The complexity of Scotland Yard

Merlijn Sevenster
sevenstr@science.uva.nl
ILLC, University of Amsterdam

March 8, 2006

Abstract

This paper discusses a case study of the board game of Scotland Yard from a computational perspective. Interestingly, Scotland Yard is a genuine "playgame" with imperfect information. For reasons not completely clear to me, games with imperfect information have escaped the interest of researchers in Algorithmic combinatorial game-theory. I show by means of a powerset argument, that Scotland Yard can also be considered a game of perfect information, that is surprisingly similar to the original game – up to isomorphism, that is. Using the powerset analysis, I show that Scotland Yard has PSPACE-complete complexity be it with or without imperfect information. In fact, imperfect information may even simplify matters: if the cops are supposed to be consequently ignorant of Mr. X's whereabouts throughout the game the complexity is 'but' NP-complete.

1 Introduction

The discipline of *Algorithmic Combinatorial Game Theory* (ACGT) deals with zero-sum games with perfect information. Although the existence of game with imperfect information is acknowledged in one of ACGT's seminal publications [1, pg. 16-7], as yet only a marginal amount of literature appeared on games with imperfect information. On the other hand, the number of publications on games with perfect information is abundant and offers a robust picture of the computational behavior of games: One-person games or *puzzles* are usually solvable in NP and many of them turn out to be complete for this class.[1] Famous examples include the game of *Minesweeper* [13] and *Clickomania* [2]. Alternation kicks in at considerable computational cost: many natural games have PSPACE-hard complexity, such as *Go* [15] and the semantic evaluation game of *quantified boolean formulae* [21, 20]. Some even have EXPTIME-complete complexity. Typical examples in this respect are the games of *Chess* [9] and *Checkers* [8]. By and large, the games with EXPTIME-complete complexity are of a *loopy* nature, that is, configurations of the game may occur over and over again. In real-life, loopy games may not be that much fun to play, as they allow for annoyingly long runs in which neither player makes any 'progress'. Loopy runs are banned from Chess by imposing that no configuration of the game occurs more than three times, roughly speaking.

Amusingly, putting an upper-bound on the duration of the game not only avoids loopy – and boring – sequences of play, but also has considerable computational impact. Papadimitriou [17, pg. 460-2] argues that every game that meets the following requirements is solvable in PSPACE:

- the length of any legal sequence of moves is bounded by a polynomial in the size of the input;
- given a 'board position' of the game there is a polynomial-space algorithm that constructs all possible next actions and board positions; or, if there is none, decides whether the board position is a win for either player.

Note that Papadimitriou does not even mention the fact that this result concerns games of perfect information. The result goes through due to the fact that the backwards induction algorithm can be run on the game's game tree in PSPACE, given that it meets the above requirements.

As for games of imperfect information some studies have been performed and their reports are basically a bad news show. In slogan one may put that imperfect information increases the computational complexity of games. Convincing results are reported in [14], in which the authors show that deciding whether either player has a winning strategy in a finite, two-player game of perfect information can be done in polynomial time in the size of

[1] To solve a game, means to determine for an instance of the game whether a designated player has a winning strategy.

the game tree. On a positive note they show that there is a P-algorithm that solves the same problem for games of imperfect information with perfect recall. However, if one of the players (or both) suffers from imperfect recall the problem of deciding whether this player has a winning strategy is NP-hard in the size of the game tree.

In [19, 18] the authors consider computation trees as game trees. This view on computation trees is adopted from [6], in which Turing machines are considered that have existential and universal states, so-called *alternating Turing machines*. The aspect of alternation is reflected in the computation tree by regarding it a game tree of a two-player game. The nodes corresponding to existential (universal) states belong to the existential (universal) player. So this makes these game two-player games with perfect information. From this viewpoint, non-deterministic Turing machines give rise to one-player game trees. In [19, 18] this idea is extended towards games of imperfect information. The authors define, amongst other devices, *private alternating Turing machines*, that give rise to computation trees that may be regarded two-player game trees in which the existential player suffers from imperfect information. It is shown that the space complexity of $f(n)$ of these machines is characterized in terms of the complexity of alternating Turing machines with space bound exponential in $f(n)$. Moreover, it is shown that private alternating Turing machines with three players – two of them are factually teaming up – can recognize undecidable problems in constant space.

Dramatic as these results may be, being general studies they cannot tell us what is the computational impact of the imperfect information found in actual games. That is, games developed to be played rather than to be analyzed.[2] It may well turn out that the imperfect information in these games have little computational impact and that the games themselves match the robust intuitions we have about the computational nature of perfect information games. As I pointed out before, there is but a small number of results games of imperfect information, let alone computational studies of real games. For this reason, I will consider the game of *Scotland Yard* that has amused game players ever since 1983.[3] The reader familiar with

[2]Fraenkel makes the distinction between "*PlayGames*" and "*MathGames*". The former being the games that "are challenging to the point that people will purchase them and play them", whereas the latter games "are challenging to a mathematician [...] to play with or ponder about"; cited from [7, pg. 476].

[3]Scotland Yard is produced by *Ravensburger/Milton Bradley* and was prestigiously

Figure 1: The box of Scotland Yard and its items, amongst which the game board, Mr. X's move board, and the players' pawns. This picture is reproduced with permission of *Ravensburger*.

Scotland Yard will acknowledge that it is the imperfect information that makes the game an enjoyable waste of time and enthusiastic accounts of players' experiences with Scotland Yard are easily found on the Internet, for instance [3].

