PSPACE suffices for deciding Nash equilibria properties for extensive games with large trees. [1]

Carme Àlvarez   Joaquim Gabarró   Maria Serna

Dep. LSI, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, Barcelona 08034, Spain.

Abstract. We study the computational complexity of deciding the existence of a Pure Nash Equilibrium or a subgame perfect equilibrium with a given payoff and other related problems in finite multi-player extensive games with perfect information. We first propose three way of representing a game with different degrees of succinctness for the components of the game. We show that when the number of moves of each player is large and the player function and the utilities are represented succinctly the considered problems are PSPACE-complete. In contraposition, when the game is described extensively by means of its associated tree all the problems are decidable in polynomial time. **Keywords**. *Extensive games, Pure Nash equilibria, Subgame perfect Nash equilibria, PSPACE.*

1 Introduction

Many situations involving interactions between independent agents can be modeled and analyzed by game theory. Game theory has been successfully applied to economics and computer science, among other disciplines. In recent times a lot of attention has been devoted to the computational aspects of game theory and specially to the relationship between Internet and games [15]. Finding Nash equilibria of games is central to game theoretical analysis and computing such equilibria is the principal aim of algorithmic game theory. The fundamental question posed by Papadimitriou [15] about the complexity of computing a Nash equilibrium in two player strategic games has initiated a line of research towards understanding the complexity of computing a pure or a mixed Nash equilibrium [5, 6, 4, 9, 2, 7, 1, 3, 16]. Recall that strategic games are one shot games in the sense that the players play simultaneously and only once.

Extensive games provide a natural way to model interactions between agents that involve sequential decisions. The game can be represented by a tree of decision nodes, and actions are represented as arcs in the tree. In a *perfect information game*, each player has complete information of the game's history. That is, each player can identify uniquely its current node at any time of the play. In an *imperfect information game*, agents are uncertain about which node they are currently in. Most of the algorithmics and complexity results on extensive games deal with imperfect information games whose tree is given as input [17, 10]. In terms of Game Theory two notions of equilibria play a fundamental role in the analysis of extensive games. The first one is that of Nash equilibria and the second one is that of subgame perfect Nash equilibria, a stronger notion of equilibrium. The fundamental question of the existence of a subgame perfect Nash equilibria, and therefore of a Nash equilibria, was answered positively as it is known that any finite extensive game with perfect information has a subgame perfect Nash equilibria [11]. Perfect information games have deserved less attention, as besides one Nash equilibrium can be compute using *backward induction* [13]. This is not the case for imperfect information games.

Backward induction has two limitations, the first one is that, in general, it is not possible to use it to obtain a Nash equilibrium with a required payoff for some player or other required

[1] Work partially supported by IST program of the EU under contract IST-2004-015964 (AEOLUS) and CoreGRID Network of Excellence and by Spanish CICYT under grant TIC2002-04498-C05-03 (Tracer).

Non-uniform representation	IsPN and IsSPN k-players ($k \geq 1$)	IsPNOut and IsSPNOut k-players ($k \geq 2$)	PNGrant and SPNGrant k-players ($k \geq 2$)
implicit	coNP-complete	PSPACE-complete	PSPACE-complete
general	coNP-complete	PSPACE-complete	PSPACE-complete
explicit	P	P	P

Fig. 1. Summary of the results of the paper

property. The second one is that the procedure needs time and space proportional to the size of the tree of the game. When one thinks on traditional games that involve sequential decisions, like like Chess, Geography and other games considers in the computational complexity community [14, 8], the tree of the game is exponentially large with respect to the size of the description of the game, thus backward induction will require exponential space. This fact leads us to study the complexity of deciding the existence of a Nash equilibrium with particular properties for the case of extensive games with perfect information when the game trees might have large size with respect to the number of moves of each player, i.e. the number of sequential decisions that can be taken by all the players.

Particular attention must be paid to the representability of the data sets appearing in the definition of a game, the player strategies and the game outcomes. We note that games computationally meaningful should have a large number of players, a large number of actions for each player, or a large number of moves. Furthermore the set of actions and the payoff functions can be given in some explicit or implicit way, using more or less space. In this paper we initiate a work towards understanding fully the components that play a fundamental role in the complexity of problems on extensive games. Following the systematic of [1] for strategic games, we consider three natural ways of giving a game as the input to a program. Each of them captures a different level of succinctness, depending on the description explicit or implicit of some of its components.

We address three basic problems, the first one asks whether a given strategy profile is a Nash equilibrium, the IsPN problem. The second one asks whether a given history, is the outcome of a Nash equilibrium, the IsPNOut problem. Finally, the third one given a value u asks whether there is a pure Nash equilibrium in which the first player gets a payoff u, the PNGrant problem. The same three questions can be asked replacing the term pure Nash equilibrium by subgame perfect Nash equilibrium, giving raise to the problems IsSPN, IsSPNOut and SPNGrant.

Our results, summarized in Figure 1, show that for games represented explicitly all the problems can be solved in polynomial time. When some of the components of the game are represented implicitly (by means of Turing machines) the IsPN and the IsSPN become coNP-complete, while the other four problems become PSPACE-complete. Furthermore, the hardness results hold also for games with two players.

The main difficulty in finding algorithms for deciding equilibria properties is that when the strategy profile is not given as a part of the input then it requires space proportional to the size of the tree of the game in order to be represented. This means that if we do not have an alternative way to represent a Nash equilibrium, then the space needed to describe a Nash equilibrium is exponential with respect to the input size when the game is given in a succinct way. We provide recursive characterizations of the existence of a pure Nash equilibrium, or a subgame perfect Nash equilibrium, with a guaranteed payoff for the first player in terms of histories instead of strategy profiles. Although those characterizations might not be useful from the point of view of game theory, they are crucial from the complexity point of view since a history requires only polynomial space in the number of sequential moves (depth of the game tree) and the length of the actions, while a strategy profile might require exponential space in the number of sequential moves.

The hardness results follows from adequate reductions so that a quantified boolean formula is described as a game in which the particular equilibria we are looking for appear only when the quantified boolean formula is satisfiable.

Finally, we want to point out the differences with the case of strategic games obtained in [1]. For strategic games, the succinctness of the representation of the set of actions is relevant since the PNE problem that asks for the existence of a pure Nash equilibrium in a given strategic game is polynomial time solvable when the representation is explicit, NP-complete when the game is given in general form, and Σ_2^p-complete when the game is given in implicit form. The number of players is also relevant, for a constant number of players $k \geq 2$ the PNE is polynomial time solvable when the game is given in general form. Contrasting with this, in the extensive games we obtain the same complexity results when the game is given in general form or in implicit form.

