


Partial Order Games

Valeria Zahoransky¹, Julian Gutierrez², Paul Harrenstein³ and Michael Wooldridge^{3,*} 

¹ Fachbereich Informatik, Université du Luxembourg, Maison du Nombre, 6 Avenue de la Fonte, L-4364 Esch-sur-Alzette, Luxembourg; valeria.zahoransky@uni.lu

² Department of Data Science & AI, Monash Data Futures Institute, Green Chemical Futures Building, 13 Rainforest Walk, Monash University, Clayton, VIC 3800, Australia; Julian.Gutierrez@monash.edu

³ Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK; paul.harrenstein@cs.ox.ac.uk

* Correspondence: mjw@cs.ox.ac.uk

Abstract: We introduce a non-cooperative game model in which players' decision nodes are partially ordered by a dependence relation, which directly captures informational dependencies in the game. In saying that a decision node v is dependent on decision nodes v_1, \dots, v_k , we mean that the information available to a strategy making a choice at v is precisely the choices that were made at v_1, \dots, v_k . Although *partial order games* are no more expressive than extensive form games of imperfect information (we show that any partial order game can be reduced to a strategically equivalent extensive form game of imperfect information, though possibly at the cost of an exponential blowup in the size of the game), they provide a more natural and compact representation for many strategic settings of interest. After introducing the game model, we investigate the relationship to extensive form games of imperfect information, the problem of computing Nash equilibria, and conditions that enable backwards induction in this new model.

Keywords: game theory; non-cooperative games; Nash equilibrium; backwards induction; computational complexity



Citation: Zahoransky, V.; Gutierrez, J.; Harrenstein, P.; Wooldridge, M. Partial Order Games. *Games* **2022**, *13*, 2. <https://doi.org/10.3390/g13010002>

Academic Editor: Ulrich Berger

Received: 19 November 2021

Accepted: 15 December 2021

Published: 21 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The two most important game models in non-cooperative game theory are normal form games and extensive form games. These games are distinguished by the information that players have about the strategies of other players. In normal form games, players must select and commit to strategies without any information relating to the strategies of others. In contrast, in extensive form games, players make moves alternately over time and, during play, their strategy may be informed by moves made previously. Variations of extensive form games—e.g., of imperfect information or recall—make it possible to capture the information available to a player when called upon to make a move.

In this article, we introduce *partial order games*. The key distinguishing feature of partial order games is that they are equipped with a *dependence relation*, which explicitly captures the informational dependencies between decision nodes. In saying that a decision node v is dependent on decision nodes v_1, \dots, v_k , we mean that the information available to a strategy making a choice at v is precisely the choices that were made at v_1, \dots, v_k . Observe that the informational dependencies captured by a dependence relation give rise to a *partial temporal order* over choices: in this case, the choices for v_1, \dots, v_k must strictly precede the choice for v . However, the temporal ordering induced in this way is indeed partial: it may be that two choices v' and v'' are independent, in which case one can say nothing about their temporal order.

In a technical sense, partial order games are no more expressive than extensive form games of imperfect information: one of our key results is to show that every partial order game can be transformed into a strategically equivalent extensive form game of imperfect information. However, for some settings, partial order games have significant advantages over their extensive form representation:

- First, informational dependencies in partial order games are *explicitly* captured, while in extensive form games, they are left *implicit* in the information sets of the game. As a consequence, some settings are much more transparently and naturally represented using partial order games, compared to the extensive form.
- Second, partial order games can be *exponentially more compact* than their extensive form. From a purely practical point of view, this means that some scenarios with a compact partial order representation cannot be handled (by people or by a computer) in their extensive form.

We analyse partial order games by means of *Nash equilibrium*—arguably game theory’s most prominent non-cooperative solution concept—and the solutions given by a natural *backwards induction procedure* for partial order games defined on the dependence relation. Here, we give special attention to the *computational complexity* of calculating such solutions in partial order games. The remainder of this article is structured as follows.

- We begin by briefly recalling some necessary concepts from graph theory and game theory, and then introduce the formal framework of partial order games.
- As we are interested in the computational properties of partial order games, we introduce a compact representation for strategies and utility functions in partial order games, based on Boolean circuits, which enables us to investigate questions about their computational complexity.
- In Section 4, we study the relationship of our game model to other game models: partial order Boolean games [1], Multi-Agent Influence Diagrams (MAIDs) [2,3], and extensive form games of imperfect information. Our main result in this section is to present a technique for translating partial order games into strategically equivalent extensive form games, although this translation comes at the cost of an exponential blowup in the size of the game. This leads us into a discussion of the use of partial order games as a compact representation of extensive form games.
- In Section 5, we investigate the problem of computing Nash equilibria in partial order games. For example, we show that checking whether a game has any pure strategy Nash equilibria is NEXPTIME-complete.
- In Section 6, we study backwards induction in partial order games. As partial order games are inherently games of imperfect information, it follows that backwards induction does not always work for partial order games. We thus investigate cases where forms of backward induction work for partial order games, and present a condition on games that we call *fit for backwards induction*, which is sufficient to allow backwards induction.
- We conclude with a brief discussion and pointers for future work.

2. Preliminary Definitions

We begin by recalling some concepts from graph theory and game theory, which we use in the remainder of the paper.

2.1. Directed Acyclic Graphs and Trees

The games in this paper are defined on *directed acyclic graphs* (DAGs). Formally, a directed acyclic graph is a pair (V, E) , where $V = \{v_1, \dots, v_{|V|}\}$ is a finite set of **vertices** or **nodes** and $E \subseteq V \times V$ a set of **directed edges** (or **arcs**) on V . We assume E to be acyclic (and thus also irreflexive). We say that vertex u is a **parent** of vertex v , and v a **child** of u , if $(u, v) \in E$. We occasionally also use infix notation and write uEv for $(u, v) \in E$. The **depth** of a vertex v , denoted by $depth(v)$, is defined recursively such that if v has no parents, $depth(v) = 0$, and otherwise, $depth(v)$ equals the maximum depth of v ’s parents plus 1. Formally,

$$depth(v) = \begin{cases} 0 & \text{if } v \text{ has no parents,} \\ \max_{(u,v) \in E} depth(u) + 1 & \text{otherwise.} \end{cases}$$

We say that vertex v is **reachable** from vertex u if $u = v$, or if there is some vertex w such that w is reachable from u and $(w, v) \in E$, that is if $u E^* v$, where E^* is the reflexive and transitive closure of E .

A **topological sorting** is a permutation $\tau = (u_1, \dots, u_{|V|})$ of the vertices $v_1, \dots, v_{|V|}$ in V such that $u_i E u_j$ implies $i < j$. Every DAG (V, E) admits at least one topological sorting of its vertices; this well-known fact will be important and useful in our study of partial order games.

A **tree** is a directed acyclic graph in which there is a unique vertex with no parents (the **root** of the tree, often denoted by v_0), and where every non-root vertex has a single parent. If a tree is finite, then, as it is acyclic, some vertices will have no children: we refer to these as the **leaves** of the tree. Observe that the root of the tree is the unique vertex of depth 0.

2.2. Normal-Form Games

We use normal-form games as the basis of our game-theoretic analysis of partial order games. Normal-form games are defined by a set of players, the strategies the players have at their disposal, and the preferences the players have over the outcomes that players choosing their strategies give rise to. Formally, a **normal-form game** is given by a tuple $(N, (S_i)_{i \in N}, (u_i)_{i \in N})$, where $N = \{1, \dots, |N|\}$ is the set of **players**, S_i is the set of (**pure**) **strategies** available to player i , and $u_i: S_1 \times \dots \times S_{|N|} \rightarrow \mathbb{R}$ is a **utility function** for each player i [4]. We refer to tuples $\mathbf{s} = (s_1, \dots, s_{|N|})$ in $S_1 \times \dots \times S_{|N|}$ as **strategy profiles**.

Nash equilibrium is the most important solution concept in non-cooperative game theory; it captures the idea of a joint course of action that is stable in the sense that no player has an incentive to deviate unilaterally from it. Furthermore, in this paper, it is one of the main analytic tools by means of which we evaluate partial order games. Formally, a strategy profile $\mathbf{s} = (s_1, \dots, s_{|N|})$ is a **Nash equilibrium** if, for all players i and strategies $s'_i \in S_i$, we have

$$u_i(\mathbf{s}) \geq u_i(\mathbf{s}_{-i}, s'_i),$$

where $(\mathbf{s}_{-i}, s'_i) = (s_1, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_{|N|})$.

We say that two normal-form games $(N, (S_i)_{i \in N}, (u_i)_{i \in N})$ and $(N', (S'_i)_{i \in N'}, (u'_i)_{i \in N'})$ are **strategically isomorphic** if there are bijections $\phi: N \rightarrow N'$ and $\psi_i: S_i \rightarrow S'_{\phi(i)}$ such that, for all players i in N and all strategy profiles $\mathbf{s} = (s_1, \dots, s_n)$ and $\mathbf{s}' = (s'_1, \dots, s'_n)$ we have

$$u_i(\mathbf{s}) \leq u_i(\mathbf{s}') \quad \text{if and only if} \quad u'_{\phi(i)}(\psi(\mathbf{s})) \leq u'_{\phi(i)}(\psi(\mathbf{s}')),$$

where $\psi(\mathbf{s}) = (\psi_1(s_1), \dots, \psi_n(s_n))$ and $\psi(\mathbf{s}') = (\psi_1(s'_1), \dots, \psi_n(s'_n))$.

Given a normal-form game, two strategies s_i, s'_i in S_i for a player i are said to be **equivalent**, in symbols $s_i \sim s'_i$, if $u_j(\mathbf{s}''_{-i}, s_i) = u_j(\mathbf{s}''_{-i}, s'_i)$ for all profiles $\mathbf{s}''_{-i} \in S_1 \times \dots \times S_{i-1} \times S_{i+1} \times \dots \times S_{|N|}$ and all players j . Let $[s_i]_{\sim}$ be the equivalence class of s_i under the equivalence relation \sim , and denote $[s]_{\sim} = ([s_1]_{\sim}, \dots, [s_{|N|}]_{\sim})$. Following the work in [5], we define the **reduced normal-form** of a normal-form game $(N, (S_i)_{i \in N}, (u_i)_{i \in N})$ as the game $(N, (S'_i)_{i \in N}, (u'_i)_{i \in N})$ where for each player i , $S'_i = \{[s_i]_{\sim} : s_i \in S_i\}$ and $u'_i([s]_{\sim}) = u_i(\mathbf{s})$ for all strategy profiles $\mathbf{s} = (s_1, \dots, s_{|N|})$. Two games are then said to be **strategically equivalent** if they induce strategically isomorphic reduced normal-forms.

2.3. Extensive-Form Games

An **extensive-form game (of imperfect information)** is based on a directed tree (V, E) and is played by a set of players N choosing from a set of actions A , starting from the root of the tree. Within the set V of vertices, we distinguish between **action or decision nodes**, which have children, and **leaf nodes**, which have no children. The set of action nodes we denote by D and the set of leaf nodes by L . With every action node v in D we associate a unique player i_v , who is then said to be *active at v* , and an action set $A_v \subseteq A$ such that every outgoing edge (v, u) is associated with a unique action in A_v . Let V_i denote the set of action nodes associated with player i , that is, $V_i = \{v \in V : i_v = i\}$, and let $A_i = \{a \in A_v : v \in V_i\}$.

Every vertex v is identified with a unique sequence of actions, called a **history**, which leads from the root v_0 to v . Histories associated with leaf nodes are referred to as **terminal histories**. We will denote the history associated with vertex v by \mathbf{h}_v . We can also think of vertices v as histories: vertex v is the sequence $\mathbf{h}_v = (a_1, \dots, a_{\text{depth}(v)})$ of actions for which there is a path $v_0 = w_0 \dots, w_{\text{depth}(v)} = v$ such that (w_i, w_{i+1}) is associated with action a_{i+1} for every $0 \leq i < \text{depth}(v)$. The root v_0 is thus the empty sequence $()$. Thus, we also find that vEw if and only if $w = va$ for some $a \in A_v$.

For every player i , we have a **partitioning** $\mathcal{S}_i = \{I_1, \dots, I_{|\mathcal{S}_i|}\}$ of their decision nodes V_i into (non-empty) **information sets**. We write $[v]$ for the information set vertex v belongs to. Here, it is understood that $A_u = A_v$, whenever u and v are in the same information set I . We then also write A_I for A_u and A_v . Information sets may be singletons. If all information sets for all players are singletons, we say the game is of **perfect information** and do not distinguish between information set $\{v\}$ and the node v itself, if no confusion is likely.

Finally, each player i 's preferences are represented by a utility value $u_i^{\mathbf{h}} \in \mathbb{R}$ at each terminal history/leaf \mathbf{h} . See Figure 1 for an example.

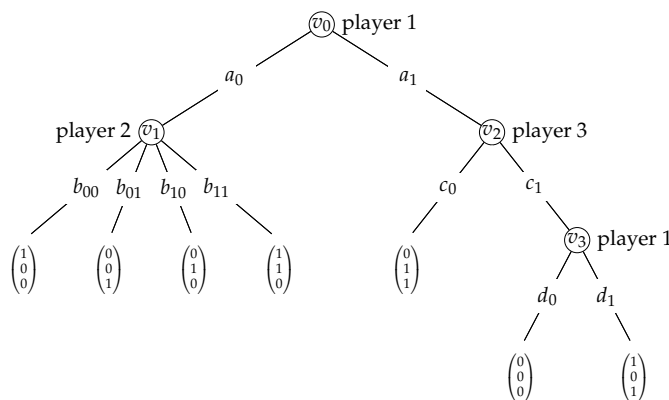


Figure 1. Extensive-form game of perfect information. The players are depicted besides the nodes at which they are active. Vertices are identified with histories, for instance, the root v_0 with $()$, vertex v_3 with a_1c_1 , and the leaf node/terminal history labelled $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ with a_0b_{00} , omitting parentheses and commas for better readability.

A **strategy** for player i in an extensive-form game is a function $s_i: \mathcal{S}_i \rightarrow A_i$ such that $s_i(I) \in A_I$ for all $I \in \mathcal{S}_i$. The set of strategies available to player i we then denote by S_i . A **strategy profile** is then a sequence $\mathbf{s} = (s_1, \dots, s_{|N|})$ of strategies, one for each player. Let $D = \{v_1, \dots, v_{|D|}\}$. Then, every strategy profile $\mathbf{s} = (s_1, \dots, s_{|N|})$ defines an **action profile** $\mathbf{a}^{\mathbf{s}} = (a_{v_1}, \dots, a_{v_{|D|}})$ such that $a_v = s_i([v])$ and where i is a player who is active at v . Observe that a strategy profile $\mathbf{s} = (s_1, \dots, s_{|N|})$ also defines a unique path w_0, \dots, w_k from the root w_0 to a leaf w_k with history $\mathbf{h}^{\mathbf{s}} = (a_1, \dots, a_k)$ defined such that, for all $0 \leq i < k$, edge (w_i, w_{i+1}) is associated with action $a_i = s_i([w_i])$, where i is the player who is active at w_i . Finally, we have for each player i a utility function $u_i: S_1 \times \dots \times S_{|N|} \rightarrow \mathbb{R}$ associating each strategy profile $\mathbf{s} = (s_1, \dots, s_{|N|})$ with a real value such that $u_i(\mathbf{s}) = u_i^{\mathbf{h}^{\mathbf{s}}}$.

The players N of an extensive-form game, together with their strategies S_i and their utilities u_i , induce a normal-form game $(N, (S_i)_{i \in N}, (u_i)_{i \in N})$. Note that there may be multiple strategy profiles giving rise to the same terminal history, that is, it may well be that $\mathbf{h}^{\mathbf{s}} = \mathbf{h}^{\mathbf{s}'}$, even though $\mathbf{s} \neq \mathbf{s}'$. Therefore, the size of this normal-form game may be exponential in the size of the underlying extensive-form game, if represented naively. This is because there are $\prod_{I \in \mathcal{S}} |A_I| = O(|A|^{|V|})$ strategy profiles, which are implicit in the definition of an extensive-form game: see Figure 2 for an illustration.

	b_{00}	b_{01}	b_{10}	b_{11}
a_0d_0	1,0,0	0,0,1	0,1,0	1,1,0
a_0d_1	1,0,0	0,0,1	0,1,0	1,1,0
a_1d_0	0,1,1	0,1,1	0,1,1	0,1,1
a_1d_1	0,1,1	0,1,1	0,1,1	0,1,1

c_0

	b_{00}	b_{01}	b_{10}	b_{11}
a_0d_0	1,0,0	1,0,1	0,1,0	0,0,1
a_0d_1	1,0,0	1,0,1	0,1,0	0,0,1
a_1d_0	0,0,0	0,0,0	0,0,0	0,0,0
a_1d_1	1,0,1	1,0,1	1,0,1	1,0,1

c_1

Figure 2. The strategic form game associated with the extensive-form game in Figure 1.

2.4. Boolean Circuits

In computer science, Boolean circuits are a well-established model for computing Boolean functions (cf., e.g., [6,7]). In this paper, we will make extensive use of them to concisely represent strategies and utility functions. We will here briefly review their definition, largely following Jukna’s exposition [7].

A k -ary **Boolean function** is a function $\beta : \{0, 1\}^k \rightarrow \{0, 1\}$, where we allow $k = 0$. The **base** of a Boolean circuit is given by a set $B = \{\beta_1, \dots, \beta_k\}$ of Boolean functions. In this paper, we will restrict attention to the set of classical Boolean functions $B = \{\neg, \wedge, \vee\}$ (“not”, “or”, and “and”, respectively), which is known to be functionally complete (i.e., sufficient to define any Boolean function).

Formally, a **Boolean circuit** (or **straight line program**) on n variables x_1, \dots, x_n over base B is given by a sequence of $t \geq n$ **gates** $\gamma_1, \dots, \gamma_t$. The first n gates are given by the variables, that is, $\gamma_1 = x_1, \dots, \gamma_n = x_n$, and are also referred to as **input gates**. Another subset of gates is singled out as the set of **output gates**. Boolean formula over variables x_1, \dots, x_n are thus represented by a Boolean circuit with only one output gate. Each subsequent gate γ_i is the application of a k -ary Boolean function β_i in the base B to k previous gates, that is, $\gamma_i = \beta_i(\gamma_{j_1}, \dots, \gamma_{j_k})$, where $j_1, \dots, j_k < i$. The variables take values in $\{0, 1\}$. Given values $\alpha_1, \dots, \alpha_n$ in $\{0, 1\}$ for the variables x_1, \dots, x_n , one can inductively associate each gate with a value $\alpha(\gamma_i)$ such that $\alpha(x_i) = \alpha_i$, if $1 \leq i \leq n$, and

$$\alpha(\beta_i(\gamma_{j_1}, \dots, \gamma_{j_k})) = \beta_i(\alpha(\gamma_{j_1}), \dots, \alpha(\gamma_{j_k})),$$

if $n < i \leq t$. A Boolean circuit C given by $\gamma_1, \dots, \gamma_t$ on n variables x_1, \dots, x_n and with m output gates $\gamma_{j_1}, \dots, \gamma_{j_m}$ then computes the function $\phi_C : \{0, 1\}^n \rightarrow \{0, 1\}^m$ that, on values $\alpha = (b_1, \dots, b_n)$ for the input gates, yields $\phi_C(\alpha) = (\alpha(\gamma_{j_1}), \dots, \alpha(\gamma_{j_m}))$.

A Boolean circuit is commonly depicted as a directed acyclic graph (V, E) , with the gates as vertices, that is, $V = \{\gamma_1, \dots, \gamma_n\}$, and $(\gamma_i, \gamma_j) \in E$ whenever $\gamma_i = \beta_i(\gamma_{j_1}, \dots, \gamma_{j_k})$ and $\gamma_j = \gamma_{j_\ell}$ for some $1 \leq \ell \leq n$. For examples of Boolean circuits, also see Figures 3 and 4, below.

The **size** of a Boolean circuit given by gates $\gamma_1, \dots, \gamma_t$ on variables x_1, \dots, x_n is given by the number $t - n$ of its gates minus the input gates. It is a well-established fact that the problem of computing $\phi_C(\alpha)$ for given values $\alpha \in \{0, 1\}^n$ for C ’s variables, which is also known as the Circuit Value Problem or Circuit Evaluation Problem, is complete for P under uniform AC_0 -reductions, and therefore can be computed in polynomial time (see in [8], p. 59).

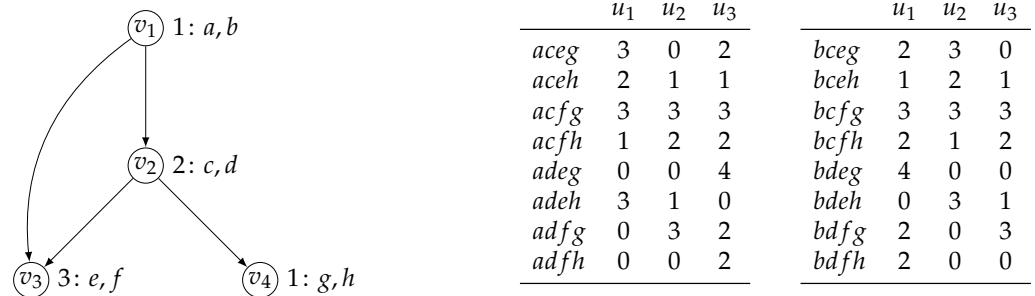


Figure 3. The Boolean circuit $\mathbb{C}_{f_{v_3}^{14}}$ representing strategy $f_{v_3}^{14}: A_{v_1} \times A_{v_2} \rightarrow A_{v_3}$ of Example 1. The value of x_0 represents the choice for a or b at v_1 , the value of x_1 the choice for c or d at v_2 , and the value of y_0 the choice for e or f at v_3 . The local indices of a, c , and e are given by 0, and those of b, d , and f by 1. The instantiation on the right thus represents that $f_{v_3}^{14}(ad) = f$.

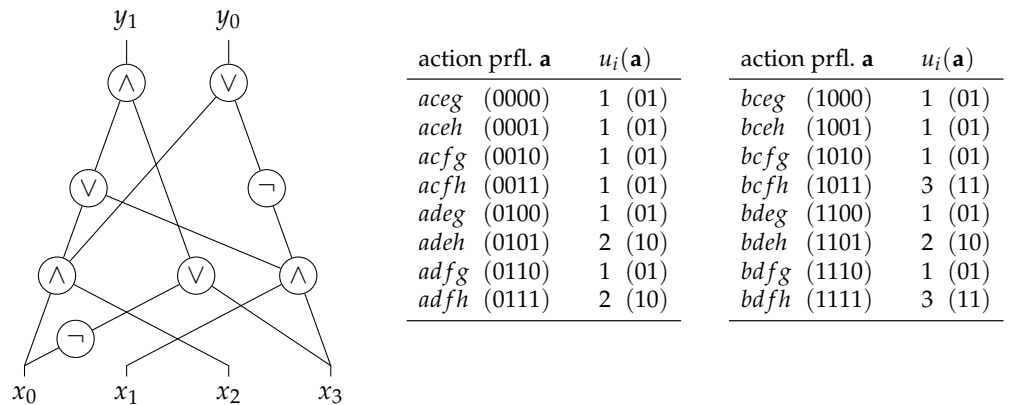


Figure 4. The Boolean circuit on the left represents the utility function u_i on the right for a player i over the action profiles of the game in Example 1. The binary representations of action profiles and numerical values are depicted in parentheses.

3. Partial Order Games

We now introduce the framework of partial order games. The basic idea is that, as in extensive form games, the game contains a number of decision nodes, which are partitioned among the players. However, play in the game is not defined by a game tree. Instead, partial order games have a binary *dependence relation* over decision nodes. This dependence defines the information available to a player when it makes a choice. If a decision node v for player i is dependent on decision nodes v_1, \dots, v_k , then this means that the information available to a strategy when making a choice at v is precisely the choices that were made at decision nodes v_1, \dots, v_k . To play a partial order game, a player must choose a selection of strategies (we call them vertex strategies), one for each of their decision nodes. A strategy for a decision node v must select a choice for that decision node *taking as input the choices that were made for the decision nodes v_1, \dots, v_k upon which it is dependent*. In this way, we have an explicit representation of the information available when making a choice, in contrast to the use of information sets in extensive form games. Utilities in partial order games are not associated with individual nodes (as in extensive form games), but derive from the total profile of actions that were made.

Our usual way of thinking about extensive form games is as players alternating to make moves, working their way down the game tree to a leaf node; upon reaching a leaf node, the game is over. Thinking about games in this way naturally gives rise to a temporal order over choices in the game: a choice is made for the root node first, and then successive nodes in the game tree. Partial order games are more abstract than this. They induce only a partial temporal ordering over decision nodes: while the choices for v_1, \dots, v_k must strictly precede the choice for v if v is dependent on v_1, \dots, v_k , it may well be that decision nodes

are independent: if nodes v and v' are in disconnected components of the dependence graph, then we can say nothing about their temporal order.

Let us consider an example.

Example 1. Consider the three-person partial order game depicted in Figure 5, with the directed acyclic graph it is based on given on the left. To the right of each vertex v , we have indicated the player i_v and the actions they have at their disposal at v . For instance, at vertex v_2 , player 2 can choose among the actions c and d . If all players make a choice at their respective vertices, an action profile results, which then is associated with a utility value. For instance, if player 1 chooses a at v_1 and h at v_4 , player 2 chooses c at v_2 , and player 3 chooses f at v_3 , the action profile (a, c, f, h) (also denoted by $acfh$) results, yielding utility values of 1 to player 1, and 2 to players 2 and 3. The table on the right summarises the utilities the players get under the different action profiles that may be played. Note that in this game there are 2^4 possible action profiles.

The crucial feature of partial order games, however, is that the players can make their choices at a given vertex v depending on the choices the other players make at the parents of v . For instance, player 2 could adopt the strategy to play c at v_2 , if player 1 chooses a at v_1 , and plays d otherwise. Accordingly, player 2 has $2^2 = 4$ strategies at their disposal at v_2 . As v_1 does not have any parents, player 1 has only two strategies at their disposal at v_1 , but they have $2^2 = 4$ strategies at v_4 . Meanwhile, player 3 has $2^4 = 16$ strategies at v_3 . A strategy profile specifies a strategy for each player at each of their vertices. Accordingly, in this game there are $2 \times 2^2 \times 2^4 \times 2^2 = 2^9 = 512$ possible strategy profiles.

The number of strategy profiles clearly outnumber the number of action profiles. Still, every strategy profile induces a unique action profile. This relationship, however, is not generally injective, as multiple strategy profiles may induce the same action profile. Assume, for instance, that the players adopt the following strategies:

- player 1 chooses a at v_1 ;
- player 2 chooses c if player 1 plays a at v_1 , and d , otherwise;
- player 3 chooses e if player 1 chooses a at v_1 and player 2 d at v_2 , and f , otherwise; and
- player 4 chooses h independently of whether player 2 chooses c or d at v_2 .

If this strategy profile is played, it can readily be seen that action profile $acfh$ results, which we have already seen gives utilities 1, 2, and 2 to the players 1, 2, and 3, respectively.

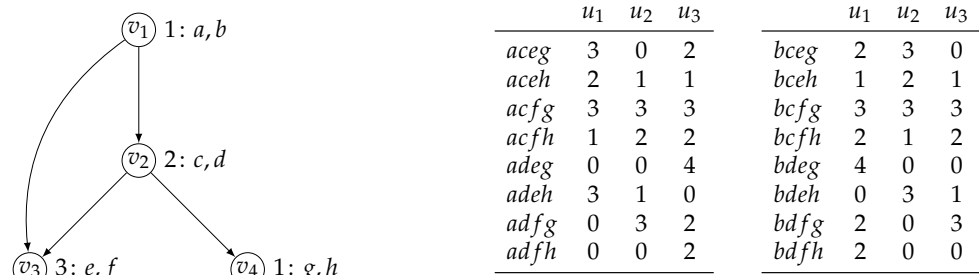


Figure 5. A simple example of a partial order game. On the left, the DAG, where each vertex v is labelled $i: x, y$, where i is the player active at the respective vertex and x and y are the actions available to i at v . On the right, the representation of the utility function which associates a numeric value with each action profile.