Scotland Yard is played on a game board, that contains approximately 200 numbered intersections of colored lines denoting available means of transportation: yellow for taxis, green for buses, and pink for underground. A game is played by two to six people, one of them being *Mr. X*, the others teaming up and thusly forming Scotland Yard. They have a shared goal: capturing Mr. X. Initially, every player gets assigned a pawn and an intersection on the game board on which his or her pawn is positioned. Before the game starts every player gets a fixed number of tickets for every means of transportation. Mr. X and the cops move alternatingly and Mr. X commences.

declared *Spiel des Jahres* in 1983.

During every stage of the game, each player – be it Mr. X or his adversaries – takes an intersection in mind connected from his or her current intersection, subject to him or her owning at least one ticket of the appropriate kind. For instance, if a player would want to use the metro from Buckingham Palace, she would have to hand in her metro ticket. If either player is out of tickets for a certain mode of transportation, he cannot travel along the related lines. The set of tickets of every player is publicly known to all players at every stage of the game.

If a cop has made up her mind on moving to an intersection, this is indicated by her moving the pawn under her control to the intersection at hand. However, if Mr. X made up his mind he secretely writes the number of the intersection at stake at the designated entry of the *move board* and covers it with the ticket used. Effectively, the cops know what means of transportation Mr. X has been using, but do not know his position. After round 3, 8, 13, 18, and 24, however, Mr. X is forced to show his whereabouts by putting his pawn on his current hideout.

The game lasts for 24 rounds during which Mr. X and the cops make their actions. If at any stage of the game, any of the cops is at the same intersection as Mr. X then the cops win. If Mr. X remains uncaught until after the last round, he wins the game. Cops who have a suspicious nature may want to check whether Mr. X's secret moves were made consistently with the lines on the game board, when the game is over. To this end, they would match the numbers on the move board with the returned tickets. If it turns out that Mr. X's cheated halfway, he loses no matter what the outcome of the game actually was.

In view of these game descriptions, the generalization of the Scotland Yard game in Definition 1 should be easy to swallow. The reader will observe that I reduced the number of means of transportation to one and that the game board is modelled by a directed graph. All results in this chapter can be generalized to hold for several means of transportation and undirected graphs, though.

Definition 1 *Let* $G = \langle V, E \rangle$ *be a finite, connected, directed graph with out-degree* ≥ 1. *Let* $u, v_1, \ldots, v_n \in V$. *Let* $f : \{1, \ldots, k\} \to \{\mathit{show}, \mathit{hide}\}$ *be the* information function, *for some integer* $2 < k < |V|$. *Then, let* $\langle G, \langle u, v_1, \ldots, v_n \rangle, f \rangle$ *be a* (Scotland Yard) instance. *Most of the time it will be convenient to abbreviate a string of vertices* $v_1, \ldots, v_n$ *by* $\vec{v}$. *Conversely,* $\vec{v}(i)$ *shall denote the ith element in* $\vec{v}$. $\{\vec{v}\}$ *denotes the set of vertices in* $\vec{v}$.

If $U \subseteq V$, then let $\{u' \in V \mid E(u, u'),$ for some $u \in U\}$ be denoted by $E(U)$. If $\vec{v}, \vec{v}' \in V^n$, then write $E(\vec{v}, \vec{v}')$ to denote that for every $1 \leq i \leq n$, $E(\vec{v}(i), \vec{v}'(i))$.

The information function f controls the imperfect information throughout the game. If round i has property $f(i) = hide$, Mr. X hides himself. As we will see the information function gives an intuitive sense to 'adding or removing' imperfect information from a Scotland Yard game. For instance, if one restricts oneself to information functions with range $\{show\}$, Mr. X shows his whereabouts after every move and one is considering a game of perfect information. Under the latter restriction, one has arrived at so-called *Pursuit* or *Cops and robbers* games. For a exposition of the literature on these games, consult [10].

The aims of this chapter are twofold. Firstly, pinpointing the computational complexity of a real game of imperfect information. Secondly, I go through a reasonable amount of effort to spell out the relation between the Scotland Yard game and a game of perfect information that is highly similar to the former game. More precisely, I show that the games' game trees are isomorphic, from a natural point of view, and that a winning strategy in the one game is a winning strategy in the other and vice versa. These similarity result may convince the reader that in some cases the wall between perfect and imperfect information is not as impenetrable as one might induce from the scarce literature on complexity of imperfect information games.

In Section 2, I define the extensive game form of the Scotland Yard game to which an instance gives rise. Next I introduce another game, related to Scotland Yard, that is of perfect information: Scotland Yard-PI.

In Section 3, I show that the justly introduced games admit for a bijection between the imperfect information game's information partitions (actually, an extension thereof) and the histories in Scotland Yard-PI. In this game, Mr. X exchanges the power of hiding for the power of moving sets of vertices.

In Section 4, the computational results are presented. In accordance with many polynomially bounded two-player games, Scotland Yard is complete for PSPACE, despite its imperfect information. That is, the computational complexity of Scotland Yard does not change when one only considers information functions with range $\{show\}$.

In fact, if one would add more imperfect information to the extent that the information flow function has range $\{hide\}$, the resulting decision problem is easier: NP-complete. This is shown in Section 5.

2 Two Scotland Yard games

Let $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ be a Scotland Yard instance as in Definition 1. Before I define the extensive game form of the Scotland Yard game to which sy gives rise, let me recap the game rules tailor-made to suit sy's particulars.

