Reaching your goals without spilling the beans: Boolean secrecy games

Nils Bulling[1], Sujata Ghosh[2] and Rineke Verbrugge[3]

[1] Clausthal University of Technology
Clausthal, Germany.
bulling@in.tu-clausthal.de
[2] Indian Statistical Institute
Chennai, India.
sujata@isichennai.res.in
[3] University of Groningen
Groningen, Netherlands.
L.C.Verbrugge@rug.nl

Abstract. Inspired by the work on Boolean games, we present turn-based games where each of the players controls a set of atomic variables and each player wants to achieve some individual goal in such a way that the other players remain unaware of the goal until it is actually achieved. We present definitions of winning such games with hidden goals for different non-cooperative settings, and discuss in which types of situations players have winning or equilibrium strategies. We also provide some complexity bounds on deciding whether a player has a winning strategy.

1 Introduction

In many intelligent interactions, there are some goals that are commonly known to all concerned, while individuals may also attempt to achieve a secret or *hidden goal* of their own. For example, in a purportedly 'win-win' *negotiation* between two companies, they aim to settle on certain issues in a way that is advantageous to both. Still, while negotiating, both parties also consider their individual benefits, possibly unknown to the other party [14, 18]. Thus, in the resulting settlements, in addition to some goals commonly known to all, some hidden goals are often achieved as well. Sometimes, these goals become common knowledge once they are reached, whereas in other cases, they remain secret forever. It may also happen that the secret goal is revealed before the actual settlement, which may even lead to the cancellation of the settlement.

Such hidden individual goals occur not only in the mixed-motive negotiations described above, where participants have both cooperative and adversarial motives. Even in the case of *teamwork*, individual team members may try to secretly achieve some individual goal while being involved in achieving the team's collective intention [6, 4]. As an example, consider *The Count of Monte Cristo* by Alexandre Dumas (père) [5]. In this story, two of the 'bad guys', Fernand Mondego and the deputy public prosecutor Villefort, achieve their common goal of imprisoning the book's hero Edmond Dantès for life in Château d'If. However, both men are driven by entirely different secret goals that they do not divulge to one another: Fernand Mondego has the hidden goal to marry Edmond Dantès' fiancee Mercédès, while Villefort is secretly driven by the goal to save his reputation and career by preventing Dantès from delivering a treasonous letter from Napoleon to Villefort's father.

Of course in openly *competitive* situations, there is even more reason to hide one's goals from opponents than in mixed-motive negotiations or seemingly cooperative cases of teamwork. For example, during the Second World War, the allied forces prepared an elaborate scheme, Operation Fortitude, to deceive the Germans into thinking that the allies were going to invade Norway and Pas de Calais, rather than Normandy [2]. The scheme to hide their real goal involved faked armies, faked wireless traffic, faked information passed on by double agents, and much more. Even after the actual landing in Normandy on June 6, 1944, the allied forces managed to delay German reinforcement in Normandy by convincing them that the landings in Normandy were meant as a diversionary attack to take attention away from Pas de Calais. In summary, from fictional adventure stories through tales of deception in wars to modern day negotiations, hidden goals may drive individuals involved in intelligent interactions.

In this paper, we aim to formalize the idea of achieving secret goals. Inspired by the work on Boolean games [10, 3], we present turn-based games where each player controls a set of atomic variables and wants to achieve some goals in such a way that its opponents do not know which goals it pursues until those goals are actually reached. This paper forms an initial investigation, intended to lead to a better understanding of ways to obtain vital information from rivals without revealing much of one's own positions; after all, knowledge is power (cf. [1]). The idea is to use very simple tools of logic and game theory to express hiding as well as gaining information in an interactive process.

2 Boolean Secrecy Games

In this section we introduce the basic definitions. A *Boolean secrecy frame* (BSF) is the basic model of our static setting. We use a *game base* for representing a specific evolution of the BSF. Finally, a *Boolean secrecy game* (BSG) consists of a BSF and a game base.

2.1 Boolean Secrecy Frames

Let $\mathcal{P}rops$ be a non-empty finite set of propositional variables and let $\mathcal{L}(\mathcal{P}rops)$ denote the set of formulas of propositional logic over $\mathcal{P}rops$, constructed with the usual propositional connectives together with the propositional constants $\top$ (truth) and $\bot$ (falsity). In the following we will simply write $\mathcal{L}$ if the set of propositions is clear from context.

We use $\mathcal{P}layers$ to refer to the set of players. Players are denoted by $1, 2, 3, \ldots$. In the following, if not said otherwise, we assume that $\mathcal{P}layers = \{1, \ldots, n\}$ and use $i, j, \ldots$ to refer to players. We use $\bar{i}$ to refer to "the opponents of i", i.e. to the set of players $\mathcal{P}layers \backslash \{i\}$. Players can decide on the truth values of specific propositions they control.

We want to model the question whether one player can achieve a certain goal formula, without the other player(s) knowing what the goal is until it becomes true. Let Γ be the set consisting of all possible goals that any of the players might want to achieve. It is common knowledge among all the players that these are the possible goals. However, these goals will certainly not all become collective goals or collective intentions [6, 4]. We consider the case where players may not be certain of the exact goal(s) that their opponents would like to achieve. The subset of the *secret goals* of a player i is given by $\Gamma_i \subseteq \Gamma$.

Similarly to [8], achieving goals may involve some costs, and sometimes players need to minimalize the cost. To this effect, we introduce costs associated with players' actions and give restraining conditions by considering cost limits.

Definition 2.1 (Boolean secrecy frame). *The sets $\mathcal{P}_i \subseteq \mathcal{P}rops$ of propositional variables stand for each player i's* set of propositions*, such that $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for all $i, j \in \mathcal{P}layers$ with $i \neq j$. Furthermore, $\neg\mathcal{P}_i$ stands for $\{\neg p : p \in \mathcal{P}_i\}$.*

For $i \in \mathcal{P}layers$, we define the set $\Sigma_i := \mathcal{P}_i \cup \neg\mathcal{P}_i \cup \{skip_i\}$ and for any $C \subseteq \mathcal{P}layers$, we define $\Sigma_C := \bigcup_{i \in C} \Sigma_i$. The set Σ_i represents the actions of player i. We refer to actions in $\mathcal{P}_i \cup \neg\mathcal{P}_i$ as propositional actions*.*

A Boolean secrecy frame (BSF) *is a tuple*

$$F = (\mathcal{P}layers, \mathcal{P}rops,\ (\mathcal{P}_i)_{i \in \mathcal{P}layers}, \Gamma,\ (\Gamma_i)_{i \in \mathcal{P}layers}, (c_i)_{i \in \mathcal{P}layers}, (C_i)_{i \in \mathcal{P}layers}),$$

where $\mathcal{P}layers$ and $\mathcal{P}rops$ are as given above; function $c_i : \Sigma_i \to \mathbb{R}^+$ represents the costs associated with the moves of player i; $C_i \in \mathbb{R}^+$ is the cost limit for player i; $\Gamma \subseteq \mathcal{L}$ is the set of possible goals for the players, which is commonly known to all players; and $\Gamma_i \subseteq \Gamma$ is the set of secret goals of player i, not known among the other players.