We now proceed to our formal definition of partial order games, which enables the application of Nash equilibrium as well as the formulation of a natural backwards induction procedure later on. As before, let $N = \{1, \dots, n\}$ be a finite set of **players**. Then a **partial order game** on a directed acyclic graph (V, E) —the game’s **dependency graph**—associates each vertex v in V with a unique player in N , denoted by i_v , and a non-empty set of **actions**, denoted by A_v .

One player may be associated with multiple vertices, that is, it may very well be that $i_u = i_v$ even if $v \neq u$. We let V_i denote the set of vertices associated with player i . We also say that player i is **active** at any vertex v in V_i . If E is also connected—that is, if vEw or wEv for all *distinct* v and w in V —we also refer to any partial order game based on it as a **total-order game**. As E is assumed to be acyclic, it follows that E is transitive in total order games. On the other end of the spectrum, we have partial order games on dependency graphs (V, E) with $E = \emptyset$. This class of games we will also refer to as **empty-order games**.

Given a subset of $W = \{w_1, \dots, w_k\}$ of vertices, an **action profile for W** is a tuple $\mathbf{a}_W = (a_{w_1}, \dots, a_{w_k})$ specifying one action a_{w_j} for each vertex w_j . In our examples, we occasionally omit parentheses and commas, and write $a_{w_1} \dots a_{w_k}$ for $(a_{w_1}, \dots, a_{w_k})$. The set of action profiles for W we denote by \mathbf{A}_W . For the set $V = \{v_1, \dots, v_{|V|}\}$ of all vertices, we write $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ for \mathbf{a}_V and refer to \mathbf{a} simply as an **action profile**. We denote the set of action profiles by $\mathbf{A} = A_{v_1} \times \dots \times A_{v_{|V|}}$. For a set $V_i = \{w_1, \dots, w_k\}$ of vertices at which a player i is active, we also write $\mathbf{a}_i = (a_{w_1}, \dots, a_{w_k})$ for \mathbf{a}_{V_i} and refer to \mathbf{a}_i as an **action profile for player i** . We let \mathbf{A}_i denote the set of actions profiles for a player i . For action profiles $\mathbf{a}_W = (a_{w_1}, \dots, a_{w_k})$ and $\mathbf{b}_U = (b_{u_1}, \dots, b_{u_m})$ for disjoint sets W and U , we denote by $(\mathbf{a}_W, \mathbf{b}_U)$ the action profile $(a_{w_1}, \dots, a_{w_k}, b_{u_1}, \dots, b_{u_m})$ for $W \cup U$. If $U = V \setminus W$ we also write $(\mathbf{a}_W, \mathbf{a}_{-W})$ for $(\mathbf{a}_W, \mathbf{a}_U)$, and if $U = V \setminus V_i$, we also write $(\mathbf{a}_i, \mathbf{a}_{-i})$. Given a topological sorting $\tau = (v_1, \dots, v_{|V|})$, we define a τ -**history** as a sequence of actions $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$ in $A_{v_1} \times \dots \times A_{v_k}$ for some $1 \leq k \leq |V|$. We stipulate the empty sequence of actions, denoted by $()$, to be a τ -history as well. For an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$, we say that $\mathbf{h} = (b_{v_1}, \dots, b_{v_k})$ is a **history of \mathbf{a}** if $(b_{v_1}, \dots, b_{v_k}) = (a_{v_1}, \dots, a_{v_k})$.

At each vertex v , player i_v can make their choice of action dependent on the actions chosen at the parents of v . A **(conditional) strategy at a vertex v** , or **vertex strategy**, is therefore a function

$$f_v: A_{y_1} \times \dots \times A_{y_k} \rightarrow A_v,$$

where y_1, \dots, y_k are the parents of v . We say that f_v is **unconditional** if it is constant, that is, if $f_v(a_1, \dots, a_k) = f_v(b_1, \dots, b_k)$ for all action profiles $\mathbf{a} = (a_1, \dots, a_k)$ and $\mathbf{b} = (b_1, \dots, b_k)$. In this case, for $a \in A_v$, we denote by f_v^a the unconditional strategy at vertex v that maps every action profile of v 's parents to action a . We will sometimes identify unconditional strategies f_v with the action $f_v(a_1, \dots, a_k)$ it specifies for *each* action profile $\mathbf{a} = (a_1, \dots, a_k)$ and write $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ for the profile of unconditional strategies $\mathbf{f}^{\mathbf{a}} = (f_{v_1}^{a_{v_1}}, \dots, f_{v_{|V|}}^{a_{v_{|V|}}})$. Note that if vertex v has no parents, then $A_{u_1} \times \dots \times A_{u_k} = \{()\}$, where $()$ is the empty tuple. Accordingly, f_v determines a single choice among A_v , namely, $f_v(())$ in this case. By F_v we denote the set of conditional strategies available at a vertex v .

By a **strategy profile** we then understand a profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ consisting of one conditional strategy for each vertex. For $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ a strategy profile and $W = \{w_1, \dots, w_k\}$ a subset of vertices in V , we let $f_W = (f_{w_1}, \dots, f_{w_k})$ denote the partial profile that is like \mathbf{f} but restricted to $W \subseteq V$. Let $\mathbf{f}^1 = (f_{v_1}^1, \dots, f_{v_{|V|}}^1)$ through $\mathbf{f}^k = (f_{v_1}^k, \dots, f_{v_{|V|}}^k)$ be k strategy profiles and $\{X_1, \dots, X_k\}$ be a k -partition of the vertex set V . Then, $(\mathbf{f}_{X_1}^1, \dots, \mathbf{f}_{X_k}^k)$ denotes the strategy profile $\mathbf{g} = (g_{v_1}, \dots, g_{v_{|V|}})$ such that, for every vertex v in V , we have $g_v = f_v^j$, if $v \in X_j$.

A **strategy for a player i** is a profile $\mathbf{f}_{V_i} = (f_{v_1}, \dots, f_{v_k})$ of vertex-strategies, where $V_i = \{v_1, \dots, v_k\}$, is the set of vertices at which player i is active. We generally denote \mathbf{f}_{V_i} by \mathbf{f}_i , and the set of strategies available to player i by \mathbf{F}_i . For strategy profiles $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ and $\mathbf{g} = (g_{v_1}, \dots, g_{v_{|V|}})$ and a player i , we also write $(\mathbf{f}_{-i}, \mathbf{g}_i)$ for the strategy profile $(\mathbf{f}_{V \setminus V_i}, \mathbf{g}_{V_i})$.

For every vertex v , a strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ defines an action $a_v^{\mathbf{f}}$, the **evaluation of \mathbf{f} at v** , recursively as follows:

$$a_v^{\mathbf{f}} = \begin{cases} f_v(()) & \text{if } \text{depth}(v) = 0, \\ f_v(a_{u_1}^{\mathbf{f}}, \dots, a_{u_k}^{\mathbf{f}}) & \text{if } \text{depth}(v) > 0 \text{ and } u_1, \dots, u_k \text{ are the parents of } v. \end{cases}$$

Observe that $a_v^{\mathbf{f}}$ is well defined because in a directed acyclic graph vertices with depth 0 exist and the parents of each vertex are of lower depth than the vertex itself. A strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ can thus be seen to induce the action profile $\mathbf{a}^{\mathbf{f}} = (a_{v_1}^{\mathbf{f}}, \dots, a_{v_{|V|}}^{\mathbf{f}})$. We also say that strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ **sustains** action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ if $\mathbf{a} = \mathbf{a}^{\mathbf{f}}$. We occasionally denote by $(\mathbf{a}_W, \mathbf{f}_U)$ the *action profile* $\mathbf{a}^{\mathbf{g}}$ where \mathbf{g} is the strategy profile $(\mathbf{f}_W^{\mathbf{a}}, \mathbf{f}_U)$ which specifies *unconditional* strategies for W . In other words, we have $(a_{v_1}, \dots, a_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ denote the *action profile* that results if on v_1 through v_k actions a_{v_1}, \dots, a_{v_k} are played, and at vertices v_{k+1} through $v_{|V|}$ the strategies $f_{v_{k+1}}$ through $f_{v_{|V|}}$.

Note the difference between actions and action profiles on the one hand, and strategies and strategy profiles on the other. Every strategy profile induces a unique action profile, whereas the same action profile may be induced by different strategy profiles. In an important sense, the action profiles are the outcomes of the game.

Therefore, we take the set of action profiles of a partial order game as its set of **outcomes**, over which the players' preferences are defined. Each player's preferences over the action profiles are given by a real-valued **utility function** $u_i: \mathbf{A} \rightarrow \mathbb{R}$, where we assume that player i *strictly prefers* action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ to action profile $\mathbf{b} = (b_{v_1}, \dots, b_{v_{|V|}})$ whenever $u_i(\mathbf{a}) > u_i(\mathbf{b})$. We extend utility functions u_i to strategy profiles \mathbf{f} , and write $u_i(\mathbf{f})$ for $u_i(\mathbf{a}^{\mathbf{f}})$. To fix concepts and notation, we consider Example 1 once more.

Example 1 (cont'd). Consider the game in Figure 5. At vertices v_1, v_2 , and v_4 , we have the following strategies:

F_{v_1}	()	F_{v_2}	a	b	F_{v_4}	c	d
$f_{v_1}^1$	a	$f_{v_2}^1$	c	c	$f_{v_4}^1$	g	g
$f_{v_1}^2$	b	$f_{v_2}^2$	c	d	$f_{v_4}^2$	g	h
		$f_{v_2}^3$	d	c	$f_{v_4}^3$	h	g
		$f_{v_2}^4$	d	d	$f_{v_4}^4$	h	h

At vertex v_3 , moreover, we have 2^4 strategies, illustrating the exponential blowup that results from strategies being represented explicitly.

F_3	ac	ad	bc	bd	ac	ad	bc	bd	ac	ad	bc	bd							
$f_{v_3}^1$	e	e	e	e	$f_{v_3}^5$	e	f	e	e	$f_{v_3}^9$	f	e	e	e	$f_{v_3}^{13}$	f	f	e	e
$f_{v_3}^2$	e	e	e	f	$f_{v_3}^6$	e	f	e	f	$f_{v_3}^{10}$	f	e	e	f	$f_{v_3}^{14}$	f	f	e	f
$f_{v_3}^3$	e	e	f	e	$f_{v_3}^7$	e	f	f	e	$f_{v_3}^{11}$	f	e	f	e	$f_{v_3}^{15}$	f	f	f	e
$f_{v_3}^4$	e	e	f	f	$f_{v_3}^8$	e	f	f	f	$f_{v_3}^{12}$	f	e	f	f	$f_{v_3}^{16}$	f	f	f	f

The (conditional) strategy profile $\mathbf{f} = (f_{v_1}^1, f_{v_2}^3, f_{v_3}^8, f_{v_4}^2)$, then, for instance, yields the evaluation $\mathbf{a}^{\mathbf{f}} = (a_{v_1}^{\mathbf{f}}, a_{v_2}^{\mathbf{f}}, a_{v_3}^{\mathbf{f}}, a_{v_4}^{\mathbf{f}}) = (a, d, f, h)$ as

$$\begin{aligned} a_{v_1}^{\mathbf{f}} &= f_{v_1}^1(()) = a & a_{v_2}^{\mathbf{f}} &= f_{v_2}^3(a_{v_1}^{\mathbf{f}}) = f_{v_2}^3(a) = d \\ a_{v_3}^{\mathbf{f}} &= f_{v_3}^8(a_{v_1}^{\mathbf{f}}, a_{v_2}^{\mathbf{f}}) = f_{v_3}^8(a, d) = f & a_{v_4}^{\mathbf{f}} &= f_{v_4}^2(a_{v_2}^{\mathbf{f}}) = f_{v_4}^2(d) = h \end{aligned}$$

Accordingly, the utilities on this profile $\mathbf{f} = (f_{v_1}^1, f_{v_2}^3, f_{v_3}^8, f_{v_4}^2)$ for these three players are therefore determined by the action profile $adf h$. Hence, $u_1(\mathbf{f}) = 0$, $u_2(\mathbf{f}) = 0$, and $u_3(\mathbf{f}) = 2$.

With the players N , their conditional strategies \mathbf{F}_i , and their preferences $u_i: \mathbf{F} \rightarrow \mathbb{R}$ over strategy profiles, a partial order game on a directed acyclic graph (V, E) can thus be seen to define immediately a game in normal-form $(N, (\mathbf{F}_i)_{i \in N}, (u_i)_{i \in N})$ (cf., Section 2.2, above). Accordingly, partial order games are directly amenable to game-theoretical analysis using the usual solution concepts, in particular, Nash equilibrium.

Conversely, every normal-form game $(N, (S_i)_{i \in N}, (u_i)_{i \in N})$ can be seen as a partial order game on the same set N of players. Its dependency graph is then given by (N, \emptyset) , associating with vertex $i \in N$, player $i \in N$ and strategy set S_i . Identifying each unconditional strategy f_i with $f_i(()) = s_i$ in S_i , we adopt the utility functions u_i unaltered for the utility functions of the partial order game.

3.1. Concise Representations for Strategies, Profiles, and Utilities

The transformation from partial order games to normal-form games comes at the cost of an exponential blowup. Observe that in the definition of partial order games, the sets of action profiles and strategy profiles are defined implicitly. Given a partial order game on a directed acyclic graph (V, E) with a set of players N and actions $A = A_{v_1} \cup \dots \cup A_{v_{|V|}}$, the number of action profiles $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ is bounded by $O(|A|^{|V|})$, and so is the number of unconditional strategy profiles $\mathbf{f}^{\mathbf{a}} = (f_{v_1}^{a_{v_1}}, \dots, f_{v_{|V|}}^{a_{v_{|V|}}})$. By contrast, the number of both conditional strategies and the number of conditional strategy profiles are bounded by $O(|A|^{|A|^{|V|}})$. It is also worth observing that the size of both a conditional strategy f and a profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ of conditional strategies is bounded by $O(|A|^{|V|})$.

Even though the dependency graph allows a concise representation of the strategies and strategy profiles in a partial order game, they remain large objects. It is therefore desirable to have concise representations of action profiles, utility functions, strategies, and strategy profiles. To this end, we take recourse to *Boolean circuits*, a formalism well established in theoretical computer science (see Section 2.4). We first show how action profiles, utility functions, and strategy profiles can be represented concisely by Boolean circuits. We then prove two lemmas that will be useful for our later complexity proofs in Sections 5 and 7.

When representing partial order games concisely, we assume that for distinct vertices v and w the action sets A_v and A_w are disjoint. Having assumed A_v to be non-empty for each vertex v , it thus follows that $|V| \leq |A|$. For ease of presentation, we also assume that $|A_v|$ is an integer power of 2 for each vertex v . In this section, we assume that for each vertex v , the elements of set A_v are enumerated as $a_0, \dots, a_{|A_v|-1}$, thus associating each action a_k with a unique **local index** k at v . Let $\text{bin}(k)$ denote the **binary representation** of the local index k , using *exactly* $\log_2 |A_v|$ digits. Thus, if $|A_v| = 16$, we have $\text{bin}(3) = 0011$.

Let v be a vertex with parents u_1, \dots, u_k . We then represent a strategy $f_v: A_{u_1} \times \dots \times A_{u_k} \rightarrow A_v$ by a Boolean circuit \mathbb{C}_{f_v} with $\log_2 |A_{u_1}| + \dots + \log_2 |A_{u_k}|$ input gates and $\log_2 |A_v|$ output gates. For an action profile $\mathbf{a} = (a_{u_1}, \dots, a_{u_k})$ in $A_{u_1} \times \dots \times A_{u_k}$ and a_v an action in A_v , we then have that $f_v(a_{u_1}, \dots, a_{u_k}) = a_v$ if and only if on input $\text{bin}(j_1) \dots \text{bin}(j_k)$ the circuit \mathbb{C}_{f_v} evaluates to outputs $\text{bin}(i)$, where j_1, \dots, j_k and i , are the local indices of a_{v_1}, \dots, a_{v_k} , and a_v at their respective vertices. As \mathbb{C}_{f_v} can be seen as combining $\log_2 |A_v|$ Boolean functions in $\log_2 |A_{u_1}| + \dots + \log_2 |A_{u_k}|$ variables and

$\log_2 |A_{u_1}| + \dots + \log_2 |A_{u_k}| \leq |A|$, we may assume that the circuit \mathbb{C}_{f_v} is of size at most exponential in $|A|$. More precisely, we may assume that the size of \mathbb{C}_{f_v} is $\Theta(\frac{2^{|A|}}{|A|})$ [9,10].

Example 2. For an example, see Figure 3, which depicts the Boolean circuit $\mathbb{C}_{f_{v_3}^{14}}$ for strategy $f_{v_3}^{14}: A_{v_1} \times A_{v_2} \rightarrow A_{v_3}$ in Example 1, given by

$$f_{v_3}^{14}(\mathbf{a}) = \begin{cases} e & \text{if } \mathbf{a} = bc, \\ f & \text{otherwise.} \end{cases}$$

Recall that $A_{v_1} = \{a, b\}$, $A_{v_2} = \{c, d\}$, and $A_{v_3} = \{e, f\}$. Let the local indices of a, c , and e be given by 0, and those of b, d , and f by 1. As $|A_{v_1}| = |A_{v_2}| = |A_{v_3}| = 2$, any action profile $\mathbf{a} \in A_{v_1} \times A_{v_2}$ can be represented by two Boolean variables x_0 and x_1 , where the value of x_0 represents the choice for a or b at v_1 , the value of x_1 the choice for c or d at v_2 . For instance, the action profile $ad \in A_{v_1} \times A_{v_2}$ can thus be represented by setting x_0 to 0 and x_1 to 1. For these values the circuit $\mathbb{C}_{f_{v_3}^{14}}$ evaluates to 1, which corresponds to $f_{v_3}^{14}(ad) = f$ and as depicted in Figure 3.

Observe that an unconditional strategy f_v^a at a vertex v with parents u_1, \dots, u_k , which maps every profile $(a_{u_1}, \dots, a_{u_k})$ invariably to action a in A_v , is represented by a Boolean circuit $\mathbb{C}_{f_v^a}$ of size at most polynomial in $|A|$. The circuit $\mathbb{C}_{f_v^a}$ will still have $\sum_{1 \leq m \leq k} \log_2 |A_{u_m}|$ inputs and $\log_2 |A_v|$ outputs, but will involve only one \perp -gate and one \top -gate. Let j be the local index of a at v and $\text{bin}(j) = b_1, \dots, b_{\log_2 |A_v|}$. Then, for $1 \leq \ell \leq \log_2 |A_v|$, connect the \perp -gate with output y_ℓ , if $b_\ell = 0$, and the \top -gate with output y_ℓ , if $b_\ell = 1$.

Similarly, a rational-valued utility function $u_i: A_{v_1} \times \dots \times A_{v_{|V|}} \rightarrow \mathbb{Q}$ can be represented by a Boolean circuit \mathbb{C}_{u_i} . This circuit will have $\sum_{v \in V} \log_2 |A_v|$ inputs and outputs $y_1, \dots, y_k, z_1, \dots, z_m$. Moreover, for $(a_{v_1}, \dots, a_{v_{|V|}})$ an action profile in $A_{v_1} \times \dots \times A_{v_{|V|}}$ with local indices $j_1, \dots, j_{|V|}$ at their respective vertices, we have $u_i(a_{v_1}, \dots, a_{v_{|V|}}) = \frac{n}{d+1}$ if and only if on input $\text{bin}(j_1), \dots, \text{bin}(j_{|V|})$ the circuit \mathbb{C}_{u_i} yields $\text{bin}(n)$ for outputs y_1, \dots, y_k and $\text{bin}(d)$ for outputs z_1, \dots, z_m (Observe that if $u_i(a_{v_1}, \dots, a_{v_{|V|}})$ is a positive integer, the outputs z_1, \dots, z_m can be dispensed with, as $\sum_{x \in \mathbb{Q}} x = 0$ and $d = \sum_{1 \leq j \leq m} z_j \cdot 2^j$). Again, as $|A_{u_1}| + \dots + |A_{u_k}| \leq |A|$, we may assume that any such circuit \mathbb{C}_{u_i} will be of size at most exponential in $|A|$.

Example 3. Figure 4 illustrates how a utility function $u_i: A_{v_1} \times A_{v_2} \times A_{v_3} \times A_{v_4} \rightarrow \mathbb{R}$ for a player i in the game of Example 1 is represented by a Boolean circuit. The utility function u_i is depicted on the right. We assume that the local indices of a, c, e , and g be given by 0, and those of b, d, f , and h by 1. As $|A_{v_1}| = |A_{v_2}| = |A_{v_3}| = |A_{v_4}| = 2$, an action profile \mathbf{a} in $A_{v_1} \times A_{v_2} \times A_{v_3} \times A_{v_4}$ is thus given by means of an assignment to four Boolean variables x_0, x_1, x_2 , and x_4 , representing the choices at the vertices v_1, v_2, v_3 , and v_4 , respectively. For instance, action profile $acfh$ is represented by setting variables x_0 and x_1 to 0, and variables x_2 and x_3 to 1, which can thus also be denoted by the binary string 0011. Evaluating the circuit for these values, we find that y_1 is set to 0 and y_0 to 1. This corresponds to $u_i(acfh) = 1$ and $\text{bin}(1) = 01$. Observe that the Boolean circuit can be evaluated in time polynomial in its size and that it is exponentially smaller than the explicit tabelling of the utility function on the right.

We conclude this section by showing two useful lemmas. The first establishes that, even if a conditional strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is represented as a Boolean circuit, the action profile $\mathbf{a}^{\mathbf{f}} = (a_{v_1}, \dots, a_{v_{|V|}})$ that \mathbf{f} gives rise to, can be computed in polynomial time.

Lemma 1. Let $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ be a conditional strategy profile represented by Boolean circuits $\mathbb{C}_{f_{v_1}}, \dots, \mathbb{C}_{f_{v_{|V|}}}$. Then, the action profile $\mathbf{a}^{\mathbf{f}} = (a_{v_1}^{\mathbf{f}}, \dots, a_{v_{|V|}}^{\mathbf{f}})$ can be computed in polynomial time.

Proof Sketch. Proceeding inductively, find for every vertex v the action a_v^f as follows. For vertices v of depth 0, the Boolean circuit \mathbb{C}_{f_v} should give the local index in binary of action a_v^f at v , and therewith, a_v^f immediately. For a vertex v of a strictly positive depth with parents u_1, \dots, u_k , we may assume that the local indices in binary $\text{bin}(j_{u_1}), \dots, \text{bin}(j_{u_k})$ of $a_{u_1}^f, \dots, a_{u_k}^f$ can be computed in polynomial time. As the evaluation problem for Boolean circuits is solvable in polynomial time, the circuit \mathbb{C}_{f_v} for inputs $\text{bin}(j_{u_1}), \dots, \text{bin}(j_{u_k})$ can also be evaluated in polynomial time, providing us with the local index of $f_v(a_{u_1}^f, \dots, a_{u_k}^f)$ at v in binary, which gives us the result. \square

Lemma 2. Let $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ be a conditional strategy profile represented by Boolean circuits $\mathbb{C}_{f_{v_1}}, \dots, \mathbb{C}_{f_{v_{|V|}}}$, and $u_i: A_{v_1} \times \dots \times A_{v_{|V|}} \rightarrow \mathbb{Q}$ be a utility function represented by a Boolean circuit \mathbb{C}_{u_i} . Then, $u_i(\mathbf{f})$ can be computed in polynomial time.

Proof Sketch. Thanks to Lemma 1, we can compute the action profile $\mathbf{a}^f = (a_{v_1}^f, \dots, a_{v_{|V|}}^f)$ in polynomial time. Let $j_1, \dots, j_{|V|}$ be the local indices of $a_{v_1}^f, \dots, a_{v_{|V|}}^f$ at their respective vertices. Using $\text{bin}(j_1), \dots, \text{bin}(j_{|V|})$ as inputs for the circuit \mathbb{C}_{u_i} , we obtain the binary encoding of $u_i(a_{v_1}^f, \dots, a_{v_{|V|}}^f)$. As the evaluation problem for Boolean circuits is solvable in polynomial time, this suffices for the result. \square

4. Related Game-Theoretic Models

In this section, we explore the interrelations between partial order games and four related game-theoretic models: *partial order Boolean games*, *Multi-Agent Influence Diagrams (MAIDs)*, *concurrent games as event structures*, and *extensive-form games*.

4.1. Boolean Games and Partial Order Boolean Games

A special class of partial order games, referred to as *partial order Boolean games*, was introduced in [1]. Partial order Boolean games extend Boolean games [11–15], where for a set Φ of propositional variables, each player i exercises unique control over the truth-values assigned to a subset Φ_i of variables and aims to satisfy a goal formula γ_i over Φ . Partial order Boolean games are Boolean games enriched with a dependency graph on Φ . To avoid confusion, we will write v_p for the propositional variable p when it occurs in the role of a vertex in the dependency graph. The player i associated with a vertex v_p coincides with the player controlling p in the Boolean game, and i can make their choice of truth-value for p depend on the values assigned to the propositional variables q_1, \dots, q_k if v_{q_1}, \dots, v_{q_k} are the parents of v_p in the dependency graph. In a Boolean game, such dependencies do not exist. Thus, Boolean games can be seen as a special class of partial order Boolean games. The class where the dependency graph does not have any edges but can be expressed by (Φ, \emptyset) .

Clearly, every partial order Boolean game $(N, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n, D)$, where $D = (\Phi, E)$ is a dependency graph on Φ , defines a partial order game with the same set of players N and the same dependency graph, where player i is associated with vertex v_p if and only if $p \in \Phi_i$. At vertex v_p , the player i associated with v_p has two actions at their disposal: setting p to true (p) and setting p to false (\bar{p}). We will assume that the local index of action p at vertex v_p is 1 and that that of action \bar{p} at v_p is 0.

In a partial order Boolean game, an individual strategy at vertex v_p , controlled by i , is represented by a so-called **choice equation** of the form $p: \varphi(q_1, \dots, q_k)$, where v_{q_1}, \dots, v_{q_k} are the parents of v_p and $\varphi(q_1, \dots, q_k)$ is a propositional logic formula over q_1, \dots, q_k , with the interpretation that i sets p to true if $\varphi(q_1, \dots, q_k)$ evaluates to true given the decisions made for q_1, \dots, q_k at v_{q_1}, \dots, v_{q_k} , and to false otherwise. Thus a choice equation $p: \varphi(q_1, \dots, q_k)$ defines an individual strategy $f_{v_p}: \{q_1, \bar{q}_1\} \times \dots \times \{q_k, \bar{q}_k\} \rightarrow \{p, \bar{p}\}$ at vertex v_p . This strategy, moreover, can be represented by a Boolean circuit $\mathbb{C}_{f_{v_p}}$ for the Boolean function $\beta_\varphi: \{0, 1\}^k \rightarrow \{0, 1\}$ expressed by the formula φ . We may assume that the circuit $\mathbb{C}_{f_{v_p}}$ is of size at most polynomial in the size of the choice equation $p: \varphi(q_1, \dots, q_k)$

and that it has exactly k inputs and one output. We may furthermore assume that the circuit $\mathbb{C}_{f_{v_p}}$ can be obtained in time polynomial in the size of φ from $p: \varphi(q_1, \dots, q_k)$.