The digraph G is the board on which the actual playing finds place. In the initial situation of the game, we find $n + 1$ pawns, named $\forall, \exists_1, \ldots, \exists_n$, positioned on the respective vertices $u_*, \vec{v}_*(1), \ldots, \vec{v}_*(n)$ in the digraph. The game is played by the two players $\exists$ and $\forall$ over k rounds, and with every round $1 \leq i \leq k$ in the game there is associated the property $f(i) \in \{show, hide\}$. Note that I converted the n-player game of Scotland Yard, where $2 \leq n \leq 6$ into a two-player game in which one player controls all pawns $\exists_1, \ldots, \exists_n$. Furthermore, for reasons of succinctness I adopt the symbol $\forall$ to refer to Mr. X and $\exists$ to refer to the player controlling Scotland Yard. Somewhat sloppily, sometimes I will not make a strict distinction between a player and (one of his or her) pawns.

First fix $i = 1$, $u = u_*$, and $\vec{v} = \vec{v}_*$; now, round i of Scotland Yard goes as follows:

1. If for some $1 \leq j \leq n$, the pawns $\forall$ and $\exists_j$ share the same vertex, i.e., $u = \vec{v}(j)$ we say that $\forall$ was *captured* (by $\exists_j$). If $\forall$ is captured the game stops and $\exists$ wins. If $\forall$ is not captured and $i > k$ the game also stops but $\exists$ loses.

2. $\forall$ chooses a vertex u', such that $E(u, u')$. If $f(i) = show$, he physically puts his pawn on u'. If $f(i) = hide$, he secretly writes u' on his move board making sure that it cannot be seen by his opponent. Set $u = u'$.

3. Player $\exists$ chooses a vector $\vec{v}' \in V^n$, such that $E(\vec{v}, \vec{v}')$, and for every $1 \leq j \leq n$, moves pawn $\exists_j$ to $\vec{v}'(j)$. Set $\vec{v} = \vec{v}'$.

4. Set $i = i + 1$.

Note that these game rules do not consider the possibility of either player getting stuck, as in not being able to move a pawn under his or her control moving along an edge. This goes without loss of generality, as the digraphs at stake are supposes to have out-degree ≥ 1.

Further, it should be borne in mind, that for $\forall$ it is not a guaranteed loss to move to a vertex occupied by one of $\exists$'s pawns. The game only terminates after $\exists$ has moved *and* one of her pawns captures $\forall$, unlike the game rules for the board game of Scotland Yard.

Scotland Yard is modelled as an extensive game with imperfect information in Definition 2. The upcoming definition and Definition 4 are notationally akin to the definitions from [16].

Definition 2 *Let* $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ *be a Scotland Yard instance. Then, let the* extensive Scotland Yard game constituted by *sy be defined as the tuple* $SY(sy) = \langle N, H, P, \sim, U \rangle$*, where*

- $N = \{\exists, \forall\}$ *is the set of players.*
- H *is the set of* histories*, that is, the smallest set containing* $\langle u_* \rangle, \langle u_*, \vec{v}_* \rangle$ *and is closed under actions taken by* $\forall$ *and* $\exists$*:*

  · *If* $h\langle u, \vec{v} \rangle \in H$*,* $\ell(h\langle u, \vec{v} \rangle) < k$*,* $u \notin \{\vec{v}\}$*, and* $E(u, u')$*, then* $h\langle u, \vec{v} \rangle \langle u' \rangle \in H$*.*

  · *If* $h\langle u, \vec{v} \rangle \langle u' \rangle \in H$ *and* $E(\vec{v}, \vec{v}')$*, then* $h\langle u, \vec{v} \rangle \langle u', \vec{v}' \rangle \in H$*.*

  If $h \in H$*, let* $\ell(h)$ *denote the number of rounds in* h*, that is the number of tuples not equal* $\langle u_*, \vec{v}_* \rangle$*. Define* $\ell(\langle u_*, \vec{v}_* \rangle) = 0$*. Somewhat unlike custom usage in game-theory, the length* $\ell(h)$ *of history* h *does not coincide with the number of plies in the game. This notation reflects my game rule saying that a history my only terminate after* $\exists$ *has moved.*

  Let $\succ$ *be the* immediate successor *relation on* H*. That is, the smallest relation closed under the following conditions:*

  · *If* $h, h\langle u \rangle \in H$*, then* $h \succ h\langle u \rangle$*.*

  · *If* $h\langle u \rangle, h\langle u, \vec{v} \rangle \in H$*, then* $h\langle u \rangle \succ h\langle u, \vec{v} \rangle$*.*

  A history that has no immediate successor we call a terminal history*. Let* $Z \subseteq H$ *be the set of terminal histories in* H*.*

- $P : H - Z \rightarrow \{\exists, \forall\}$ *is the* player function *that decides who is to move in a non-terminal history. Due to the notational convention, the value of* P *is easily determined by the history's form, in the sense that* $P(h\langle u \rangle) = \exists$ *and* $P(h\langle u, \vec{v} \rangle) = \forall$*.*

- $\sim$ *is the* indistinguishability relation *that formalizes the imperfect information in the game. It is defined such that for any pair of histories $h, h' \in H$, where*

$$h = \langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \ldots \langle u_i \rangle \quad \textit{and} \quad h' = \langle u_*, \vec{v}_* \rangle \langle u'_1, \vec{v}'_1 \rangle \ldots \langle u'_i \rangle \tag{1}$$

  it is the case that $h \sim h'$, if

  (a) $\vec{v}_j = \vec{v}'_j$, for every $1 \leq j \leq i-1$

  (b) $u_j = u'_j$, for every $1 \leq j \leq i$ such that $f(j) = show$.