In contrast to [8] and similarly to [9], we do not require that the players' sets of propositions form a partition of $\mathcal{P}rops$, so not necessarily $\bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i = \mathcal{P}rops$. Propositions not contained in $\bigcup_i \mathcal{P}_i$ are controlled by some external entity not being part of the game. We also refer to the elements of $\mathcal{P}_i \cup \neg\mathcal{P}_i \cup \{skip_i\}$ as *actions* of player i. An action $p \in \mathcal{P}_i$ corresponds to agent i making p true, and action $\neg p \in \neg\mathcal{P}_i$ corresponds to i making p false; alternatively, when agents have run out of propositions—a proposition can only be set once—they can *skip*.

As a running example for our subsequent discussions, let us consider the misinformed German high command and the British command during Operation Fortitude. We can set variables p_1 for the British troops preparing for some attack, p_2 for the British attacking Normandy, and p_3 for the British attacking Pas de Calais, p_4 for the German troops preparing for defending Normandy, p_5 for the German troops preparing for defending Pas de Calais.[4] Here follows the relevant Boolean secrecy frame.

Example 2.1 (Boolean secrecy frame). We consider a BSF with two players $\{1$ (*British*), 2 (*German*)$\}$ and the set of propositions $\mathcal{P}rops = \{p_1, \ldots, p_5\}$. The set of propositions controlled by the players are given by $\mathcal{P}_1 = \{p_1, p_2, p_3\}$, and $\mathcal{P}_2 = \{p_4, p_5\}$. The set of possible goals $\Gamma = \{p_1 \wedge (p_4 \to p_3) \wedge (p_5 \to p_2), (p_1 \wedge p_4 \wedge p_2) \vee (p_1 \wedge p_5 \wedge p_3)\}$. Player 1's set of secret goals is given by $\Gamma_1 = \{(p_1 \wedge (p_4 \to p_3) \wedge (p_5 \to p_2)\}$; player 2's secret goal set is empty.

[4] For simplicity, we leave out Norway from the example.

The first goal formula, which is also the secret goal for the British, says that wherever the Germans put up their defense, the British will attack at the other point. The other goal formula says that the troops from both sides will meet at one of the regions, which corresponds more to the assumed (non-secret) German goal. In this way, we represent a simplification of Operation Fortitude.

In the dynamic Boolean secrecy games that we will define, agents do not win simply by reaching their goals: they must do so without spilling the beans; that is, without the others knowing about their secret goals.

2.2 Histories and Propositional Truth Assignments

In the following, if not said otherwise we assume that $F = (\mathcal{P}layers, \mathcal{P}rops, (\mathcal{P}_i)_{i\in\mathcal{P}layers}, \Gamma, (\Gamma_i)_{i\in\mathcal{P}layers}, (c_i)_{i\in\mathcal{P}layers}, (C_i)_{i\in\mathcal{P}layers})$ is a Boolean secrecy frame (BSF). A BSF models the initial setting and the player characteristics. Now, the players try to realize their hidden goals by following a specific course of action. The performance of all players over time yields a finite sequence of actions which we call a *history*. Histories encode possible dynamic evolutions in the given BSF. We only impose the restriction that the action of making a particular atomic proposition p in $\mathcal{P}rops$ true or false can be executed at most once along any history. Histories that contain a propositional action p or $\neg p$ for each proposition p in $\bigcup_{i\in\mathcal{P}layers}\mathcal{P}_i$ are of particular interest and are called *complete*.

Definition 2.2 (History, subhistory, complete history). *An F-*history *is a finite sequence $h = (a_j)_{j=0,\dots k} \in (\bigcup_{i\in\mathcal{P}layers}\Sigma_i)^*$, $k \geq 0$ (or, $h = \epsilon$) such that:*

1. *if $a_l \in \{p, \neg p\}$ for some $p \in \bigcup_{i\in\mathcal{P}layers}\mathcal{P}_i$ and $0 \leq l \leq k$, then there is no $l' \neq l$ with $0 \leq l' \leq k$ and $a_{l'} \in \{p, \neg p\}$;*
2. *if $a_l = skip_i$, then for all $p \in \mathcal{P}_i, \exists j : 0 \leq j < l$ and $(a_j = p$ or $a_j = \neg p)$.*

The length *of h, $|h|$, is defined as $|h| = k + 1$. We write $h[l]$ to refer to the lth action a_l on h where $0 \leq l < |h|$. For two histories h, h' we say that h' is a* subhistory *of h, denoted by $h' \leq h$, if h' is an initial segment of h. We write $h' < h$ if $h' \leq h$ and $h' \neq h$. The set of propositions occurring in h is referred to as $\mathcal{P}rops(h)$; i.e. $\mathcal{P}rops(h) = \{p \mid \exists j : 0 \leq j < |h| and\ (a_j = p\ or\ a_j = \neg p)\}$. The* concatenation *of two finite sequences $h, h' \in (\bigcup_{i\in\mathcal{P}layers}\Sigma_i)^*$ is denoted by hh'.*

*Let $C \subseteq \mathcal{P}layers$. A history h is said to be C-*complete *if $\bigcup_{i\in C}\mathcal{P}_i \subseteq \mathcal{P}rops(h)$. As abbreviations, we say that h is i-*complete *if h is $\{i\}$-complete, and that h is* complete *if it is $\mathcal{P}layers$-complete.*

According to the definition, each history can consist of at most $|\mathcal{P}rops|$-many propositional actions, and never contains a literal p or $\neg p$ for which $p \in \mathcal{P}rops \setminus \bigcup_{i\in\mathcal{P}layers}\mathcal{P}_i$.

A history is one possible evolution of a BSF; usually, there are many possible evolutions. Firstly, a player i can select any action from Σ_i. Secondly, the order in which players act need not be fixed. One option, although not always realistic in dynamic environments, would be to fix such an ordering. Instead, we impose some constraints on the possible dynamic evolutions. A sensible requirement could for instance be that each player is allowed to make all her controlled propositions true or false, that is, each history could be *Players*-complete. A player being allowed to make all her controlled propositions true or false in one block of actions—,one after the other—is not sensible, however, as then she might reveal what her secret goal is. In a two-player setting one could assume that the players act alternately. We propose two possible intuitive fairness conditions:

Definition 2.3 (Fair and alternating histories). *Let h be an F-history. (a) h is said to be* fair *if it is $\mathcal{P}layers$-complete. (b) h is said to be* π-alternating *where π is a permutation on $\mathcal{P}layers$ if for all $i \in \mathcal{P}layers$ and all $l \in \mathbb{N}$ with $l \cdot |\mathcal{P}layers| + \pi(i) \leq |h|$, we have $h[l \cdot |\mathcal{P}layers| + \pi(i)] \in \Sigma_i$.*

Finally, we relate histories to valuations of propositional variables. Each history gives rise to a *partial valuation* of the propositions occurring in it. A complete history corresponds to a complete valuation of $\bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$.