Similar considerations apply to the representation of the players' utility functions. In partial order Boolean games, the preferences of each player i are represented by a goal formula γ_i in the propositional language over Φ . Every action profile $\mathbf{a} = (a_{v_{p_1}}, \dots, a_{v_{p_{|\Phi|}}})$ of a Boolean partial order game defines a **valuation** $\alpha_{\mathbf{a}}: \Phi \rightarrow \{0, 1\}$ and yields player i utility 1 if $\alpha_{\mathbf{a}}$ satisfies γ_i , and utility 0 otherwise. The Boolean function $\beta_{\gamma_i}: \{0, 1\}^{|\Phi|} \rightarrow \{0, 1\}$ expressed by γ_i can be represented by a Boolean circuit \mathbb{C}_{γ_i} with at most $|\Phi|$ inputs and exactly one output. As at every vertex v_p , the assigned player has exactly two actions at their disposal and utility values are integers, we find that \mathbb{C}_{γ_i} is exactly of the form suggested for the representation of players' utilities in partial order games in the previous section. Moreover, \mathbb{C}_{γ_i} will be of size at most exponential in $|\Phi|$ and of size polynomial in γ_i . We may furthermore assume that the circuit $\mathbb{C}_{f_{\gamma_i}}$ can be obtained from γ_i in polynomial time.

4.2. Multi-Agent Influence Diagrams (MAIDs)

Multi-agent Influence Diagrams (MAIDS) were proposed in [2,3] and later extended from a game-theoretic perspective in [16]. MAIDs trace their origins to *Bayesian Networks* (cf., [17]) and *influence diagrams*, a decision theoretic extension of Bayesian networks proposed by Howard and Matheson [18].

Like partial order games, MAIDs involve a finite set $\mathcal{A} = \{a_1, \dots, a_n\}$ of **agents** and are defined in directed acyclic graphs, where three different types of vertices are distinguished: a set of **chance nodes** \mathcal{X} , a set of **decision nodes** \mathcal{D}_a for each agent a in \mathcal{A} , and a set of **utility nodes** \mathcal{U}_a also for each agent a . In line with the Bayesian network literature, every vertex is taken to be a random variable X with possible values in a finite domain $dom(X)$. It is generally assumed that utility nodes cannot be parents of other nodes, and that their domain is a finite set of real numbers. (Hammond et al. [16] relax this condition, and also allow utility nodes to have outgoing edges.)

The values any variable in a MAID can take may depend on the values assumed by that variable's parents in the DAG. This is very much in line with how a choice of action at a node may depend on the decisions taken at parent nodes in partial order games. In the MAIDs framework, the parents of a decision node X are denoted by $Pa(X)$ and an **instantiation**, commonly denoted by \mathbf{pa} , for $Pa(X) = \{Y_1, \dots, Y_k\}$ defines a value from the domain of each of these variables, that is, $\mathbf{pa} \in dom(Y_1) \times \dots \times dom(Y_k)$. A **conditional probabilistic distributions (CPD)** for a decision or utility node X now assigns a probability $P(X | \mathbf{pa})$ for each instantiation \mathbf{pa} of $Pa(X)$, and a **decision rule** defines a conditional probabilistic distribution for one particular decision variable. Sometimes it is required that CPDs for utility nodes be *deterministic*, arguing that all of the stochasticity should be subsumed into chance variables. This convention, however, is not universally adopted in the literature on MAIDs. We refer to the work in [19] for a good in-depth study. On this basis, a **strategy** δ_a for an agent a defines a conditional probability distribution for each decision variable D in \mathcal{D}_a . A **strategy profile** is then a tuple $(\delta_{a_1}, \dots, \delta_{a_n})$ of strategies, one for each agent. Given a strategy profile $(\delta_{a_1}, \dots, \delta_{a_n})$, the MAID reduces to a Bayesian network, and as such defines a joint probability distribution $P_{\delta_{a_1}, \dots, \delta_{a_n}}$ over all of its variables. Accordingly, every agent a can be assigned a **expected utility** defined as follows:

$$\mathbb{E}_a(\delta_{a_1}, \dots, \delta_{a_n}) = \sum_{U \in \mathcal{U}_a} \sum_{x \in dom(U)} P_{\delta_{a_1}, \dots, \delta_{a_n}}(U = x) \cdot x.$$

With players, their strategies, and their utilities being defined, a MAID defines a strategic-form game, and is amenable to analysis by the usual game-theoretic solution concepts.

It is not hard to see how every partial order game can be seen as a MAID without chance variables, with 'hidden' utility nodes, and only allowing for deterministic decision rules. Formally, let $N = \{1, \dots, n\}$ be the set of players; V_1, \dots, V_n be the sets of nodes assigned to the players 1 through n , respectively; (V, E) be a directed acyclic graph with

$V = V_1 \cup \dots \cup V_n$; and u_1, \dots, u_n be utility functions for the players 1 through n . Then, the partial order game defined from this can be seen as a MAID with set of agents $\mathcal{A} = N$ and decision variables V_1, \dots, V_n such that $dom(V_v) = A_v$ for all vertices v in V . There are no chance variables, and one utility variable U_i for each player i that is a child of *all* decision nodes. The MAID’s partial order is then given by (V', E') where $V' = V \cup \{U_1, \dots, U_n\}$ and $E' = E \cup V \times \{U_1, \dots, U_n\}$. Thus, the instantiations of the parents of each utility variable, the set of instantiations of all decision variables, corresponds to the set of all action profiles of the partial order game. Each of these instantiations $(a_{v_1}, \dots, a_{v_{|V|}})$, in turn, corresponds to a *deterministic* strategy profile $(\delta_1, \dots, \delta_n)$ in the MAID, which allows us to complete the model by setting $\mathbb{E}_i((\delta_1, \dots, \delta_n)) = u_i((a_{v_1}, \dots, a_{v_{|V|}}))$.

4.3. True Concurrency and Games as Event Structures

Event structures [20] are the so-called “true concurrency” analogue of trees: just as transition systems unfold to trees, so some models of true concurrency, such as Petri nets and asynchronous transition systems, unfold to event structures [21]. Similarly, in the same way that *sequential* games can be represented by *trees*, certain *concurrent* games can be represented by *event structures*—where plays in this much more general setting determine *partial* instead of *total* orders of moves; cf., see in [22,23].

An *event structure* is a triple (E, \leq, Con) , consisting of a set E of *events* that are partially ordered by \leq , the *causal dependency relation*, and a nonempty *consistency relation* Con over finite subsets of E , which satisfy four conditions:

1. $\{e' : e' \leq e \text{ is finite for all } e \in E\}$,
2. $\{e\} \in \text{Con}$ for all $e \in E$,
3. $Y \subseteq X \in \text{Con}$ implies $Y \in \text{Con}$, and
4. $X \in \text{Con}$ and $e \leq e' \in X$ implies $X \cup \{e\} \in \text{Con}$.

The states of an event structures E are called *configurations*, denoted by $\mathcal{C}(E)$, and consist of those subsets $x \subseteq E$ which are both

- *consistent*: $\forall X \subseteq x. X \text{ is finite implies } X \in \text{Con}$; and
- *down-closed*: $\forall e, e'. e' \leq e \in x \text{ implies } e' \in x$.

Configurations can be finite or infinite. Concurrency, then, is naturally modelled as follows: two events, say e and e' , which are both consistent and incomparable with respect to causal dependency are seen as *concurrent*, and assumed to be independent in that they can happen in parallel. In a game-theoretic context, it also means that such two events can be played or executed in parallel.

In the context of concurrent games as event structures, we consider only two players, named Player (the system) and Opponent (the environment), who own disjoint sets of events which they can play (execute). They do so asynchronously in an attempt to reach a state in their “winning set” of configurations. In these games, players are allowed to execute an event, say e , only if all events on which e causally depends have been executed. In this setting, events do not have a Boolean or otherwise interpretation; they are simply available actions that a player can execute—and informally are intended to represent observable events in a computer system. The Nash equilibrium of concurrent games as event structures have been studied in the past, and fully characterised in [23] for two-player general-sum games with players’ goals given by Borel sets of winning configurations. However, their main application is in the field of formal semantics for programming languages and logical systems; cf, see in [22,24,25].

4.4. Extensive-Form Games

Games in extensive-form are the canonical game-theoretic model to account for the strategic interactions that result if decisions are made in a prescribed order. Their use in the game-theoretic literature is ubiquitous. In this section, we show that partial order games constitute a concisely represented class of extensive-form games.

4.4.1. Partial Order Games as Extensive Games of Imperfect Information

Given a topological sorting of the decision nodes, there is a natural transformation of partial order games to extensive form games of imperfect information. Let the DAG of the partial order game be given by (V, E) , where $V = \{v_1, \dots, v_{|V|}\}$, and the utility function for each player i by $u_i: A_{v_1} \times \dots \times A_{v_{|V|}} \rightarrow \mathbb{R}$. First, assume a topological order τ to linearise the DAG, which, for ease of presentation, we assume to be $\tau = (v_1, \dots, v_{|V|})$. Figure 6 illustrates this construction for the partial order game in Figure 5 under topological order τ .

The extensive-form game of imperfect information representing the partial order game is then defined for the same set N of players. We define the game tree (V', E') such that V' is the set of all prefixes of terminal histories in $A_{v_1} \times \dots \times A_{v_{|V|}}$, with $()$ being the root node. The player active at each history/vertex $\mathbf{h}_{v_k} = (a_{v_1}, \dots, a_{v_k})$ in V' , we identify with the player that is active at v_{k+1} in the partial order game. At each of these vertices/histories this player has $A_{v_{k+1}}$ as the set of actions to chose from. In particular, i_{v_1} is the player active at $()$ and has A_{v_1} as action set. Accordingly, all prefixes/histories of the same length are assigned to the same player and each of them have the same action set. For $v, w \in V'$, we then have $vE'w$ if and only if $w = va$ for some $a \in A_v$, as expected.

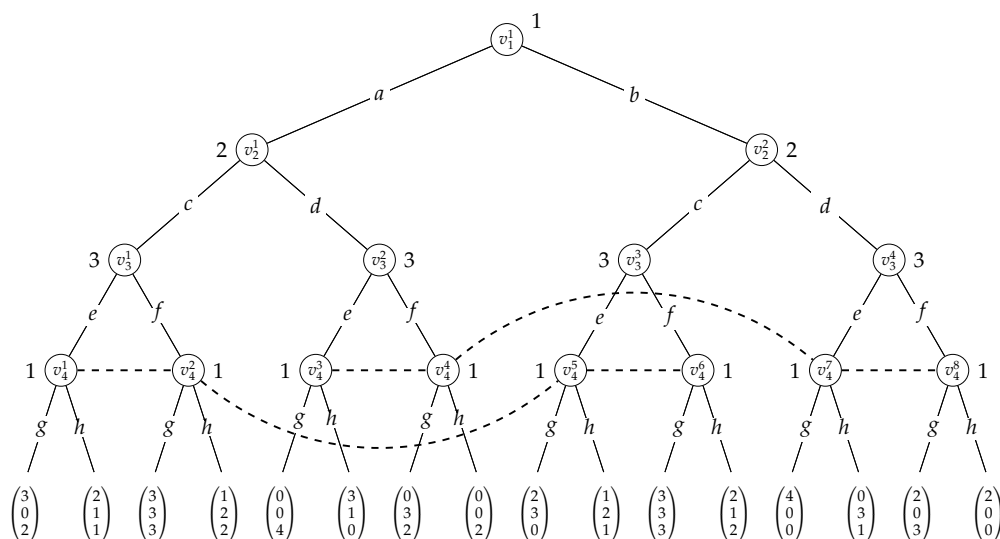


Figure 6. Extensive-form game of imperfect information representing the partial order game in Figure 5 assuming topological order (v_1, v_2, v_3, v_4) . Here, for instance, $v_1^1 = ()$, $v_3^2 = ad$, and $v_4^4 = adf$. The dashed lines connecting vertices indicate the information sets.

Every leaf/terminal history $\mathbf{h} = (a_{v_1}, \dots, a_{v_{|V|}})$ and player i we associate with the utility value $u_i^{\mathbf{h}} = u_i(a_{v_1}, \dots, a_{v_{|V|}})$, where u_i denotes player i 's utility function in the partial order game. Note that in our construction, every terminal history corresponds with a full action profile.

The definition of the information sets is crucial. For every internal vertex/history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$, we formally define these such that

$$[(a_{v_1}, \dots, a_{v_k})] = \{(a'_{v_1}, \dots, a'_{v_k}) : a'_y = a_y \text{ for all parents } y \text{ of } v_{k+1}\}.$$

Intuitively, a player can only distinguish a vertex/history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$ where it is active from those vertices/histories of the same length that differ on the choice of action in at least one if its parents in the partial order game. All other vertices/histories of the same length belong to the same information set. For an example, consider Figure 6, where vertices $v_4^2 = acf$ and $v_4^5 = bce$ are in the same information set for player 1. This is because, in the underlying partial order game (depicted in Figure 5), vertex v_2 is the only parent of v_4 and the histories acf and bce both specify action c for v_2 . Finally, we define the set of player i 's information sets as

$$\mathcal{I}_i = \{[v'] : v' \in V' \text{ and } i \text{ is active at } v'\}.$$

To see that the extensive-form game defined through this transformation is strategically equivalent to the original partial order game, first consider an arbitrary vertex v_j along with its parents Y in the partial order game. Together with v_j itself, each profile \mathbf{a}_Y in \mathbf{A}_Y defines a unique information set for i defined as

$$I(v_j, \mathbf{a}_Y) = \{\mathbf{a}' \in A_{v_1} \times \dots \times A_{v_{j-1}} : \mathbf{a}_Y = \mathbf{a}'_Y\}.$$

Let θ be the function that maps each pair (v_j, \mathbf{a}_Y) to information set $I(v_j, \mathbf{a}_Y)$.

Observe that defined thus, θ is both injective and surjective. We now find that we can extend θ to a function $\hat{\theta}$ that maps each conditional strategy $\mathbf{f}_i = (f_{v_{i_1}}, \dots, f_{v_{i_k}})$ in the partial order game to a strategy $\hat{\mathbf{f}}_i : \mathcal{I}_i \rightarrow A_i$ for player i in the extensive-form game. To do so, define $\hat{\mathbf{f}}_i$ such that for each each information set $I \in \mathcal{I}_i$ with $\theta^{-1}(I) = (v_j, \mathbf{a}_Y)$

$$\hat{\mathbf{f}}_i(I) = f_{v_j}(\mathbf{a}_Y).$$

Observe that the extended function $\hat{\theta}$ is also bijective. Moreover, some reflection reveals that the full action profile $\mathbf{a}^f = (a_{v_1}^f, \dots, a_{v_{|V|}}^f)$ determined by conditional strategy profile $\mathbf{f} = (f_1, \dots, f_{|N|})$ in the partial order game will be identical to the one determined by the strategy profile $\hat{\mathbf{f}} = (\hat{\mathbf{f}}_1, \dots, \hat{\mathbf{f}}_{|N|})$ in the extensive-form game. Having defined the players' utilities in the extensive-form game as we did, we may conclude that the partial order game is strategically equivalent to the extensive game of imperfect information. Formally:

Proposition 1. *A partial order game and the extensive-form game obtained from it on basis of a given topological sorting, as described in this section, are strategically equivalent.*

This result means that, with respect to game theoretic analysis, partial order games are in a precise technical sense (i.e., with respect to strategic equivalence), no more expressive than extensive form games: any scenario we can model with a partial order game can also be modelled with an extensive form game. However, this does not mean that partial order games are redundant. The translation from partial order game to extensive form game comes at the expense of a blow-up in the size of the game: if $|A_v| \geq 2$ for every vertex v , then $V' = \Omega(2^{|V|})$. The practical upshot of this is that there are situations that we can capture using the partial order model that would be infeasible to capture with the extensive form. In addition, we argue that the partial order representation can in some cases be much more comprehensible than the extensive form: compare the partial order game in Figure 5 with its extensive form representation in Figure 6. Understanding what is going on (and in particular, the informational dependencies present) in the latter representation requires much more work than in the former.

Note that, for total-order games, the construction yields (finite) extensive-form games of *perfect information* (see Figures 7 and 8 below). This can easily be appreciated by recalling that a total order game allows for only one topological order, say $(v_1, \dots, v_{|V|})$. Thus, for every $1 \leq j \leq |V|$ we have that the parents of v_j are all vertices in $Y = \{v_1, \dots, v_{j-1}\}$. However, then, obviously, $I(v_j, \mathbf{a}_Y) = \{\mathbf{a}' \in A_{v_1} \times \dots \times A_{v_{j-1}} : \mathbf{a}_Y = \mathbf{a}'_Y\}$ will be a singleton for every action profile \mathbf{a}_Y , and the resulting game one of perfect information. As finite extensive-form games of perfect information can be solved by *backwards induction* (cf. e.g., in [4], Chapter 7), it follows as a corollary that total-order games always admit Nash equilibria in general.

Proposition 2. *Total-order games always have at least one Nash equilibrium.*

We conclude this section with a remark about topological sortings. The DAG of a partial order game may not have a unique topological sorting over its nodes and, under

the transformation described above, different topological sortings may lead to different extensive-form games. This is illustrated by Figures 6 and 9, which both depict extensive-form games obtained from the partial order game in our first example in Figure 5. These extensive-form games we obtain using our transformation for the topological sortings (v_1, v_2, v_3, v_4) and (v_1, v_2, v_4, v_3) , respectively, and are clearly distinct. Yet, they are strategically equivalent. A formal proof that any two extensive-form games that are obtained using our transformations for different topological orders will be strategically equivalent is beyond the scope of this paper. It suffices to say that it is due to the so-called *Interchange of Moves* principle, which is one of the four Thompson transformations [26] and informally captures the idea that in extensive-form games of imperfect information “the order of play is immaterial if one player does not have any information about the other player’s action when making his choice” (see in [5], page 224).

4.4.2. Partial Order Games as a Concise Representation of Extensive-Form Games

As we noted above, the transformation of partial order games to extensive-form games presented in the previous section may give rise to an exponential blow-up. Reasoning conversely, this raises the claim that partial order games could be seen as providing a *concise representation* of extensive form games. In this section, we argue that *if the utilities in partial order games are represented by Boolean circuits*, then this claim is supported.

In Section 2.3, we recalled the well-known fact that transforming an extensive-form game to a normal-form game leads to an exponential blow-up. One reason for this is that, in extensive-form games, a player’s utilities are represented by an association of utility values to leaf nodes/terminal histories. One leaf node, however, may be reached by playing different strategy profiles, or even different action profiles, whereas in normal-form games utilities are specified for each strategy/action profile separately.

In partial order games, the players’ utilities are also modelled as an association of action profiles (not strategy profiles) and utility values. We find, however, that an exponential blow-up can be avoided when the players’ utility functions are represented by Boolean circuits.

The main idea behind this can be explained by means of an example. Consider again the extensive form in Figure 1, where each action has already been labelled with its local index in binary. For each internal vertex/non-terminal history v with action set A_v , we introduce $\lceil \log_2 |A_v| \rceil$ Boolean variables $x_1^v, \dots, x_{\lceil \log_2 |A_v| \rceil}^v$. In the example, we introduced variables x for v_0 , y and y' for v_1 , z for v_2 , and x' for v_3 . A truth-value assignment α to these variables for all vertices then defines an action profile $\mathbf{a}^\alpha = (a_{v_1}^\alpha, \dots, a_{v_{|V|}}^\alpha)$ such that a_j^α is the action in A_v with numerical index $num(\alpha(x_1), \dots, \alpha(x_{\lceil \log_2 |A_v| \rceil}))$, for each vertex v . In our example, for instance, the assignment α^* that sets x and y' to false and y, z , and x' to true, corresponds to the action profile (a_0, b_{10}, c_1, d_1) . With all utilities in our example being either 0 or 1, we can now associate a Boolean formula φ_{u_i} that characterises each player’s utility function, in the sense that assignment α satisfies φ_{u_i} if and only if $u_i(\mathbf{a}^\alpha) = 1$. In our example, such a formula φ_{u_1} representing player 1’s binary utility function could then for instance be obtained by first defining recursively a formula φ_v for each vertex v as follows. First, for each leaf node/terminal history \mathbf{h} , we set $\varphi_{\mathbf{h}} = \top$, if $u_i^{\mathbf{h}} = 1$, and $\varphi_{\mathbf{h}} = \perp$, if $u_i^{\mathbf{h}} = 0$. In our example, we thus have, for instance, $\varphi_{a_0 b_{01}} = \perp$ and $\varphi_{a_1 c_1 d_1} = \top$. For every internal vertex/non-terminal history v , furthermore, we then set

$$\varphi_v = \bigvee_{a \in A_v} (\chi_a \wedge \varphi_{va}),$$

where $\chi_a = \tilde{x}_1^v \wedge \dots \wedge \tilde{x}_{\lceil \log_2 |A_v| \rceil}^v$ characterises the local index k of a , that is $\tilde{x}_j^v = x_j^v$, if $b_j = 1$, and $\tilde{x}_j^v = \neg x_j^v$ if $b_j = 0$, where $bin(k) = b_1 \dots b_{\lceil \log_2 |A_v| \rceil}$. Then, set $\varphi_i = \varphi_{\cdot}$. Thus, in our example we get, subsequently,

$$\begin{aligned} \varphi_{v_3} &= (\bar{x}' \wedge \perp) \vee (x' \wedge \top) \\ \varphi_{v_2} &= (\bar{z} \wedge \perp) \vee (z \wedge \varphi_{v_3}) \\ \varphi_{v_1} &= (\bar{y}\bar{y}' \wedge \top) \vee (\bar{y}y' \wedge \perp) \vee (yy' \wedge \top) \vee (yy' \wedge \perp) \\ \varphi_{v_0} &= (\bar{x} \wedge \varphi_{v_1}) \vee (x \wedge \varphi_{v_2}) \end{aligned}$$

Finally, we obtain $\varphi_{u_1} = \varphi_{v_0}$. In our example, we now find, for instance, that assignment α^* does not satisfy φ_{u_i} and that $u_i(\mathbf{a}^*) = 0$.

This procedure is general and can be applied to every extensive-form game. The crucial thing to observe is that each variable x occurs at most $2^{\lceil \log_2 |A_v| \rceil} = O(|A_v|)$ times in φ_{u_i} . Therefore, the size of φ_{u_i} is still polynomial in the size of game and, thus, there is a polynomial-sized Boolean circuit C_{u_i} representing the Boolean function $\beta_{\varphi_{u_i}}$. Finally, any rational-valued utility function $u_i: A_{v_1} \times \dots \times A_{v_{|V|}} \rightarrow \mathbb{Q}$ can be represented by a linear combination of such circuits, and we may conclude that u_i can be represented by a Boolean circuit whose size is polynomial in the size of the extensive-form game, giving us the following lemma.

Lemma 3. *Let $u_i: S_1 \times \dots \times S_{|N|} \rightarrow \mathbb{Q}$ be the utility function of a player i in an extensive-form game. Then, u_i can be represented by a Boolean circuit that is of a size polynomial in the size of the extensive-form game.*

Recall that an extensive-form game and a partial order game are said to be strategically equivalent if they give rise to strategically equivalent normal-form games. We now find that, *provided that utilities may be represented by Boolean circuits*, every extensive-form game can be represented by a strategically equivalent partial order game *without giving rise to more than an at most a polynomial blowup*.

Proposition 3. *For every extensive-form game, there is a strategically equivalent partial order game whose size is at most polynomially larger than that of the extensive-form game.*

Sketch of Proof. Let an extensive form for players in a set N be based on a tree (V, E) . Let \mathcal{I} be the information sets of this game. We construct a partial order game on the trivial DAG (\mathcal{I}, \emptyset) , which is obviously polynomial in the size of the extensive-form game. We associate with each information set I in \mathcal{I} the same player and the same action set A_I as in the extensive-form game. For every player i with $\mathcal{I}_i = \{I_1, \dots, I_k\}$, we can define a bijection that maps every strategy $s_i: \mathcal{I}_i \rightarrow A_i$ for i in the extensive-form game to a strategy $\hat{f}_i = (\hat{f}_{I_1}, \dots, \hat{f}_{I_k})$ for player i in the partial order game such that \hat{f}_{I_j} is the strategy that maps the empty sequence $()$ to action $s_i(I_j)$. By virtue of Lemma 3 we may assume the utility function of each player in the extensive-form game can be represented by a polynomially sized Boolean circuit. Accordingly, the the size of the whole partial order game constructed thus is polynomial in the size of the extensive-form game. Conclude by observing that the extensive-form game and the partial order game induce strategically equivalent normal-form games, as desired. \square

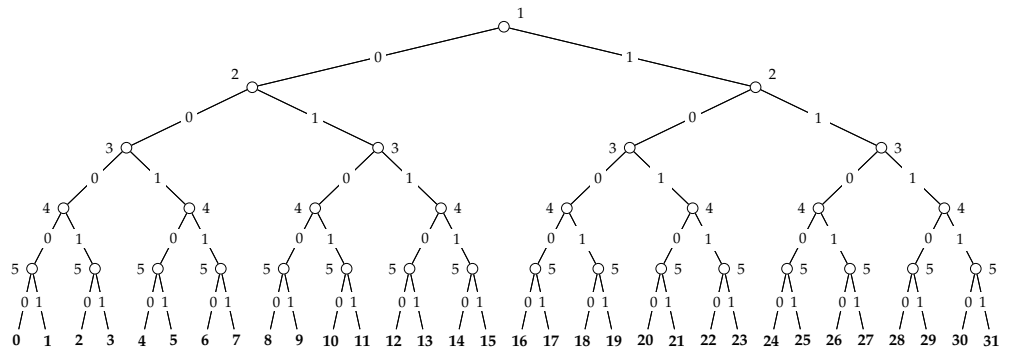


Figure 7. Extensive-form game of perfect information illustrating Proposition 4. for every $0 \leq n \leq 31$, we have \mathbf{n} denote the vector $(n, n, n, n, n)^T$.

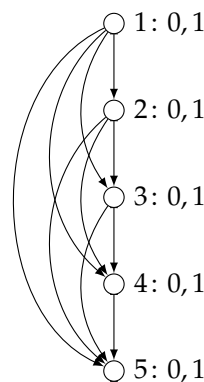


Figure 8. The DAG for the partial order game with five players illustrating Proposition 4.

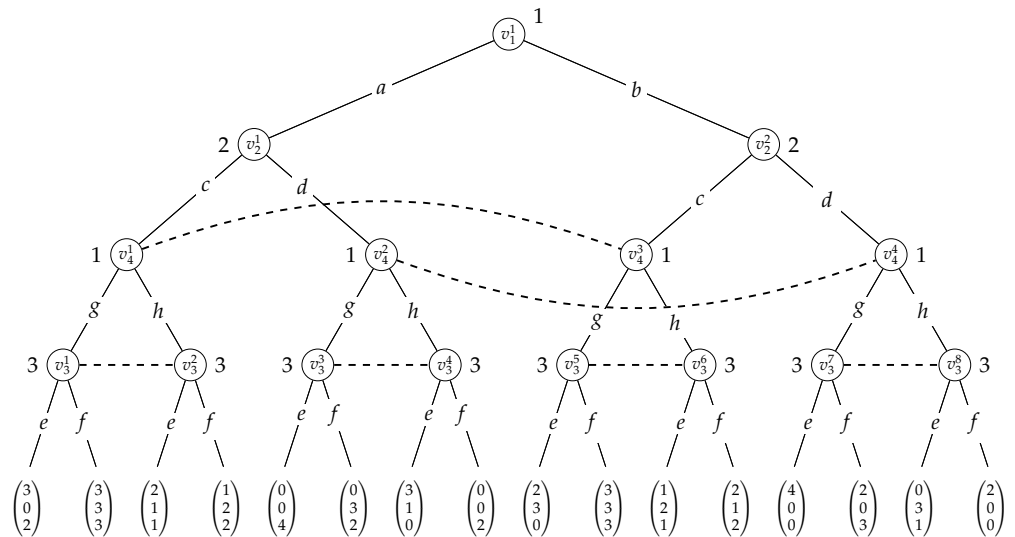


Figure 9. Extensive-form game of imperfect information representing the partial order game in Figure 5 assuming topological ordering (v_1, v_2, v_4, v_3) . The dashed lines connecting vertices indicate the information sets.