2 Extensive games with perfect information

We start introducing some operations and notation on languages and functions. Given a language $L \subseteq \Sigma^*$ and a word $a \in \Sigma^*$ we define the languages $aL = \{aw \mid w \in L\}$ and $a^{-1}L = \{w \mid aw \in L\}$. Given a function $\tau : L \to B$ and a symbol $a \in \Sigma$ we define the functions $a\tau : aL \to B$, where $a\tau(aw) = \tau(w)$, and $a^{-1}\tau : a^{-1}L \to B$, where $a^{-1}\tau(w) = \tau(aw)$. The previous operations can be extended to prefix formed by words in the usual way, $(az)L = a(zL)$ and $(za)^{-1}L = z^{-1}(a^{-1}L)$ and similarly for functions. Given a collection of pairwise disjoint languages $(L_1, \ldots, L_k)$ and a set of functions $\tau_1, \ldots, \tau_k$ where, for each $1 \leq j \leq k$, $\tau_j : L_j \to B$, their *union* is the function $\tau = (\tau_1, \ldots, \tau_k)$ defined on $\cup_{i=1}^k L_i$ as $\tau(w) = \tau_i(w)$ for $w \in L_i$. Given a function $\tau = (\tau_1, \ldots, \tau_k)$, for any i, $1 \leq i \leq k$, τ_{-i} is the function $(\tau_1, \ldots, \tau_{i-1}, \tau_{i+1}, \ldots, \tau_k)$. For an element $b \in B$, the function τ_b is defined on $\{\lambda\}$ as $\tau_b(\lambda) = b$.

Now we state the definition of finite extensive games with perfect information as it is given in [13]. We have expressed some conditions using standard Computer Science notation for alphabets, words, and concatenation.

Definition 1. *A* finite extensive game with perfect information *is a structure* $\Gamma = (N, (A_i)_{i \in N}, H, Z, P, (u_i)_{i \in N})$ *where*

- *$N = \{1, \ldots n\}$ is the set of players. For each player $i \in N$, A_i is a finite set of actions.*
- *H is a finite set of words defined on the alphabet $A_1 \cup \cdots \cup A_n$. Each element of H is called a* history*. H satisfies the following properties: The empty word λ is a member of H and if $h \in H$ all the prefixes of h belong to H.*
- *$Z \subseteq H$ is formed by the terminal histories. A history $h \in H$ is* terminal *if h cannot be extended into a larger history.*
- *P is the* player function*. P assigns to each nonterminal history, $h \in H \setminus Z$, a player. $P(h)$ is the player who is allowed to take an action after the history h. Thus, for any $h \in H \setminus Z$, if for some action a we have that $h \cdot a \in H$ then $a \in A_{P(h)}$.*
- *For each player $i \in N$, a utility function u_i assigning to each terminal history a rational value.*

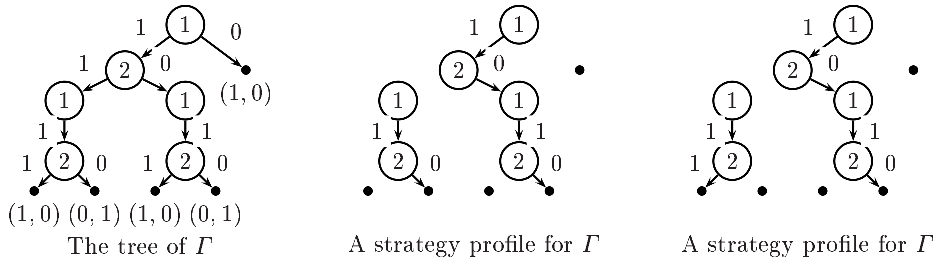

Fig. 2. The tree and a strategy profile for the game $\Gamma = (N, A_1, A_2, H, Z, (u_i)_{i\in N})$ where $N = \{1, 2\}$, $A_1 = A_2 = \{0, 1\}$, $H = \{0, 1, 11, 10, 111, 101, 1111, 11110, 1011, 1010\}$, and $Z = \{0, 1111, 1110, 1011, 1010\}$.

The simplest extensive game is an *empty game* denoted by $\Gamma_\lambda(v_1, \dots, v_n)$. This game has n players and $H = Z = \{\lambda\}$, therefore Γ_λ has only one history that is terminal, and, for any $1 \le i \le n$, $u_i(\lambda) = v_i$. Observe that no player plays, but player i gets immediately the payoff $u_i(\lambda)$. The *length of a game*, $\mathsf{length}(\Gamma)$, is the length of the longest history in the game. Thus $\mathsf{length}(\Gamma_\lambda(v_1, \dots, v_n)) = 0$.

An alternative definition of game is done by means of a rooted tree. The *tree of a game* is a rooted tree with labeled nodes and arcs. We assume that the arcs of the tree are directed from parent to child. We require all the labels in the outgoing arcs of a node to be different. Such a tree defines a game $\Gamma = (N, (A_i)_{i\in N}, H, Z, P, (u_i)_{i\in N})$ as follows. N is the set of the labels of the internal nodes. For $i \in N$, A_i is the set formed with the labels of the outgoing arcs of a node labeled i. The word obtained by the concatenation of the labels in the path from the root to a node u is the history associated to u. H is the set of histories associated to the tree nodes and Z is the set of histories associated to the leaves. The player function assigns to a non terminal history the label of its associated node. The leaves of the tree hold a table storing one rational for each player, thus defining the utility of the associated terminal history for each player. The tree associated to an empty game is a leaf holding the utility values associated to the history λ. In the following we formalize the notion of subgame.

Definition 2. *Given an extensive game $\Gamma = (N, (A_i)_{i\in N}, H, Z, P, (u_i)_{i\in N})$, the* subgame *of Γ after history h is the extensive game $h^{-1}\Gamma = (N, (A_i)_{i\in N}, h^{-1}H, h^{-1}Z, h^{-1}P, (h^{-1}u_i)_{i\in N})$.*

The following result follows from the definitions.

Lemma 1. *Given $h \in H\backslash Z$ and an action a such that $ha \in H$ it holds $(ha)^{-1}\Gamma = a^{-1}(h^{-1}\Gamma)$.*

Our second operation is the composition of games. The intented meaning of the composition is "player i starts the game and chooses an action, from a set of ℓ actions, each selection makes the players continue playing a particular game among a set of ℓ games".

Definition 3. *Given a collection of ℓ games $(\Gamma_1, \dots, \Gamma_\ell)$ where, for any $1 \le \alpha \le \ell$, $\Gamma_\alpha = (N, (A_i)_{i\in N}, H^\alpha, Z^\alpha, P^\alpha, (u_i^\alpha)_{i\in N})$, a player $i \in N$, and ℓ different actions from A_i, $a_{i_1}, \dots a_{i_\ell}$, the* composite game *$\Gamma(i, a_{i_1}\Gamma_1 \dots a_{i_\ell}\Gamma_\ell)$ is the game $(N, (A_i)_{i\in N}, H, Z, P, (u_i)_{i\in N})$, where $H = \{\lambda\} \cup a_{i_1}H^1 \cup \dots \cup a_{i_\ell}H^\ell$, $Z = a_{i_1}Z^1 \cup \dots \cup a_{i_\ell}Z^\ell$, $P = (\tau_i, a_{i_1}P^1, \dots, a_{i_\ell}P^\ell)$, and for every player $j \in N$, $u_j = (a_{i_1}u_j^1, \dots, a_{i_\ell}u_j^\ell)$,*

Over the tree of the game, the composition takes the tree of the component games, creates a new root with label i and join the new root to the root of game Γ_α with an arc labeled a_{i_α}.

The composition operation allows us to decompose games with positive length recursively from subgames. Given a nonterminal history h, let $\mathsf{next}(h) = \{a \in A_{P(h)} \mid ha \in H\}$.