As a reminder, a (truth) *valuation* is a function $v : \mathcal{P}rops \rightarrow \{\mathbf{t}, \mathbf{f}\}$ which assigns a truth value to each proposition in $\mathcal{P}rops$. In order to evaluate Boolean formulas we lift a valuation v to formulas in the standard way. We write $v(\varphi) = \mathbf{t}$ (resp. $v(\varphi) = \mathbf{f}$) to denote that valuation v makes φ true (resp. false). A *P-valuation* is a partial valuation defined on a subset $P \subseteq \mathcal{P}rops$. Clearly, a valuation is a $\mathcal{P}rops$-valuation and vice versa.

In order to check whether a player i can achieve a given goal, we introduce an *i-extension* of a partial valuation, which extends it to a partial valuation that specifies truth values for all propositions controlled by i. Thus, an i-extension fixes all of i's variables. Finally, a *completion* of a partial valuation extends it to a *complete* one by assigning truth values to *all* propositions from $\mathcal{P}rops$ not yet specified by the original partial valuation:

Definition 2.4 (Induced valuation, i-extension, completion). *Given an F-history h, the h-*induced valuation *is the $\mathcal{P}rops(h)$-valuation v_h with $v(p) = \mathbf{t}$ (resp. $v(p) = \mathbf{f}$) if $p = a_j$ (resp. $\neg p = a_j$) for some j with $0 \leq j < |h|$. Let $P, P' \subseteq \bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$. We say that a P-valuation v* agrees with *a P'-valuation v' if for all $p \in P \cap P'$, we have that $v(p) = v'(p)$. An i-*extension *v' of a P-valuation v is a $(P \cup \mathcal{P}_i)$-valuation that agrees with v on P. A* completion *v'* of *a P-*valuation *v is a $\mathcal{P}rops$-valuation v' that agrees with v on P.*

Note that an i-extension of an h-induced valuation is indeed a partial valuation (see Definition 2.2). It is not difficult to see that for any fair history h and any players i, j with $i \neq j$, history h gives a j-extension for a $\mathcal{P}_i$-valuation agreeing with the $\mathcal{P}rops(h)$-valuation v_h. If $\cup_{i \in \mathcal{P}layers} \mathcal{P}_i = \mathcal{P}rops$ then any fair history h gives a completion for each $\mathcal{P}_i$-valuation agreeing with $\mathcal{P}rops(h)$-valuation v_h.

2.3 Extensive Form Boolean Secrecy Game

We have just introduced Boolean secrecy frames (BSF) and histories. Each history represents one possible course of action, corresponding to some specific combination of strategies (strategy profile) of the players. Initially, if there are no fixed strategies, many histories may still be possible. We call the set that contains all possible evolutions a *game base*.

Definition 2.5 (Game base, i-history). *Let H be a non-empty set of F-histories. We say that H is an F-*game base *if it satisfies the following conditions:*

1. *$\epsilon \in H$ (i.e., H contains the empty history);*
2. *if $h \in H$ then every subhistory h' of h is in H (i.e., H is downwards closed);*
3. *for all histories $h \in H$ there is a player $i \in \mathcal{P}layers$ such that for each history h' that has h as strict subhistory (i.e., $h < h'$), we have that $h'[|h|] \in \Sigma_i$; for this player i who is about to play, we call h an i-*history*; and*
4. *all maximal histories $h \in H$ are complete; here, a history h is* maximal *if there is no history $h' \in H$ with $h < h'$.*

*We use H_i to denote the set of all i-histories in H. Moreover, H is π-*alternating *if all histories $h \in H$ are π-alternating. Note that all maximal histories are fair.*

In particular, requirement 3 ensures that the game base can be seen as the underlying structure of a turn-based extensive form game. At each history (h), it is the turn of the player who "owns" all the directly succeeding nodes ($h'[|h|]$). Requirement 4 enforces that players eventually assign a truth value to all the variables they control.

Definition 2.6 (Full-branching game base). *An F-game base H is said to be* full-branching *if for all $i \in \mathcal{P}layers$, the following holds for the set H_i of all i-histories in H: If $h \in H_i$, then for all $p \in \mathcal{P}_i \setminus \mathcal{P}rops(h)$, both $hp \in H$ and $h\neg p \in H$.*

Note that in every F-game base H there is a unique player $i \in \mathcal{P}layers$ such that $\epsilon \in H_i$. We refer to i as the *initial player*. It is also easy to see that each combination of a BSF F and permutation π gives rise to a unique π-alternating full-branching game base H.
Finally, we are ready to introduce *Boolean secrecy games*. They model all possible evolutions of a BSF regarding a given F-game base without fixing any truth assignment of any player in advance. Essentially, a *Boolean secrecy game* (BSG) encodes an extensive form game, well-known from game theory [12]. However, in order to define the players' preference relations, we need some additional notions.

Definition 2.7 (Boolean secrecy game). *A* Boolean secrecy game *(BSG) is given by $G = (F, H)$, where F is a Boolean secrecy frame and H is a full-branching F-game base. We say that G is the H-*based game over *F. Moreover, we lift the properties of H given in Definition 2.5 and Definition 2.6 to G.*

A Boolean secrecy game $G = (F, H)$ gives rise to a canonical extensive form game frame $E(G)$. Histories of H correspond to nodes in $H(G)$. A player function, indicating which player's turn it is, can be extracted from the set of i-histories: it is player i's turn at history h if and only if h is an i-history.

Example 2.2 (Boolean secrecy game). The BSF F in Example 2.1 and the order $(1, 2)$ give rise to a unique full-branching $(1, 2)$-alternating BSG $G = (F, H)$, as shown in Figure 1(a).

The strategy s_1 for player 1 makes her play p_1 at the first node, and then allows her to respond with p_2 to all possible moves for player 2. Player 1 could have given different responses to different moves of player 2. Note that decisions are irreversible. Thus, after a player chooses to play p_i or $\neg p_i$, these possibilities disappear from the subsequent play, while literals corresponding to not-yet-played controlled variables remain possible for that player.

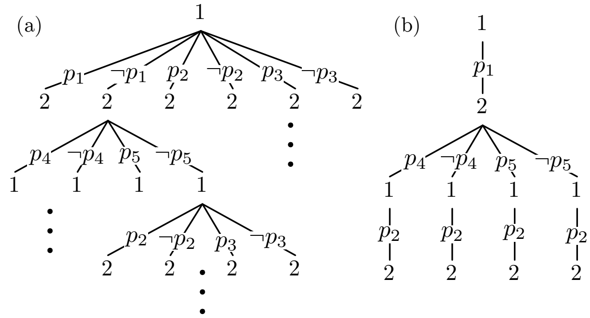

Fig. 1. (a) A diagrammatic representation of the full branching F-game base, with F given in Example 2.2. (b) A strategy s_1 for player 1 and corresponding game base $H|_{s_1}$.