Proposition 3 shows that every extensive-form game of perfect-information can be represented by a partial order game at the cost of an at most polynomial blow-up. In some cases, moreover, we find that partial order games may be exponentially smaller than any extensive-form game of perfect information that is strategically equivalent. Consider the following extensive-form game with n players, ordered from 1 to n . Every player has two actions, 0 and 1, at their disposal, but can make their choice dependent on the players that occur before them in the ordering. Each strategy profile \mathbf{s} gives rise to a unique terminal

history $\mathbf{h}^s = (b_1, \dots, b_n)$ of actions in $\{0, 1\}$, where each player i plays b_i . Accordingly, there are exactly $|\{0, 1\}^n| = 2^n$ terminal histories or leaves in this extensive-form game. Let the preferences for each player i at each of these terminal histories $\mathbf{h} = (b_1, \dots, b_n)$ be given by the utility function such that $u_i^{\mathbf{h}} = \text{num}(b_1, \dots, b_n)$, where $\text{num}(b_1, \dots, b_n)$ is the numerical value of the sequence b_1, \dots, b_n conceived as an integer in binary. For instance, $\text{num}(01101) = 13$. Then, for each player, every terminal history or leaf yields a unique utility value. See Figure 7 for an illustration of this game for $n = 5$. To represent this game, an extensive-form game needs to have at least 2^n leaf nodes to account for all the different payoffs a player may get, and thus is of size at least exponential in the number of players.

By contrast, one can represent this game as a partial order game on a transitive graph (V, E) , where $V = \{v_1, \dots, v_n\}$ and $v_i E v_j$ if and only if $i < j$. Let player i be assigned to vertex v_i and $A_{v_i} = \{0, 1\}$ for all $1 \leq i \leq n$. For each player i , the utility function $u_i(b_1, \dots, b_n) = \text{num}(b_1, \dots, b_n)$ can clearly be represented by a Boolean circuit with V input variables, and as many output variables, with the m th input gate immediately leading to the m th output gate with no intermediate logic gates. See Figure 8 for an illustration of the partial order game representing the extensive-form game in Figure 7. Therefore, we have the following result establishing that partial order games can be seen as presenting a concise representation of extensive-form games. Moreover, this observation still holds if attention is restricted to games of perfect information.

Proposition 4. *There exist partial order games for which every strategically equivalent representation as an extensive-form game of imperfect information is at least exponentially as large.*

5. Nash Equilibria

We saw in Section 2.2 how every partial order game defines a normal-form game. Accordingly, partial order games are amenable to game-theoretic analysis using the standard non-cooperative solution concepts that are available for normal-form games. In this section, we consider several complexity problems surrounding Nash equilibrium in partial order games.

We have the following lemma that will be useful for proving the complexity results in this section. It states that, if a player has a profitable deviation to a conditional strategy, then they also have a profitable deviation to an unconditional strategy. In other words, if a player has a best response, then they also have an unconditional best response (This is reminiscent of mixed strategies for (finite) normal-form games, where, due to the linearity of expected utility, a player having a best response implies their having a pure best response).

Lemma 4. *Let $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_n)$ be a profile of conditional strategies. Then, if there is some strategy g_i for player i with $u_i(\mathbf{f}_{-i}, \mathbf{g}_i) > u_i(\mathbf{f})$, there also is an unconditional strategy \mathbf{g}'_i such that $u_i(\mathbf{f}_{-i}, \mathbf{g}'_i) > u_i(\mathbf{f})$. Therefore, $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a Nash equilibrium if and only if $u_i(\mathbf{f}) \geq u_i(\mathbf{f}_{-i}, \mathbf{g}'_i)$ for all players i and unconditional strategies \mathbf{g}'_i for i .*

Proof. Assume that there is some strategy g_i for player i with $u_i(\mathbf{f}_{-i}, \mathbf{g}_i) > u_i(\mathbf{f})$. Let v_{i_1}, \dots, v_{i_k} be the vertices assigned to player i . Moreover, let $\mathbf{a}^{(\mathbf{f}_{-i}, \mathbf{g}_i)} = (a_{v_1}^*, \dots, a_{v_{|V|}}^*)$. Now define $\mathbf{g}'_i = (g'_{v_{i_1}}, \dots, g'_{v_{i_k}})$ such that for every $1 \leq j \leq k$, and every profile $(a_{u_1}, \dots, a_{u_\ell})$, where u_1, \dots, u_ℓ are the parents of v_{i_j} ,

$$g'_{v_{i_j}}(a_{u_1}, \dots, a_{u_\ell}) = g_{v_{i_j}}(a_{u_1}^*, \dots, a_{u_\ell}^*).$$

Thus, \mathbf{g}'_i is clearly an unconditional strategy for player i . Let $\mathbf{a}^{(\mathbf{f}_{-i}, \mathbf{g}'_i)} = (a_{v_1}^{**}, \dots, a_{v_{|V|}}^{**})$. By a straightforward induction on $\text{depth}(w)$ it can then easily be shown that $a_w^{**} = a_w^*$ for every vertex w . And thus, $u_i(\mathbf{f}_{-i}, \mathbf{g}'_i) > u_i(\mathbf{f})$, giving us the result. \square

The key point about this lemma is that unconditional strategies are *small*, in the sense that unconditional strategies can be represented by Boolean circuits whose size is

polynomial in the set of actions (see Section 3.1, above). Thus, when we are considering whether a player has a beneficial deviation, we can without loss of generality restrict our attention to small strategies. This has implications for the complexity of the decision problems we consider.

The dependency graph provides a concise representation of the sets of conditional strategies available to the players. A single strategy f_v for the player i playing at vertex v , however, has to take into account all profiles $(a_{y_1}, \dots, a_{y_k})$ in $A_{y_1} \times \dots \times A_{y_k}$, where y_1, \dots, y_k are the parents of v . The number of these profiles tends to be exponential in k , the number of parents, as in all non-trivial cases generally $|A_v| \geq 2$. Similarly, a naive representation of the player’s utilities for the different action profiles tends to be exponential in the number of actions available to the players in the game. However, as the utilities of players are represented by Boolean circuits, there is a straightforward polynomial transformation of Boolean games and Boolean partial order games to general partial order games. This enables us to leverage hardness results for Boolean and partial order Boolean games to obtain hardness results for general partial order games.

We first consider the decision problem of determining whether a given strategy profile is a Nash equilibrium for a given partial order game. Here, we assume that the utility function u_i of each player is represented by a Boolean circuit \mathbb{C}_{u_i} and the strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is represented by a sequence of $|V|$ Boolean circuits $\mathbb{C}_{f_{v_1}}, \dots, \mathbb{C}_{f_{v_{|V|}}}$.

IS-NASH

Given: Partial order game G and strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$
Problem: Is \mathbf{f} a Nash equilibrium of G ?

We find that IS-NASH is intractable for partial order games.

Theorem 1. IS-NASH is coNP-complete. The problem remains coNP-hard for empty-order games and total-order games.

Proof. For membership in coNP, let partial order game G and conditional strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ be given. We can guess a player i and a strategy $\mathbf{g}_i = (g_{v_1}, \dots, g_{v_m})$, where v_1, \dots, v_m are the vertices assigned to i . By virtue of Lemma 4, we may assume that strategies g_{v_1}, \dots, g_{v_m} are all *unconditional* strategies. As we saw in Section 3.1, the Boolean circuits $\mathbb{C}_{g_{v_1}}, \dots, \mathbb{C}_{g_{v_m}}$ can therefore each be assumed to be of size polynomial in $|A|$. Now, Lemma 2 allows us to find in polynomial time the utilities $u_i(\mathbf{f}_{-i}, \mathbf{g}_i)$ and $u_i(\mathbf{f}_{-i}, \mathbf{f}_i)$. Then, $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is *not* a Nash equilibrium if and only if $u_i(\mathbf{f}_{-i}, \mathbf{g}_i) > u_i(\mathbf{f}_{-i}, \mathbf{f}_i)$, and so we may conclude that IS-NASH is in coNP.

For coNP-hardness, we reduce from IS-NASH for Boolean partial order games, which is known to be coNP-hard (see in [1], Proposition 1). Thus, let an instance of IS-NASH for Boolean partial order games be given by a Boolean game $(N, \Phi, \Phi_1, \dots, \Phi_{|N|}, \gamma_1, \dots, \gamma_{|N|})$, a dependency graph (Φ, E) , and a strategy profile $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_{|N|})$, where for each player i strategy \mathbf{f}_i is given by a sequence of choice equations $p_1: \varphi_1$ through $p_m: \varphi_m$, where $\Phi_i = \{p_1, \dots, p_m\}$. In Section 4.1, we argued how this Boolean partial order game *coincides with* a partial order game with a slightly different representation of strategies and utilities. A similar remark concerns the strategy profile $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_{|N|})$. As we found that the transformation of the representation of the game as a Boolean partial order game to the representation of the same game as a partial order game can be effected in polynomial time, we obtain our result. Moreover, because Boolean games are a special type of empty-order games, the problem remains hard for empty-order games as well.

To see that IS NASH also remains coNP-hard for total-order games, we adapt the proof for the IS NASH problem for Boolean games as presented in (Wooldridge et al. [15], Proposition 1). We reduce from the complement of SATISFIABILITY, the problem of determining if a given Boolean formula φ is satisfiable. To this end, let φ a Boolean formula over propositional variables p_1, \dots, p_m . We construct a total-order Boolean game with one player i controlling p_1, \dots, p_m along with an additional variable p_0 . The dependency graph (V, E)

is then given by $V = \{v_{p_0}, \dots, v_{p_m}\}$ with relation E such that $p_j E p_k$ if and only if $j < k$. Player i has as goal $\gamma_i = \varphi \wedge p_0$. Now, consider the strategy profile $\mathbf{f}^0 = (f_{v_{p_0}}^0, f_{v_{p_1}}^0, \dots, f_{v_{p_m}}^0)$ that sets all variables to false, that is, \mathbf{f}^0 is given by choice equations $p_j: \perp$ for all $0 \leq j \leq m$. Now observe that, as we are dealing with a one-player game, there is a natural surjection that maps each of player i 's strategies \mathbf{f} to valuation $\alpha_{\mathbf{f}}: \{p_0, \dots, p_m\} \rightarrow \{0, 1\}$. It can then easily be appreciated that \mathbf{f}^0 is a Nash equilibrium in the game constructed if and only if φ is satisfiable, as desired. \square

Another canonical problem is more general, in that it asks whether a partial order game has any Nash equilibria at all, as opposed to whether or not a specific strategy profile is a Nash equilibrium.

NON-EMPTINESS

Given: Partial order game G

Problem: Does G have a Nash equilibrium?

In view of Proposition 2, NON-EMPTINESS is vacuous for total-order games, as in this class of games Nash equilibria are guaranteed to exist. The problem is considerably more difficult, namely NEXPTIME-complete, for general partial order games. It also seems a fair conclusion to draw from this contrast that increase in computational complexity arises from the structure of the dependency graph.

Theorem 2. NON-EMPTINESS is NEXPTIME-complete.

Proof. A NEXPTIME algorithm to decide NON-EMPTINESS can be designed along the following lines. First, guess a strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$. Given that strategies f_v are given by a Boolean circuit \mathbb{C}_{f_v} , this can be achieved in time not more than exponential in $|A|$, the size of the set of actions. Second, to decide whether $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a Nash equilibrium, by virtue of Lemma 4, it suffices to check, for all players i and all unconditional strategies $\mathbf{g}_i = (g_{v_1}, \dots, g_{v_{|V_i|}})$ for i , that $u_i(\mathbf{f}_{-i}, \mathbf{f}_i) \geq u_i(\mathbf{f}_{-i}, \mathbf{g}_i)$. For player i , there are $|A_{v_1} \times \dots \times A_{v_{|V_i|}}|$ unconditional strategies which is upper bounded by $|A|^{|V|}$. Moreover, as we have seen in Section 3.1, each of these unconditional strategies can be represented by a sequence of Boolean circuits of size polynomial in $|A|$. Furthermore, the action profiles $\mathbf{a}^{(\mathbf{f}_{-i}, \mathbf{f}_i)}$ and $\mathbf{a}^{(\mathbf{f}_{-i}, \mathbf{g}_i)}$ can be computed in time exponential in $|A|$, by virtue of Lemma 1 and each circuit involved being at most exponential in the size of $|A|$. Finally, on the basis of Lemma 2, we can check in time polynomial in $|A|$ the utilities for player i on action profiles $\mathbf{a}^{(\mathbf{f}_{-i}, \mathbf{f}_i)}$ and $\mathbf{a}^{(\mathbf{f}_{-i}, \mathbf{g}_i)}$, and therefore also whether $u_i(\mathbf{f}_{-i}, \mathbf{f}_i) \geq u_i(\mathbf{f}_{-i}, \mathbf{g}_i)$. Altogether, the algorithm runs in non-deterministic exponential time.

A proof of NEXPTIME-hardness of NON-EMPTINESS can be achieved by a reduction from the NON-EMPTINESS problem for Boolean partial order games. We rely here on the same direct reduction as in the proof of Theorem 1, above. NEXPTIME-hardness for Boolean partial order games was established by [1], giving us the result. \square

If we restrict attention to empty-order games, however, NON-EMPTINESS has considerably lower computational costs, even though the problem still remains Σ_2^P -hard. Key to this result is the observation that for empty-order games all strategies are unconditional, and, thus, they can be represented by Boolean circuits of polynomial size (see Section 3.1).

Theorem 3. For empty-order games, NON-EMPTINESS is Σ_2^P -complete.

Proof. To see that NON-EMPTINESS is in Σ_2^P , recall that $\Sigma_2^P = \text{NP}^{\text{coNP}}$, that is, the set of problems that can be solved in polynomial time on a non-deterministic Turing machine with a coNP-oracle. Furthermore, recall that strategy profiles in empty-order games are unconditional and are of the form $f_v: \{()\} \rightarrow A_v$. Moreover, they can be represented by

Boolean circuits of polynomial size (see Section 11, above). Accordingly, given an empty-order game, we can guess an unconditional strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$, and consult the coNP-oracle to check whether \mathbf{f} is a Nash equilibrium. Theorem 1 guarantees that the latter is feasible.

For Σ_2^P -hardness, recall that Boolean games constitute a subclass of empty-order partial order games. Σ_2^P -hardness then follows immediately from NON-EMPTYNESS being Σ_2^P -hard for Boolean games (see in [13], Proposition 5). \square

Recall that in partial order games, strategy profiles and action profiles are essentially different objects. As a natural counterpart to the IS NASH problem, we therefore now consider the decision problem whether a given action profile is sustained by a Nash equilibrium in a partial order game.

IS NASH ACTIONS

Given: Partial order game G and action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$
Problem: Is \mathbf{a} sustained by a Nash equilibrium?

In sharp contrast to coNP-completeness of IS NASH, we find that IS NASH ACTIONS is NEXPTIME-complete. From the perspective of computational complexity, IS NASH ACTIONS appears to be more kindred to the NON-EMPTYNESS problem for partial order games. In this connection, it is worth observing that the proof of Bradfield et al. regarding the NEXPTIME-completeness of NON-EMPTYNESS for Boolean partial order games in [1] relied on a reduction from DEPENDENCY QUANTIFIER BOOLEAN FORMULA GAME, which is defined as follows.

An instance of DQBF is a tuple $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$, where φ is a Boolean formula φ and $\{X_1, X_2, Y_1, Y_2\}$ a partition of the propositional variables over which φ is defined. DQBF then concerns the following game with three players— B ('Black'), W_1 ('White 1'), and W_2 ('White 2')—where B forms one team and W_1 and W_2 another team W . Player B chooses an assignment for the variables in $X_1 \cup X_2 = \{x_1^1, \dots, x_{|X_1|}^1\} \cup \{x_1^2, \dots, x_{|X_2|}^2\}$, player W_1 for those in $Y_1 = \{y_1^1, \dots, y_{|Y_1|}^1\}$, and player W_2 for those in $Y_2 = \{y_1^2, \dots, y_{|Y_2|}^2\}$. Player B chooses first, then W_1 and W_2 choose, on the understanding that W_1 can only see the assignment B chooses for W_1 and W_2 only the assignment B chooses for Y_2 . Team B aims to make φ true, whereas team W 's goal is to make φ false. If the overall assignment for $X_1 \cup X_2 \cup Y_1 \cup Y_2$ satisfies φ , then B wins, otherwise W . A positive instance of DQBF is when team W has a winning strategy. DEPENDENCY QUANTIFIER BOOLEAN FORMULA GAME was shown to be NEXPTIME-complete by Hearn and Demaine in [27].

Theorem 4. IS NASH ACTIONS is NEXPTIME-complete.

Proof. For membership in NEXPTIME, let $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ be the action profile that is given as input. Recall from the proof of Theorem 2 that we can guess a strategy profile \mathbf{f} and check whether it is a Nash equilibrium in exponential time, provided that \mathbf{f} is represented by a Boolean circuit. By Lemma 1 we know we can additionally compute $\mathbf{a}^{\mathbf{f}}$ in polynomial time and check whether $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$. As the latter can also be achieved in polynomial time, we obtain our result.

For NEXPTIME-hardness, we reduce from DEPENDENCY QUANTIFIER BOOLEAN FORMULA GAME (DQBF). Given an instance $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ of DQBF we construct a partial order Boolean game with the same three players, with B controlling the variables in $X_1 \cup X_2 \cup \{x_0\}$, player W_1 those in Y_1 , and player W_2 those in Y_2 , where x_0 is a "fresh" variable not in $X_1 \cup X_2 \cup Y_1 \cup Y_2$. Let the players' goals be given by

$$\gamma_B = x_0 \wedge \varphi \qquad \gamma_{W_1} = \gamma_{W_2} = \perp$$

The dependency graph $(X_1 \cup X_2 \cup Y_1 \cup Y_2 \cup \{x_0\}, E)$ is defined such that

- (i) $x_1 E y_1$, for all $x_1 \in X_1$ and $y_1 \in Y_1$,

- (ii) $x_2 E y_2$, for all $x_2 \in X_2$ and $y_2 \in Y_2$,
- (iii) $x_0 E z$, for all $z \in X_1 \cup X_2 \cup Y_1 \cup Y_2$.

Now, consider the action profile

$$\bar{\mathbf{a}} = (\bar{x}_0, \bar{x}_1^1, \dots, \bar{x}_{|X_1|}^1, \bar{x}_1^2, \dots, \bar{x}_{|X_2|}^2, \bar{y}_1^1, \dots, \bar{y}_{|Y_1|}^1, \bar{y}_1^2, \dots, \bar{y}_{|Y_2|}^2),$$

which sets all variables, including x_0 , to false. Note that none of the three players win if $\bar{\mathbf{a}}$ is played. We are now in a position to prove that $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ is a positive instance of DQBFG if and only if $\bar{\mathbf{a}}$ is sustained by a Nash equilibrium in the partial order Boolean game constructed.

First assume that $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ is positive instance of DQBFG. Then, the white team has a joint winning strategy in the original DQBFG-game given by Boolean functions β_1 and β_2 on the variables X_1 and X_2 , respectively. Then, define strategies $\mathbf{f}_{W_1}^*$ and $\mathbf{f}_{W_2}^*$ for W_1 and W_2 , respectively, in the Boolean partial order game, that are given by the following choice equations for $i = 1, 2$ and $j = 1, \dots, |Y_i|$:

$$y_j^i: (x_0 \rightarrow \varphi_{\beta_i}(x_1^i, \dots, x_{|X_i|}^i)) \wedge (\bar{x}_0 \rightarrow \perp)$$

As the dependency graph (V, E) respects the information dependencies of the DQBFG-instance, observe that $\mathbf{f}_{W_1}^*$ and $\mathbf{f}_{W_2}^*$ together embody a winning strategy to render φ false, if x_0 is played. Let B 's unconditional strategy \mathbf{f}_B^* be defined by the choice equations of the form $x_j^i: \perp$, setting x_j^i to false for all for $i = 1, 2$ and $j = 1, \dots, |X_i|$.

Observe that, defined thus, $\mathbf{a}^{\mathbf{f}^*} = \bar{\mathbf{a}}$. Also note that $\mathbf{f}^* = (\mathbf{f}_B^*, \mathbf{f}_{W_1}^*, \mathbf{f}_{W_2}^*)$ is a Nash equilibrium. As x_0 will be set to false, all players lose when \mathbf{f}^* is played, but none has an incentive to deviate either. Player B could only hope to win by setting at least x_0 to true. If so, however, W_1 and W_2 are playing a winning strategy against B , dashing all the latter's hopes to win after all. Players W_1 and W_2 will lose, no matter which strategy they adopt. As a consequence, they do not want to deviate either, and we may conclude that \mathbf{f}^* is a Nash equilibrium sustaining action profile $\bar{\mathbf{a}}$.

Finally, assume that $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ is a negative instance of DQBFG. Then, team W does not have a winning strategy, meaning that for every strategy $\mathbf{f}_W = (\mathbf{f}_{W_1}, \mathbf{f}_{W_2})$, player B has a best response \mathbf{f}'_B such that B wins under $\mathbf{f} = (\mathbf{f}_{W_1}, \mathbf{f}_{W_2}, \mathbf{f}'_B)$. Now, consider the partial order Boolean game and an arbitrary strategy profile $\bar{\mathbf{f}}$ with $\mathbf{a}^{\bar{\mathbf{f}}} = \bar{\mathbf{a}}$. Some reflection reveals that under these circumstances, player B has a strategy \mathbf{f}'_B at their disposal setting x_0 to true and incorporating a winning response to $\bar{\mathbf{f}}_W = (\bar{\mathbf{f}}_{W_1}, \bar{\mathbf{f}}_{W_2})$. Therefore, player B has an incentive to unilaterally deviate from $\bar{\mathbf{f}}$ and play \mathbf{f}'_B instead. We may conclude that $\bar{\mathbf{f}}$ is not a Nash equilibrium, as desired. \square

For the two extremal classes of empty-order games and total order games, IS NASH ACTIONS are less computationally demanding. More precisely, the problem is coNP-complete for empty-order games and PSPACE-complete for total-order games. The proof of the former statement is relatively straightforward, when one realises that in empty-order games all strategies are unconditional and that, consequently, there is a natural bijection between action profiles and strategy profiles.

Theorem 5. For empty-order games, IS NASH ACTIONS is coNP-complete.

Proof. First recall that, in empty-order games, there is a natural bijection between action profiles and strategy profiles, mapping each action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ to strategy profile $\mathbf{f}^{\mathbf{a}} = (f_{v_1}^{\mathbf{a}}, \dots, f_{v_{|V|}}^{\mathbf{a}})$, where $f_v^{\mathbf{a}}(()) = a_v$. Thus, all strategy profiles of an empty-

order game are of this form. Moreover, for arbitrary action profiles $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ and strategy profiles $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$, we have that

$$\begin{aligned} \mathbf{a} = \mathbf{a}^{\mathbf{f}} & \text{ if and only if } f_v^{a_v} = a_v, \text{ for all } v \in V \\ & \text{if and only if } \mathbf{f} = \mathbf{f}^{\mathbf{a}} \end{aligned}$$

Moreover, recall that unconditional strategies are represented by polynomial-sized Boolean circuits. Theorem 1 established that IS NASH is coNP-complete for empty-order games. Altogether, it suffices to show that, for all action profiles $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ of a empty-order game,

$$\mathbf{a} \text{ is sustained by a Nash equilibrium if and only if } \mathbf{f}^{\mathbf{a}} \text{ is a Nash equilibrium.}$$

The “if”-direction is immediate. If $\mathbf{f}^{\mathbf{a}}$ is a Nash equilibrium, then, by observing that $\mathbf{a}^{\mathbf{f}^{\mathbf{a}}} = \mathbf{a}$, action profile \mathbf{a} is sustained by some Nash equilibrium, namely $\mathbf{f}^{\mathbf{a}}$. For the “only if”-direction, assume there is some strategy profile \mathbf{f}^* with $\mathbf{a} = \mathbf{a}^{\mathbf{f}^*}$. By our earlier observation we find that $\mathbf{f}^* = \mathbf{f}^{\mathbf{a}}$. Hence, $\mathbf{f}^{\mathbf{a}}$ is a Nash equilibrium, as desired. \square

The proof of PSPACE-completeness of IS NASH ACTIONS for total-order games is considerably more involved, and we defer it to the very end of Section 7.2. where the necessary proof elements are in place. Still, we conclude this section with its statement.

Theorem 6. For total-order games, IS NASH ACTIONS is PSPACE-complete.

6. Backwards Induction

Backwards induction is the most fundamental technique for the analysis of extensive form games. The basic backward induction algorithm for extensive form games of perfect information runs in time polynomial in the size of the game tree, and computes Nash equilibrium strategy profiles (which are guaranteed to exist in games of perfect information). It is therefore very natural to ask whether approaches based on backward induction might work for partial order games. However, backward induction is not applicable in extensive form games of imperfect information, and as partial order games correspond to imperfect information games, it follows that the technique is not always applicable. This section introduces a backward induction procedure to find pure Nash equilibria in partial order games. Games in which the procedure is well defined—in the sense that the procedure produces at least one strategy profile—we denote as being *fit for backwards induction*. We relate this latter concept to an informational notion we refer to as *scrutability*. For games that are fit for backwards induction, we prove that our backwards induction procedure is guaranteed to produce a Nash equilibrium outcome.

6.1. Backwards Induction

Due to their acyclic nature, it would seem that a natural concept of backward induction procedure can straightforwardly be defined for partial order games. Our aim is thus to define a strategy profile $\mathbf{f}^* = (f_{v_1}^*, \dots, f_{v_{|V|}}^*)$ that is obtained in the following fashion. One starts with a vertex v of maximal depth with set of parents $Y = \{y_1, \dots, y_k\}$, and one inspects the possible *actions*—rather than their *strategies*—the players active at the parents of v can play. Let $\mathbf{a}_Y = (a_{y_1}, \dots, a_{y_k})$ be any such an action profile. suppose i is the player active at v . Then, the action $a_v^* = f_v^*(a_{y_1}, \dots, a_{y_k})$ selected by strategy f_v^* from A_v should maximise i 's utility against \mathbf{a}_Y and *all* possible choices of action \mathbf{b}_X at the vertices in $X = V \setminus (Y \cup \{v\})$, provided any such action a_v^* exists. Subsequently, with the strategies for the vertices of greatest depth thus fixed, the strategies in \mathbf{f}^* should recursively find optimal strategies at vertices of lesser depth in a similar fashion.

In the case of partial order games, it is not strictly necessary to proceed recursively on the *depth* of the vertices. We can, instead, use any topological sorting $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$ of the vertices that respects the graph and, starting with the vertex v_{τ_j} with maximal

index $\tau_j = |V|$, we iterate through τ until we reach the vertex with minimal topological index. In this section, we will develop this more general concept of backward induction for partial order games that employs a topological order.

Our **backwards induction procedure** defines recursively a strategy profile $\mathbf{f}^* = (f_{v_1}^*, \dots, f_{v_{|V|}}^*)$ for each vertex v relative to a topological sorting $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$, and is formally defined as follows. Let v be a vertex with topological index τ_j and let i be the player to move at v . Let $Y = \{y_1, \dots, y_k\}$ be the set of parents of v , $Z = \{z_1, \dots, z_m\}$ be the set of vertices with a topological index strictly greater than τ_j , and $X = \{x_1, \dots, x_\ell\}$ be given by $V \setminus (Y \cup Z \cup \{v\})$. For every profile $\mathbf{a}_Y = (a_{y_1}, \dots, a_{y_k})$, we define $f_v^*(a_{y_1}, \dots, a_{y_k})$ as an action a_v^* in A_v such that, for all profiles $\mathbf{b}_X = (b_{x_1}, \dots, b_{x_\ell})$,

$$a_v^* \in \arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{f}_Z^*), \tag{1}$$

where the basis of this recursion is provided by the case where $Z = \emptyset$ (Here we exploit the notation $\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{f}_Z^*$ to refer to the *action profile* that results if after τ -history $(\mathbf{b}_X, \mathbf{a}_Y, a_v$ the strategy profile \mathbf{f}^* is played over the vertices in Z . Formally, $\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{f}_Z^*$ is the action profile \mathbf{a}^g for strategy profile $\mathbf{g} = (\mathbf{f}_X^{\mathbf{b}_X}, \mathbf{f}_Y^{\mathbf{a}_Y}, f_v^{a_v}, \mathbf{f}_Z^*)$, where $(\mathbf{f}_X^{\mathbf{b}_X}, \mathbf{f}_Y^{\mathbf{a}_Y}, f_v^{a_v})$ is a profile of unconditional strategies (also see page 3, above)). To illustrate this definition, we have the following example.