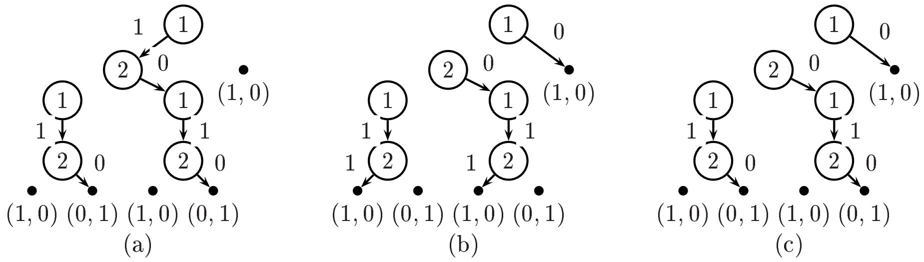

Fig. 3. Some strategy profiles for the game Γ given in Figure 2. The strategy given in (a) is not a Nash equilibrium. The strategy given in (b) is a Nash equilibrium, but not a subgame perfect Nash equilibrium. Finally, the strategy (c) is a subgame perfect Nash equilibrium.

Note that, an extensive game with perfect information Γ with $length(\Gamma) > 0$ can be factorized as $\Gamma = (i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$ where $\mathsf{next}(\lambda) = \{a_{i_1}, \ldots, a_{i_\ell}\}$, $i = P(\lambda)$, and, for any $1 \le \alpha \le \ell$, $\Gamma_\alpha = a_{i_\alpha}^{-1}\Gamma$.

Now we turn our attention to how the players play. Let Γ be an extensive game with perfect information, for any player $i \in N$, define $H_i = \{h \in H \mid P(h) = i\}$. Notice that the languages in the collection $(H_i)_{i\in N}$ are pairwise disjoint. A *strategy of player $i \in N$ in Γ* is a function $s_i : H_i \to A_i$. A *strategy profile* $s = (s_1, \ldots, s_n)$ is the union of a set of n strategies, one for each player. Note that over the tree of the game a strategy profile is a selection of an outgoing arc for every internal node (see Figure 2). When $\Gamma = \Gamma_\lambda(v_1, \ldots, v_n)$ the players do not play ($H_i = \emptyset$), or in other words, they play the *empty strategy* and thus the game has a unique strategy profile, the *empty strategy profile*.

For a player $i \in N$, the *adversary team* is represented by $-i$ meaning all the other players. Given a strategy profile s the set of strategies for the team $-i$ is represented by s_{-i}. Observe that in our notation s_{-i} is a function defined on $(H\backslash Z)\backslash H_i$ as $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$.

For each *strategy profile* $s = (s_1, \ldots s_n)$ the *outcome* of s, $O(s)$, is the terminal history that results when each player $i \in N$ follows the precepts of s_i. That is, $O(s) = a_1 \cdots a_k$ where $a_1 = s_{P(\lambda)}(\lambda)$ and, for any $1 \le i < k$, $a_{i+1} = s_{P(a_1 \ldots a_i)}(a_1 \ldots a_i)$. Looking at the tree of the game, the outcome of a strategy profile is the history associated to the unique leaf accessible from the root following the arcs selected by the strategy profile (see Figure 2). Note that different strategy profiles s and s' might determine the same outcome as it it shown in Figure 2.

Finally, we define the *utility for player i* of a strategy profile s as the utility of the outcome determined by s, that is $u_i(s) = u_i(O(s))$.

Let us consider strategies in front of subgames. Let Γ be a game with positive length and let $s = (s_1, \ldots, s_n)$ be a strategy profile. The strategy profile induced by s in the game after history h, $h^{-1}\Gamma$, is the function $h^{-1}s$. Assuming that Γ factorizes as $(i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$, for any $1 \le \alpha \le \ell$, the strategy profile profile induced by s on Γ_α is $a_{i_\alpha}^{-1}s$.

On the reverse side we can define strategies by extension of strategies on subgames. Assume that $\Gamma = (i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$. Given β, $1 \le \beta \le \ell$, and for any α, $1 \le \alpha \le \ell$, a strategy profile s^α. The *composite strategy profile* is $s = (\tau_{a_{i_\beta}}, a_{i_1}s^1, \ldots, a_{i_\ell}s^\ell)$.

We have given the definitions for strategy profiles, they can be easily extended to player strategies, replacing s by s_i, or to strategy profiles for the adversary teams, replacing s by s_{-i}. Finally, we state the definitions of Pure Nash equilibrium and subgame perfect Nash equilibrium as it is done in Game Theory.

Definition 4. *A strategy profile s^* in Γ is a* Nash equilibrium *if for every player $i \in N$ we have $u_i(s^*_{-i}, s^*_i) \ge u_i(s^*_{-i}, s_i)$, for every strategy s_i of player i.*

Observe that in an empty game the empty strategy profile is a Nash equilibrium.

Definition 5. *A strategy profile s^* in Γ is a* subgame perfect Nash equilibrium *if for every history h, the strategy profile $h^{-1}s^*$ is a Nash equilibrium in $h^{-1}\Gamma$.*

Note that every subgame perfect Nash equilibrium is also a Nash equilibrium but the reverse is not always true (see Figure 3). Furthermore, for every history h, the strategy profile $h^{-1}s^*$ is a subgame perfect Nash equilibrium in $h^{-1}\Gamma$. Observe that for an empty game Γ_λ the empty strategy profile is a subgame perfect Nash equilibrium, and thus Nash equilibrium.

As we have pointed out any finite extensive game has a strategy profile that is a subgame perfect Nash equilibrium [11], and therefore also a pure Nash equilibrium. Therefore, we are interested in deciding the existence of Nash equilibria or subgame perfect Nash equilibria that verify additional properties.

3 Problems on games and their input description

We are interested in the study of the computational complexity of the following problems.

Is pure Nash equilibrium? (IsPN). Given an extensive game Γ and a strategy profile s, decide whether s is a Nash equilibrium of Γ.

Is pure Nash equilibrium outcome? (IsPNOut). Given an extensive game Γ and a history h, decide whether h is the outcome of a Nash equilibrium of Γ.

Pure Nash equilibrium with guarantee (PNGrant). Given an extensive game Γ and a positive number u, decide whether Γ has a Nash equilibrium s such that $u_{P(\lambda)}(s) = u$.

Is subgame perfect Nash equilibrium? (IsSPN). Given an extensive game Γ and an strategy profile s, decide whether s is a subgame perfect Nash equilibrium of Γ.

Is subgame perfect Nash equilibrium outcome? (IsSPNOut). Given an extensive game Γ and an history h, decide whether h is the outcome of a subgame perfect Nash equilibrium of Γ.

Subgame perfect Nash equilibrium with guarantee (SPNGrant). Given an extensive game Γ and a positive number u, decide whether Γ has a Nash equilibrium s such that $u_{P(\lambda)}(s) = u$.

The first decision to take is about the representation of a game as an input to a program. Some of the components of a game can be given with different degrees of succinctness. For example the player and the utility functions can be given tabulated or by means of Turing machines. In the following definitions P and M denote the player and utility TMs and t denotes the computation time bound. The *player machine* P implements a player function according to the following convention: Given an input sequence h, P computes in time at most t an integer i, whenever $P(h) \in \{1, \dots, n\}$ it is interpreted as the player function on history h; $P(h) = 0$ indicates that $h \in Z$; otherwise P indicates that $h \notin H$. The *utility* machine M on input (h, i) outputs in at most t steps $u_i(h)$. We consider three natural ways of giving a game, according to the succinctness of actions, player function, and utilities. The criteria are similar to those proposed for strategic games in [1]. In the following definitions t is the computation time bound for any TM appearing in the description.