2.4 Strategies, Histories, and Winning Criteria

We introduce the notion of *strategy*, prescribing how a player acts.

Definition 2.8 (Strategy). *Let* $G = (F, H)$ *be a Boolean secrecy game. An* i-strategy (in G) *is a function* $s_i : H_i \to \Sigma_i$ *such that if* $s_i(h) \in \{p, \neg p\}$ *then* $s_i(h') \notin \{p, \neg p\}$ *for all histories* h' *that strictly extend* h *(i.e.,* $h < h'$*); and such that if* $s_i(h) = skip_i$*, then* h *is* i*-complete.*

A C-joint strategy *for a coalition* $C \subseteq \mathcal{P}layers$ *is a tuple of strategies, one for each player in* C*. A* strategy profile *is a* $\mathcal{P}layers$*-joint strategy.*

Now, not all histories are compatible with a player's strategy, only those that respect the actions specified by it. We say that such histories *agree* with a strategy. Formally, we have:

Definition 2.9 (Agreeing, $H|_s$). *A history* $h = (a_j)_{j=0,...k}$ agrees with *an* i*-strategy* s_i *if for all* $h' < h$ *with* $h' \in H_i$ *we have that* $h[|h'|] = s_i(h')$ *(i.e. the action prescribed at* h' *by* s_i *is the next action extending* h' *in* h*). Similarly, for* $C \subseteq \mathcal{P}layers$*, we say that* h agrees with *a* C*-joint strategy* $s_C = (s_{i_1}, \ldots, s_{i_{|C|}})$ *if* h *agrees with* s_{i_j} *for* $j = 1, \ldots, |C|$*. We use* $H|_{s_C}$ *to denote the set of all histories from* H *agreeing with* s_C *and we write* $H|_{s_i}$ *for* $H|_{\{s_i\}}$*.*

Note that, if s_i is a member of the tuple s_C, then $H|_{s_C} \subseteq H|_{s_i}$.

Example 2.3 (Strategy).
A 1-strategy in the BSG given in Example 2.2 is shown in Figure 1(b).

The idea of the player 1 (British) strategy is to attack wherever the player 2 (Germans) is not building up their defense. The play given by the sequence $p_1, p_5, p_2, \ldots$ models the actual history to some extent [2].

When is a strategy "winning" for a player? We are interested in strategies that keep the set of its intended goals secret, in the sense that the opponents should not become fully aware of a subset of goals before they are achieved. We capture this idea by introducing *goal-achieving* strategies. Suppose we are given an i-strategy s_i. In order for s_i to be goal-achieving, player i must be able to guarantee that some goal formula $\varphi \in \Gamma_i$ becomes true; that is, φ must eventually become true on *all* histories agreeing with s_i. Although this guarantees the truth of a secret goal, it does not yet preserve its secrecy. What we want to model is the following question: *"Can player i achieve some member of Γ_i, that is, can the agent make a certain goal formula true, without the other players knowing what the goal is until it becomes true?"*

For preserving secrecy, we also require that for each history h agreeing with s_i there is a non-goal formula $\varphi' \in \Gamma \backslash \Gamma_i$ that can be guaranteed by i to become true at an extension of h: from $\bar{i}$'s point of view, φ' could also be a goal of i. Formally, there should exist an appropriate i-extension extending the choices made so far.

Although these points capture the basic idea of keeping the goals secret, we are not yet done. Consider the case where $\Gamma = \{a \to (b \vee c), (b \vee c)\}$, $\Gamma_i = \{b \vee c\}$, $s_i(\epsilon) = a$ and $s_i(ad) = b$, where d is a move of a player in $\bar{i}$ during their turn. Clearly, s_i satisfies both conditions mentioned above. After the first step, however, it should be clear for $\bar{i}$ that i has the *subgoal* to make $b \vee c$ true. Although $\bar{i}$ is still not sure whether i's actual goal is $a \to (b \vee c)$ or $b \vee c$, they have a clear idea of what comes next. To avoid this, we require that the deceiving goal φ' is sufficiently different from the actual goal at each step. Finally, we capture these ideas formally:

Definition 2.10 (Goal-achieving). *Let $G = (F, H)$ be a BSG. An i-strategy s_i is* goal-achieving *if the following holds. For all F-histories $h \in H|_{s_i}$ there is a subhistory $h' \leq h$ and a formula $\varphi \in \Gamma_i$ such that:*

1. $v(\varphi) = \mathbf{t}$ for all completions v of the h'-induced valuation (i.e. goal φ is guaranteed to become true);

and for all $h'' < h'$ the following conditions hold:

2. there is a formula $\varphi' \in \Gamma \backslash \Gamma_i$ such that
(a) there is an i-extension v_1 of the h''-induced valuation $v_{h''}$ such that for all completions v_2 of v_1, we have $v_2(\varphi') = \mathbf{t}$ (i.e. φ' is a possible goal that i could enforce);
(b) there is at least one completion v_3 of the h''-induced valuation $v_{h''}$ such that $v_3(\varphi') = \mathbf{f}$ (i.e. φ' is not yet guaranteed to be true);
(c) there is a completion v_4 of the h''-induced valuation $v_{h''}$ such that $v_4(\varphi') \neq v_4(\varphi)$ (i.e. φ' is sufficiently different from φ).

3. for all $\psi \in \Gamma_i$ there exists a completion v_5 of the h''-induced valuation $v_{h''}$ such that $v_5(\psi) = \mathbf{f}$ (i.e. no goal of agent i has been guaranteed to be true before φ at h').

Figure 2 illustrates the definition of goal-achieving strategies. There can be more than one goal-achieving strategy. Naturally, which one to choose should

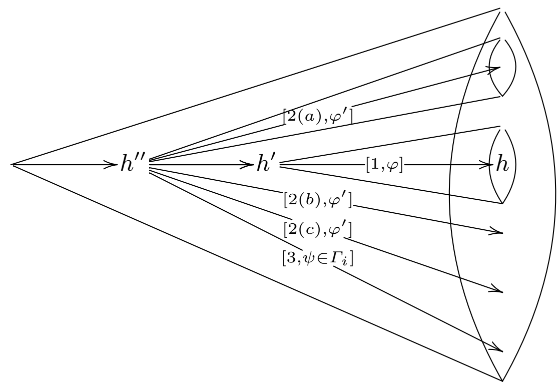

Fig. 2. Goal-achieving strategies as described in Definition 2.10. The figure shows some history $h \in H|_{s_i}$ and a subhistory h'. All completions of h' (cf. the part labeled [1, φ]) make φ necessarily true. Moreover, before h', at any subhistory h'' of h', no secret goal formula of i is allowed to be necessarily true (existence of such completions shown by the arrow labeled [3, $\psi \in \Gamma_i$]); in particular, φ is not allowed to be necessarily true at h''. Finally, there must also be a potential goal formula φ', sufficiently different from φ (cf. arrow labelled [2(c), φ']), that is not a secret goal of i (cf. arrow labeled [2(b), φ']) and that can be made true by i (cf. arrow labelled [2(a), φ']).

also depend on the costs of executing a strategy; in particular, the execution may exceed the cost limit of a player.