Example 4. Consider the utility function for the game depicted in Figure 10, and fix a topological sorting (v_1, v_2, v_3, v_4) . From inspection, it can easily be appreciated that the utility to player 1 is only dependent on their own choices at vertices v_1 and v_4 , and player 2's at vertex v_2 . At this moment, we can apply the backwards induction procedure as follows. Inspecting their utility function, we find that player 3 chooses as follows at vertex v_3 :

$$f_{v_3}(ac) = f \quad f_{v_3}(ad) = e \quad f_{v_3}(bc) = f \quad f_{v_3}(bd) = f$$

At vertex v_4 , player 1, can set their strategy such that $f_{v_4}(c) = g$ and $f_{v_4}(d) = h$, because they are indifferent to any choice of actions by the other players. Player 2 can have their strategy at v_2 depend on player 1's choice at v_1 . If player 1 chooses a at v_1 , then by playing c at v_2 , action profile $acfg$ will result and player 2 will obtain utility 3. Whereas, by playing d , action profile $adeh$ will ensue, with a utility of 1 for player 2. Therefore, $f_{v_2}(a) = c$. A similar reasoning yields that, if player 1 chooses b at v_1 , by playing c at v_2 , action profile (b, c, f, g) results with utility 0 for player 2, whereas, by playing d , action profile (b, d, f, h) ensues with utility 2 for player 2. Accordingly, $f_{v_2}(b) = d$. Finally, by playing a at v_1 , player 1 obtains utility 3, as action profile (a, c, f, g) would be played. Observe that by playing b at v_1 , player 1 obtains utility 2, as in that case action profile (b, d, f, h) would result. Accordingly, $f_{v_1}(\cdot) = a$. We thus obtain the strategy profile $\mathbf{f} = (f_{v_1}^1, f_{v_2}^2, f_{v_3}^{12}, f_{v_4}^2)$ yielding action profile (a, c, f, g) as the backwards induction solution of this game.

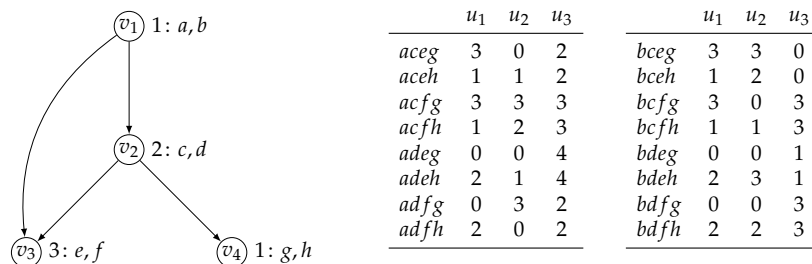


Figure 10. The partial order game from Example 1 with a slightly different utility function.

Some reflection reveals that on *total-order games* the procedure defined thus mimics standard backwards induction on the corresponding extensive-form games of perfect information as described in Section 4.4. It is important to note, however, that, on *general*

partial order games, our backwards induction procedure does not always yield a strategy profile \mathbf{f}^* . This is because the outcome of a partial order game—and therefore also the players’ utilities as well as the possibility of finding a utility maximising action—need not be fully determined by the actions chosen at a given vertex v , at v ’s parent vertices, and at the vertices with a larger topological index than v .

The outcome may also depend on the actions chosen at vertices of an equal or lesser depth that are not parents of the respective vertex. If so, it may happen that, at some stage of the procedure and at some vertex, no action can be singled out as an unequivocal optimal choice based on the choices at the parent nodes and those at the vertices with a greater topological index alone. This would cause the procedure to stall. More formally, this happens if for some vertex v , there are two different profiles $\mathbf{b}_X = (b_{w_1}, \dots, b_{w_\ell})$ and $\mathbf{b}'_X = (b'_{w_1}, \dots, b'_{w_\ell})$ such that $\arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{f}_Z^*)$ and $\arg \max_{a_v \in A_v} u_i(\mathbf{b}'_X, \mathbf{a}_Y, a_v, \mathbf{f}_Z^*)$ are disjoint. The following example illustrates this point.

Example 5. Consider again the game of Example 1, depicted in Figure 5. Let us first consider the strategies that player 3 can choose at vertex v_3 . This player has to specify an action from among e and f , for each of the possible action profiles player 1 and 2 can choose at vertices v_1 and v_2 , respectively, that is, for the action profiles ac , ad , bc , and bd . Inspecting player 3’s utilities, player 3 should arguably choose $f_{v_3}(ac) = f$, as no matter what action player 1 chooses at v_4 , player 3 is better off playing f than playing e . However, how to choose a value for $f_{v_3}(ad)$ is not so obvious, as the optimal choice depends on whether player 1 chooses g or h at vertex v_4 ; this is something player 3 may not let their strategy depend upon as not v_3Ev_4 .

To formalise this issue, we say that a partial order game is **fit for backwards induction** under a topological sorting $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$ if, for every vertex v , and every action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$,

$$\bigcap_{\mathbf{b}_X \in \mathbf{A}_X} \arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{a}_Z) \neq \emptyset,$$

where Y is the set of parents of v , Z the set of vertices with a topological index that is not smaller than that of v , and $X = V \setminus (Y \cup Z \cup \{v\})$. This ensures that, in Equation (1), the optimal action a_v^* is defined for every vertex v . At this point, it is worth observing that for total-order games, the set X will be empty and that they allow for only one topological sorting. It follows that total-order games vacuously satisfy the fit-for-backwards-induction condition under every topological sorting.

We are now in a position to show that our backwards induction procedure always yields a Nash equilibrium in games that are fit for backwards induction.

Theorem 7. Let G be a partial order game that is fit for backwards induction under a topological order τ . Then, every strategy profile $\mathbf{f}^* = (f_{v_1}^*, \dots, f_{v_{|V|}}^*)$ generated by the backwards induction procedure under τ is a Nash equilibrium.

Proof. Let $\mathbf{f}^* = (f_{v_1}^*, \dots, f_{v_{|V|}}^*)$ be a strategy profile that our backwards induction procedure yields for topological order $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$. Without loss of generality, we may assume that $\tau_j = j$ for all $1 \leq j \leq |V|$, that is, $v_1, \dots, v_{|V|}$ are already topologically ordered. Assume for contradiction that \mathbf{f}^* is not a Nash equilibrium, that is, there is some strategy \mathbf{g}_i for some player i such that $u_i(\mathbf{f}_{-i}^*, \mathbf{g}_i) > u_i(\mathbf{f}_{-i}^*, \mathbf{f}_i^*)$. Let $\mathbf{g} = (\mathbf{f}_{-i}^*, \mathbf{g}_i)$.

Now, there is at least one vertex v_j with $1 \leq j \leq |V|$ that is controlled by player i with parents $Y = \{y_1, \dots, y_k\}$ for which there is an action profile $\mathbf{x} = (x_{v_1}, \dots, x_{v_{j-1}})$ in $A_{v_1} \times \dots \times A_{v_{j-1}}$ such that:

$$u_i(x_{v_1}, \dots, x_{v_{j-1}}, a_{v_j}, f_{v_{j+1}}^*, \dots, f_{v_{|V|}}^*) < u_i(x_{v_1}, \dots, x_{v_{j-1}}, b_{v_j}, g_{v_{j+1}}, \dots, g_{v_{|V|}}), \tag{2}$$

where $a_{v_j} = f_{v_j}^*(x_{y_1}, \dots, x_{y_k})$ and $b_{v_j} = g_{v_j}(x_{y_1}, \dots, x_{y_k})$ (To see this, let $\mathbf{a}^{\mathbf{f}^*} = (a_{v_1}, \dots, a_{v_{|V|}})$ and $\mathbf{a}^{\mathbf{g}} = (b_{v_1}, \dots, b_{v_{|V|}})$). Then observe that there must be a smallest index $1 \leq j \leq |V|$ such

that $v_j \in V_i$ with $a_{v_j} \neq b_{v_j}$ and $a_{v_m} = b_{v_m}$ for all $1 \leq m < j$, which provides us with the desired v_j). Now we may assume that j is the largest index with $v_j \in V_i$ for which such an action profile, or τ -history, $\mathbf{x} = (x_{v_1}, \dots, x_{v_{j-1}})$ can be found. Therefore,

$$u_i(x_{v_1}, \dots, x_{v_{j-1}}, b_{v_j}, g_{v_{j+1}}, \dots, g_{v_{|V|}}) \leq u_i(x_{v_1}, \dots, x_{v_{j-1}}, b_{v_j}, f_{v_{j+1}}^*, \dots, f_{v_{|V|}}^*). \tag{3}$$

Combining inequalities (2) and (3), we obtain that

$$u_i(x_{v_1}, \dots, x_{v_{j-1}}, a_{v_j}, f_{v_{j+1}}^*, \dots, f_{v_{|V|}}^*) < u_i(x_{v_1}, \dots, x_{v_{j-1}}, b_{v_j}, f_{v_{j+1}}^*, \dots, f_{v_{|V|}}^*).$$

Having assumed $(v_1, \dots, v_{|V|})$ to be ordered topologically, the set $Y = \{y_1, \dots, y_k\}$ of parents of v_j is a subset of $\{v_1, \dots, v_{j-1}\}$. Let further $X = \{v_1, \dots, v_{j-1}\} \setminus Y$ and $Z = \{v_{j+1}, \dots, v_{|V|}\}$. Recall that $a_{v_j} = f_{v_j}^*(x_{y_1}, \dots, x_{y_k})$. We then find that, for profile \mathbf{x}_Y , there is some profile, namely \mathbf{x}_X , such that

$$f_{v_j}^*(x_{y_1}, \dots, x_{y_k}) \notin \arg \max_{a_{v_j} \in A_{v_j}} u_i(\mathbf{x}_X, \mathbf{x}_Y, a_{v_j}, \mathbf{f}_Z^*).$$

It follows that our backwards induction procedure does not provide us with $f_{v_j}^*$, which contradicts our initial assumption regarding the profile $\mathbf{f}^* = (f_{v_1}^*, \dots, f_{v_{|V|}}^*)$. \square

We also have the following lemma, which is a version of the *one-stage-deviation property* for partial order games, and which is an almost immediate consequence of the definition of our backwards induction procedure (also see in [28] (pages 108–110) and Osborne and Rubinstein [5] (page 98)). This property will prove to be of great help proving the computational complexity results in Section 7. Intuitively it says that to establish that a strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a backwards induction solution, it suffices to look at deviations by a player from \mathbf{f} at a single vertex v_k given a single history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$. The property is perhaps best understood by considering its contrapositive: strategy profile \mathbf{f} is *not* a backwards induction solution if there is a single player who has an incentive to deviate from \mathbf{f} at a single vertex v_k given a single history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$. Note that we here assume that the game in question is fit for backwards induction.

Lemma 5 (One-stage-deviation property). *Let the vertices of a partial order game be sorted according to a topological sorting $\tau = (v_1, \dots, v_{|V|})$. Then, for games that are fit for backwards induction, profile $f = (f_{v_1}, \dots, f_{v_{|V|}})$ is a backwards induction solution for τ if and only if for every τ -history $(a_{v_1}, \dots, a_{v_{k-1}})$ with $1 \leq k \leq |V|$ we have for all $b_{v_k} \in A_{v_k}$:*

$$u_i(a_{v_1}, \dots, a_{v_{k-1}}, f_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}}) \geq u_i(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}}),$$

where i is the player assigned to v_k .

Proof. First assume that $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a backwards induction solution, and consider an arbitrary τ -history $(a_{v_1}, \dots, a_{v_{k-1}})$ and equally arbitrary b_{v_k} in A_{v_k} . Let Y be the parents of v_k and $X = \{v_1, \dots, v_{k-1}\} \setminus Y$. Then,

$$u_i(\mathbf{b}_X, \mathbf{a}_Y, f_{v_k}(\mathbf{a}_Y), f_{v_{k+1}}, \dots, f_{v_{|V|}}) \geq u_i(\mathbf{b}_X, \mathbf{a}_Y, b_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}}), \tag{4}$$

for all profiles \mathbf{b}_X . Now observe that inequality (4) holds for in particular for $\mathbf{b}_X = \mathbf{a}_X$.

For the “if”-direction assume for contraposition that $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is not a backwards induction solution for τ . Then, there is a vertex v_k with minimal topological index, parents Y , and $X = \{v_1, \dots, v_{k-1}\} \setminus Y$ such that for some profile \mathbf{a}_Y , we have for every a_{v_k} in A_{v_k} that there is some profile \mathbf{b}_X such that

$$u_i(\mathbf{b}_X, \mathbf{a}_Y, a_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}}) < u_i(\mathbf{b}_X, \mathbf{a}_Y, b_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}}). \tag{5}$$

At this point, consider any a_{v_k} in A_{v_k} and let \mathbf{b}_X be the accompanying profile for which Equation (5) holds. We can conclude the proof by considering the τ -history $(\mathbf{b}_X, \mathbf{a}_Y)$. \square

6.2. Games of Adequate Information

In this section, we propose the class of *adequate information games*, which are guaranteed to be fit for backwards induction, and, thus, always have a Nash equilibrium. As an auxiliary concept we first introduce the notion of *scrutability*, an informational criterion specifying on which choices of which players a player can have its decision depend when choosing an action at a specific vertex.

6.2.1. Scrutability

We introduce *scrutability* as a technical term to reason about the strategic information that is available to the players at each stage of the backwards induction process. Intuitively, let v and w be a vertices at which, respectively, players i and j are active. Then, assuming that the game is fit for backwards induction, a vertex w is *scrutable* to v if j 's choices of action at w can make a difference as to the optimality of i 's choice of action at v when deciding on a backwards induction strategy at v . Thus, all vertices reachable from v are scrutable, as their backwards induction choices will already have been made. Moreover, the player active at v can have their choice of action depend on the choices of action at the parent nodes of v .

Formally, we say that a vertex w is **scrutable** to vertex v if either

- (i) w is reachable from v , or
- (ii) wEv , that is w is one of v 's parents.

We refer to a vertex that is not scrutable to v as being **inscrutable** to v . Recall the reachability relation is reflexive, and, thus, vertex v is always scrutable to v itself. Observe that the relations of scrutability and inscrutability are not necessarily symmetric. For instance, if a vertex v is reachable from w , but w is not a parent of v , then v is scrutable to w , but not vice versa. If v is inscrutable to w , and w is also inscrutable to v , we say that v and w are **independent**.

We differentiate between a vertex being scrutable to a *vertex* and a vertex being scrutable to a *player*. For $V_i = \{v_1, \dots, v_m\}$ player i 's vertices, we denote by I_v the set of vertices that are inscrutable to v , and by S_v the set of vertices that are scrutable to v . Furthermore, let $I_i = I_{v_1} \cup \dots \cup I_{v_m}$ and $S_i = V \setminus I_i$ denote the sets of vertices that are inscrutable and scrutable to player i , respectively. That is, if a vertex v is inscrutable to a player i , if v is inscrutable to at least one vertex at which i is active. Vertex v is scrutable to player i , if it is not inscrutable to i , that is, if v is scrutable to all vertices at which i is active. It may be worth observing that in total-order games all vertices will be scrutable from any other vertex.

Furthermore, we introduce a notion of scrutability that is relative to a topological sorting of the vertices. Therefore, let $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$ be a topological sorting. Then, a vertex w with topological index τ_w is **scrutable under** τ to vertex v with topological index τ_v , if either

- (i) $\tau_w \geq \tau_v$, or
- (ii) wEv , that is w is one of v 's parents.

Any vertex that is not scrutable under τ is referred to as being **inscrutable under** τ . For $V_i = \{v_1, \dots, v_m\}$ and topological sorting $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$, we furthermore denote by I_v^τ the set of vertices that are inscrutable to v under τ , and by S_v^τ the set of vertices that are scrutable to v under τ .

Moreover, we define $I_i^\tau = I_{v_1}^\tau \cup \dots \cup I_{v_m}^\tau$ and $S_i^\tau = V \setminus I_i^\tau$. Note that the set of vertices scrutable to a player i need not include the set V_i of vertices where i is active.

Given a topological sorting τ and a vertex v , by Y , we will generally adhere to the following notational conventions. The parents of v we generally denote by Y , the set of vertices inscrutable to v by X , and the set of all vertices that are not parents of v but still scrutable to v by Z . The following example illustrates these concepts.

Example 6. Consider the game depicted in Figure 10. Then, we have the following sets of scrutable and inscrutable vertices.

$$\begin{array}{llll}
 S_{v_1} = \{v_1, v_2, v_3, v_4\} & I_{v_1} = \emptyset & S_1 = \{v_2, v_4\} & I_1 = \{v_1, v_3\} \\
 S_{v_2} = \{v_1, v_2, v_3, v_4\} & I_{v_2} = \emptyset & S_2 = \{v_1, v_2, v_3, v_4\} & I_2 = \emptyset \\
 S_{v_3} = \{v_1, v_2, v_3\} & I_{v_3} = \{v_4\} & S_3 = \{v_1, v_2, v_3\} & I_3 = \{v_4\} \\
 S_{v_4} = \{v_2, v_4\} & I_{v_4} = \{v_1, v_3\} & &
 \end{array}$$

For the topological sorting $\tau = (v_1, v_2, v_3, v_4)$, we get

$$\begin{array}{llll}
 S_{v_1}^\tau = \{v_1, v_2, v_3, v_4\} & I_{v_1}^\tau = \emptyset & S_1^\tau = \{v_2, v_4\} & I_1^\tau = \{v_1, v_3\} \\
 S_{v_2}^\tau = \{v_1, v_2, v_3, v_4\} & I_{v_2}^\tau = \emptyset & S_2^\tau = \{v_1, v_2, v_3, v_4\} & I_2^\tau = \emptyset \\
 S_{v_3}^\tau = \{v_1, v_2, v_3, v_4\} & I_{v_3}^\tau = \emptyset & S_3^\tau = \{v_1, v_2, v_3, v_4\} & I_3^\tau = \emptyset \\
 S_{v_4}^\tau = \{v_2, v_4\} & I_{v_4}^\tau = \{v_1, v_3\} & &
 \end{array}$$

By contrast, under topological sorting $\tau' = (v_1, v_2, v_4, v_3)$, we obtain

$$\begin{array}{llll}
 S_{v_1}^{\tau'} = \{v_1, v_2, v_3, v_4\} & I_{v_1}^{\tau'} = \emptyset & S_1^{\tau'} = \{v_2, v_3, v_4\} & I_1^{\tau'} = \{v_1\} \\
 S_{v_2}^{\tau'} = \{v_1, v_2, v_3, v_4\} & I_{v_2}^{\tau'} = \emptyset & S_2^{\tau'} = \{v_1, v_2, v_3, v_4\} & I_2^{\tau'} = \emptyset \\
 S_{v_3}^{\tau'} = \{v_1, v_2, v_3\} & I_{v_3}^{\tau'} = \{v_4\} & S_3^{\tau'} = \{v_1, v_2, v_3\} & I_3^{\tau'} = \{v_4\} \\
 S_{v_4}^{\tau'} = \{v_2, v_3, v_4\} & I_{v_4}^{\tau'} = \{v_1\} & &
 \end{array}$$

For another example, consider Figure 11 along with topological order $\tau = (v_1, v_2, v_3, v_4)$. We then obtain for the vertices the following sets of scrutable and inscrutable vertices for vertices as well as for players:

$$\begin{array}{llll}
 S_{v_1}^\tau = \{v_1, v_2, v_3, v_4\} & I_{v_1}^\tau = \emptyset & S_1^\tau = \{v_1, v_2, v_3, v_4\} & I_1^\tau = \emptyset \\
 S_{v_2}^\tau = \{v_2, v_3, v_4\} & I_{v_2}^\tau = \{v_1\} & S_2^\tau = \{v_2, v_4\} & I_2^\tau = \{v_1, v_3\} \\
 S_{v_3}^\tau = \{v_1, v_2, v_3, v_4\} & I_{v_3}^\tau = \emptyset & & \\
 S_{v_4}^\tau = \{v_1, v_2, v_4\} & I_{v_4}^\tau = \{v_3\} & &
 \end{array}$$

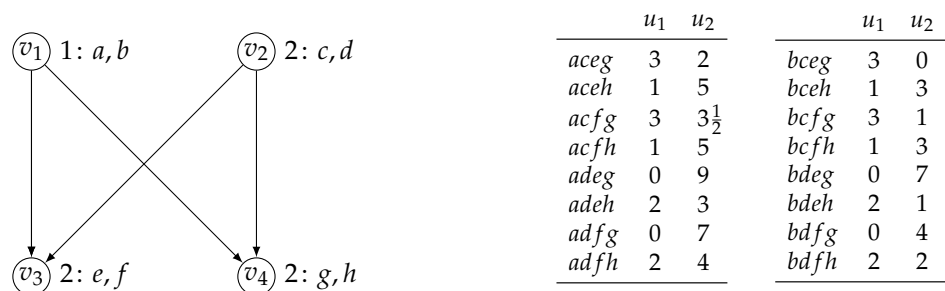


Figure 11. Another example of a partial order game.

Observe that w being reachable from v implies $\tau_w \geq \tau_v$, in other words, $S_v \subseteq S_v^\tau$ for every topological order τ . Accordingly, plain scrutable is stronger than scrutable under a topological sorting. The following lemma lays down the precise formal interrelationship between scrutable and scrutable under a topological order.

Lemma 6. Let i be a player of a partial order game and v a vertex controlled by i . Then, for all vertices w ,

- (i) w is scrutable to v if and only if w is scrutable to v under all topological orders τ .
- (ii) w is inscrutable to v if and only if w is inscrutable to v under some topological order τ .

- (iii) w is scrutable to i if and only if w is scrutable to i under all topological orders τ .
- (iv) w is inscrutable to i if and only if w is inscrutable to i under some topological order τ .

Proof. We prove part (i) first. Then, parts (ii) through (iv) follow by straightforward set-theoretic reasoning.

- (i) For the “only if”-direction, first assume that w is scrutable to v and consider an arbitrary topological sorting τ with topological indices τ_i and τ_j for v and w , respectively. Then, either wEv or w is reachable from v , that is, vE^*w . In the former case, we have immediately that w is scrutable to v under τ . In the latter case, observe that then $\tau_i < \tau_j$ by definition of a topological sorting. It then also follows that w is scrutable to v under τ .

For the “if”-direction, assume for contraposition that w is not scrutable to v . Then, neither wEv nor w being reachable from v hold. In case v is reachable from w , let τ be any topological order with τ_i and τ_j the topological indices for v and w . Then, $\tau_j < \tau_i$, and it follows that w is not scrutable under τ . In case v is not reachable from w , let (V, E) be the DAG underlying the partial order game. Then, consider the edge set $E' = E \cup \{(w, v)\}$. The graph (V, E') must be acyclic because if (V, E') contained a directed cycle, there would be a simple path from v to w not involving the edge (w, v) . Therefore, w would also be reachable from v in (V, E) , a contradiction. Therefore, (V, E') is acyclic and has a topological sorting τ . Observe that τ is also a topological order for (V, E) . Let τ_i and τ_j be the topologic al indices for v and w , respectively. As $wE'v$, we have $\tau_j < \tau_i$ and again we may conclude that w is not scrutable to v under τ .

- (ii) This part follows immediately from part (i), by observing the following equivalences:

w is inscrutable to v
 if and only if w is not scrutable to v
 if and only if w is not scrutable to v under some topological order τ
 if and only if w is inscrutable to v under some topological order τ

- (iii) Let v_1, \dots, v_m be the vertices assigned to player i , and let τ_1, \dots, τ_k be the topological & sortings for the game. From (ii) we obtain that $I_{v_j} = I_{v_j}^{\tau_1} \cup \dots \cup I_{v_j}^{\tau_k}$ for every $1 \leq j \leq m$.
- (iv) Therefore,

$$I_i = \bigcup_{1 \leq j \leq m} I_{v_j} = \bigcup_{1 \leq j \leq m} I_{v_j}^{\tau_1} \cup \dots \cup I_{v_j}^{\tau_k} = \bigcup_{1 \leq \ell \leq k} I_{v_1}^{\tau_\ell} \cup \dots \cup I_{v_m}^{\tau_\ell} = \bigcup_{1 \leq \ell \leq k} I_i^{\tau_\ell},$$

and it follows that w is inscrutable to i if and only if w is inscrutable to i under some topological sorting. Part (iii) then is an immediate consequence of de Morgan’s law:

$$S_i = V \setminus I_i = V \setminus (I_i^{\tau_1} \cup \dots \cup I_i^{\tau_k}) = V \setminus (I_i^{\tau_1} \cap \dots \cap I_i^{\tau_k}) = S_{v_1}^{\tau_1} \cap \dots \cap S_i^{\tau_k} = \bigcap_{1 \leq \ell \leq k} S_i^{\tau_\ell}.$$

Therefore, w is scrutable to i if and only if i is scrutable to i under all topological sortings.

This concludes the proof. \square

6.2.2. Games of Adequate Information

In this section, we define a class of partial order games, which we refer to as games of *adequate information*. We prove that games of adequate information are fit for backward induction, and thus, on account of Theorem 7, allow for Nash equilibria. For this, we require that each player at each vertex assigned to them is in a position to determine their optimal choices if *they are given only the choices at their parents and those at the nodes with a greater topological index*.

Formally, we introduce the concept of a utility function being of adequate information and a variant of the notion parametrised by a topological order τ . These concepts only

differ in the vertices that are considered to be scrutable and inscrutable. Let i be a player with utility function $u_i: A_{v_1} \times \dots \times A_{v_{|V|}} \rightarrow \mathbb{R}$. Without loss of generality, we assume that $V_i = \{v_1, \dots, v_m\}$. Let S be the set of vertices scrutable to i (under τ) and I the set of vertices inscrutable to i (under τ). Then, player i 's utility function $u_i: A_V \rightarrow \mathbb{R}$ is said to be of **adequate information (under τ)** or **adequately informed (under τ)**, if there is a function $u_i^0: \mathbf{A}_S \rightarrow \mathbb{R}$ such that for all profiles $\mathbf{a}_S, \mathbf{a}'_S \in \mathbf{A}_S$ and $\mathbf{b}_I \in \mathbf{A}_I$

$$u_i(\mathbf{a}_S, \mathbf{b}_I) \geq u_i(\mathbf{a}'_S, \mathbf{b}_I) \quad \text{if and only if} \quad u_i^0(\mathbf{a}_S) \geq u_i^0(\mathbf{a}'_S).$$

If this is the case, we say that player i 's utility function $u_i: \mathbf{A} \rightarrow \mathbb{R}$ (**ordinally**) **reduces to** $u_i^0: \mathbf{A}_S \rightarrow \mathbb{R}$.