  The previous condition, considering histories as in (1), defines $\sim$ only as a relation between histories h in which $\exists$ has to move: $P(h) = \exists$. This reflects the fact that it is $\exists$ who experiences the imperfect information while playing the game. Somewhat unusual, I extend $\sim$ to histories in which $\forall$ has to move. The reader is urged to take this extension as a technicality, and not to start looking for deeper explanations (after reading Theorem 19). I put as follows: for any pair of histories $h, h' \in H$, where

$$h = \langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \ldots \langle u_i, \vec{v}_i \rangle \quad \textit{and} \quad h' = \langle u_*, \vec{v}_* \rangle \langle u'_1, \vec{v}'_1 \rangle \ldots \langle u'_i, \vec{v}'_i \rangle$$

  it is the case that $h \sim h'$, if

  (a) $\vec{v}_j = \vec{v}'_j$, for every $1 \leq j \leq i$

  (b) $u_j = u'_j$, for every $1 \leq j \leq i$ such that $f(j) = show$.

- *$U : Z \rightarrow \{win, lose\}$ is the function that decides whether a terminal history $h\langle u, \vec{v} \rangle$ is won or lost for $\exists$. Formally,*

$$U(h\langle u, \vec{v} \rangle) = \begin{cases} \textit{win} & \textit{if } u \in \{\vec{v}\} \\ \textit{lose} & \textit{if } u \notin \{\vec{v}\}. \end{cases}$$

  Usually, one has a utility function per player, but as the game is win-loss one had just as well stick to one function.

Since $\sim$ is reflexive, symmetric, and transitive it defines an equivalence relation on H. Let us write $\mathcal{H} \subseteq \wp(H)$ for the set of equivalence classes, or *information cells*, in which H is partitioned by $\sim$. That is, $\mathcal{H} = \{C_1, \ldots, C_m\}$, where $C_1 \cup \ldots \cup C_m = H$ and for every $1 \leq i \leq m$, if $h \in C_i$ and $h \sim h'$, then $h' \in C_i$. A standard inductive argument suffices to see that for every $C_i \in \mathcal{H}$ and pair of histories $h, h' \in C_i$, the length of h and h' coincides and $P(h) = P(h')$.

I lift the relation $\succ$ to $\mathcal{H}$, using the same symbol: For any pair $C, C' \in \mathcal{H}$, I write $C \succ C'$ if there exists histories $h \in C$ and $h' \in C'$ such that $h \succ h'$. It is easy to see that if h, h' are histories in a cell $C \in \mathcal{H}$, then $P(h) = P(h')$. Thus, the player function is meaningfully lifted as follows: if $C \in \mathcal{H}$ and h is a history in C, then $P(C) = P(h)$. Call a cell $C \in \mathcal{H}$ *terminal* if all its histories $h \in C$ are terminal.

Since I study an extension of $\sim$, the set $\mathcal{H}$ partitions all histories in H. As I pointed out in the definition of $\sim$, if histories h and h' stand in the $\sim$ relation and belong to $\forall$, this should not be taken to reflect any conceptual consideration about $\exists$'s experiences, as it is merely a technicality. Yet, if h and h' belong to $\exists$, to write $h \sim h'$ reflects genuine indistinguishability for player $\exists$ between the two histories h and h'. In this manner, we see that a *subset* of $\mathcal{H}$ is an object familiar from game-theory. Consider the set $\mathcal{H}_\exists = \{C \in \mathcal{H} \mid P(C) = \exists\}$, that partitions the set of histories that belong to $\exists$. I claim that $\mathcal{H}_\exists$ is an *information set.* To prove this claim it suffices to show that for every information cell $C \in \mathcal{H}_\exists$ no two histories $h, h' \in C$ can be distinguished on the basis of the actions that $\exists$ can take at h and h'. Formally, for every $C \in \mathcal{H}$ and every pair of histories $h, h' \in C$ it is the case that $A(h) = A(h')$.

To this end, let

$$A(h\langle u, \vec{v}\rangle\langle u'\rangle) = \{\vec{v}' \in V^n \mid h\langle u, \vec{v}\rangle\langle u'\rangle \succ h\langle u, \vec{v}\rangle\langle u', \vec{v}'\rangle\} = \{\vec{v}' \in V^n \mid E(\vec{v}, \vec{v}')\}$$

define the actions available to $\exists$ after $h\langle u, \vec{v}\rangle\langle u'\rangle$. Let h and h' be histories as in (1) sitting in the same cell $C \in \mathcal{H}$. Then, by (a) $\vec{v}_{i-1} = \vec{v}'_{i-1}$ and therefore $A(h) = A(h')$. Hence, $\mathcal{H}_\exists$ is an information set. (Note that information cells in $\mathcal{H}_\exists$ are usually called *information partitions.*) Thus, by modelling the imperfect information in $SY(sy)$ by the extended $\sim$ relation an object is obtained that is still highly similar to the customary object $\langle N, H, P, \langle \mathcal{I}_i\rangle_{i\in N}, U\rangle$ modelling the game of Scotland Yard induced by sy.