Extensive games in implicit form. A game is a tuple $\Gamma = \langle 1^n, 1^m, 1^h, P, M, 1^t \rangle$. This game has n players. For any i, $1 \le i \le n$, the set of actions of player i is Σ^m. The set H of histories is a subset of Σ^{mh}.

Extensive games in general form. A game is a tuple $\Gamma = \langle 1^n, A_1, \ldots, A_n, 1^h, P, M, 1^t\rangle$. The game has n players. For any i, $1 \leq i \leq n$, the set of actions of player i is A_i, which is given explicitly. The set H of histories is a subset of $(\cup_{1\leq i\leq n} A_i)^h$.

Extensive games in explicit form. A game is a tuple $\Gamma = \langle 1^n, T\rangle$. The game has n players and T is the game tree.

Observe that an explicit description of a strategy requires space proportional to the size of the tree associated to a game. Therefore, since strategies can be given either implicitly or explicitly we take the following natural convention. When a game Γ is described in a succinct (implicit or general) form, the strategies will be given implicitly. A strategy is represented implicitly by a TM S, *the global strategy* TM so that on any non terminal history $h \in H \setminus Z$ outputs the action to be undertaken by the player that is allowed to play, formally $S(h) \in A_{P(h)}$. In the case of games given in explicit form, the strategy will be also represented in explicit form. This is a natural decision, having a strategy given in explicit form as input will allow us to compute a explicit form of the game in polynomial time. For sake of simplicity whenever we want to represent a game and a strategy we will incorporate a unique time bound that will be used in all the TMs involved in the input's description.

In the following sections we analyze the computational complexity of this set of six problems depending on the form in which the input game is given.

4 A characterization of the outcomes of Nash equilibria

We provide first a characterization of the existence of a Pure Nash equilibrium with a particular outcome. This result will be crucial in the classification of problems because it reduces the computation space from exponential to polynomial. Our recursive characterization will allow us to place the problem in PSPACE. We start with one technical definition.

Definition 6 ((i,u)-blocking). *Let Γ be an extensive game, i a player, and u a value. We say that Γ is* (i,u)-blocking *if $\mathsf{length}(\Gamma) = 0$ and $u_i(\lambda) \leq u$, or if $\mathsf{length}(\Gamma) > 0$ and either*

- *$P(\lambda) \neq i$ and there is $a \in \mathsf{next}(\lambda)$ such that $a^{-1}\Gamma$ is (i,u)-blocking, or*
- *$P(\lambda) = i$ and, for all $a \in \mathsf{next}(\lambda)$, the game $a^{-1}\Gamma$ is (i,u)-blocking.*

Our first result is a characterization of the previous property in terms of the existence of an adversarial team strategy s_{-i} that keeps player i benefit at most u. The proof is given in appendix A.

Lemma 2. *Given an extensive game Γ, a player i and value u, Γ is (i,u)-blocking if and only if there is a strategy profile s_{-i} such that, for any s_i it holds $u_i(s_{-i}, s_i) \leq u$.*

Our second definition is the key to capture through a recursive definition the concept of outcome of a Nash equilibrium as it is shown in the next result.

Definition 7 (Γ-stable). *Let Γ be an extensive game and h be a terminal history in Γ. We say that h is Γ-stable if $|h| = 0$ or, otherwise, assuming that $\Gamma = (i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$ and that $h = a_{i_1}h'$,*

- *h' is Γ_1-stable and*
- *for all $1 < \alpha \leq \ell$, Γ_α is $(i, u_i(h))$-blocking.*

Theorem 1. *Let Γ be an extensive game with perfect information. A terminal history h is the outcome of some Nash equilibrium iff h is Γ-stable.*

Proof. We proceed by induction on the length of Γ. When $\mathsf{length}(\Gamma) = 0$ the unique history λ is both the outcome of the empty strategy, which is a Nash equilibrium, and then Γ-stable by definition. Assume that $\Gamma = (i, a_{i_1}\Gamma_{i_1}, \ldots, a_{i_\ell}\Gamma_{i_\ell})$ has length k and that the theorem holds for any game with length less than k. Without loose of generality we assume that $h = a_{i_1}h'$.

Let $s = (s_{-i}, s_i)$ be a Nash equilibrium in Γ such that $h = O(s)$. Consider any α, $1 < \alpha \leq \ell$, and the adversary team $-i$ strategy profile s^α_{-i}, that is the concretion of s_{-i} to Γ_α. Take any strategy σ^α_i for player i in Γ_α and consider the joint strategy $\sigma^\alpha = (s^\alpha_{-i}, \sigma^\alpha_i)$. Notice that $u_i(a_{i_\alpha}O(\sigma^\alpha)) \leq u_i(h)$. Otherwise player i could change s_i and get a better payoff, therefore s will not be a Nash equilibrium. From the subgame point of view, by Lemma 2 we have that Γ_α is $(i, u_i(h))$-blocking. Furthermore, player i has no incentive to change their initial selection of a_{i_1}.

Consider the strategy s^1 in Γ_1, note that s^1 is a Nash equilibrium in Γ_1, otherwise some player will be able to deviate unilaterally and get a better payoff, but given the initial selection of i, their payoff will also be improved in Γ and thus s will not be a Nash equilibrium. Thus h fulfills the recursive definition and thus is Γ-stable.

Assume that h is Γ-stable. Therefore, there is a Nash equilibrium s^1 of Γ^1 for which $h' = O(s^1)$. Furthermore, for any α, $1 < \alpha \leq \ell$, Γ_α is $(i, u_i(h))$-blocking, according to Lemma 2 the team $-i$ has a blocking strategy s^α_{-i} for Γ_α. Let s^α_i be any strategy for player i in Γ_α and set $s^\alpha = (s^\alpha_{-i}, s^\alpha_i)$. Consider the strategy profile $\sigma = (\tau_{a_{i_1}}, s^1, \ldots, s^\ell)$.

It remains to show that σ is a Nash equilibrium. Consider a player $j \neq i$, as s^1 is a Nash equilibrium in Γ_1 and player i is selecting this game through action a_{i_1}, player j cannot change their strategy in Γ_1 and get a better utility. From the point of view of player i, for any $\alpha \neq 1$, Γ_α is $(i, u_i(h))$ blocking so the utility that player i can get from α is at most $u_i(h)$ regardless of player i strategy. Furthermore, since s^1 is a Nash equilibrium in Γ_1, then player i cannot change their strategy in Γ_1 and get a better payoff. Therefore σ is a Nash equilibrium. □

Our PSPACE membership results for PNOut and PNGrant problems when the input is given in general or in implicit form come from the fact that to verify that a history h is the outcome of a Nash equilibria can be done recursively in polynomial space. In order to do this we use Γ-stability recursive characterization which requires a number of successive recursive calls bounded by $\mathsf{length}(\Gamma)$ and for each call we only need to store a suffix of h. Moreover it is also needed to verify the (i, u)-blocking of each subgame Γ_α of Γ. These can also be done in polynomial space since the number of successive recursive calls is bounded by $\mathsf{length}(\Gamma)$ and for each recursive call we need to store an action. Note that to verify that for each $a \in \mathsf{next}(\lambda)$, $a^{-1}\Gamma$ is (i, u)-blocking, the space of each successive call can be reused.