A **game of adequate information under τ** is, then, a partial order game in which each player's utility function is of adequate information under τ . We also define a concept of adequate information that is irrespective of the topological sorting τ in the straightforward manner. Thus, a **game of adequate information simpliciter** is a partial order game in which each player's utility function is of adequate information. Observe that total-order games constitute a subclass of adequate-information games, since such games do not allow for inscrutable vertices. To illustrate, we have the following example:

Example 7 (Game of adequate information). Consider the game depicted in Figure 10 again. Under topological sorting $\tau = (v_1, v_2, v_3, v_4)$, player 1 has scrutables $S_1^\tau = \{v_2, v_4\}$ and inscrutables $I_1^\tau = \{v_1, v_3\}$. For player 1, observe that for all $\mathbf{a} \in A_{v_2} \times A_{v_4}$ and all $\mathbf{b}, \mathbf{b}' \in A_{v_1} \times A_{v_3}$ we have that $u_1(\mathbf{a}, \mathbf{b}) = u_1(\mathbf{a}, \mathbf{b}')$. For instance, $u_1(cg, ae) = u_1(cg, af) = 3$ and $u_1(dg, af) = u_1(dg, bf) = 0$. Accordingly, define $u_1^0: A_{v_2} \times A_{v_4} \rightarrow \mathbb{R}$ such that

$$u_1^0(cg) = 3 \qquad u_1^0(ch) = 1 \qquad u_1^0(dg) = 0 \qquad u_1^0(dh) = 2,$$

and find that u_1 ordinally reduced to u_1^0 and, hence, that player 1 is adequately informed in this game. For players 2 and 3, we saw that $S_2^\tau = S_3^\tau = \{v_1, v_2, v_3, v_4\}$ and $I_2^\tau = I_3^\tau = \emptyset$. As some reflection reveals, u_2 and u_3 thus vacuously reduce to themselves. We therefore immediately see that their utility functions are of adequate information under τ as well.

Finally, consider once more the game depicted in Figure 11 along with the natural topological order $\tau = (v_1, v_2, v_3, v_4)$. Under τ , this game is of adequate information. As $I_1^\tau = \emptyset$, this is easy to see for player 1. For player 2 we have $S_2^\tau = \{v_2, v_4\}$ and $I_2^\tau = \{v_1, v_3\}$. Now, we can define u_2^0 such that

$$u_2^0(cg) = 1 \qquad u_2^0(ch) = 4 \qquad u_2^0(dg) = 8 \qquad u_2^0(dh) = 2$$

It then only requires an easy check to see that u_2 ordinally reduces to u_2^0 . For instance, $u_2^0(cg) < u_2^0(dg)$ and also

$$\begin{aligned} u_2(cg, ae) &= 2 < 9 = u_2(dg, ae) & u_2(cg, af) &= 3\frac{1}{2} < 7 = u_2(dg, af) \\ u_2(cg, be) &= 2 < 7 = u_2(dg, be) & u_2(cg, bf) &= 2 < 4 = u_2(dg, bf) \end{aligned}$$

By contrast, under topological order $\tau'' = (v_2, v_1, v_4, v_3)$, player 1's utility function is not of adequate information. To see this, first observe that $S_1^{\tau''} = \{v_1, v_3, v_4\}$ and $I_1^{\tau''} = \{v_2\}$. Now, $u_1(aeg, c) = 2 < 9 = u_1(aeh, c)$, whereas $u_1(aeg, d) = 5 > 3 = u_1(aeh, d)$. Had there been a function $u_1^0: \mathbf{A}_{S_1^{\tau''}} \rightarrow \mathbb{R}$ such that u_1 ordinally reduces to u_1^0 , then both $u_1^0(aeg) < u_1^0(aeh)$ and $u_1^0(aeg) > u_1^0(aeh)$, a contradiction.

The following result establishes that games of adequate information (under a topological sorting τ) are fit for backwards induction (under τ).

Theorem 8. Let $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$ be a topological sorting. Then,

- (i) games of adequate information under τ are guaranteed to have a Nash equilibrium, which can be found using our backward induction procedure under τ , and
- (ii) games of adequate information simpliciter are guaranteed to have a Nash equilibrium, which can be found using our backward induction procedure under any topological sorting.

Proof. Let $\tau = (v_{\tau_1}, \dots, v_{\tau_{|V|}})$ be an arbitrary topological sorting. For part (i), due to Theorem 7, it suffices to prove that games of adequate information under τ are fit for backward induction under τ . To this end, consider an arbitrary game of adequate information under τ and consider an arbitrary vertex v . Let i be the player assigned to v . Furthermore, let $I_i^\tau = \{w_1, \dots, w_k\}$ denote the set of vertices that are inscrutable to i under τ and $S_i^\tau = \{w_{k+1}, \dots, w_{|V|}\}$ denote the set of vertices that are scrutable to i under τ . We let Y denote the set of parents of v , and Z the set of vertices with a strictly greater topological index than v . Then, set $X = V \setminus (Y \cup Z \cup \{v\})$ and let $X = \{x_1, \dots, x_{|X|}\}$. Observe that X is the set of vertices that are inscrutable to v under τ , and thus all vertices in X are inscrutable to i under τ , that is, $X \subseteq I_i^\tau$.

By assumption, player i 's utility function $u_i: \mathbf{A} \rightarrow \mathbb{R}$ is adequately informed under τ . In particular, we may assume that u_i ordinally reduces to a function $u_i^0: \mathbf{A}_{S_i^\tau} \rightarrow \mathbb{R}$.

Now, consider an arbitrary action profile $\mathbf{a} = (a_1, \dots, a_{|V|})$. Having assumed \mathbf{A} to be finite, there must be an action a_v^* in A_v with

$$a_v^* \in \arg \max_{a_v \in A_v} u_i(\mathbf{a}_X, \mathbf{a}_Y, a_v, \mathbf{a}_Z).$$

Let $\mathbf{a}^* = (\mathbf{a}_X, \mathbf{a}_Y, a_v^*, \mathbf{a}_Z)$.

Consider an arbitrary a_v'' in A_v , and let $\mathbf{a}'' = (\mathbf{a}_X, \mathbf{a}_Y, a_v'', \mathbf{a}_Z)$. Then, $u_i(\mathbf{a}^*) \geq u_i(\mathbf{a}'')$. Having assumed that the game is of adequate information we may assume that there is a function $u_i^0: \mathbf{A}_{S_i^\tau} \rightarrow \mathbb{R}$ such that

$$u_i^0(\mathbf{a}_{S_i^\tau}^*) \geq u_i^0(\mathbf{a}_{S_i^\tau}''). \tag{*}$$

Consider an arbitrary profile \mathbf{b} in \mathbf{A} . Then, $u_i(\mathbf{b}_{I_i^\tau}, \mathbf{a}_{S_i^\tau}^*) \geq u_i(\mathbf{b}_{I_i^\tau}, \mathbf{a}_{S_i^\tau}'')$. Now, either v is scrutable to i or v is inscrutable to i . If the latter, $I_i^\tau = X \cup \{v\}$. Then, $u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v^*, \mathbf{a}_Z) = u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v'', \mathbf{a}_Z)$. With a_v'' having been chosen arbitrarily, $a_v^* \in \arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{a}_Z)$. If the former, $I_i^\tau = X$, and we find that $u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v^*, \mathbf{a}_Z) \geq u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v'', \mathbf{a}_Z)$. Again, we may conclude that $a_v^* \in \arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{a}_Z)$. In either case, and with \mathbf{b} having been chosen arbitrarily, $a_v^* \in \bigcap_{\mathbf{b}_X \in \mathbf{A}_X} \arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{a}_Z)$, which proves the case.

Part (ii) follows by an argument that runs along analogous lines as that for part (i), reasoning about the vertices that are inscrutable to i simpliciter rather than under a specific topological sorting τ . In this context, observe that, by virtue of Lemma 6, the set I_i^τ of vertices inscrutable to i under a topological sorting τ constitutes invariably a subset of the set I_i of vertices inscrutable to i simpliciter. \square

7. Computing Backwards Induction Solutions for Partial Order Games

In this section, we explore a number of natural computational problems surrounding our backwards induction procedure for partial order games and games of adequate information. We first conduct an asymptotic analysis of our backwards induction procedure. We find that the procedure runs in exponential time in the general case, but that a considerable speed up is possible if the in-degree of the vertices or the number of their parents is bounded appropriately. Subsequently, we investigate the computational complexity of deciding whether a partial order game is fit for backwards induction, that is if the precondition for our procedure to produce a Nash equilibrium holds. In a similar vein, we also address the closely related question of whether a given partial order game is a game of adequate information. We find that both these problems are coNP-complete. Finally, we come to consider the problems of deciding whether a given strategy profile

is a backwards induction solution and whether a given action profile is sustained by a backwards induction solution. We show that the latter problem is NEXPTIME-complete for the general case, but that it is PSPACE-complete for total-order games.

7.1. Asymptotic Complexity of Backwards Induction

Due to the compactness of the partial order game representation, our backwards induction procedure runs in time exponential in the number $|V|$ of vertices. Let $v_1, \dots, v_{|V|}$ be ordered according to the topological sorting τ under which backwards induction is conducted. Throughout this section, as we did before, for each vertex v , by Y we denote the parents of v , by Z the vertices with a greater topological index than v , and let $X = V \setminus (Y \cup Z \cup \{v\})$. Then, for each vertex v_j , where $1 \leq j \leq |V|$, with parents Y and τ -inscrutable vertices X we have to find, an action $a_{v_j}^*$ in A_{v_j} such that, for i the active player at v_j ,

$$a^* \in \bigcap_{\mathbf{b}_X \in \mathbf{A}_X} \arg \max_{a_{v_j} \in A_{v_j}} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_{v_j}, \mathbf{f}_Z^*),$$

and abort if no such action exists. This requires for each profile a_Y in A_Y the computation of $|A_{v_j}| \cdot \prod_{u \in X} |A_u|$ utility values $u_i(\mathbf{a}_X, \mathbf{a}_Y, a_{v_j}, \mathbf{a}_Z^*)$. Letting $\bar{A} = \max_{v \in V} |A_v|$, this makes for a total number of utility computations of

$$\sum_{j=1}^{|V|} \prod_{k=1}^j |A_k| \leq \sum_{j=1}^{|V|} \prod_{k=1}^j \bar{A} = \sum_{j=1}^{|V|} \bar{A}^j = \sum_{j=0}^{|V|} \bar{A}^j - 1 = \frac{|\bar{A}|^{|V|+1} - 1}{|\bar{A}| - 1} - 1 = O(|\bar{A}|^{|V|}).$$

Assuming that the players' utilities are represented by a Boolean circuit, each of these can be achieved in time polynomial in $|A|$. Following the definition, we can therefore compute a backwards induction solution in $O(|A|^{|V|})$ time.

It is not hard to see that, if we restrict ourselves to partial order games of *adequate information* in which all vertices have at most $\log_2(|A|)$ parents, a backwards induction profile can be computed in time pseudo-polynomial in $|A|$. If, in addition to their having at most $\log_2(|A|)$ parents, every vertex v of a game of adequate information has at most two actions, that is, $|A_v| \leq 2$, then a backwards induction profile can be computed in time polynomial in $|A|$. Summarising, we have the following proposition.

Proposition 5. *Let Y_v denote the set of parents of a vertex v . For a partial order game of adequate information, a backwards induction profile can then be computed in time:*

- (i) $O(|A|^{|V|})$, in the general case,
- (ii) $O(|A|^{\log_2(|A|)})$, if $|Y_v| \leq \log_2(|A|)$ for all vertices v ,
- (iii) $O(|A|)$, if $|Y_v| \leq \log_2(|A|)$ and $|A_v| \leq 2$ for all vertices v .

7.2. Computational Complexity of Backwards Induction

In this section, we investigate the natural computation problems of deciding whether a backwards induction solution exists in a given partial order game for a given topological order. Furthermore, we analyse how hard it is to decide whether a given strategy profile incorporates a backwards induction solution. Finally, recalling that action profiles and strategy profiles are essentially different objects in the setting of partial order games, we explore how computationally complex it is to decide whether a given action profile is *sustained* by a backwards induction solution, that is, whether, for a given action profile \mathbf{a} , there is a backwards induction solution \mathbf{f} such that $\mathbf{a} = \mathbf{a}^{\mathbf{f}}$. Throughout this section, we assume that strategies and utilities are concisely represented as Boolean circuits as outlined in Section 3.1.

First, we consider the following computational problem concerning the conditions under which a partial order game is fit for backwards induction. Formally:

FIT FOR BACKWARDS INDUCTION

Given: Partial order game G and a topological sorting τ of its vertices

Problem: Is G fit for backwards induction under τ ?

For total-order games, FIT FOR BACKWARDS INDUCTION is vacuous, as they are all fit for backwards induction. For general partial order games, however, we have the following.

Theorem 9. FIT FOR BACKWARDS INDUCTION is coNP-complete.

Proof. To see that FIT FOR BACKWARDS INDUCTION is in coNP, let τ be a topological sorting of the game’s vertices. A certificate for a counterexample is then given by

- (i) a vertex v , with $A_v = \{a_1, \dots, a_k\}$, and X, Y , and Z the set of inscrutable vertices with a lower topological index, the set of v ’s parents, and the vertices with a higher topological index than v , respectively;
- (ii) a profile \mathbf{a}_Y in \mathbf{A}_Y ;
- (iii) a profile \mathbf{a}_Z in \mathbf{A}_Z ;
- (iv) a tuple (a'_1, \dots, a'_k) of k actions a'_j in A_k ; and
- (v) a tuple $(\mathbf{b}_X^1, \dots, \mathbf{b}_X^k)$ of k profiles \mathbf{b}_X^j in \mathbf{A}_X .

It is easy to verify that such a certificate is still of polynomial size. Observe that, if $u_i(\mathbf{b}_X^j, \mathbf{a}_Y, a'_j, \mathbf{a}_Z) > u_i(\mathbf{b}_X^j, \mathbf{a}_Y, a_j, \mathbf{a}_Z)$ holds for some $1 \leq j \leq k$, we may conclude that

$$a_j \notin \bigcap_{\mathbf{b}_X \in \mathbf{A}_X} \arg \max_{a_v \in A_v} u_i(\mathbf{b}_X, \mathbf{a}_Y, a_v, \mathbf{a}_Z).$$

Therefore, such a certificate is a counterexample to the game being fit for backwards induction under τ if and only if $u_i(\mathbf{b}_X^j, \mathbf{a}_Y, a'_j, \mathbf{a}_Z) > u_i(\mathbf{b}_X^j, \mathbf{a}_Y, a_j, \mathbf{a}_Z)$ for every $1 \leq j \leq k$. Observe that determining the sets X, Y , and Z can be achieved in polynomial time. Having assumed, moreover, that the utility function u_i is part of the input and represented by a Boolean circuits and evaluation problem for Boolean circuits is solvable in polynomial time, we find that this is a property that can be checked in polynomial time.

For coNP-hardness, we reduce the complement of SATISFIABILITY. Let φ be a propositional formula in the propositional variables p_1, \dots, p_m . Let q and r be two additional, fresh, propositional variables. We construct a partial order Boolean game with three players, player 1, player 2, and player 3 on a DAG (V, E) with $V = \{v_q, v_r, v_{p_1}, \dots, v_{p_m}\}$ and E such that $(v, u) \in E$ whenever $v = v_q$ and $u = v_{p_j}$ for some $1 \leq j \leq m$, or $v = v_{p_j}$ for some $1 \leq j \leq m$ and $u = v_r$. Furthermore, see Figure 12 for an illustration of the construction of such a DAG in the case φ is defined over five propositional variables. Let v_q be assigned to player 1, the vertices v_{p_1}, \dots, v_{p_m} to player 2, and v_r to player 3. Furthermore, assume that $\gamma_1 = \gamma_2 = \top$ and $\gamma_3 = \varphi \rightarrow (q \leftrightarrow r)$. Consider an arbitrary topological sorting τ of the vertices.

First, assume that φ is not satisfiable. Then, all players are indifferent between all outcomes, and it can easily be seen that the game is fit for backwards induction under τ .

Finally, assume that assignment $\alpha: \{p_1, \dots, p_m\} \rightarrow \{0, 1\}$ satisfies φ . Observe that v_q is inscrutable to v_r and has a lower topological rank under τ . Moreover, v_{p_1}, \dots, v_{p_m} are the parents of v_r . Now, let $a_{v_{p_1}}, \dots, a_{v_{p_m}}$ be such that $a_{p_j} = p_j$ if $\alpha(p_j) = 1$, and $a_{p_j} = \bar{p}_j$ if $\alpha(p_j) = 0$. Furthermore, let $b_{v_q} = q$ and $b'_{v_q} = \bar{q}$, and $a_{v_r} = r$ and $a'_{v_r} = \bar{r}$. Then,

$$u_3(q, a_{v_{p_1}}, \dots, a_{v_{p_m}}, r) > u_3(\bar{q}, a_{v_{p_1}}, \dots, a_{v_{p_m}}, \bar{r})$$

but

$$u_3(\bar{q}, a_{v_{p_1}}, \dots, a_{v_{p_m}}, r) < u_3(q, a_{v_{p_1}}, \dots, a_{v_{p_m}}, \bar{r}).$$

Accordingly, we may conclude that the game is not fit for backwards induction. \square

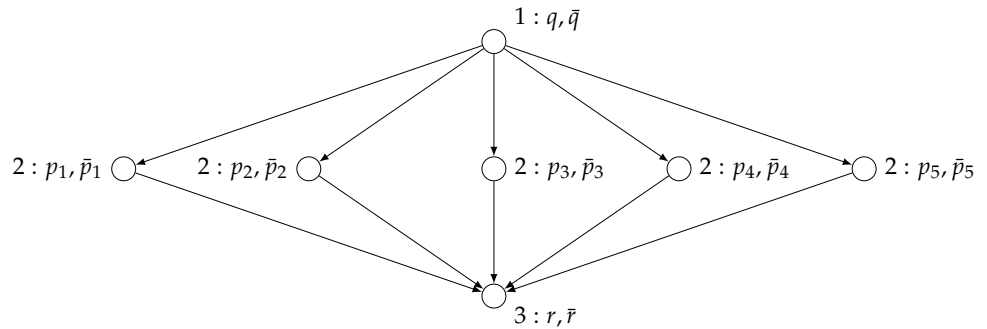


Figure 12. Construction of the DAG for the coNP-hardness proof for FIT FOR BACKWARDS INDUCTION in the case of the formula φ being defined over five variables p_1 through p_5 .

The closely related decision problem of whether a given partial order game is of adequate information, can be formulated as follows.

ADEQUATE INFORMATION

Given: Partial order game G and topological sorting τ

Problem: Is G a game of adequate information under τ ?

Again, note that ADEQUATE INFORMATION is vacuous for total-order games. For general partial order games, we find that the problem of ADEQUATE INFORMATION is not easier than FIT FOR BACKWARDS INDUCTION. The similarity of the two problems—and so also the closeness of the concepts of fitness of backwards induction and adequate information—is revealed by the closeness of their respective proofs of coNP-completeness.

Theorem 10. ADEQUATE INFORMATION is coNP-complete.

Proof. For membership in coNP, let a certificate for a counterexample consist of a player i , a pair of profiles \mathbf{a}_S and \mathbf{a}'_S in \mathbf{A}_S , and a pair of profiles \mathbf{b}_I and \mathbf{b}'_I in \mathbf{A}_I , where S is the set of vertices that are scrutable to i under τ and I the set of vertices that are inscrutable to i under τ . Observe that such a certificate is of polynomial size and that the sets S and I can be computed in polynomial time. Then, such a certificate $(\mathbf{a}_S, \mathbf{a}'_S, \mathbf{b}_I, \mathbf{b}'_I)$ defines a counterexample against the game being of adequate information whenever it is *not* the case that $u_i(\mathbf{a}_S, \mathbf{b}_I) \leq u_i(\mathbf{a}'_S, \mathbf{b}_I)$ if and only if $u_i(\mathbf{a}_S, \mathbf{b}'_I) \leq u_i(\mathbf{a}'_S, \mathbf{b}'_I)$. Having assumed that the utility function u_i is part of the input and represented by a Boolean circuit and evaluation problem for Boolean circuits is solvable in polynomial time, we find that this is a property that can be checked in polynomial time (see Lemma 2).

The proof of coNP-hardness is by a reduction from the complement of SATISFIABILITY, and runs along analogous lines as the proof of coNP-hardness of FIT FOR BACKWARDS INDUCTION. Thus, given a propositional formula φ in the variables p_1, \dots, p_k , construct the partial order Boolean game as in the proof of Proposition 9.

Then, if φ is not satisfiable, we find that for all three players, each action profile yields a utility of 1. It thus immediately follows that the game is of adequate information, as we can set, for instance, $u_i^0(\mathbf{a}_S) = 1$ for all players i with set S of scrutable vertices under τ .

For the opposite direction, assume that assignment $\alpha: \{p_1, \dots, p_m\} \rightarrow \{0, 1\}$ witnesses the satisfiability of φ . Observe that v_q is inscrutable to player 3 under τ , whereas $v_q, v_{p_1}, \dots, v_{p_m}$ are scrutable to player 3. Now let $a_{v_{p_1}}, \dots, a_{v_{p_m}}$ be such that $a_{p_j} = p_j$ if $\alpha(p_j) = 1$, and $a_{p_j} = \bar{p}_j$ if $\alpha(p_j) = 0$. Then,

$$u_3(q, a_{v_{p_1}}, \dots, a_{v_{p_m}}, r) > u_3(q, a_{v_{p_1}}, \dots, a_{v_{p_m}}, \bar{r})$$

but

$$u_3(\bar{q}, a_{v_{p_1}}, \dots, a_{v_{p_m}}, r) < u_3(\bar{q}, a_{v_{p_1}}, \dots, a_{v_{p_m}}, \bar{r}).$$

Assuming that the game be of adequate information, there then should be a utility function $u_3^0: \mathbf{A}_S \rightarrow \mathbb{R}$ such that from the former it would follow that $u_3^0(a_{v_{p_1}}, \dots, a_{v_{p_m}}, r) >$

$u_3^0(a_{v_{p_1}}, \dots, a_{v_{p_m}}, \bar{r})$, whereas from the latter $u_3^0(a_{v_{p_1}}, \dots, a_{v_{p_m}}, r) < u_3^0(a_{v_{p_1}}, \dots, a_{v_{p_m}}, \bar{r})$, a contradiction. Accordingly, we may conclude that the game is not of adequate information. \square

Using Theorems 7 and 8, we know that games that are fit for backwards induction, like games of adequate information, have Nash equilibria. Thus, Theorems 9 and 10 could be seen to prove coNP-completeness of weak versions of the non-emptiness problem for backwards induction. This still leaves the question of whether a given strategy profile is a backwards induction solution and of whether a given action profile is sustained by a backwards induction solution, which can be seen as variants of the non-emptiness problem for backwards induction. The first decision problem can be formulated formally as follows.

IS BACKWARD INDUCTION

- Given:* Partial order game G that is fit for backwards induction under topological sorting τ and strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$
Problem: Is \mathbf{f} a backwards induction solution of G under τ ?

We find that IS BACKWARD INDUCTION is coNP-complete. For the membership part, recall that Lemma 5 showed that our backwards induction procedure has the *one-stage-deviation property*. Let τ be a topological sorted and assume $(v_1, \dots, v_{|V|})$ be sorted according to τ . The contrapositive of the one-stage-deviation property says that, if a profile \mathbf{f} is not a backwards induction solution, there is one player, who, after some τ -history, wants to unilaterally deviate and play another action than the one prescribed by \mathbf{f} , but otherwise has no incentive to change their strategy. This prepares the ground for the following result.

Theorem 11. IS BACKWARD INDUCTION is coNP-complete. The problem remains coNP-hard for total-order games.

Proof. For membership in coNP, one can design a non-deterministic algorithm along the following lines. Assume $v_{v_1}, \dots, v_{v_{|V|}}$ is ordered according to τ . Guess a vertex v_k assigned to player i along with action profiles $\mathbf{a} = (\mathbf{a}_X, \mathbf{a}_Y, a_{v_k}, \mathbf{a}_Z)$ and $\mathbf{a}' = (\mathbf{a}_X, \mathbf{a}_Y, a'_{v_k}, \mathbf{a}'_Z)$ (thus, $\mathbf{a}'_X = \mathbf{a}_X$ and $\mathbf{a}'_Y = \mathbf{a}_Y$). Let Y denote the parents of v_k , Z the vertices with a greater topological index than v_k , and $X = V \setminus (Y \cup Z \cup \{v_k\})$. Then, check whether

- (i) the actions $a_{v_k}, a_{v_{k+1}}, \dots, a_{v_{|V|}}$ are in accordance with $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$, that is, for every $k \leq m \leq |V|$, we have $a_{v_m} = f_{v_m}(\mathbf{a}_{Y_m})$, where Y_m is the set of parents of v_m ;
- (ii) the actions $a'_{v_{k+1}}, \dots, a'_{v_{|V|}}$ are in accordance with $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$, that is, for every $k < m \leq |V|$, we have $a'_{v_m} = f_{v_m}(\mathbf{a}'_{Y_m})$, where Y_m is the set of parents of v_m ; and
- (iii) $u_i(\mathbf{a}') > u_i(\mathbf{a})$.

In virtue of Lemmas 1 and 2, each of these checks can be performed in polynomial time in the size of the input. Moreover, if all of these checks are positive, we have found a counterexample against \mathbf{f} being a backwards induction solution.

For coNP-hardness, we reduce the complement of SATISFIABILITY. Given an Boolean formula φ on propositional variables p_1, \dots, p_k , we construct a partial order Boolean game G_φ with one player i with goal formula $\gamma_i = \varphi \wedge p_0$, where p_0 is a “fresh” variable distinct from any p_1, \dots, p_k . Let the vertices v_{p_0}, \dots, v_{p_k} be ordered accordingly, that is, $p_\ell \in p_m$ if and only if $\ell < m$. Thus, G_φ is a totally-ordered Boolean game, and, consequently, also fit for backwards induction. Now, consider the strategy profile $\mathbf{f}^0 = (f_{v_{p_0}}^0, \dots, f_{v_{p_k}}^0)$ where each $f_{v_{p_j}}^0$ is represented by the choice equation $p_j: \perp$, that is, \mathbf{f}^0 unconditionally sets every propositional variable to false. Note that \mathbf{f}^0 yields player i a utility of 0, that is $u_i(\mathbf{f}^0) = 0$.

Now, if φ is unsatisfiable, so is i 's goal γ_i , and all strategy profiles yield i the same payoff. Therefore, \mathbf{f}^0 is vacuously a backwards induction solution. If, on the other hand, φ is satisfiable, say in virtue of assignment α , let $\mathbf{f}^\alpha = (f_{v_{p_0}}, \dots, f_{v_{p_k}})$ be the unconditional strategy profile represented by the choice equations $p_0 = \top$ and, for each $1 \leq m \leq k$, $p_m: \top$, if $\alpha(p_m) = 1$, and $p_m: \perp$, if $\alpha(p_m) = 0$. It can then easily be verified that player i has $u_i(\mathbf{f}^\alpha) = 1$. Accordingly, player i has an incentive to deviate from \mathbf{f}^0 to \mathbf{f}^α , indicating

that \mathbf{f}^0 is not a Nash equilibrium. By Theorem 7, it then follows that \mathbf{f}^0 is not a backwards induction solution either. \square

Finally, we come to consider the natural counter-part of the IS NASH ACTIONS problem for the backwards induction procedure, which can formally be stated as follows.

IS BACKWARD INDUCTION ACTIONS

Given: Partial order game G , topological sorting τ , and action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$

Problem: Is \mathbf{a} sustained by a backwards induction solution under τ in G ?

We find that BACKWARDS INDUCTION ACTIONS is NEXPTIME-complete for general partial order games. The hardness part of this result follows again from a reduction from DEPENDENCY QUANTIFIER BOOLEAN FORMULA GAME (DQBFG).

Theorem 12. For partial order games, BACKWARD INDUCTION ACTIONS is NEXPTIME-complete.

Proof. A NEXPTIME algorithm to decide BACKWARD INDUCTION ACTIONS can be designed along the following lines. Given a partial order game G , a topological-ordering $\tau = (v_1, \dots, v_{|V|})$ of its vertices, and an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$, first, guess a strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$. Given that strategies f_v are given by a Boolean circuit \mathcal{C}_{f_v} , this can be achieved in time not more than exponential in $|A|$, the size of the set of actions. Second, check whether $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a backwards induction solution. The inductive definition of a backwards induction solution suggests a procedure for how this can be achieved in exponential time. Finally, check whether action profile \mathbf{a} as induced by strategy profile \mathbf{f} coincides with the action profile \mathbf{a} as it is given in the input. This can be achieved in time polynomial in the size of the Boolean circuit representing \mathbf{f} . Altogether, the algorithm runs in non-deterministic exponential time.