For future reference, lay down the following Proposition:

Proposition 3 *Let $SY(sy) = \langle N, H, P, \sim, U\rangle$ be the Scotland Yard game constituted by sy. Then, the following statements hold:*

1. *If* $h_1\langle u_1\rangle \sim h_2\langle u_2\rangle$ *and* $f(\ell(h_1\langle u_1\rangle)) = \mathit{hide}$*, then* $h_1 \sim h_2$.

2. *If* $h_1\langle u_1\rangle \sim h_2\langle u_2\rangle$ *and* $f(\ell(h_1\langle u_1\rangle)) = \mathit{show}$*, then* $h_1 \sim h_2$ *and* $u_1 = u_2$.

3. *If* $h_1\langle u_1, \vec{v}_1\rangle \sim h_2\langle u_2, \vec{v}_2\rangle$*, then* $h_1\langle u_1\rangle \sim h_2\langle u_2\rangle$ *and* $\vec{v}_1 = \vec{v}_2$.

4. *If* $h_1 \not\sim h_2$ *and* $h_1\langle u_1\rangle, h_2\langle u_2\rangle \in H$*, then* $h_1\langle u_1\rangle \not\sim h_2\langle u_2\rangle$.

Proof. Readily observed from the definition of $\sim$ in Definition 2. □

As an illustration of modelling a Scotland Yard instance as an extensive game with imperfect information, consider the digraph $G^\times = \langle V^\times, E^\times\rangle$, where

$$\begin{array}{rcl} V^\times & = & \{u_*, v_*, a, b, A, B, 1, 2, 3\} \\ E^\times & = & \{\langle u_*, a\rangle, \langle u_*, b\rangle, \langle a, 1\rangle, \langle b, 2\rangle, \langle b, 3\rangle, \\ & & \langle v_*, A\rangle, \langle v_*, B\rangle, \langle A, 1\rangle, \langle B, 2\rangle, \langle B, 3\rangle\}. \end{array}$$

For a depiction of $G^\times$, see Figure 2. Let $f^\times$ be a function such that $f^\times(1) = \mathit{hide}$ and $f^\times(2) = \mathit{show}$. Let us conclude the construction of the Scotland Yard instance $sy^\times$, by putting u_* and v_* as the initial vertices of $\forall$ and $\exists$, respectively. In $SY(sy^\times)$, the set of histories H contains exactly the following histories:

$$\begin{array}{lll} \langle u_*, v_*\rangle & & \\ \langle u_*, v_*\rangle\langle a\rangle & \langle u_*, v_*\rangle\langle a, B\rangle\langle 1\rangle & \langle u_*, v_*\rangle\langle a, B\rangle\langle 1, 3\rangle \\ \langle u_*, v_*\rangle\langle b\rangle & \langle u_*, v_*\rangle\langle b, A\rangle\langle 2\rangle & \langle u_*, v_*\rangle\langle b, A\rangle\langle 2, 1\rangle \\ \langle u_*, v_*\rangle\langle a, A\rangle & \langle u_*, v_*\rangle\langle b, A\rangle\langle 3\rangle & \langle u_*, v_*\rangle\langle b, A\rangle\langle 3, 1\rangle \\ \langle u_*, v_*\rangle\langle a, B\rangle & \langle u_*, v_*\rangle\langle b, B\rangle\langle 2\rangle & \langle u_*, v_*\rangle\langle b, B\rangle\langle 2, 2\rangle\ ! \\ \langle u_*, v_*\rangle\langle b, A\rangle & \langle u_*, v_*\rangle\langle b, B\rangle\langle 3\rangle & \langle u_*, v_*\rangle\langle b, B\rangle\langle 2, 3\rangle \\ \langle u_*, v_*\rangle\langle b, B\rangle & \langle u_*, v_*\rangle\langle a, A\rangle\langle 1, 1\rangle\ ! & \langle u_*, v_*\rangle\langle b, B\rangle\langle 3, 2\rangle \\ \langle u_*, v_*\rangle\langle a, A\rangle\langle 1\rangle & \langle u_*, v_*\rangle\langle a, B\rangle\langle 1, 2\rangle & \langle u_*, v_*\rangle\langle b, B\rangle\langle 3, 3\rangle\ ! \end{array}$$

The terminal histories marked with an exclamation mark are winning histories for $\exists$. Because $f^\times(1) = \mathit{hide}$, the game that we are dealing with is a genuine game of imperfect information. This fact is reflected in the set of information cells $\mathcal{H}$, containing the following three non-singletons: $\{\langle u_*, v_*\rangle\langle a\rangle, \langle u_*, v_*\rangle\langle b\rangle\}$, $\{\langle u_*, v_*\rangle\langle a, A\rangle, \langle u_*, v_*\rangle\langle b, A\rangle\}$, and finally there is $\{\langle u_*, v_*\rangle\langle a, B\rangle, \langle u_*, v_*\rangle\langle b, B\rangle\}$. (Note that under the customary definition of $\sim$, one would not have the latter two information cells, as they belong

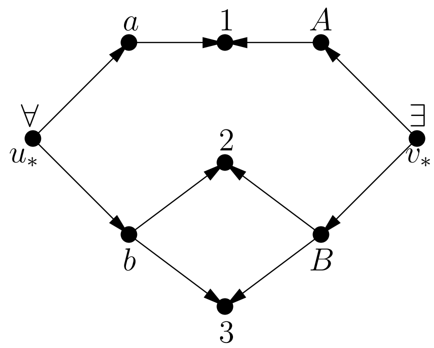


Figure 2: The digraph $G^{\times}$, allowing for a two-round Scotland Yard game.

to ∀.) Game-theorists often find it convenient to present extensive games as trees, see Figure 3.