Our PSPACE hardness results follows from reductions from the PSPACE-complete QBF problem: Given a boolean formula F over n variables decide whether the quantified boolean formula $\Phi \equiv \exists\, x_1\ \forall\, x_2 \ldots Qx_n\ F(x_1, \ldots, x_n)$ is satisfiable [8]. Given a QBF formula Φ we define two extensive games $\Gamma(\Phi)$ and $\Gamma'(\Phi)$.

$\Gamma(\Phi)$ has two players and $A_1 = A_2 = \{0, 1\}$. Player 1 controls all the variables with an existential quantifier and player 2 controls the other set of variables, they alternate assigning a truth value to the variables in order. The set of histories is $H = \{h \in \{0,1\}^* \mid |h| \leq n\}$ and $Z = \{0,1\}^n$. The utility function is given by $u_1(h) = F(h)$ and $u_2(h) = 1 - u_1(h)$.

$\Gamma'(\Phi)$ has two players and $A_1 = \{0,1\}$ and $A_2 = \{A, B, 0, 1\}$. Let $\Gamma_1 = \Gamma_\lambda(1,0)$ and set $\Gamma'(\Phi) = (2, A\Gamma_1, B\Gamma(\Phi))$.

Theorem 2. *Let Φ be a quantified boolean formula on n variables and let $\Gamma = \Gamma(\Phi)$ and $\Gamma' = \Gamma'(\Phi)$. The following statements are equivalent*

- *Φ is satisfiable.*
- *Γ has a subgame perfect Nash equilibrium in which the first player gets utility 1.*
- *Γ has a Nash equilibrium in which the first player gets utility 1.*
- *A is the outcome of some subgame perfect Nash equilibrium of Γ'.*
- *A is the outcome of some Nash equilibrium of Γ'.*

The next theorem, whose proof is is given in Appendix C, summarizes our complexity results.

Theorem 3. *When the game is given in explicit form, the* IsPN*, the* PNOut *and the* PNGrant *problems belongs to* P*. When the game is given in general or implicit form the* IsPN *problem is* coNP*-complete while the* PNOut *and the* PNGrant *problems become* PSPACE*-complete.*

5 A characterization of the outcomes of subgame perfect Nash equilibria

An alternative characterization of subgame perfect Nash is given in [12]. Consider the *one-deviation property:* no player i can increase their payoff by changing their action at the start of any subgame in which player i is the first mover, given the other player's strategies and the rest of player i strategy.

Lemma 3 (Proposition 438.1 of [12]). *A strategy profile in a finite extensive game with perfect information is a subgame perfect Nash equilibrium iff it satisfies the one-deviation property.*

For our complexity results we need a further refinement of the above property to transfer the existence of a strategy profile to the existence of a terminal history that is the outcome of some subgame perfect Nash equilibrium. Consider the following property defined recursively.

Definition 8 (Γ-fixable). *Let Γ be an extensive game and h be a terminal history in Γ. We say that h is Γ-fixable if $|h| = 0$ or, otherwise, assuming that $\Gamma = \Gamma(i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$ and that $h = a_{i_1}h'$,*

- *h' is Γ_1-fixable and*
- *for all $1 < \alpha \leq \ell$, there is $h^\alpha \in a_{i_\alpha}^{-1}Z$ such that $u_i(a_{i_\alpha}h^\alpha) \leq u_i(h)$ and h^α is Γ_α-fixable.*

Now we can state the characterization of the outcomes of a subgame perfect Nash equilibrium.

Theorem 4. *Let Γ be an extensive game with perfect information. A terminal history h is the outcome of some subgame perfect Nash equilibrium of Γ iff h is Γ-fixable.*

Proof. We proceed by induction on the length of the game. Assume that h is the outcome of some subgame perfect Nash equilibrium of Γ. When $\mathsf{length}(\Gamma) = 0$ the empty strategy profile is the unique subgame perfect Nash equilibrium, it has outcome λ, and, by Definition 8, it is Γ-fixable. Assume that $\Gamma = (i, a_{i_1}\Gamma_1, \dots, a_{i_\ell}\Gamma_\ell)$ has length $k > 1$ and that the theorem holds for games with length at most k. Without loose of generality, we assume $h = a_{i_1}h'$.

Let s be a subgame perfect Nash equilibrium of Γ for which $h = O(s)$. From the definition of subgame perfect Nash equilibrium, we know that, for any $1 \leq \alpha \leq l$, $a_{i_\alpha}^{-1}s$ is a subgame perfect Nash equilibrium of Γ_α, so taking $h^\alpha = O(a_{i_\alpha}^{-1}s)$, by induction hypothesis, we have that h^α is Γ_α-fixable. By Lemma 3, we know that the utility for player i is maximized by $u_i(a_{i_1}h')$ so, for any $1 \leq \alpha \leq \ell$, we have $u_i(a_{i_\alpha}h^\alpha) \leq u_i(a_{i_1}h')$. Therefore, we have that h is Γ-fixable.

For the reverse implication, as h is Γ-fixable, we know that h' is Γ_1-fixable and that, for any $1 < \alpha \leq \ell$, there is $h^\alpha \in a_{i_\alpha}^{-1}Z$ Γ_α-fixable for which $u_i(a_{i_\alpha}h^\alpha) \leq u_i(h)$. By induction hypothesis, for any $1 \leq \alpha \leq \ell$, there is a subgame perfect Nash equilibrium s_i^α of Γ_α for which $O(s_i^\alpha) = h^\alpha$. The additional condition $u_i(a_{i_\alpha}h^\alpha) \leq u_{i_1}(h')$ guarantees that the strategy profile $(\tau_{a_{i_1}}, s_i^1, \dots, s_i^\ell)$ for Γ satisfies the one-deviation property and therefore it is a subgame perfect Nash equilibrium whose outcome, by construction is h. □

The next theorem summarizes our complexity results. Their proof is given in Appendix D and it uses arguments similar to those of the proof of Theorem 3.

Theorem 5. *When the game is given in explicit form, the* IsSPN*, the* SPNOut *and the* SPNGrant *problems belongs to* P*. When the game is given in general or implicit form the* IsSPN *problem is* coNP*-complete while the* SPNOut *and the* SPNGrant *problems become* PSPACE*-complete.*

References

1. C. Àlvarez, J. Gabarro, and M. Serna. Pure nash equilibrium in strategic games with a large number of actions. In *Mathematical Foundations of Computer Science, MFCS 2005*, LNCS, 2005. To appear.
2. V. Conitzer and T. Sandholm. Complexity results about Nash equilibra. In *IJCAI 2003*, pages 765–771, 2003.
3. K. Daskalakis and C. Papadimitriou. The complexity of games on highly regular graphs. In *ESA 05*, 2005.
4. A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *STOC 2004*, pages 604–612, 2004.
5. D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. In *ICALP 2002*, pages 123–134, 2002.
6. D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. In *ICALP 2004*, pages 593–605, 2004.
7. M. Gairing, T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. Nash equilibria in discrete routing games with convex latency functions. In *ICALP 2004*, pages 645–657, 2004.
8. Garey and Johnson. *Computers and intractability. A guide to the NP-completeness.* Freeman, 1979.
9. G. Gottlob, G. Greco, and F. Scarcello. Pure Nash equilibria: Hard and easy games. In *Theoretical Aspects of Rationality and Knowledge.*, pages 215–230, 2003.
10. D. Koller, N. Megiddo, and B. Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:247–259, 1996.
11. H.W. Kuhn. Extensive games and the problem of information. In H.W. Kuhn and A.W. Tucker, editors, *Contribution to the theory of games, Volume II*, number 28 in Annals of Mathematics Studies, pages 193–216. Princeton University press, 1953.
12. M.J. Osborne. *A Introductions to Game Theory.* Oxford University Press, 2004.
13. M.J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.
14. C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

15. C. Papadimitriou. Algorithms, games and the internet. In *STOC 2001*, pages 4–8, 2001.
16. G.R. Schoenebeck and S. Vadham. The complexity of Nash equilibria in concisely represented games. Technical Report 52, Electronic Colloquium on Computational Complexity, 2005.
17. B. Stengel. Computational complexity of correlated equilibria for extensive games. Technical Report LSE-CDAM-2001-03, CDAM, 2001.