Before defining the cost of a strategy, we observe that a strategy profile s_C for $C \neq \mathcal{P}layers$ usually identifies a *set* of histories in the secrecy game, namely $H|_{s_C}$. Furthermore, note that although a complete profile s fixes all players' choices, the variables controlled by the environment are not yet set–again, resulting in a set of possible histories. As a consequence, when defining a player's *costs* of a strategy we need to consider a set of possible histories. We take on the worst-case perspective and define the cost as the maximal cost caused by any history agreeing with the strategy.

Definition 2.11 (Cost of a strategy). *Let $G = (F, H)$ be a BSG. We define the* cost of player i *of an F-history h, denoted $c_i(h)$, inductively as follows: $c_i(\epsilon) = 0$ (i.e., the empty history is cost-free); and for $h'a \leq h$, $c_i(h'a) = c_i(h')$ if $a \in \Sigma_{\bar{i}}$ or h' satisfies conditions (1) and (2) of Def. 2.10; and $c_i(h'a) = c_i(h') + c_i(a)$ otherwise.*

The cost of a set $H' \subseteq H$ of F-histories for player i is defined as $c_i(H') := \max_{h \in H'} c_i(h)$. Finally, if s_C is a C-joint strategy for $C \subseteq \mathcal{P}layers$, we define the cost of player i corresponding to the strategy s_C as $c_i(s_C) = c_i(H|_{s_C})$. Thus, $c_i(s_i) = c_i(H|_{s_i})$.

Note that, if s_i is a member of the tuple s_C, then $c_i(s_C) \leq c_i(s_i)$. We have used the intuitive additive model to define costs of strategies in terms of cost of individual actions. The effect of different kinds of cost functions on the determination of winning strategies is left for future work.

Definition 2.12 (Winning). *Given a* BSG $G = (F, H)$*, an* i*-strategy* s_i *is* winning *in* G *iff it is goal achieving and* $c_i(s_i) \le C_i$*. Player* i *is winning iff there is a winning* i*-strategy.*

Note that more than one player can have a winning strategy in a given game G. Also, it is possible that no player can win, see Section 2.5.

One could think of different kinds of winning conditions, for example: one that guarantees that nobody else's goal is satisfied; one that finds out another player's goal before the goal becomes true; one that satisfies one's goal formulas (while possibly allowing others to satisfy theirs); one that satisfies one's goal formula and nobody else does; and various other possibilities. For now, we restrict ourselves to the winning condition of Definition 2.12. Ultimately, we are interested in the question whether players have a strategy to keep their hidden goals secret given a (static) BSF. This crucially depends on the order in which players move. Formally, this is captured by game bases.

Definition 2.13 (Winning in Boolean secrecy frames). *We say that a player* i *wins in the Boolean secrecy frame* F *if* i *wins in the* H*-based game of* F *for every full-branching* F*-game base* H*. If we consider complete* π*-alternating game bases only, then we say that* i *wins in the* π*-alternating Boolean secrecy frame* F*.*

2.5 Non-Determinacy and Importance of Order

We show the existence of a Boolean secrecy game $G = (F, H)$ in which no player has a winning strategy. We consider a real-life situation. A hiring committee for a faculty position, consisting of a mathematician with some expertise of biology and a physicist. The committee has to hire a theoretical biology expert with good managerial skills. Both committee members, however, are commonly known not to be able to evaluate candidates' managerial skills (there is an outside expert for that). If the two committee members want to hire a candidate in their own expert area, there is no chance of keeping that secret in committee discussions, as they can only mention someone's scientific expertise as argument in favor of a candidate. We now construct a BSF with as intuitive meanings of the propositional atoms: p_1 means that the chosen candidate is an expert on mathematics and biology; p_2 means that the chosen candidate is an expert on bio-physics; and p_3 means that the chosen candidate has good managerial skills. Consider the frame $F = (\mathcal{P}layers, \mathcal{P}rops, (\mathcal{P}_i)_{i \in \mathcal{P}layers}, \Gamma, (\Gamma_i)_{i \in \mathcal{P}layers}, (c_i)_{i \in \mathcal{P}layers}, (C_i)_{i \in \mathcal{P}layers})$, instantiated as follows: $\mathcal{P}layers = \{1, 2\}$; $\mathcal{P}rops = \{p_1, p_2, p_3\}$; $\mathcal{P}_1 = \{p_1\}$; $\mathcal{P}_2 = \{p_2\}$; $\Gamma = \{p_1 \wedge p_3, p_2 \wedge p_3\}$; $\Gamma_1 = \{p_1 \wedge p_3\}$; $\Gamma_2 = \{p_2 \wedge p_3\}$; $c_i : \Sigma_i \to \{1\}$ for each i; $C_i = 2$ for each i.

Because neither player has control over p_3, it can be shown that neither player has a winning strategy in any full-branching F-game base H. Note that the non-determinacy depends on the fact that $\bigcup_{i \in \mathcal{P}layers} P_i$ is a proper subset of $\mathcal{P}rops$.

Perhaps surprisingly, it turns out that the order of the players' moves matters a lot when it comes to winning. One might think that for a full-branching π-alternating game-base H, if i is winning in (F, H) then i could also win in all

π'-alternating game bases H', as long as π and π' are equivalent with respect to i, that is $\pi(i) = \pi'(i)$. But this is not the case, as we now show by an example.

Consider a BSF with three players $\{1, 2, 3\}$ and set of propositions $\mathcal{P}rops = \{p_1, \ldots, p_5\}$. The set of propositions controlled by the players are given by $\mathcal{P}_1 = \{p_1, p_3\}$, $\mathcal{P}_2 = \{p_2\}$, $\mathcal{P}_3 = \{p_4, p_5\}$. The set of possible goals $\Gamma = \{p_1 \wedge p_2, p_3 \wedge \neg p_2, p_4 \vee p_5\}$. Player 1's set of secret goals is given by $\Gamma_1 = \{p_1 \wedge p_2, p_3 \wedge \neg p_2\}$; player 2's and player 3's secret goal sets are empty. We let $C_1 = C_2 = C_3 = 10$ and $c_i : \Sigma_i \to \{1\}$ for $i \in \{1, 2\}$. If player 2's first move is before player 1's first move, then player 1 has a winning strategy: "if player 2 has made p_2 true, then make p_1 true; if player 2 has made p_2 false, then make p_3 true".[5] However, if player 1 has her first move before player 2, then player 1 has no winning strategy. So, in this three player frame F, if H is the $(2, 1, 3)$-alternating full branching game base, then player 1 will have a winning strategy in $G = (F, H)$, whereas she will not have any winning strategy in $G = (F, H')$ where H' denotes the $(3, 1, 2)$-alternating game base. Note that the position of player 1 is the same in both cases.