I observed that Scotland Yard is a game with imperfect information and in Definition 2 I modelled it as an extensive game with imperfect information. This model one may find Scotland Yard's canonical means of analysis, for admittedly, it gives a natural account of the imperfect information that makes Scotland Yard such a fun game to play. Canonical or not, this does not imply, of course, that Scotland Yard can *only* be analyzed as an imperfect information game. In the remainder of this section I will show how a Scotland Yard instance may also give rise to a game of perfect information. The underlying idea is that during rounds in which ∀ hides his whereabouts, he picks up a *set of vertices* that contain all vertices where he can possibly be. In case ∀ has to show himself, he selects one vertex from the current set of vertices and announces this vertex as his new location.

Formally, ∀'s powers are lifted from the level of picking up vertices to the level of picking up sets of vertices. ∃'s power remain unaltered, as compared to the game with imperfect information that was explicated above.

Modelling imperfect information by means of a powerset construction – as I am about to do – is by no means new. For instance, the reader may find this idea occurring in the computational analyzes of games with imperfect information [19, 18]. In logic, the idea of evaluating a formula from Independence friendly logic with respect to a set of assignments underlies Hodges' *trump semantics* [11], a variant of which appeared in [4]. An Ehrenfeucht-Fraïssé game for IF logic with perfect information was defined in [22]. In automata

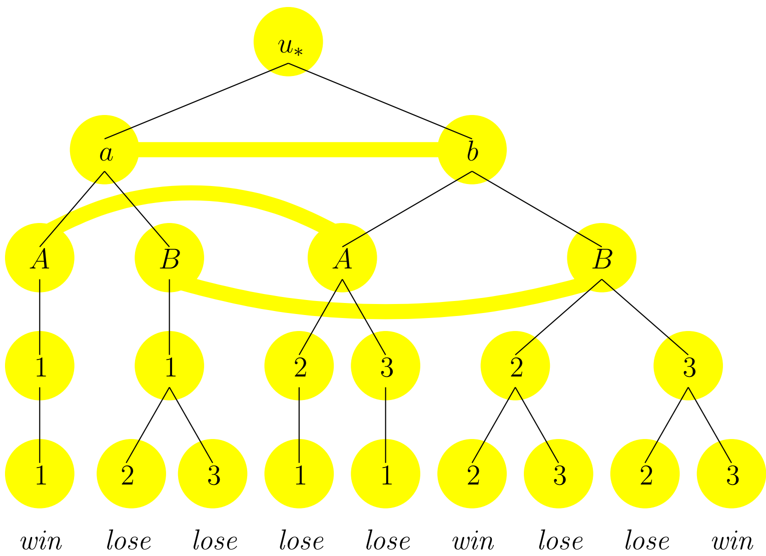


Figure 3: A graphical representation of the Scotland Yard game played on the game board constituted by the digraph $G^{\times}$ from Figure 2. A path from the root to any of its nodes represents a history in the game. For instance, the path $u_*, b, A, 2$ corresponds with the history $\langle u_*, v_* \rangle \langle b, A \rangle \langle 2 \rangle$. The information cells are indicated by the shaded areas.

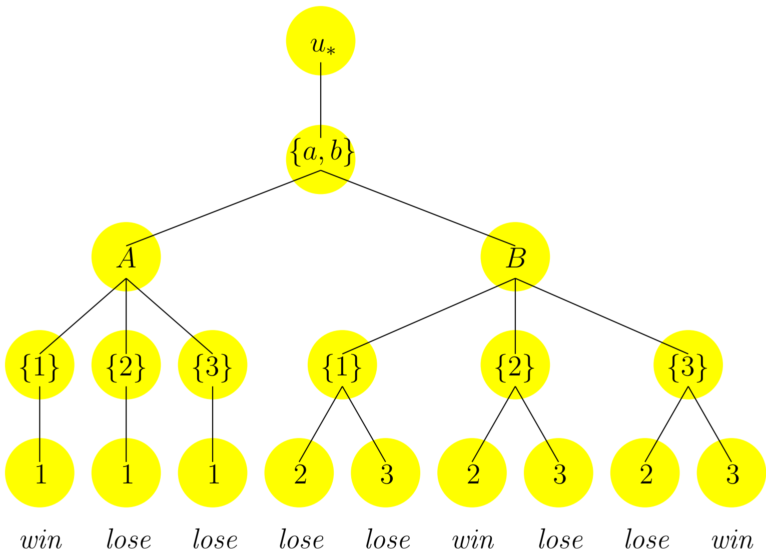


Figure 4: A graphical representation of the Scotland Yard-PI game played on the game board constituted by the digraph $G^{\times}$ from Figure 2. A path from the root to any of its nodes represents a history in the game. For instance, the path $u_*, \{a, b\}, A, \{2\}$ corresponds with the history $\langle\{u_*\}, v_*\rangle\langle\{a, b\}, A\rangle\langle\{2\}\rangle$.

theory, the move to powersets is made when converting a non-deterministic finite automaton to a deterministic one, see [12].

In all three disciplines, however, observe that the phenomenon that was analyzed through powersets is substantially more powerful than the original phenomenon. For instance, in [18] it was shown that three-player games with imperfect information can be undecidable. In the realm of IF logic it was proven [5] that no compositional semantics can be given based on single assignments only. And it is well-known that in the worst case converting a non-deterministic finite automaton makes the number of states increase exponentially.