A Proof of Lemma 2

Proof. Our proof is by induction on the length of the game. For games with length 0 the lemma trivially follows. In this case we have an empty game, when $u_i(\lambda) \leq u$, the empty strategy (with no choices at all) fulfills both conditions. Assume now that the lemma holds for games with length k. Let Γ be a game with length $k+1$.

Let us prove (1) $\implies$ (2). There are two cases depending on whether the initial player is i or not.

Case 1: $P(\lambda) = j$ with $j \neq i$. We have the factorization $\Gamma = (j, a_{j_1}\Gamma_1, \ldots, a_{j_\ell}\Gamma_\ell)$. As Γ is (i,u)-blocking, we know that at least one of the games, lets say Γ_1, is (i,u)-blocking. By the induction hypothesis we know that, in Γ_1, there is a strategy profile s^1_{-i} such that for any strategy s^1_i it holds $u^1_i(s^1_{-i}, s^1_i) \leq u$. Taking $s_{-i} = (\tau_{a_{j_1}}, a_{j_1}s^1_{-i}, \ldots, a_{j_\ell}s^\ell_{-i})$ where for any $\alpha \neq j$, s^α_{-i} is any strategy for the team $-i$ in Γ_α, for any s_i, it holds $u_i(s_{-i}, s_i) = u^1_i(s^1_{-i}, a^{-1}_{j_1}s_i) \leq u$. In words, the adversary j chooses a_{j_1} forcing the players to enter into the (i,u)-blocking game Γ_1. Inside this game adversaries $-i$ will play according to the existing strategy profile s^1_{-i}, playing against this strategy, player i will not be allowed to win more than u.

Case 2: $P(\lambda) = i$. We have the factorization $\Gamma = (i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$. Moreover Γ is (i,u)-blocking, so we know that all the subgames Γ_α, $1 \leq \alpha \leq \ell$, are (i,u)-blocking, therefore for any $1 \leq \alpha \leq \ell$ there is a strategy profile for the team $-i$, s^j_{-i}, such that, for any possible choice s^α_i allowed to player i, it holds $u^\alpha_i(s^\alpha_{-i}, s^\alpha_i) \leq u$.

Define $s_{-i} = (a_{i_1}s^1_{-i}, \cdots, a_{i_\ell}s^\ell_{-i})$. Let s_i be any strategy for player i in Γ. Let β be the subgame for which $s_i(\lambda) = a_{i_\beta}$ and let $s = (s_{-i}, s_i)$. Note that $u_i(s) = u^\beta_i(s^\beta_{-i}, a^{-1}_{i_\beta}s_i)$. Therefore, $u_i(s) \leq u$ and the claim follows. In words any selection of an action a_{i_β} of player i will lead to an (i,u)-blocking game Γ_β, there adversaries will play the accorded s^α_{-i} strategy, and player i will not be allowed to gain more than u.

We prove now (2) $\implies$ (1). As before we have two cases.

Case 1: $P(\lambda) = j$ with $j \neq i$. We have the factorization $\Gamma = (j, a_{j_1}\Gamma_1, \ldots, a_{j_\ell}\Gamma_\ell)$. Let s_{-i} be the strategy profile for team $-i$ verifying (2). Without loose of generality we assume that $s_{-i}(\lambda) = a_{j_1}$. Let $s^1_{-i} = a^{-1}_{j_1}s_{-i}$, let s^1_i be any strategy for player i in Γ_1, and let s_i be any strategy for player i in Γ for which $s^1_i = a^{-1}_{j_1}s_i$. We have $u_i(s_{-i}, s_i) \leq u$ and it holds $u_i(s_{-i}, s_i) = u_i(s^1_{-i}, s^1_i)$, therefore subgame Γ_1 verifies also (2) and by induction hypothesis it is (i,u)-blocking, therefore Γ is also (i,u)-blocking.

Case 2: $P(\lambda) = i$. We have the factorization $\Gamma = (i, a_{i_1}\Gamma_1, \ldots, a_{i_\ell}\Gamma_\ell)$. Let s_{-i} be the strategy profile for team $-i$ verifying (2). For any α, $1 \leq \alpha \leq \ell$, let $s^\alpha_{-i} = a^{-1}_{i_\alpha}s_{-i}$. Fix β, $1 \leq \beta \leq \ell$, let s^β_i be any strategy for player i in Γ_β, and let s_i be any strategy for player i in Γ for which $s^\beta_i = a^{-1}_{i_\beta}s_i$ and $s_i(\lambda) = a_{i_\beta}$. We know, by (2), that $u_i(s_{-i}, s_i) \leq u$, but, by construction $u_i(s_{-i}, s_i) = u_i(s^\beta_{-i}, s^\beta_i)$, therefore subgame Γ_β verifies also (2). Therefore by induction hypothesis, Γ_β is (i,u)-blocking. As β was arbitrary we have that all Γ_β are (i,u)-blocking and therefore Γ is (i,u)-blocking. □

B Proof of Theorem 2

We split the proof of Theorem 2 in two pieces, we prove first the part of the theorem related to the game $\Gamma(\Phi)$ and after the second part of the theorem related to game $\Gamma'(\Phi)$. In the following we note $\overline{0} = 1$ and $\overline{1} = 0$.

Lemma 4. *If Φ is a satisfiable quantified boolean formula on n variables, then $\Gamma(\Phi)$ has a subgame perfect Nash s for which $u_1(s) = 1$.*

Proof. The proof is by induction on the number n of variables in Φ.

When $n = 0$ all formulas are satisfiable. In such a case the associated game is an empty game and its unique strategy is a subgame perfect Nash in which player 1 gets payoff 1.

When $n = 1$, consider the strategy profile s in $\Gamma(\Phi)$ defined s as τ_0 when $F(0) = 1$, otherwise τ_1. As either $F(0)$ or $F(1)$ are true, the value received by player 1 is maximum among all possible histories, therefore s is a subgame perfect Nash equilibrium by Lemma 3.

Assume that the conditions in the lemma are true for any QBF on $2 \leq k < n$ variables. Given two boolean values a and b, $\Phi(a, b)$ represents the formula in which the two first quantifiers have been removed and the formula has been simplified by setting $x_1 = a$ and $x_2 = b$.