3 Computational Complexity

An interesting question is whether a player can win in a BSG (see Definitions 2.12 and 2.13). Here, we present some results on the complexity of such problems.

Firstly, let us consider the representation of the input. We measure the size $|F|$ of a BSF,

$$F = (\mathcal{P}layers, \mathcal{P}rops, (\mathcal{P}_i)_{i \in \mathcal{P}layers}, \Gamma, (\Gamma_i)_{i \in \mathcal{P}layers}, (c_i)_{i \in \mathcal{P}layers}, (C_i)_{i \in \mathcal{P}layers})$$

as the sum of the sizes of all elements in F; that is, $|F| = |\mathcal{P}layers| + |\mathcal{P}rops| + \sum_{i=1}^{|\mathcal{P}layers|} \mathcal{P}_i + \sum_{\gamma \in \Gamma} |\gamma| + \sum_{\gamma \in \Gamma_i, i \in \mathcal{P}layers} |\gamma| + \sum_{i \in \mathcal{P}layers} (|c_i| + |C_i|)$, where $|\gamma|$ denotes the length of the formula γ and we assume, omitting the details, that $|c_i|$ and $|C_i|$ refer to some reasonable encoding of the functions c_i and numbers C_i, respectively. $|H|$ refers to the cardinality of set H.

The *size* of a BSG $G = (F, H)$ is defined as $|F| + |H|$. In this representation, the size of H is usually exponential in the size of F. Hence, *compact representations* are of more interest. Instead of taking H as input, we only fix the *structure* of the game base according to Definition 2.3; for example, we can consider only π-alternating game bases. Then, following Definition 2.13, we would like to determine whether a player is winning in a frame together with a structural description of the game base. For example, the input might be given by (F, π) and the question to be answered is whether a player is winning in all BSG's (F, H) where H is a π-alternating F-game base. In the case of π-alternating game bases, we even have that the BSG (F, H) is unique and that in most non-trivial cases the size of the input (F, π) is exponentially smaller than the size of the corresponding explicit input (F, H), where H is the unique π-alternating F-game base. Naturally, complexity results for input (F, π) are more insightful than for (F, H).

[5] Note that different histories agreeing with this strategy may incur different costs for player 1.

To start with, we show that the question whether a player is winning can be solved efficiently for the class of games $G = (F, H)$ for which the input is given in explicit form *and* all propositions are controlled by the players, that is, $\mathcal{P}rops = \bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$. We denote this class of games by $\mathcal{G}^-$. The general case, in which not necessarily $\mathcal{P}rops = \bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$, is more complex. We have the following complexity results:

Proposition 3.1. *The problem whether a player is winning in $G \in \mathcal{G}^-$ (in explicit form) is in* $\mathbf{P}$ *with respect to the size of G.*

Proof. Let $F = (G, H)$ and player i be given. We propose a labeling procedure to determine whether i has a winning strategy. For ease of understanding, we interpret the game base H as a tree and use standard vocabulary: each history h represents a *node* and each minimal extension ha of h is a (direct) *child* of h, the empty history ϵ is the root, etc. We use four types of labels for nodes $h \in H$, where $\varphi \in \Gamma$: φ, standing for "φ is true on all completions of v_h"; $[\varphi]$, for "there is an i-strategy in h that guarantees that φ will be true"; $\langle \neg\varphi \rangle$, for "there is a completion of v_h that makes φ false"; and G_φ, for "a node labelled φ is reachable". We use $L(h)$ to denote the set of labels of node h. We apply the following steps to H. For each formula $\psi \in \Gamma$:
(1) Label all leaf nodes h (i.e. maximal histories) with $[\psi]$ and ψ if $v_h(\psi) = \mathbf{t}$ and with $\langle \neg\psi \rangle$ if $v_h(\psi) = \mathbf{f}$. (This is possible because $\mathcal{P}rops = \bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$.)

Now, we apply the following steps as long as possible:
(2) Let $h \in H \backslash H_i$. Label h with $[\psi]$ (resp. ψ) if all children (i.e. direct successors) h' of h are labelled with $[\psi]$ (resp. ψ); otherwise, label h with $\langle \neg\psi \rangle$.
(3) Let $h \in H_i$. Label h with $[\psi]$ (resp. $\langle \neg\psi \rangle$) if there is a child labelled $[\psi]$ (resp. $\langle \neg\psi \rangle$) (i.e. player i can make ψ (resp. $\neg\psi$) true) and with ψ if all children are labelled with ψ .[6]

Now we have for $\varphi \in \Gamma$: *There is an i-strategy s_i such that for all F-histories $h \in H|_{s_i}$ we have $v_h(\varphi) = \mathbf{t}$ iff the root node is labeled $[\varphi]$, i.e. $[\varphi] \in L(\epsilon)$.*

The following steps label some subtrees as invalid (we use a label $\bot$)—we cannot immediately delete these trees due to technical reasons. For all nodes h in the tree do the following:
(4) If $\varphi \in L(h) \cap \Gamma_i$, then label all children of h with $\bot$. [This ensures Condition 3 of Def. 2.10, identifies the lowest occurrence of a node labelled $\varphi \in \Gamma_i$.]
(5) If there is no $\psi \in \Gamma \backslash \Gamma_i$ with $\{[\psi], \langle \neg\psi \rangle\} \subseteq L(h)$, then label all children $\bot$. [Condition 2(a) and 2(b) of Def. 2.10.]
(6) If $\varphi \in L(h) \cap \Gamma_i$ and $\bot \notin L(h)$, then label all predecessors of h with G_φ.
(7) Check whether there is a $\varphi \in \Gamma_i$ and $\psi \in \Gamma \backslash \Gamma_i$ with $\{G_\varphi, [\varphi], [\psi], \langle \neg\psi \rangle\} \subseteq L(h)$ and a leaf node h' reachable from h with $\{\langle \neg\psi \rangle, [\varphi]\} \subseteq L(h')$ or $\{[\psi], \langle \neg\varphi \rangle\} \subseteq L(h')$. If this is not the case, then label h and all children with $\bot$ [Condition 2(c) of Def. 2.10.]
(8) Remove all labels G_φ from all nodes and apply the following again: For all h, if $\varphi \in L(h) \cap \Gamma_i$ and $\bot \notin L(h)$, then label all predecessors of h with G_φ.

[6] If for some formula ψ both $\psi, \neg\psi \in \Gamma$, a node could have all labels $[\psi]$, $[\neg\psi]$, $\langle \neg\psi \rangle$, $\langle \neg\neg\psi \rangle$; double negations cannot be removed! Also, $\psi \in L(h)$ iff $\langle \neg\psi \rangle \notin L(h)$.

Now, we remove all nodes labelled $\bot$ from H and observe: *There is a goal-achieving i-strategy s_i iff $\{[\varphi], G_\varphi\} \subseteq L(\epsilon)$ for some $\varphi \in \Gamma_i$.*

Finally, we need to consider costs. We introduce new labels C_x^φ for the nodes where $\varphi \in \Gamma_i$ and $x \in \mathbb{R} \cup \{\infty\}$.[7] The intuitive reading is that φ can be guaranteed with cost x.