In view of these results it is striking that one can define a Scotland Yard game with perfect information using a powerset argument, that is highly similar to the same Scotland Yard game with imperfect information. What is meant by 'highly similar' is made precise in Section 3. First let me postulate the game rules for the Scotland Yard game with perfect information and define its extensive game form in Definition 4.

Let $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ be a Scotland Yard instance as in Definition 1. The initial position of the Scotland Yard-PI game constituted by sy is similar to the initial position of the Scotland Yard game that sy constitutes. That is, a $\forall$ pawn is positioned on u_* and for every $1 \leq j \leq n$, the $\exists_j$ pawn is positioned on $\vec{v}(j)$. In Scotland Yard-PI, $\forall$ doesn't have one pawn at his disposal but as many as there are vertices in G. First fix $i = 1$, $U = \{u_*\}$, and $\vec{v} = \vec{v}_*$; round i of Scotland Yard-PI goes as follows:

1-PI. If $U - \{\vec{v}\} = \varnothing$, then the game stops and $\exists$ wins. If $U - \{\vec{v}\} \neq \varnothing$ and $i > k$ the game also stops but $\exists$ loses.

2-PI. Let $U' = E(U - \{\vec{v}\})$. If $f(i) = \textit{hide}$, then set $U = U'$ and $\forall$ positions a $\forall$ pawn on every vertex v in U. If $f(i) = \textit{show}$, then $\forall$ picks a vertex $u' \in U'$, removes all his pawns from the board, and puts one pawn on u'. Set $U = \{u'\}$.

3-PI. Player $\exists$ chooses a vector $\vec{v}' \in V^n$, such that $E(\vec{v}, \vec{v}')$, and for every $1 \leq j \leq n$, moves pawn $\exists_j$ to $\vec{v}'(j)$. Set $\vec{v} = \vec{v}'$.

4-PI. Set $i = i + 1$.

Clearly, for arbitrary sy, the Scotland Yard-PI game constituted by sy is a game of perfect information. For this reason a natural means of analysis is to model it as an extensive game.

Definition 4 *Let* $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ *be a Scotland Yard instance. Then, let the* extensive Scotland Yard-PI game constituted by sy *be defined as the tuple* $SY\text{-PI}(sy) = \langle N_{\mathrm{PI}}, H_{\mathrm{PI}}, P_{\mathrm{PI}}, U_{\mathrm{PI}} \rangle$*, where*

- $N_{\mathrm{PI}} = \{\exists, \forall\}$ *is the set of players.*

- H_{PI} *is the set of* histories*, that is, the smallest set containing the strings* $\langle \{u_*\} \rangle, \langle \{u_*\}, \vec{v}_* \rangle$*, that furthermore is closed under taking actions for* $\exists$ *and* $\forall$*:*

  · *If* $h\langle U, \vec{v} \rangle \in H_{\mathrm{PI}}$*,* $\ell(h\langle U, \vec{v} \rangle) \leq k$*,* $f(\ell(h\langle U, \vec{v} \rangle) + 1) = \textit{hide}$*, and* $U - \{\vec{v}\} \neq \varnothing$*, then* $h\langle U, \vec{v} \rangle \langle E(U - \{\vec{v}\}) \rangle \in H_{\mathrm{PI}}$*.*

  · *If* $h\langle U, \vec{v} \rangle \in H_{\mathrm{PI}}$*,* $\ell(h\langle U, \vec{v} \rangle) \leq k$*, and* $f(\ell(h\langle U, \vec{v} \rangle) + 1) = \textit{show}$*, then* $\{h\langle U, \vec{v} \rangle \langle \{u'\} \rangle \mid u' \in E(U - \{\vec{v}\})\} \subseteq H_{\mathrm{PI}}$*.*

  · *If* $h\langle U, \vec{v} \rangle \langle U' \rangle \in H_{\mathrm{PI}}$ *and* $E(\vec{v}, \vec{v}')$*, then* $h\langle U, \vec{v} \rangle \langle U', \vec{v}' \rangle \in H_{\mathrm{PI}}$*.*

  Let $\succ_{\mathrm{PI}}$ *be the* immediate successor *relation on* H_{PI}*. That is, the smallest relation closed under the following conditions:*

  · *If* $h, h\langle U \rangle \in H_{\mathrm{PI}}$*, then* $h \succ_{\mathrm{PI}} h\langle U \rangle$*.*

  · *If* $h\langle U \rangle, h\langle U, \vec{v} \rangle \in H_{\mathrm{PI}}$*, then* $h\langle U \rangle \succ_{\mathrm{PI}} h\langle U, \vec{v} \rangle$*.*

  A history that has no immediate successor we call a terminal history*. Let* $Z_{\mathrm{PI}} \subseteq H_{\mathrm{PI}}$ *be the set of terminal histories in* H_{PI}*.*

- $P_{\mathrm{PI}} : H_{\mathrm{PI}} - Z_{\mathrm{PI}} \to \{\exists, \forall\}$ *is the player function that decides who is to move in a non-terminal history. Due to the notational convention, the value of* P *is determined by the history's form, in the sense that* $P(h\langle U \rangle) = \exists$ *and* $P(h\langle U, \vec{v} \rangle) = \forall$*.*

- $U_{\mathrm{PI}} : Z_{\mathrm{PI}} \to \{\textit{win}, \textit{lose}\}$ *is the function that decides whether a terminal history* $h\langle U, \vec{v} \rangle$ *is won or lost for* $\exists$*. Formally,*

$$U(h\langle U, \vec{v} \rangle) = \begin{cases} \textit{win} & \textit{if } U - \{\vec{v}\} = \varnothing \\ \textit{lose} & \textit{if } U - \{\vec{v}\} \neq \varnothing. \end{cases}$$

These definitions may be best appreciated by checking SY-PI($sy^{\times}$), where $sy^{\times} = \langle G^{\times}, \langle u_*, \vec{v}_* \rangle, f^{\times} \rangle$ and $G^{\times}$ is the digraph depicted in Figure 2. I skip writing down all histories in this particular game, leaving the reader with a graphical representation of its game tree in Figure 4.