Let Φ be a satisfiable QBF formula on n variables and $\Gamma = \Gamma(\Phi)$. As Φ is satisfiable there is $a \in \{0, 1\}$ such that, for any $b \in \{0, 1\}$, $\Phi(a, b)$ is satisfiable. By induction hypothesis $\Gamma(\Phi(a, b))$ has a subgame perfect Nash equilibrium s_b such that $u_1(s_b) = 1$. Let s' be any subgame perfect Nash equilibrium in the game $\overline{a}^{-1}\Gamma$. Define s in such a way that s coincides with s' in $\overline{a}^{-1}\Gamma$, for any $b \in \{0, 1\}$, with s_b in $(ab)^{-1}\Gamma$ and with $s(\lambda) = a$. As s verifies the one deviation property we have that s is a subgame perfect Nash equilibrium of Γ furthermore $u_1(s) = 1$. □

Lemma 5. *If $\Gamma(\Phi)$ has a Nash equilibrium s for which $u_1(s) = 1$ then Φ is satisfiable.*

Proof. Let s be a Nash equilibrium, for which $u_1(s) = 1$, and let $h = a_1 \dots a_n = O(s)$. Consider any prefix za of h for which $P(z) = 2$. Notice that $u_1((z\overline{a})^{-1}s) = 1$ otherwise player 2 will have an incentive to change their strategy and s will not be a Nash equilibrium. Furthermore, $(z\overline{a})^{-1}s$ is a Nash equilibrium in $(z\overline{a})^{-1}\Gamma$, notice that the utility that gets player 1 is maximum and therefore player 1 cannot increase its payoff. Player 2 is bound to get payoff 0, otherwise s will not be a Nash equilibrium.

Now we proceed by induction. It is easy to see that the statement of the lemma is true for games of length 0 or 1.

Assume that Γ has length n and let $h = a_1 \dots a_n$ be the outcome of a Nash equilibrium s with $u_1(s) = 1$ in $\Gamma(\Phi)$. By the previous observation we know that s induces a Nash equilibrium in both $(a_1 a_2)^{-1}\Gamma$ and $(a_1 \overline{a_2})^{-1}\Gamma$. By induction hypothesis this means that both $\Phi(a_1, a_2)$ and $\Phi(a_1, \overline{a_2})$ are satisfiable. Therefore Φ is also satisfiable. □

Theorem 6. *Let Φ be a quantified boolean formula on n variables and let $\Gamma = \Gamma(\Phi)$. The following statements are equivalent*

- *Φ is satisfiable.*
- *Γ has a subgame perfect Nash equilibrium in which the first player gets utility 1.*
- *Γ has a Nash equilibrium in which the first player gets utility 1.*

¿From Theorem 6 and the definition of Nash equilibrium we have the following result.

Lemma 6. *If Φ is satisfiable the A is the outcome of a subgame perfect Nash equilibrium in $\Gamma'(\phi)$.*

Lemma 7. *If A is the outcome of a Nash equilibrium in $\Gamma'(\phi)$ then Φ is satisfiable.*

Proof. Let s' be a Nash equilibrium of $\Gamma' = \Gamma'(\Phi)$ such that $O(s') = A$. Let $s = B^{-1}s'$ the strategy played in $\Gamma = B^{-1}\Gamma'$, the Nash condition implies that $u_1(s) = 1$ in Γ.

Observe that s gives to player 1 payoff 1 and hence player one will not deviate unilaterally from its strategy. But for player two, it is not possible to change its strategy and get a better payoff, otherwise s' will not be a Nash equilibrium in Γ'. Therefore, s is a Nash equilibrium in Γ and the utility for player one is 1, according to Theorem6, Φ is satisfiable. □

Theorem 7. *Let Φ be a quantified boolean formula on n variables and let $\Gamma' = \Gamma'(\Phi)$. The following statements are equivalent*

- *Φ is satisfiable.*
- *A is the outcome of some subgame perfect Nash equilibrium of Γ'.*
- *A is the outcome of some Nash equilibrium of Γ'.*

C Proof of Theorem 3

The following theorem give us precise limits about the complexity of testing for a Nash equilibrium.

Theorem 8. *The* IsPN *problem is* coNP*-complete when the game is given in implicit or general form and the strategy is also represented implicitly.*

Proof. We prove that when the game is given in implicit form the problem belongs to coNP and that the problem is coNP-hard when the game is given in general form.

Given the input $\langle 1^n, 1^m, 1^h, P, M, 1^t, S\rangle$, we can test in nondeterministic polynomial time whether S is not a pure Nash. Suppose that $s = (s_1, \ldots, s_n)$ is the strategy profile associated to S. Given a player i, the strategy profile factorizes as $s = (s_{-i}, s_i)$. The strategy s is not a pure Nash, if it exists a player i having strategy s'_i such that $u_i(s_{-i}, s_i) < u_i(s'_{-i}, s_i)$. Consider the following nondeterministic algorithm

$i =$ choose non deterministically$\{1, \ldots, n\}$
$h = \lambda$
while $h \notin Z$ **do**
  if $i = P(h)$
    then $a =$ choose(A_i)
    else $a = S(h)$
  end if
  $h = ha$
end while
test $u_i(s) < u_i(h)$

In order to prove hardness we consider a reduction from coNP-complete VALIDITY problem: Given a boolean formula F decide whether F is satisfiable by all truth assignments[8]. Given a formula F with n variables we define an extensive game $\Gamma(F)$ and a strategy S such that F is valid iff S is a pure Nash. Γ has just one player that plays n times. The set of histories is $H = \{V\} \cup \{h \in \{0,1\}^n \mid |h| \leq n\}$ where V is a special symbol meaning "valid" and $Z = \{V\} \cup \{0,1\}^n$. Initially the player has to choose between $V, 0, 1$ meaning "valid', "assign 0 to the first variable" or "assign 1 to the first variable". In any other move the player has to choose the corresponding truth value (it is not possible to choose V anymore). The utility function is given by.

$$u_1(h) = \begin{cases} 0 & \text{if } h = V \text{ or } F(h) = 1 \\ 1 & \text{if } F(h) = 0 \end{cases}$$

Consider the strategy

$$s_1(h) = \begin{cases} V & \text{if } h = \lambda \\ 0 & \text{if } h \in H \setminus Z \text{ and } h \neq \lambda \end{cases}$$

Note that $O(s_1) = V$ and $u_1(s_1) = 0$. If F is valid, 0 is the best value, therefore s_1 is a Nash equilibrium. If F is invalid there exists a succinct disqualification h' such that $F(h') = 0$. Suppose $h' = b_1 b_2 \dots b_n$, this case the player has incentive to change into the strategy

$$s_1'(h) = \begin{cases} b_1 & \text{if } h = \lambda \\ b_{i+1} & \text{if } h = b_1 b_2 \dots b_i \text{ and } i < n \\ 0 & \ h \text{ is not a prefix of } h' \text{ and } h \neq Z \end{cases}$$

because $O(s_1') = h'$ and thus $u_1(s_1') = 1$.

Finally, note that it is easy to construct a TM implementing S working in linear time, and that a general or implicit form description of the game $\Gamma(F)$ can be obtained in polynomial time. □

Theorem 9. *The* IsPN *problem for games given in explicit form is in* P.

Proof. In this case we have as part of the input the tree of the game and the strategy. Notice that for each player i we have to perform the following procedure.

- Set $u = u_i(S)$.
- If for all leaves, the value associated to i is $\leq u$, return true. Otherwise,
- For each node labeled $j \neq i$ with associated history h remove from T all the edges with label different from $S(h)$.
- Check whether in the remaining graph the root of T is connected to a leaf in which the value associated to i is greater than u.
- If there is such a leaf output no.

Clearly the above procedure needs only polynomial time. When the N test succeed the given strategy is a Nash equilibrium. □

Before analyzing the complexity of the other two problems we provide some additional results.