(9) If h is a leaf node with $\varphi \in L(h) \cap \Gamma_i$, then label it with $C^\varphi_{\hat{c}_i(h)}$ where $\hat{c}_i(a_1 \dots a_n) = c_i(a_1) + \dots + c_i(a_n)$; otherwise with C_∞.

In the following we say that a node is labelled with $C_?^\varphi$ if it is labelled with a label of the above type C_x^φ for some $x \in \mathbb{R} \cup \{\infty\}$. Finally, we apply the following steps to H for as long as possible:

(10) If $h \in H \backslash H_i$ is not labelled $C_?^\varphi$ and all children of h are labelled with $C_?^\varphi$, then label h with $C^\varphi_{\max\{x \mid h' \in S \text{ and } h' \text{ is labelled } C_x^\varphi\}}$, where S is the set of all children of h. [Computes the costs, cf. Def. 2.12.]

(11) If $h \in H_i$ is not labelled $C_?^\varphi$, and all children of h are labelled $C_?^\varphi$, then label h with $C_{\min\{x \mid h' \in S \text{ and } h' \text{ is labelled } C_x^\varphi\}}$, where S is the set of all children of h labelled $C_?^\varphi$. [Ensures Def. 2.12.]

Now, *s_i is winning iff $\{[\varphi], G_\varphi, C_x^\varphi\} \subseteq L(\epsilon)$ with $x \leq C_i$ for some $\varphi \in \Gamma_i$.* The described algorithm runs in time polynomial in $|G|$. □

Proposition 3.2. *The general problem whether a player is winning in a BSG G is in $\mathbf{\Delta_2^P}$ and is* **coNP***-hard with respect to the size of G.*

Proof. We modify the algorithm given in the proof of Proposition 3.1. We observe that we only need to modify step 1 of the algorithm, since the propositions in $\mathcal{P}rops \backslash \bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$ never occur on a history. Now, let $\psi[v_h]$ be ψ but with each proposition $p \in \bigcup_{i \in \mathcal{P}layers} \mathcal{P}_i$ in ψ replaced by $\top$ (resp. $\bot$) if $v_h(p) = \mathbf{t}$ (resp. if $v_h(p) = \mathbf{f}$). Then in step 1, we replace $v_h(\psi) = \mathbf{t}$ by $\models \psi[v_h]$ and $v_h(\psi) = \mathbf{f}$ by $\not\models \psi[v_h]$. The modified algorithm can be implemented by a deterministic Turing machine with **NP**-oracle.

coNP-hardness is shown by a reduction of SAT to the complement of our problem. Suppose $\psi \equiv \exists x_1, \dots x_n \varphi(x_1, \dots, x_n)$ is a SAT instance. We define $\mathcal{P}rops = \{x_1, \dots, x_n\}$, $\mathcal{P}layers = \{1\}$, $\mathcal{P}_1 = \emptyset$, $\Gamma = \Gamma_1 = \{\neg\varphi\}$, omitting the other elements of F. Then there is only a single game base $H = \{\epsilon\}$ and we have that i is not winning iff $v(\neg\varphi) = \mathbf{f}$ for some valuation of $\mathcal{P}rops$ iff ψ is true [cf. condition 1 of Def. 2.10[8]]. □

The next result sheds light on the interesting case of compact representations. In this paper, we only consider the case of π-alternating (full-branching) game bases. Note that there is a gap between lower and upper bound.

Proposition 3.3. *The problem whether a player is winning in a frame F over π-alternating F-game bases is in $\mathbf{\Sigma_4^P}$ and is $\mathbf{\Sigma_2^P}$-hard in the size of F and π.*

[7] ∞ has its standard meaning; in particular, $x < \infty$ for all $x \in \mathbb{R}$.

[8] Note that in this case conditions 2 and 3 of Def. 2.10 are vacuously true, as ϵ has no proper subhistories.

Proof. **Membership**: Firstly, we sketch an algorithm checking whether i is winning in (F, π) (cf. Def. 2.10). We proceed bottom-up and assume that h, h', h'' are given as in Definition 2.10.

The problem given in item 3 of Def. 2.10, denoted as the language L_3, can be solved by a non-deterministic TM in polynomial time by guessing a completion v'' of the h''-induced valuation and because Γ_i is part of the input and can be traversed in polynomial time. The same holds for the problem given in item 2(b) of Def. 2.10, which we denote by L_{2b}. Thus, $L_{2b}, L_3 \in \mathbf{NP}$.

The problem given in 2(a) of Def. 2.10, denoted L_{2a}, can be solved by a non-deterministic **NP**-oracle TM in polynomial time by first guessing an i-extension v' of the h''-induced valuation and checking whether for all completions v of v', $v(\varphi') = \mathbf{t}$. The latter can be implemented by an oracle guessing a completion v, checking whether $v(\varphi') = \mathbf{f}$ and reverting the answer. This shows that $L_{2a} \in \mathbf{\Sigma_2^P}$.

Problem L_{2c} corresponding to 2(c) can also be solved analogously to 2(b) by guessing an appropriate completion; thus, $L_{2c} \in \mathbf{NP}$.

Summing up, item 2, denoted by problem L_2, can be solved by a deterministic TM with an oracle solving L_{2a}, L_{2b}, L_{2c}. Thus, $L_2 \in \mathbf{\Sigma_2^P}$.

Analogously, the complement of the problem in item 1 of Def. 2.10, $\bar{L}_1$, can be solved by a non-deterministic TM in polynomial time: firstly, a completion v of the h'-induced valuation is guessed and then it is verified whether $v(\varphi) = \mathbf{f}$. Hence, $L_1 \in \mathbf{coNP}$.

Now, for a given s_i the question whether for all F-histories $h \in H|_{s_i}$ there is a subhistory $h' \leq h$ and a formula $\varphi \in \Sigma_i$ such that the conditions 1, 2 and 3 hold and can be solved in $\mathbf{\Sigma_3^P}$. We denote the problem by L. To see this we construct a non-deterministic TM with a $\mathbf{\Sigma_2^P}$ oracle accepting the complement of L. ($\star$) Firstly, the machines guesses an F-history h and checks whether for all $h' \leq h$ (there are only polynomially many) and all $\varphi \in \Gamma_i$ (Γ_i is part of the input), conditions 1, 2 or 3 are violated. The latter can be determined by a query to a $\mathbf{\Sigma_2^P}$ oracle following our previous considerations. So, $L \in \mathbf{coNP}^{\mathbf{\Sigma_2^P}} = \mathbf{\Pi_3^P}$.

