_{kj}$ is equal to $\sum_{j,m_{kj} \in M_1' \cap \{\Delta_1 \cup \Delta_2 \cup \Delta_3\}} m_{kj} - \sum_{j,m_{kj} \in M_2' \cap \{\Delta_1 \cup \Delta_2 \cup \Delta_3\}} m_{kj}$ that, by above assumption, belongs to $\{-1, 0, 1\}$. This proves that, if constraints (27) are satisfied for the first three blocks of columns, then such constraints can be satisfied also for all the blocks of columns with an opportune assignment of the columns of the last two blocks. □

By Lemma A.11, M is totally unimodular if matrix $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$ is totally unimodular. Thus, from here on, we study only the total unimodularity of $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$.

Lemma A.12. *Matrix $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$ is totally unimodular.*

Proof. We build M_1' and M_2' as follows. Assign all the columns of $M' \cap \{\Delta_2 \cup \Delta_3\}$ to M_1', while the columns of $M' \cap \Delta_1$ will be assigned to M_1' or M_2' to satisfy the total unimodularity condition as described below. Consider the rows belonging to Λ_4 and Λ_5 and sum the entries on these rows in M_1' independently of whether the columns of $M' \cap \Delta_1$ are assigned to M_1' or M_2': given that all the columns of $M' \cap \{\Delta_2 \cup \Delta_3\}$ are assigned to M_1' and all the entries of columns Δ_1 are zeros, the sum belongs to $\{-1, 0, 1\}$ (we recall, as discussed above, that G and G' have one '1' per row, while K and K' have one '−1' per row). Thus, the rows belonging to Λ_4 and Λ_5 satisfy constraints (27) independently of whether the columns of $M' \cap \Delta_1$ are assigned to M_1' or M_2'. Consider the columns of M_1' belonging to Δ_2: the sum of the entries of the rows belonging to Λ_3 can be $\{-1, 0, 1\}$ (we recall, as discussed above, that E has one '1' in the first row and one '1' and one '−1' in all the other rows). It can be easily seen that, D having no more than one '1' per column and per row, we can always assign the columns of Δ_1 to M_1' or M_2' to make that constraints (27) are satisfied on the rows belonging to Λ_3. Finally, we observe that constraints (27) are always satisfied on the rows belonging to Λ_1 and Λ_2 given that C and C' have one '1' per row. Therefore, matrix $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$ is totally unimodular. □

Since, by Lemma A.11, if $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$ is totally unimodular, then M is totally unimodular and, by Lemma A.12, $\{\Delta_1 \cup \Delta_2 \cup \Delta_3\}$ is totally unimodular, we have that M is totally unimodular. This completes the proof of the proposition. □