For hardness, we reduce from DEPENDENCY QUANTIFIER BOOLEAN FORMULA GAME (DQFG) using a similar construction of a Boolean partial order game as in the proof of Theorem 4. Given an instance $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ of DQFG with $X_1 = \{x_1^1, \dots, x_{|X_1|}^1\}$, $X_2 = \{x_1^2, \dots, x_{|X_2|}^2\}$, $Y_1 = \{y_1^1, \dots, y_{|Y_1|}^1\}$, and $Y_2 = \{y_1^2, \dots, y_{|Y_2|}^2\}$. Let x_0 be a ‘fresh’ variable x_0 not in $X_1 \cup X_2 \cup Y_1 \cup Y_2$. We define a Boolean partial order game on dependency graph (V, E) such that $V = \{x_0\} \cup X_1 \cup X_2 \cup Y_1 \cup Y_2$ and where E is defined as follows:

- (i) $x_0 E z$ for all $z \in X_1 \cup X_2 \cup Y_1 \cup Y_2$,
- (ii) $x_k^i E x_m^j$ for all $x_k^i \in X_i$ and $x_m^j \in X_j$ with $i = j$ and $k < m$, or $i \leq j$,
- (iii) $x_1 E x_2$, for all $x_1 \in X_1$ and $x_2 \in X_2$,
- (iv) $x_1 E y_1$, for all $x_1 \in X_1$ and $y_1 \in Y_1$,
- (v) $x_2 E y_2$, for all $x_2 \in X_2$ and $y_2 \in Y_2$.

Observe that under this definition on $X_1 \cup X_2$, the relation E is complete and transitive. We extend E to a topological sorting τ such that, for instance,

$$\tau = (x_0, x_1^1, \dots, x_{|X_1|}^1, x_1^2, \dots, x_{|X_2|}^2, y_1^1, \dots, y_{|Y_1|}^1, y_1^2, \dots, y_{|Y_2|}^2).$$

The players of the Boolean game are the same as in the instance of DQBFG, where B controls $\{x_0\} \cup X_1 \cup X_2$, W_1 the variables in Y_1 , and W_2 those in Y_2 . Their goals are given by

$$\gamma_B = x_0 \wedge \varphi \qquad \gamma_{W_1} = \gamma_{W_2} = \perp.$$

Observe that all vertices are scrutable to player B under τ and that the white players have the same utility for every outcome. It therefore follows that the game constructed is of adequate information and hence fit for backwards induction.

We are now in a position to show that the profile

$$\bar{\mathbf{a}} = (\bar{x}_0, \bar{x}_1^1, \dots, \bar{x}_{|X_1|}^1, \bar{x}_1^2, \dots, \bar{x}_{|X_2|}^2, \bar{y}_1^1, \dots, \bar{y}_{|Y_1|}^1, \bar{y}_1^2, \dots, \bar{y}_{|Y_2|}^2),$$

which sets all variables, including x_0 , to false, is sustained by a Nash equilibrium in G' if and only if $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ is a postive instance of DQBFG.

First, assume $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ is a postive instance of DQBFG. Then, the white team has a joint winning strategy in the original DQBFG-game given by Boolean functions β_1 and β_2 on the variables X_1 and X_2 , respectively. Next, define strategies $\mathbf{f}_{W_1}^*$ and $\mathbf{f}_{W_2}^*$ for W_1 and W_2 in the Boolean partial order game that are given by the following choice equations for $i = 1, 2$ and $j = 1, \dots, |Y_i|$,

$$y_j^i: (x_0 \rightarrow \varphi_{\beta_i}(x_1^i, \dots, x_{|X_i|}^i)) \wedge (\bar{x}_0 \rightarrow \perp)$$

As the dependency graph (V, E) respects the information dependencies of the DQBFG-instance, observe that \mathbf{f}_{W_1} and \mathbf{f}_{W_2} together embody a winning strategy to render φ false if x_0 is played. Let B 's unconditional strategy \mathbf{f}_B^* be defined by the choice equations of the form $x_j^i: \perp$, setting x_j^i to false for all for $i = 1, 2$ and $j = 1, \dots, |X_i|$. Observe that for the strategy profile $\mathbf{f}^* = (\mathbf{f}_B^*, \mathbf{f}_{W_1}^*, \mathbf{f}_{W_2}^*)$, we have that $\mathbf{a}^{\mathbf{f}^*} = \bar{\mathbf{a}}$, as desired. As W_1 and W_2 will not have their goal achieved under any action profile, their strategies trivially conform with the backwards induction procedure. Observe furthermore that B does not get their goal satisfied no matter which strategy plays against $\mathbf{f}_{W_1}^*$ and $\mathbf{f}_{W_2}^*$. If B sets x_0 to false, their goal is not satisfied in a self-defeating fashion. By setting x_0 to true, B will induce W_1 and W_2 to play a winning strategy rendering $\neg\varphi$ false and therewith γ_B as well. This in particular holds for any backwards induction strategies B has at their disposal, and it follows that \mathbf{f}^* is a backwards induction solution sustaining $\bar{\mathbf{a}}$.

For the opposite direction, assume that $\langle \varphi, X_1, X_2, Y_1, Y_2 \rangle$ is a negative instance of DQBFG. Analogously to the proof of Theorem 4, it can then be shown that the constructed partial order Boolean game does not have a Nash equilibrium sustaining $\bar{\mathbf{a}}$. In virtue of the contrapositive of Theorem 7, there is no backwards induction solution sustaining $\bar{\mathbf{a}}$, giving us the result. \square

The computational complexity of BACKWARD INDUCTION ACTIONS reduces considerably when we restrict attention to total-order games. Still, the problem is PSPACE-complete for this class of games. To prove this result, we first introduce some auxiliary notation and terminology.

Recall that if the dependency graph (V, E) of a total-order game is such that $v_0 E \dots E v_{|V|}$, then $\tau = (v_0, \dots, v_{|V|})$ is the only topological sorting of the vertices. Moreover, the parents of each vertex v_k are all vertices with a lower topological index. The set $\{v_1, \dots, v_{k-1}\}$ of parents of vertex v_k we will denote by Y_k , on the understanding that $Y_k = \emptyset$, if $k = 0$. Similarly, we will denote by $Z_k = \{v_{k+1}, \dots, v_{|V|}\}$ the set of vertices with a higher topological index than v_k , here on the understanding that $Z_k = \emptyset$, if $k = |V|$.

For $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ an action profile and $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$ a τ -history of \mathbf{a} , we now introduce the auxiliary concept of \mathbf{a} being sustained by a backwards induction solution at history \mathbf{h} of \mathbf{a} . This enables us to reason recursively about action profiles being sustained by backwards induction solutions without having to make explicit reference to specific underlying strategy profiles. Formally, for action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ and τ -history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$ of \mathbf{a} , we say that \mathbf{a} is sustained (by backwards induction) at \mathbf{h} , if either $\mathbf{h} = \mathbf{a}$, or $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$ for some $0 \leq k < |V|$ and the following two conditions hold:

- (a.i) action profile \mathbf{a} is sustained by backwards induction at history $(a_{v_1}, \dots, a_{v_k}, a_{v_{k+1}})$,
- (a.ii) for every $b_{v_{k+1}} \in A_{v_{k+1}}$, there is an action profile $\mathbf{a}' = (a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, b_{v_{k+2}}, \dots, b_{v_{|V|}})$ that is sustained by backwards induction at history $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}})$ and which is such that $u_i(\mathbf{a}) \geq u_i(\mathbf{a}')$, where i is the player active at v_{k+1} .

We now have the following lemma.

Lemma 7. For total-order games, an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ is sustained by a backwards induction solution if and only if \mathbf{a} is sustained by backwards induction at all histories of \mathbf{a} . Moreover,

\mathbf{a} is sustained by a backwards induction solution only if \mathbf{a} is sustained by backwards induction at the empty history $()$.

Proof. As we are dealing with total-order games, first observe that, for each vertex v , the set X of vertices that are neither a parent of v nor have a greater topological index than v is empty. Thus, Condition (1) in the definition of a backwards induction solution \mathbf{f}^* for partial order games reduces to

$$f_{v_{k+1}}^*(a_{v_1}, \dots, a_{v_k}) \in \arg \max_{b_{v_{k+1}} \in A_{v_{k+1}}} u_i(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}}), \tag{BI'}$$

for all $0 \leq k < |V|$, and where i is the player active at v_{k+1} .

As an auxiliary concept, we define a strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ to sustain action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ (by backwards induction) at history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$, if either $\mathbf{a} = \mathbf{h}$, or both

- (f.i) \mathbf{f} sustains \mathbf{a} at $(a_{v_1}, \dots, a_{v_k}, a_{v_{k+1}})$,
- (f.ii) $f_{v_{k+1}}(a_{v_1}, \dots, a_{v_k}) = a_{v_{k+1}}$, and
- (f.iii) for all $b_{v_{k+1}} \in A_{v_{k+1}}$ with $b_{v_{k+1}} \neq a_{v_{k+1}}$, we have that \mathbf{f} sustains action profile $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}})$ at history $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}})$ and,

$$u_i(a_{v_1}, \dots, a_{v_k}, a_{v_{k+1}}, a_{v_{k+2}}, \dots, a_{v_{|V|}}) \geq u_i(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, f_{v_{k+1}}, \dots, f_{v_{|V|}}),$$

where i is the player active at v_{k+1} .

The we split the proof up in three parts, where we prove, respectively, the following statements (I), (II), and (III), below. In conclusion then put them together again, to prove the lemma.

- (I) for all $0 \leq k \leq |V|$ and all action profiles $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$, \mathbf{a} is sustained at $(a_{v_1}, \dots, a_{v_k})$ if and only if some strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ sustains \mathbf{a} at $(a_{v_1}, \dots, a_{v_k})$.
- (II) strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ sustains action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ at history $()$ if and only if $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$ and for every $0 \leq k \leq |V|$ and all action profiles $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$, strategy profile \mathbf{f} sustains \mathbf{x} at $(x_{v_1}, \dots, x_{v_k})$.
- (III) for every $0 \leq k \leq |V|$ and every action profile $\mathbf{x}' = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$, strategy profile \mathbf{f} sustains \mathbf{x} at history $\mathbf{h} = (x_{v_1}, \dots, x_{v_k})$ if and only if strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a backwards induction solution.

Part (I) is by induction on $|V| - k$. For the basis, where $k = |V|$, consider an arbitrary action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ and arbitrary history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$. In this case, we find that $\mathbf{h} = \mathbf{a}$ and immediately that \mathbf{a} is sustained at \mathbf{h} . Moreover, every strategy profile sustains \mathbf{a} at \mathbf{h} in this case.

For the induction step, consider an arbitrary $1 \leq k < |V|$, and equally arbitrary action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ and history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$.

First, assume that \mathbf{a} is sustained at (a_{v_1}, \dots, v_k) , and let i be the player active at v_{k+1} . Then, \mathbf{a} is also sustained at $(a_{v_1}, \dots, v_{k+1})$. By the induction hypothesis, there is a strategy profile \mathbf{f} that sustains \mathbf{a} at $(a_{v_1}, \dots, v_{k+1})$. Moreover, as $k < |V|$, for each $b_{v_{k+1}} \in A_{v_{k+1}}$ with $b_{v_{k+1}} \neq a_{v_{k+1}}$, there is an action profile $\mathbf{a}' = (a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, b_{v_{k+2}}, \dots, b_{v_{|V|}})$ that is sustained at history $(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, b_{v_{k+1}})$ such that $u_i(\mathbf{a}) \geq u_i(\mathbf{a}')$. Let $A_{v_{k+1}} = \{b_{v_{k+1}}^1, \dots, b_{v_{k+1}}^{|A_{v_{k+1}}|}\}$. By the induction hypothesis, for each $1 \leq j \leq |A_{v_{k+1}}|$, there is a strategy profile $\mathbf{g}^j = (g_{v_1}^j, \dots, g_{v_{|V|}}^j)$ sustaining action profile $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}^j, g_{v_{k+2}}^j, \dots, g_{v_{|V|}}^j)$ at history $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}^j)$, and which is such that

$$u_i(\mathbf{a}) \geq u_i(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}^j, g_{v_{k+1}}^j, \dots, g_{v_{|V|}}^j).$$

Now, define the strategy profile $\mathbf{f}^* = (f_{v_1}^*, \dots, f_{v_{|V|}}^*)$ such that for $1 \leq k \leq |V|$ and every history $\mathbf{x} = (x_{v_1}, \dots, x_m)$, where $0 \leq m \leq |V|$,

$$f_{v_m}^*(\mathbf{x}) = \begin{cases} f_{v_m}^j(\mathbf{x}) & \text{if } x_{v_{k+1}} = a_{v_{k+1}}^j \text{ and } m > k, \\ g_{v_m}^j(\mathbf{x}) & \text{if } x_{v_{k+1}} = b_{v_{k+1}}^j \text{ and } m > k, \\ a_{v_m} & \text{otherwise.} \end{cases}$$

It is now easy to check that, by construction, strategy profile \mathbf{f}^* sustains \mathbf{a} at history $(a_{v_1}, \dots, a_{v_k})$, as desired. In particular, observe that $f_{v_{k+1}}^*(a_{v_1}, \dots, a_{v_k}) = a_{v_{k+1}}$.

For the opposite direction, assume that strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ sustains action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ at history $(a_{v_1}, \dots, a_{v_k})$, where $k < |V|$. Then, by definition, strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ also sustains action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ at history $(a_{v_1}, \dots, a_{v_{k-1}})$. By the induction hypothesis, we then immediately obtain that \mathbf{a} is sustained at history $(a_{v_1}, \dots, a_{v_{k+1}})$. Now, consider an arbitrary $b_{v_{k+1}} \in A_{v_{k+1}}$ with $a_{v_{k+1}} \neq b_{v_{k+1}}$, along with history $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}})$. Then, \mathbf{f} sustains action profile $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}})$ at history $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}})$. We have, moreover, that $u_i(\mathbf{a}) \geq u_i((a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}}))$. By the induction hypothesis, we find that action profile $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}})$ is sustained at history $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}})$, which proves the case.

For part (II), first assume that $\mathbf{a}^f = \mathbf{a}$ and that for every $0 \leq k \leq |V|$, strategy profile \mathbf{f} sustains action profile $(x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at history $(x_{v_1}, \dots, x_{v_k})$. The latter then holds in particular for $k = 0$, that is, for history $()$. Accordingly, \mathbf{f} sustains action profile $(f_{v_1}, \dots, f_{v_{|V|}}) = \mathbf{a}^f$ at history $()$. Our initial assumption that \mathbf{a}^f then gives the result for this direction.

For the opposite direction, assume that strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ sustains action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ at history $()$. First, we find that by repeatedly applying condition (f.ii) gives us $\mathbf{a}^f = \mathbf{a}$. Now, consider an arbitrary $0 \leq k \leq |V|$ and action profile $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$. We prove by induction on k that \mathbf{f} sustains $(x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at history $\mathbf{h} = (x_{v_1}, \dots, x_{v_k})$.

First assume that $k = 0$, that is, $\mathbf{h} = ()$. In that case, $\mathbf{x} = (f_{v_1}, \dots, f_{v_{|V|}}) = \mathbf{a}^f$. We have already seen that $\mathbf{a}^f = \mathbf{a}$, and, thus, the base case immediately follows from the assumption that $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ sustains action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ at history $()$.

For the induction step, let $k > 0$. In case $k = |V|$ we have $\mathbf{h} = \mathbf{a}$ and we are done immediately. Otherwise, by the induction hypothesis, we may assume that \mathbf{f} sustains action profile $(x_{v_1}, \dots, x_{v_{k-1}}, f_{v_k}, \dots, f_{v_{|V|}})$ at history $(x_{v_1}, \dots, x_{v_{k-1}})$. If $f_{v_k}(x_{v_1}, \dots, x_{v_{k-1}}) = x_k$, then it follows from condition (f.i) that \mathbf{f} sustains $(x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at $(x_{v_1}, \dots, x_{v_k})$. If $f_{v_k}(x_{v_1}, \dots, x_{v_{k-1}}) \neq x_k$, then still $f_{v_k}(x_{v_1}, \dots, x_{v_{k-1}}) \in A_{v_k}$. Now, it follows from condition (f.iii) that \mathbf{f} sustains $(x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at $(x_{v_1}, \dots, x_{v_k})$.

For part (III), first assume, for every $0 \leq k \leq |V|$ and action profile $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$, that strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ sustains \mathbf{x} at history $\mathbf{h} = (x_{v_1}, \dots, x_{v_k})$. Now, consider an arbitrary history (z_1, \dots, z_k) for some $1 \leq k < |V|$. Then, by assumption, \mathbf{f} sustains action profile $(z_{v_1}, \dots, z_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at history $(z_{v_1}, \dots, z_{v_k})$. As $k < |V|$, we find that $(z_1, \dots, z_k) \neq (z_1, \dots, z_k, f_{v_{k+1}}, \dots, f_{v_{|V|}})$. Therefore, by condition (f.iii), we then find that

$$f_{v_{k+1}}(z_{v_1}, \dots, z_{v_k}) \in \arg \max_{b_{v_{k+1}} \in A_{v_{k+1}}} u_i(z_{v_1}, \dots, z_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}}),$$

where i is the player active at v_{k+1} . It thus follows that \mathbf{f} is a backward induction solution.

For the opposite direction, assume that $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ is a backwards induction solution. Now, consider an arbitrary $0 \leq k \leq |V|$. We prove by induction on $|V| - k$, that,

for every action profile $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$, strategy profile \mathbf{f} sustains \mathbf{x} at history $\mathbf{h} = (x_{v_1}, \dots, x_{v_k})$.

For the basis assume let $k = |V|$. Consider an arbitrary action profile $\mathbf{x} = (x_{v_1}, \dots, x_{v_k})$ along with history $\mathbf{h} = (x_{v_1}, \dots, x_{v_k})$. Obviously, $\mathbf{h} = \mathbf{x}$, and, thus, immediately that \mathbf{f} sustains \mathbf{x} at history \mathbf{h} .

For the induction step, let $k < |V|$. Again, consider an arbitrary action profile $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ along with history $\mathbf{h} = (x_{v_1}, \dots, x_{v_k})$. By the induction hypothesis, we find, for all action profiles $\mathbf{y} = (y_{v_1}, \dots, y_{v_{k+1}}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$, that \mathbf{f} sustains \mathbf{y} at history $(y_{v_1}, \dots, y_{v_{k+1}})$. This holds in particular for histories $(x_{v_1}, \dots, x_{v_k}, b_{v_{k+1}})$ where $b_{v_{k+1}} \in A_{v_{k+1}}$. For $b_{v_{k+1}} = f_{v_{k+1}}(x_{v_1}, \dots, x_{v_k})$, we obtain condition (f.i). Condition (f.ii) is trivially satisfied. Finally, having assumed that \mathbf{f} is a backwards induction solution, we find that,

$$u_i(x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}}) \geq u_i(x_{v_1}, \dots, x_{v_k}, b_{v_{k+1}}, f_{v_{k+2}}, \dots, f_{v_{|V|}}),$$

for all $b_{v_{k+1}} \in V_{v_{k+1}}$ with $b_{v_{k+1}} \neq f_{v_{k+1}}(x_{v_1}, \dots, x_{v_k})$ and i the player active at v_{k+1} . We may therefore conclude that \mathbf{f} sustains $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at $(x_{v_1}, \dots, x_{v_k})$, proving part (III).

With (I) through (III) in place, we conclude the proof as follows. Consider an arbitrary action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$. First, assume, for all $0 \leq k \leq |V|$, that \mathbf{a} is sustained by backwards induction at $(a_{v_1}, \dots, a_{v_k})$. Then by (I), for all $0 \leq k \leq |V|$, there is some strategy profile $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ that sustains \mathbf{a} by backwards induction at $(a_{v_1}, \dots, a_{v_k})$. For the particular case wherein $k = 0$, we find that this strategy profile \mathbf{f} that sustains \mathbf{a} by backwards induction at (\cdot) . By repeated application of (f.ii), moreover, it can easily be established that $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$. Part (II) then yields that for every $0 \leq k \leq |V|$ and all action profiles $\mathbf{x} = (x_{v_1}, \dots, x_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$, strategy profile \mathbf{f} sustains \mathbf{a} at history $(x_{v_1}, \dots, x_{v_k})$. Finally, by virtue of (III), we may conclude that \mathbf{f} is a backwards induction solution that sustains \mathbf{a} .

For the opposite direction, assume that \mathbf{f} is a backwards induction solution sustaining \mathbf{a} . Therefore, $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$, and, accordingly, $(a_{v_1}, \dots, a_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}}) = (a_{v_1}, \dots, a_{v_{|V|}})$ for all $0 \leq k \leq |V|$. Part (III) yields furthermore that for every $0 \leq k \leq |V|$, strategy profile \mathbf{f} sustains action profile $\mathbf{a}' = (a_{v_1}, \dots, a_{v_k}, f_{v_{k+1}}, \dots, f_{v_{|V|}})$ at history $(a_{v_1}, \dots, a_{v_k})$. It now follows, for every $0 \leq k \leq |V|$, that \mathbf{f} sustains \mathbf{a} at history $(a_{v_1}, \dots, a_{v_k})$. With part (I), we may now conclude that, for all $0 \leq k \leq |V|$, action profile \mathbf{a} is sustained by backwards induction at $(a_{v_1}, \dots, a_{v_k})$, as desired.

Finally, observe that the second part of the lemma also follows from the above argument. \square

The intuition behind Lemma 7 is that, for a strategy profile \mathbf{f} sustaining an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ to be a backward induction solution, it does not suffice just to induce \mathbf{a} , in the sense that $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$. It should, in addition, incorporate a *threat strategy* at every vertex v_k that succeeds in deterring the player active at v_k to play any action other than a_{v_k} , that is, the action prescribed by \mathbf{a} . The lemma, moreover, shows that a strategy profile that achieve this can be constructed from the strategy profiles that achieve this at each history.

We introduce the following auxiliary decision problem, solving which is an important subroutine in our proof of PSPACE-membership of BACKWARDS INDUCTION ACTIONS for total-order games.

BACKWARD INDUCTION GUARANTEE

- Given:* Total-order game G , action profiles $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ and history $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$, player i , and $r \in \mathbb{R}$.
- Problem:* Is \mathbf{a} sustained by backwards induction at \mathbf{h} such that $u_i(\mathbf{a}) \leq r$?

We find that BACKWARD INDUCTION GUARANTEE is in PSPACE for total-order games.

Lemma 8. *For total-order games, BACKWARD INDUCTION GUARANTEE is in PSPACE, that is, the problem can be solved in space polynomial in the size of the game.*

Proof. Let a total-order game be given together with an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$. We prove by induction on $|V| - k$, that for all histories $\mathbf{h} = (a_{v_1}, \dots, a_{v_k})$ of \mathbf{a} , all players i , and all values $r \in \mathbb{R}$ the problem BACKWARD INDUCTION GUARANTEE can be solved in polynomial space.

For the basis, assume $k = |V|$. Then, $\mathbf{h} = \mathbf{a}$, and consider an arbitrary player i and arbitrary value $r \in \mathbb{R}$. In this case, it suffices to compute $u_i(\mathbf{a})$ and check whether $u_i(\mathbf{a}) \leq r$. By Lemma 2, this achieved in polynomial time, and hence in polynomial space as well.

For the induction step, let $k < |V|$. Consider an arbitrary player j and an arbitrary value $r \in \mathbb{R}$. Let, furthermore, i be the player active at a_{v_k} . Now, we first check whether \mathbf{a} is sustained by backward induction at history $\mathbf{h}a_{v_{k+1}} = (a_{v_1}, \dots, a_{v_{k+1}})$. By virtue of the induction hypothesis, this can be achieved in polynomial space by computing BACKWARD INDUCTION GUARANTEE for \mathbf{a} , history $\mathbf{h}a_{v_{k+1}}$, player i , and value $r = u_i(\mathbf{a})$. If this check is negative, we can conclude immediately and output “no”.

Otherwise, we deploy two binary counters with, respectively, $\log_2 |A(x)|$ and $\log_2 |H| = O(|V| \log_2 |A|)$ digits to run through all actions $b_{v_{k+1}}$ in A_{v_k} and all action profiles

$$(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, b'_{k+1}, \dots, b'_{v_{|V|}}),$$

respectively. We then subsequently check for each $b_{v_{k+1}}$ in $A_{v_{k+1}}$ whether some action profile $\mathbf{b} = (a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}}, b'_{k+1}, \dots, b'_{v_{|V|}})$ is sustained by backwards induction at $(a_{v_1}, \dots, a_{v_k}, b_{v_{k+1}})$ with $u_i(\mathbf{b}) \leq u_i(\mathbf{a})$. Then, \mathbf{a} is sustained by backwards induction a history $\mathbf{h} = (a_{v_1}, \dots, a_{v_{k+1}})$ if and only if for each $b_{v_{k+1}}$ in $A_{v_{k+1}}$ at least one of these checks is positive. Moreover, as some reflection reveals, all of these checks are independent and, by the induction hypothesis, we may assume that all of them can be achieved in polynomial space, we may conclude that the subsequent execution of these checks can be performed in polynomial space as well, giving us the result. \square

With Lemma 8 in place, we are now in a position to now demonstrate the last main result of this section.

Theorem 13. *For total-order games, BACKWARDS INDUCTION ACTIONS is PSPACE-complete.*

Proof. For membership in PSPACE, let a total order game with $v_1 E \dots E v_{|V|}$ and an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ be given. Observe that the topological order $\tau = (v_1, \dots, v_{|V|})$ is fixed by E . By virtue of Lemma 7, it suffices to check whether \mathbf{a} is sustained by backwards induction at the empty history $()$ for value $k = u_i(\mathbf{a})$, where i is the player active at v_1 . Lemma 8 ensures that this can be achieved in space polynomial in the input.

For PSPACE-hardness, we reduce from QBF, which is known to be PSPACE-hard. Let $\Phi = Q_1 x_1, \dots, Q_{|X|} x_{|X|} \varphi(x_1, \dots, x_k)$ be a fully quantified propositional formula over $X = \{x_1, \dots, x_{|X|}\}$, where each Q_i is either \exists or \forall . We let X_\forall and X_\exists denote the sets of universally quantified variables and existentially quantified variables in Φ , respectively. Furthermore, for each variable x_i with $1 \leq i \leq |X|$, we let B_i denote the set of Boolean functions on the variables x_1, \dots, x_{i-1} . With each profile $\beta = (\beta_1, \dots, \beta_{|X|})$ in $B_1 \times \dots \times B_{|X|}$, we associate an a valuation $\alpha_\beta: X \rightarrow \{1, 0\}$ such that

$$\alpha_\beta(x_1) = \beta_1 \qquad \alpha_\beta(x_{i+1}) = \beta_{i+1}(\alpha_\beta(x_1), \dots, \alpha_\beta(x_{i-1})).$$

We construct a total-order Boolean game G^Φ with two players: player 1 and player 2. The propositional variables our game is based on are given by $x_1, \dots, x_{|X|}$ together with an additional ‘fresh’ variable x_0 . The players’ preferences are captured by

$$\gamma_1 = x_0 \wedge \neg \varphi \qquad \qquad \gamma_2 = x_0 \wedge \varphi.$$

We assign control over variable x_i to player 1, if $Q_i = \forall$, and to player 2, if $Q_i = \exists$. Moreover, the auxiliary variable x_0 is assigned to the control of player 1. The dependency graph $(\{x_0, \dots, x_k\}, E)$ is defined such that $x_i E x_j$ if and only if $i < j$. The game G^Φ is a total-order game and, thus, fit for backwards induction. Backwards induction solutions are therefore bound to exist. We now demonstrate that $\Phi = Q_1 x_1, \dots, Q_{|X|} x_{|X|} \varphi$ evaluates to true if and only if the action profile $\bar{\mathbf{x}} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{|X|})$ is sustained by a backwards induction solution in G^Φ .