Lemma 8. *Given an extensive game Γ in explicit form a player i and a value u, there is a polynomial time algorithm that checks whether Γ is (i, u)-blocking.*

Proof. Note that in this case we have as input the tree describing the game. In order to check the property consider the following algorithm with input the tree T of the game, a player i and a rational u,

- Construct a tree T', with the same vertices and edges as T, but without labels.
- All the leaves holding a value x_i for player i such that $x_i \leq u$ get label 1, the remaining leaves get label 0.

– All the internal nodes of T' that have label i in T get label $\wedge$.
– All the internal nodes of T' that have label $j \neq i$ in T get label $\vee$.
– Return the value at the root after evaluating the and/or tree T'.

According to the recursive definition of (i, u)-blocking given in Page 7, the above algorithm decides whether T is (i, u)-blocking. □

¿From the recursive definition of (i, u)-blocking we get the following result.

Lemma 9. *Given an extensive game Γ in general or implicit form a player i and a value u, there is a polynomial space algorithm that checks whether Γ is (i, u)-blocking.*

Theorem 10. *The* IsPNOut *problem for extensive games given in general or implicit form is* PSPACE*-complete, even when the number of players is* 2.

Proof. For the positive part. Assume that $\Gamma = \langle 1^n, 1^m, 1^h, P, M, 1^t \rangle$ with $| \Sigma | > 1$ and $m > 1$ and that a terminal history z is also given. We test if the property given in Theorem 1 holds. Notice that the property given in Theorem 1 can be reduced to a sequence of (i, u)-blocking tests for different trees. To identify the actual tree we have to keep information on the pieces of path leading to it. In any case this additional space never exceeds a constant factor of $\mathsf{length}(\Gamma)$. Therefore, it can be performed using only polynomial space.

For the hardness we consider the reduction from QBF that associates to a formula Φ the game $\Gamma'(\Phi)$. As it is shown in Theorem 7 Φ is satisfiable iff history A is the outcome of a Nash equilibrium in $\Gamma'(\Phi)$. It is easy to show that a general or implicit description of the game can be obtained in polynomial time. Taking into account that QBF is PSPACE-complete, we have the desired hardness result. □

Theorem 11. *The* IsPNOut *problem for extensive games given in explicit form is in* P.

Proof. According to Theorem 1 we have to perform a polynomial number of (i, u)-blocking test. Thus from Lemma 8 the final running time will be polynomial. □

Taking into account that in the game constructed in the reduction from QBF has only two players we get the next result.

Theorem 12. *The* PNGrant *problem for extensive games in implicit or general form is* PSPACE*-complete, even when the number of players is a fixed constant $k \geq 2$.*

Proof. The hardness part follows from a Theorem 6 and the fact that a representation of $\Gamma(\Phi)$ can be obtained in polynomial time.

Assume that $\Gamma = \langle 1^n, 1^m, 1^h, P, M, 1^t \rangle$ with $| \Sigma | > 1$ and $m > 1$ and that a positive number u is also given. To decide whether Γ has a Nash equilibrium s such that $u = u_i(s)$ we generate in PSPACE all the terminal histories of Γ. For a particular $z \in Z$ we check whether $u_{P(\lambda)}(z) = u$ and if so we test if the property given in Theorem 1 holds. Notice that the property given in Theorem 1 can be expressed recursively by a formula with a polynomial (in h) number of alternations. Therefore it can be checked using only polynomial space. □

Finally, we can show that for extensive games given in explicit form the PNGrant problem can be solved in polynomial time.

Theorem 13. *The* PNGrant *problem for extensive games in explicit form is in* P.

Proof. Note that in this case we have as input the tree describing the game. In order to check the property given in Definition 6, according to Theorem 1 we have to perform for each terminal history h with $u_1(h) = u$ a collection of $|h|$ tests, one from each of the associated subgames. As both the number of terminal histories and the length of them are bounded by a polynomial in the game description, the complexity follows. □

D Proof of Theorem 5

Theorem 14. *The* IsSPN *problem for extensive games given in general or implicit form is* coNP*-complete when the strategy is also represented implicitly.*

Proof. Accordingly to Lemma 3 to show that s is not a subgame perfect Nash equilibrium it is enough to conjecture a non terminal history h and an action a_i, where $i = P(h)$ and $a_i \neq s_i(h)$, for which $u_i((ha_i)^{-1}s) \geq u_i(h^{-1}s)$. It is straightforward to show that computing the outcome of an strategy profile given implicitly can be obtained in polynomial time. Therefore, we have that for games given in implicit or general form, the problem is in coNP.

For the hardness part, consider the reduction given in Theorem 9. It is straightforward to see that when the formula is valid, then the proposed strategy profile verifies the one-deviation property and thus it is a subgame perfect Nash equilibria. When the formula is not valid the proposed strategy profile is not a Nash equilibrium and therefore it is not a subgame perfect Nash equilibrium. □

Theorem 15. *The* IsSPN *problem for extensive games given in explicit form belongs to* P*.*

Proof. It is enough to check that the given strategy satisfies the one-deviation property. This can be done in polynomial time in a two traversal steps. The first one goes bottom up, computing the value obtained by the players at every internal node. The second one goes top down checking that at every node the selected branch verifies the one deviation property.

Theorem 16. *The* IsSPNOut *problem for extensive games given in general or implicit form is* PSPACE*-complete.*

Proof. For the positive part it is enough to check whether the given terminal history h is Γ-fixable. According to Definition 8 the property can be decomposed as a polynomial sequence of alternated quantifiers, ending in a poly time checkable property. Therefore, the problem belongs to PSPACE.

Consider the reduction from QBF that, given a formula Φ, computes the representation of $\Gamma'(\Phi)$. Such function can be computed in polynomial time. According with Theorem 7 Φ is satisfiable iff A is the outcome of a subgame perfect Nash equilibrium of $\Gamma'(\Phi)$. □

Theorem 17. *The* IsSPNOut *problem for extensive games given in explicit form belongs to* P*.*

Proof. We have to compute for every subgame Γ' a table with an entry for every terminal history z with in Γ', telling whether z is Γ' fixable.

We start by evaluating all the subgames of length 0. For this case the table can be computed easily in constant time. For subgames of length greater than 0, the table can be computed from the table of the game components in polynomial time, taking into account the definition of Γ-fixable. Once we have the table for the root of the game tree we check whether h is Γ-fixable. □

Theorem 18. *The* SPNGRANT *problem for extensive games given in general or implicit form is* PSPACE*-complete.*

Proof. The positive part follows from the positive part of 16 and the fact that we have to perform the above test for any terminal history h with $u_i(h) = u$. Since all the terminal histories can be enumerated one after the other in polynomial space, then it follows that the SPNG is in PSPACE for extensive games given in general or implicit form.

For the hardness part, consider the reduction from QBF that, given a formula Φ, computes the representation of $\Gamma(\Phi)$. Such function can be computed in polynomial time. According with Theorem 6 Φ is satisfiable iff $\Gamma(\Phi)$ subgame perfect Nash equilibrium in which the first player get payoff 1. □

Theorem 19. *The* SPNGRANT *problem for extensive games given in explicit form belongs to* P*.*

Proof. As in the proof of Theorem 17 we compute a table holding the information about all the terminal histories that are Γ-fixable. After that we check whether some terminal history z with $u_{P(\lambda)}(h) = u$ is Γ-fixable. This final test can be performed in polynomial time.