Finally, to check whether i is winning we use a non-deterministic $\mathbf{\Pi_3^P}$-oracle TM to guess a strategy s_i and to verify whether it satisfies the conditions of Definition 2.10 and whether $c(s_i) \leq C_i$. We have just shown the verification of the former can be done in $\mathbf{\Pi_3^P}$. Whether s_i adheres to the cost limit can be incorporated in ($\star$). This shows that whether there is a winning strategy can be solved in $\mathbf{\Sigma_4^P}$.

Hardness: Next, we prove $\mathbf{\Sigma_2^P}$-hardness by a reduction of Q_2SAT [13]. Let $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1 \ldots y_m\}$ and let $x \notin X \cup Y$ be a fresh variable. Suppose that $\psi \equiv \exists X \forall Y \varphi(X, Y)$ is a Q_2SAT instance.[9] We define $\mathcal{P}layers = \{1\}$, $\Gamma = \{x\} \cup \Gamma_1$, $\Gamma_1 = \{\varphi\}$, $\mathcal{P}rops = X \cup Y \cup \{x\}$, $\mathcal{P}_1 = X \cup \{x\}$, and no cost limits.

"$\Rightarrow$:" Suppose 1 is winning. Then, on all histories (there is only one for a given strategy!) there is a (sub)history $h' = x_{i_1}, \ldots, x_{i_k}$ such that it is true that $\forall X \backslash \{x_{i_1}, \ldots, x_{i_k}\} \forall Y \varphi[v_{h'}]$; here, $\varphi[v_{h'}]$ is φ but with proposition $p \in$

[9] Here, $\exists X \xi(X)$ abbreviates $\exists x_1 \ldots \exists x_n \xi(x_1, ..., x_n)$.

$\{x_{i_1}, \ldots, x_{i_k}\}$ in φ replaced by $\top$ if $v_{h'}(p) = \mathbf{t}$ and by $\bot$ if $v_{h'}(p) = \mathbf{f}$. However, this implies that $\exists X \backslash \{x_{i_1}, \ldots, x_{i_k}\} \forall Y \varphi[v_{h'}]$ holds and thus that ψ is true.

"$\Leftarrow$:" Suppose ψ is true and let v_X be a witnessing truth assignment of the variables in X. Let s_X be the strategy that assigns truth values to propositions according to v_X with respect to the order $x_1, \ldots, x_n$ and makes x true afterwards. We show that s_X is a winning strategy (again, note that there is only one history for a fixed s_X). Then there is a minimal subsequence $h' = x_1, \ldots, x_k$ with $\forall X \backslash \{x_1, \ldots, x_k\} \forall Y \varphi[v_{h'}]$. Such a sequence exists because the history $h = x_1, \ldots, x_n, x$ satisfies φ (for, ψ is true). We need to show that all the other conditions of a goal-achieving strategy are satisfied. We consider h' and first assume that $h \neq h'$ (i.e. $x_k \neq x$) and consider $x \in \Gamma$. In this case, 3 of Def. 2.10 is true by definition. Condition 2(a) of Def. 2.10 holds because the 1-extension in which x is set $\mathbf{t}$ is a witness. For condition 2(b), any completion where x is set $\mathbf{f}$ is sufficient; and for 2(c), payer 1 has to choose an appropriate truth value for x. For $h = h'$, the same argument holds for all subhistories, and in the last step the goal of 1 is already true. $\square$

4 Conclusion

In this article, we propose a game-like model to describe how an agent can go about trying to achieve a goal without letting the others know until the goal has been reached. The turn-based Boolean games used in our setting facilitate modeling situations where a player can play and strategize based on how the others have acted in the history of the game, in order to keep the player's intended goals secret. The point of *trying to achieve something in secret* would be lost if we considered normal form games instead of turn-based ones.

Various recent work [9, 8] has focused on variants of cooperative Boolean games, introduced in [7]. The current work has taken some inspiration from those articles, but we used the idea of Boolean games that was introduced in [10] for modeling interactive situations.

We have addressed the question whether a player is winning in a game and have analyzed the computational complexity with respect to explicit and compact game representations. In our future research, we would like to close the gaps in the complexity results and to consider more sophisticated solution concepts. We also plan to elaborate on other compact game representations, not only π-alternating game bases.

For future work, it would also be interesting to combine the notion of secret goals with other forms of uncertainty; for example, agents could have incomplete information about which other agent controls which variables, as in [16]. One issue that has not been touched in this work is that of 'cooperativeness' [7]. In what ways can some of the players cooperate to achieve their goals? We would like to propose characterizations and complexity results for cooperativeness in our setting. Finally, it would be interesting to investigate whether one can adapt a dynamic framework similar to [11] to model the idea of achieving secret goals.

Acknowledgements We would like to thank the three anonymous reviewers for their useful suggestions. This research was supported by Vici grant NWO 277-80-001 awarded to Rineke Verbrugge.

References

1. T. Ågotnes, W. van der Hoek, and M. Wooldridge. Scientia potentia est. In Sonenberg et al. [15], pages 735–742.
2. M. Barbier. *D-Day Deception: Operation Fortitude and the Normandy Invasion*. Greenwood Press, Westport, CT, 2007.
3. E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini. Boolean games revisited. In G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, editors, *ECAI*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 265–269. IOS Press, 2006.
4. F. Dignum, B. Dunin-Keplicz, and R. Verbrugge. Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9(2):289–304, 2001.
5. A. Dumas. *Le Comte de Monte-Cristo*. Gallimard, Paris, 1844.
6. B. Dunin-Kęplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley, Chichester, 2010.
7. P. E. Dunne, W. van der Hoek, S. Kraus, and M. Wooldridge. Cooperative boolean games. In L. Padgham, D. C. Parkes, J. P. Müller, and S. Parsons, editors, *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 1015–1022. IFAAMAS, 2008.
8. U. Endriss, S. Kraus, J. Lang, and M. Wooldridge. Incentive engineering for Boolean games. In Walsh [17], pages 2602–2607.
9. J. Grant, S. Kraus, M. Wooldridge, and I. Zuckerman. Manipulating Boolean games through communication. In Walsh [17], pages 210–215.
10. P. Harrenstein, W. van der Hoek, J.-J. Meyer, and C. Witteveen. Boolean games. In J. van Benthem, editor, *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 287–298, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
11. A. Herzig, E. Lorini, F. Moisan, and N. Troquard. A dynamic logic of normative systems. In Walsh [17], pages 228–233.
12. M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.
13. C. Papadimitriou. *Computational Complexity*. Addison Wesley : Reading, 1994.
14. H. Raiffa, J. Richardson, and D. Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Belknap Press of Harvard Univ. Press, Cambridge (MA), 2002.
15. L. Sonenberg, P. Stone, K. Tumer, and P. Yolum, editors. *Proceedings 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*. IFAAMAS, 2011.
16. W. van der Hoek, N. Troquard, and M. Wooldridge. Knowledge and control. In Sonenberg et al. [15], pages 719–726.
17. T. Walsh, editor. *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. IJCAI/AAAI, 2011.
18. G. Zlotkin and J. S. Rosenschein. Incomplete information and deception in multi-agent negotiation. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 225–231, 1991.