First assume for the “only if”-direction, that Φ holds. Then, there is a profile of Boolean functions β_i for the variables in X_\exists , such that for all profiles Boolean functions β_j for the variables $x_j \in X_\forall$, such that the assignment $\alpha_{\beta_1, \dots, \beta_{|V|}} : X \rightarrow \{0, 1\}$ satisfies φ . On this basis we define a strategy profile $\mathbf{f}^* = (\mathbf{f}_1^*, \mathbf{f}_2^*)$, where \mathbf{f}_1^* and \mathbf{f}_2^* are given by the following choice equations for each $x_i \in X_\forall \cup \{x_0\}$ (controlled by player 1) and each $x_j \in X_\exists$ (controlled by player 2):

$$x_i : \perp \qquad \qquad x_j : x_0 \wedge \varphi_{\beta_j}(x_1, \dots, x_{j-1}).$$

Here, φ_{β_j} denotes the Boolean formula representing Boolean function β_j . Note, furthermore, that $\mathbf{a}^{\mathbf{f}^*} = \bar{\mathbf{x}}$.

First observe that for all action profiles $\mathbf{a} = (a_{x_0}, \dots, a_{x_{|X|}})$ with $a_{x_0} = \bar{x}_0$, the valuation $\alpha_{\mathbf{a}}$ will satisfy neither γ_1 nor γ_2 . Some reflection, thus reveals that \mathbf{f}^* sustains $\bar{\mathbf{x}}$ at all histories $\mathbf{h} = (a_{x_0}, \dots, a_{x_k})$ with $a_{v_{x_0}} = \bar{x}_0$, particular at history $a_{v_{x_0}}$. Now observe that \mathbf{f}_2 incorporates a winning strategy for player 2 in the “subgame” that results if player 1 sets x_0 to true. Therefore, φ will be satisfied by the valuations induced by the Nash equilibria in which x_0 is set to true—if any such Nash equilibria exist. As, by Theorem 7, every backwards induction solution is a Nash equilibrium, it follows that every action profile $\mathbf{a} = (a_{x_0}, \dots, a_{x_{|X|}})$ with $a_{x_0} = x_0$ that is sustained by backwards induction will render φ true, and, hence, player 1’s goal γ_1 false. Now let $\mathbf{f}' = (\mathbf{f}'_1, \mathbf{f}'_2)$ be a strategy profile that sustains some action profile $\mathbf{a}' = (a'_{x_0}, \dots, a'_{x_{|X|}})$ with $a'_{x_0} = x_0$ by backwards induction at history $\mathbf{h}' = x_0$. Observe that we may assume the existence of such a strategy profile. At this juncture, consider the strategy profile $\mathbf{f}^{**} = (f_{x_0}^{**}, \dots, f_{x_{|X|}}^{**})$ defined such that for all histories $\mathbf{h} = (a_{x_0}, \dots, a_{x_{k-1}})$ with $0 \leq k < |X|$ and $i \in \{1, 2\}$,

$$f_i^{**}(\mathbf{h}) = \begin{cases} f_1^*(\mathbf{h}) & \text{if } a_{x_0} = \bar{x}_0, \\ f'_2(\mathbf{h}) & \text{otherwise.} \end{cases}$$

Using the auxiliary formal apparatus introduced in the proof of Lemma 7, we find that \mathbf{f}^{**} sustains action profile $\bar{\mathbf{x}}$ at \bar{x}_0 and action profile $(x_0, f_{x_1}^{**}, \dots, f_{|V|}^{**})$ at x_0 . Moreover, $u_1(\bar{\mathbf{x}}) \geq u_1(x_0, f_{x_1}^{**}, \dots, f_{|X|}^{**})$. As player 1 controls x_0 , we may therefore conclude that \mathbf{f}^{**} sustains $\bar{\mathbf{x}}$ at (\cdot) . By Lemma 7, we may now conclude that $\bar{\mathbf{x}}$ is sustained by a backwards induction solution, as desired.

For the “if”-direction, assume that Φ does not hold. In this case, it can be seen that player 2 does not have a winning strategy in the “subgame” that results if x_0 is set to true. Now, consider an arbitrary strategy profile $\bar{\mathbf{f}} = (\bar{\mathbf{f}}_1, \bar{\mathbf{f}}_2)$ that sustains $\bar{\mathbf{x}}$. Then, player 1’s goal is not satisfied if $\bar{\mathbf{f}}$ is played. Moreover, as player 2 does not have a winning strategy in the “subgame” after x_0 is set to true, player 1 has some strategy \mathbf{f}_i^* with which they set x_0 to true and such that $(\mathbf{f}_1^*, \bar{\mathbf{f}}_2)$ guarantees γ_1 to be satisfied. Accordingly, player 1 will want to deviate to \mathbf{f}_1^* and strategy profile $\bar{\mathbf{f}}$ is not a Nash equilibrium. By Theorem 7, strategy

profile $\bar{\mathbf{f}}$ is not a backwards induction solution either. Having chosen $\bar{\mathbf{f}}$ arbitrarily, we get the result. \square

With this result in place, we can now present the proof of Theorem 6, which stated PSPACE-completeness of IS NASH ACTIONS for total-order games. We show that, given an action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$, a given total-order game G can be transformed into a total-order game $G^{\mathbf{a}}$ in which \mathbf{a} is sustained by a backwards induction solution if and only if \mathbf{a} is sustained by a Nash equilibrium in the original game G . We then leverage Theorem 13 to prove PSPACE-membership of IS NASH ACTIONS.

Formally, let G be a partial order game defined on dependency graph (V, E) , and let $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ be an action profile. Then, we define $G^{\mathbf{a}}$ as exactly the same game as G , be it with the following binary utility functions $u_i^{\mathbf{a}}: \mathbf{A} \rightarrow \{0, 1\}$. For a given action profile $\mathbf{b} = (b_{v_1}, \dots, b_{v_{|V|}})$, we say that player j is the first to deviate from \mathbf{a} (to \mathbf{b}) if, for some $1 \leq k \leq |V|$, player j is active at v_k , $b_{v_m} = a_{v_m}$ for all $m < k$, and $b_{v_k} \neq a_{v_k}$. For each player i , we set $u_i^{\mathbf{a}}(\mathbf{a}) = 1$ and, for every action profile $\mathbf{b} \neq \mathbf{a}$ where player j is the first player deviate from \mathbf{a} to \mathbf{b} , we define

$$u_i^{\mathbf{a}}(\mathbf{b}) = \begin{cases} 0 & \text{if } i = j \text{ and } u_j(\mathbf{b}) \leq u_j(\mathbf{a}), \\ 2 & \text{if } i = j \text{ and } u_j(\mathbf{b}) > u_j(\mathbf{a}), \\ 2 & \text{if } i \neq j \text{ and } u_j(\mathbf{b}) \leq u_j(\mathbf{a}), \\ 0 & \text{if } i \neq j \text{ and } u_j(\mathbf{b}) > u_j(\mathbf{a}). \end{cases}$$

We now have the following two lemmas, the first of which shows that the game $G^{\mathbf{a}}$ preserves the Nash equilibria of G .

Lemma 9. *Let $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$ be an action profile of a total-order game G . Then, \mathbf{a} is sustained by a Nash equilibrium in G if and only if \mathbf{a} is sustained by a Nash equilibrium in $G^{\mathbf{a}}$.*

Proof. First assume that $\mathbf{f} = (f_{v_1}, \dots, f_{v_{|V|}})$ be a Nash equilibrium sustaining \mathbf{a} in G . Therefore, $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$. For contradiction, also assume that \mathbf{f} does not do so in $G^{\mathbf{a}}$. Then, there is some player i and some strategy \mathbf{g}_i , such that $u_i^{\mathbf{a}}(\mathbf{a}) < u_i^{\mathbf{a}}(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)})$. Obviously, player i is the first to deviate from \mathbf{a} to $\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}$. Thus, $u_i^{\mathbf{a}}(\mathbf{a}) = 1$ and $u_i^{\mathbf{a}}(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}) = 2$, and, by definition of $u^{\mathbf{a}}$, we find that $u_i(\mathbf{a}^{\mathbf{f}}) < u_i(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)})$. It therefore follows that \mathbf{f} is not a Nash equilibrium of G , a contradiction.

Now assume that \mathbf{f} is a Nash equilibrium sustaining \mathbf{a} in $G^{\mathbf{a}}$. Therefore, $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$. It suffices to show that \mathbf{f} is also a Nash equilibrium in G . To this end, consider an arbitrary player i and strategy \mathbf{g}_i for i . If $\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)} = \mathbf{a}^{\mathbf{f}}$, we are done immediately. Otherwise, i is the first player to deviate from $\mathbf{a}^{\mathbf{f}}$ to $\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}$. Because \mathbf{f} is a Nash equilibrium in $G^{\mathbf{a}}$, we find that $u_i^{\mathbf{a}}(\mathbf{a}^{\mathbf{f}}) \geq u_i^{\mathbf{a}}(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)})$. As $\mathbf{a}^{\mathbf{f}} = \mathbf{a}$, we have $u_i^{\mathbf{a}}(\mathbf{a}^{\mathbf{f}}) = 1$. Observe that it now follows that $u_i^{\mathbf{a}}(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}) = 0$. Accordingly, $u_i(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}) \leq u_i(\mathbf{a})$, and thus $u_i(\mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}) \leq u_i(\mathbf{a}^{\mathbf{f}})$. We may conclude that \mathbf{f} is a Nash equilibrium in G , as desired. \square

For total-order game $G^{\mathbf{a}}$, we can furthermore show that if there is a Nash equilibrium that sustains action profile \mathbf{a} , then there is also a backwards induction that does the same. The converse of this statement is immediate, as every backwards induction solution is also a Nash equilibrium (Theorem 7).

Lemma 10. *Let \mathbf{a} be an action profile of a total-order game G . Then, \mathbf{a} is sustained by a Nash equilibrium in $G^{\mathbf{a}}$ if and only if \mathbf{a} is sustained by a backwards induction solution in $G^{\mathbf{a}}$.*

Proof. By virtue of Theorem 7, the “if”-direction of the lemma is immediate. For the opposite direction, assume that \mathbf{a} is sustained by a Nash equilibrium \mathbf{f}^* . For contradiction, also assume that \mathbf{a} is not sustained by any backwards induction solution. As we are dealing with total-form games, we may assume the existence of backwards induction solutions. These we

may also assume to be Nash equilibria in virtue of Theorem 7. Now, let $\mathbf{g} = (g_{v_1}, \dots, g_{v_{|V|}})$ be any backwards induction solution and let $\mathbf{a}^{\mathbf{g}} = \mathbf{b} = (b_{v_1}, \dots, b_{v_{|V|}})$. Furthermore, let $1 \leq k \leq |V|$ be the *greatest* index such that $g_{v_k}(a_{v_1}, \dots, a_{v_{k-1}}) \neq f_{v_k}(a_{v_1}, \dots, a_{v_{k-1}})$ and assume i to be the player active at v_k . We may also assume that \mathbf{f} is the backwards induction solution for which there are the *fewest* histories $\mathbf{h} = (x_{v_1}, \dots, x_{v_\ell})$ such that some player $j \neq i$ is active at v_ℓ and $f_{v_\ell}(\mathbf{h}) \neq g_{v_\ell}(\mathbf{h})$. Furthermore, observe that, by choice of k , we have that $\mathbf{a} = (a_{v_1}, \dots, a_{v_k}, g_{v_{k+1}}, \dots, g_{v_{|V|}})$. Having assumed that \mathbf{g} is a backwards induction solution, we find moreover that

$$1 = u_i(\mathbf{a}) = u_i(\mathbf{a}^{\mathbf{f}}) < u_i(\mathbf{a}^{\mathbf{g}}) = u_i(\mathbf{b}) = 2.$$

Now, observe that $\mathbf{a}^{\mathbf{g}} \neq \mathbf{a}^{(\mathbf{f}-i\mathbf{g}_i)}$; otherwise, \mathbf{f} would not be a Nash equilibrium. Accordingly, there must be some player $j \neq i$ and some vertex $v_{k'}$ with $k < k' \leq |V|$ and at which player j is active such that

$$f_{v_{k+1}}(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{k'-1}) \neq g_{v_{k+1}}(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{k'-1}).$$

Having assumed that \mathbf{g} is a backwards induction solution, we find that

$$u_j(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}}, a'_{v_{k'}}, g_{v_{k'+1}}, \dots, g_{v_{|V|}}) \leq u_j(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}}, b_{v_{k'}}, g_{v_{k'+1}}, \dots, g_{v_{|V|}}) = u_j(\mathbf{b}).$$

where $a'_{v_{k'}} = f_{v_{k'}}(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}})$. Because $a_{v_k} \neq b_{v_k}$ and $u_i(\mathbf{b}) = 2$, we have $u_j(\mathbf{b}) = 0$. It now follows that $u_j(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}}, a'_{v_{k'}}, g_{v_{k'+1}}, \dots, g_{v_{|V|}}) = 0$ as well. Accordingly,

$$u_j(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}}, a'_{v_{k'}}, g_{v_{k'+1}}, \dots, g_{v_{|V|}}) \geq u_j(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}}, b_{v_{k'}}, g_{v_{k'+1}}, \dots, g_{v_{|V|}}).$$

At this juncture, consider the strategy profile $\mathbf{g}' = (g'_{v_1}, \dots, g'_{v_{|V|}})$ defined such that for all vertices v_ℓ and all histories $\mathbf{h} = (x_{v_1}, \dots, x_{v_{\ell-1}})$,

$$g'_v(\mathbf{h}) = \begin{cases} a'_{v_{k'}} & \text{if } \ell = k' \text{ and } \mathbf{h} = (a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}}), \\ g_{v_\ell}(\mathbf{h}) & \text{otherwise.} \end{cases}$$

The key observation to make at this point is that \mathbf{g}' is a backwards induction solution just as well as \mathbf{g} is. As $a'_{v_{k'}} = f_{v_{k'}}(a_{v_1}, \dots, a_{v_{k-1}}, b_{v_k}, \dots, b_{v_{k'-1}})$, however, we find that for \mathbf{g}' the number of histories $\mathbf{h} = (x_{v_1}, \dots, x_{v_\ell})$ such that there some player $j \neq i$ that is active at v_ℓ and $f_{v_\ell}(\mathbf{h}) \neq g'_{v_\ell}(\mathbf{h})$ is one fewer than the same number of such histories for \mathbf{g} . This contradicts our minimality assumption regarding the latter. This concludes the proof. \square

We are now finally in a position to prove Theorem 6, and establish that IS NASH ACTIONS is PSPACE-complete for total-order games.

Proof of Theorem 6. For PSPACE-membership, first observe that, given total-order game G and action profile $\mathbf{a} = (a_{v_1}, \dots, a_{v_{|V|}})$, we can construct $G^{\mathbf{a}}$ in polynomial time. To see this, note for each player i , we have to transform the circuit C_{u_i} computing i 's utility function u_i in G to a circuit $C_{u_i^{\mathbf{a}}}$. This can be achieved by adding a polynomial number of gates to each C_{u_i} . We then check whether \mathbf{a} is sustained by a backwards induction solution in $G^{\mathbf{a}}$. Theorem 13 guarantees that this can be achieved in polynomial space. Lemma 10 then yields the result.

To prove PSPACE-hardness, we adapt the proof of Theorem 6 and reduce from QBF. Given an instance $\Phi = Q_1x_1, \dots, Q_{|X|}x_{|X|} \varphi(x_1, \dots, x_k)$ of QBF, we construct the game G^Φ as in Theorem 6. We demonstrate that $\Phi = Q_1x_1, \dots, Q_{|X|}x_{|X|} \varphi$ evaluates to true if and only if the action profile $\bar{\mathbf{x}} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{|X|})$ is sustained by a Nash equilibrium.

First assume that Φ holds. Then, there is a profile of Boolean functions β_i for the variables in X_\exists , such that for all profiles of Boolean functions β_j for the variables $x_j \in X_\forall$,

such that the assignment $\alpha_{\beta_1, \dots, \beta_{|V|}} : X \rightarrow \{0, 1\}$ satisfies φ . As in the proof of Theorem 6, we consider profile $\mathbf{f}^* = (\mathbf{f}_1^*, \mathbf{f}_2^*)$, where \mathbf{f}_1^* and \mathbf{f}_2^* are given by the following choice equations for each $x_i \in X_{\forall} \cup \{x_0\}$ (controlled by player 1) and each $x_j \in X_{\exists}$ (controlled by player 2):

$$x_i : \perp \qquad x_j : x_0 \wedge \varphi_{\beta_j}(x_1, \dots, x_{j-1}).$$

Note that $\mathbf{a}^{\mathbf{f}^*} = \bar{x}$, and so it suffices to show that \mathbf{f}^* is a Nash equilibrium. Recall that \mathbf{f}_2 incorporates a winning strategy for player 2 in the “subgame” that results if player 1 sets x_0 to true. It follows that for each of player 1’s strategies \mathbf{g}_1 , the assignment $\alpha_{\mathbf{a}(\mathbf{g}_1, \mathbf{f}_2^*)}$ associated with action profile $\mathbf{a}(\mathbf{g}_1, \mathbf{f}_2^*)$ either satisfies φ or does not satisfy x_0 . In either case, player 1’s goal is not satisfied, and hence does not want to deviate from \mathbf{f}^* . Player 2 does not want to deviate either, as, with player 1 playing \mathbf{f}_1^* , variable x_0 will be set to false and player 2’s goal γ_2 will not be satisfied no matter which strategy player 2 chooses. We may conclude that \mathbf{f}^* is a Nash equilibrium, as desired.

Finally, assume that Φ does not hold. In that case, player 2 does not have a winning strategy in the “subgame” that results if x_0 is set to true. Now consider an arbitrary strategy profile $\bar{\mathbf{f}} = (\bar{\mathbf{f}}_1, \bar{\mathbf{f}}_2)$ that sustains \bar{x} , and observe that $\bar{\mathbf{f}}$ does not lead to player 1’s goal being satisfied. As player 2 does not have a winning strategy when x_0 is set to true, there must be a strategy \mathbf{f}_1^* for player 1 such that $(\mathbf{f}_1^*, \bar{\mathbf{f}}_2)$ leads to player 1’s goal γ_1 becoming true. Accordingly, player 1 would like to deviate to \mathbf{f}_1^* , and $\bar{\mathbf{f}}$ is not a Nash equilibrium. We may therefore conclude that \bar{x} is not sustained by a Nash equilibrium, as desired. \square

8. Conclusions

Game theorists have developed a huge range of game models since the field was initiated nearly a century ago. Of these, the most important and prominent are normal form games and extensive form games. These models differ in the assumptions they make about the information available to players and their strategies. In this article, we have introduced a game model in which informational dependencies between decision nodes in a game are captured in what we call a dependence relation. This dependence relation explicitly states what information is available when making a choice at a decision node. Although our partial order games are strictly no more expressive than extensive form games of imperfect information, they can be exponentially more compact, and we believe they are much more transparent with respect to the information flow in a game.

The compact representation of non-cooperative settings as partial order games naturally evokes various questions concerning the complexity of computing solution concepts in this model. We have focussed on Nash equilibrium and backwards induction. We explored six decision problems surrounding these two solution concepts—NON-EMPTINESS, IS NASH, IS NASH ACTIONS, FIT FOR BACKWARDS INDUCTION, IS BACKWARDS INDUCTION, and IS BACKWARDS INDUCTION ACTIONS—for partial-games as well as for the two extremal subclasses of empty-order games and total-order games. Our results are summarised in Tables 1 and 2. For each of these problems, we find that computational costs are highest for general partial order games, rather than for the two extremal classes. This suggests that the structure of the dependence relation is a major source of complexity.

Table 1. Overview of complexity results surrounding Nash equilibrium in partial order games.

	NON-EMPTINESS	IS-NASH	IS-NASH ACTIONS
empty-order games	Σ_2^P -compl.	coNP-compl.	coNP-compl.
partial order games	NEXPTIME-compl.	coNP-compl.	NEXPTIME-compl.
total-order games	constant time	coNP-compl.	PSPACE-compl.

Table 2. Overview of complexity results surrounding backwards induction in partial order games.

	FIT FOR B-I	IS B-I	IS B-I ACTIONS
partial order games	coNP-compl.	coNP-compl.	NEXPTIME-compl.
total-order games	constant time	coNP-compl.	PSPACE-compl.

Furthermore, note the contrast in complexity between IS NASH and IS BACKWARDS INDUCTION, on the one hand, and IS NASH ACTIONS and IS BACKWARDS INDUCTION ACTIONS on the other. Other than in normal-form games, and arguably similar to extensive-form games, partial order games evince a clear distinction between strategy profiles and action profiles, with the former being much (exponentially) ‘larger’ objects than the latter, and of which there can also be exponentially more. Thus, IS NASH and IS NASH ACTIONS are similar problems, which basically only differ in that the former has a strategy profile as part of its input and the latter an action profile. We propose that the difference in size and number between strategy profiles and action profiles accounts for much of the contrast in computational cost between IS NASH and IS NASH ACTIONS. A similar remark applies to FIT FOR BACKWARDS INDUCTION and IS BACKWARDS INDUCTION ACTIONS.

For future work, it would be useful to develop software tools to support reasoning in partial order games, and from a theoretical perspective, it would also be interesting to consider restrictions on dependence graph structures that might lead to more tractable decision problems. Finally, of course, it would be valuable to look in more detail at applications of our model.

Author Contributions: Conceptualization, J.G., P.H., M.W., and V.Z.; methodology, J.G., P.H., M.W., and V.Z.; formal analysis, P.H., M.W., and V.Z.; investigation, P.H. and V.Z.; writing—original draft preparation, P.H. and V.Z.; writing—review and editing, all authors; supervision, M.W.; project administration, M.W.; funding acquisition, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: Wooldridge and Harrenstein were supported by the UKRI under a Turing AI World Leading Researcher Fellowship (EP/W002949/1) awarded to Wooldridge.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank James Fox, Lewis Hammond, and Thomas Steeples for discussions on this work, in particular on the relationship between partial order games and MAIDs.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bradfield, J.; Gutierrez, J.; Wooldridge, M. Partial-order Boolean games: Informational independence in a logic-based model of strategic interaction. *Synthese* **2016**, *193*, 781–811.
2. Koller, D.; Milch, B. Multi-agent influence diagrams for representing and solving games. In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), Seattle, WA, USA, 4–10 August 2001; pp. 1027–1034.
3. Koller, D.; Milch, B. Multi-agent influence diagrams for representing and solving games. *Games Econ. Behav.* **2003**, *45*, 181–221.
4. Maschler, M.; Solan, E.; Zamir, S. *Game Theory*; Cambridge University Press: Cambridge, UK, 2013.
5. Osborne, M.J.; Rubinstein, A. *A Course in Game Theory*; MIT Press: Cambridge, MA, USA, 1994.
6. Vollmer, H. *Introduction to Circuit Complexity: A Uniform Approach*; Texts in Theoretical Computer Science; Springer: Berlin/Heidelberg, Germany, 1999.
7. Jukna, S. *Boolean Function Complexity: Advances and Frontiers*; Algorithms and Combinatorics; Springer: Berlin/Heidelberg, Germany, 2012; Volume 27.
8. Greenlaw, R.; Hoover, H.J.; Ruzzo, W.L. *Limits to Parallel Computation: P-Completeness Theory*; Oxford University Press: Oxford, UK, 1995.
9. Muller, D.E. Complexity in electronic switching circuits. *IRE Trans. Electron. Comput.* **1956**, *1*, 15–19.
10. Lupanov, O.B. A method of circuit synthesis. *Izvestiya VUZ Radiofiz* **1958**, *1*, 120–140. (In Russian)

11. Harrenstein, P.; van der Hoek, W.; Meyer, J.J.; Witteveen, C. Boolean Games. In Proceeding of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII), Certosa Di Pontignano, Italy, 8–10 July 2001; van Benthem, J., Ed.; Morgan Kaufmann: Burlington, MA, USA, 2001; pp. 287–298.
12. Harrenstein, P. Logic in Conflict. Ph.D. Thesis, Utrecht University, Utrecht, The Netherlands, 2004.
13. Bonzon, E.; Lagasquie, M.; Lang, J.; Zanuttini, B. Boolean games revisited. In Proceedings of the ECAI 2006, Riva del Garda, Italy, 28 August–1 September 2006; Volume 141.
14. Bonzon, E.; Lagasquie-Schiex, M.C.; Lang, J.; Zanuttini, B. Compact preference representation and Boolean games. *Auton. Agents Multi-Agent Syst.* **2009**, *18*, 1–35.
15. Wooldridge, M.; Endriss, U.; Kraus, S.; Lang, J. Incentive engineering for Boolean games. *Artif. Intell.* **2013**, *195*, 418–439.
16. Hammond, L.; Fox, J.; Everitt, T.; Abate, A.; Wooldridge, M. Equilibrium Refinements for Multi-Agent Influence Diagrams: Theory and Practice. In Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'21), Online, 3–7 May 2021; Endriss, U., Nowé, A., Dignum, F., Lomuscio, A., Eds.; Available online: <https://arxiv.org/pdf/2102.05008.pdf> (accessed: 17 December 2021).
17. Pearl, J. *Probabilistic Reasoning in Intelligent Systems*; Morgan Kaufmann: Burlington, MA, USA, 1988.
18. Howard, R.A.; Matheson, J.E. Influence Diagrams. In *Readings on the Principles and Applications of Decision Analysis*; Strategic Decisions Group: Menlo Park, CA, USA, 1994; Volume 2, pp. 721–762.
19. Kjaerulff, U.B.; Madsen, A.L. *Bayesian Networks and Influence Diagrams*; Springer Science+ Business Media 200: Berlin/Heidelberg, Germany 2008.
20. Nielsen, M.; Plotkin, G.D.; Winskel, G. Petri Nets, Event Structures and Domains, Part I. *Theor. Comput. Sci.* **1981**, *13*, 85–108.
21. Sassone, V.; Nielsen, M.; Winskel, G. A Classification of Models for Concurrency. In Proceedings of the CONCUR '93, 4th International Conference on Concurrency Theory, Hildesheim, Germany, 23–26 August 1993; Best, E., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1993; Volume 715, pp. 82–96.
22. Clairambault, P.; Gutierrez, J.; Winskel, G. The Winning Ways of Concurrent Games. In Proceedings of the IEEE Symposium on Logic in Computer Science, Dubrovnik, Croatia, 25–28 June 2012; pp. 235–244. doi:10.1109/LICS.2012.34.
23. Gutierrez, J.; Wooldridge, M. Equilibria of concurrent games on event structures. In Proceedings of the CSL-LICS, Vienna, Austria, July 14–18, 2014, 2014; ACM: New York, NY, USA, 2014; pp. 46:1–46:10.
24. Alcolei, A.; Clairambault, P.; Hyland, M.; Winskel, G. The True Concurrency of Herbrand's Theorem. In Proceedings of the CSL, Birmingham, UK, 4–8 September 2018; Ghica, D.R., Jung, A., Eds.; Schloss Dagstuhl: Wadern, Germany, 2018; Volume 119, pp. 5:1–5:22.
25. Castellan, S.; Clairambault, P.; Winskel, G. Thin Games with Symmetry and Concurrent Hyland-Ong Games. *Log. Methods Comput. Sci.* **2019**, *15*, 18:1–18:85, doi:10.236338/LMCS-15(1:18)2019.
26. Thompson, F.B. *Equivalence of Games in Extensive Form*; Research Memorandum RM-759; Rand Corporation: Santa Monica, CA, USA, 1952.
27. Hearn, R.A.; Demaine, E.D. *Games, Puzzles and Computation*; A.K. Peters: Natick, MA, USA, 2009.
28. Fudenberg, D.; Tirole, J. *Game Theory*; The MIT Press: Cambridge, MA, USA, 1991.