3 An effective equivalence

In this section, the similarity between Scotland Yard, the game with imperfect information, and its perfect information variant is established. Making use of this similarity, I prove that for any instance sy, $\exists$ has a winning strategy in $SY(sy)$ iff she has one in SY-PI(sy), cf. Theorem 19. In order to prove this result, I go about as follows: Firstly, it will be shown in Lemma 10 that the structures $\langle \mathcal{H}, \succ \rangle$ and $\langle H_{\mathrm{PI}}, \succ_{\mathrm{PI}} \rangle$ are isomorphic. Secondly, I formally introduce the notion of a winning strategy and the backwards induction algorithms for $SY(sy)$ and SY-PI(sy). This algorithm typically labels every history with *win* or *lose*, starting with the terminal histories. Crucially, I show that the backwards induction algorithms correctly compute whether $\exists$ has a winning strategy in the respective game. Finally, I show that for every history h in $SY(sy)$, the label assigned to it by the backwards induction algorithm for Scotland Yard corresponds with the label assigned to it by the backwards induction algorithm for Scotland Yard-PI. The claim then follows, as the initial histories $\langle u_*, \vec{v}_* \rangle$ and $\langle \{u_*\}, \vec{v}_* \rangle$ carry the same label.

3.1 Scotland Yard and Scotland Yard-PI are isomorphic

Main result of this subsection resides in Lemma 10, saying that the structures $\langle \mathcal{H}, \succ \rangle$ and $\langle H_{\mathrm{PI}}, \succ_{\mathrm{PI}} \rangle$ are isomorphic. The witness of this isomorphism is the bijection β, defined in Definition 5 below. As some of the intermediate results that bring us to the bijection lemma are not very insightful, I defer them to Appendix A.

Definition 5 *Let $SY(sy)$ and SY-PI(sy) be games constituted by sy. Define the function $\beta : H_{\text{PI}} \to \wp(H)$ inductively as follows:*

$$
\begin{array}{rcl}
\beta(\langle\{u_*\}\rangle) & = & \{\langle u_*\rangle\} \\
\beta(\langle\{u_*\}, \vec{v}_*\rangle) & = & \{\langle u_*, \vec{v}_*\rangle\} \\
\beta(h\langle U\rangle) & = & \{g\langle u\rangle \in H \mid g \in \beta(h), u \in U\} \\
\beta(h\langle U, \vec{v}\rangle) & = & \{g\langle u, \vec{v}\rangle \in H \mid g\langle u\rangle \in \beta(h\langle U\rangle)\}.
\end{array}
$$

The function β is (partially) depicted in Figure 5 mapping the histories from SY-PI$(sy^\times)$ to sets of histories from $SY(sy^\times)$. The reader may find it useful to return to this figure to strengthen his or her intuitions.

Proposition 6 states that if in a history $h \in H_{\text{PI}}$ a pawn (owned by either player) is positioned on a vertex, then also in $\beta(h)$ there exists a history in which this vertex is occupied by a pawn.

Proposition 6 *For every history $h' \in H_{\text{PI}}$, the following hold:*

1. *If $h' = h\langle U\rangle$ and $f(\ell(h\langle U\rangle)) =$ hide, then it is the case that $U = \{u \mid g\langle u\rangle \in H$, for some $g \in \beta(h)\}$.*
2. *If $P(h') = \forall$ and $f(\ell(h') + 1) =$ show, then it is the case that $\{u \mid h' \succ h'\langle\{u\}\rangle$, for some $h'\langle\{u\}\rangle \in H_{\text{PI}}\} = \{u \mid g\langle u\rangle \in H$, for some $g \in \beta(h')\}$.*
3. *If $h' = h\langle U\rangle \in H_{\text{PI}}$ and $u \in U$, then there exists a history $g \in \beta(h)$ such that $g\langle u\rangle \in H$.*
4. *If $h' = h\langle U, \vec{v}\rangle \in H_{\text{PI}}$, then it is the case that $\beta(h\langle U, \vec{v}\rangle) = \{g\langle u, \vec{v}\rangle \mid g\langle u\rangle \in \beta(h\langle U\rangle)\}$.*

Proposition 7 is the converse of the previous Proposition, as it links up histories in H with histories in H_{PI}.

Proposition 7 *For every $g' \in H$, the following hold:*

1. *If $g' = g\langle u\rangle \in H$, then there exists a $h\langle U\rangle \in H_{\text{PI}}$ such that $g \in \beta(h)$ and $u \in U$.*
2. *If $g' = g\langle u, \vec{v}\rangle \in H$, then there exists a $h\langle U, \vec{v}'\rangle \in H_{\text{PI}}$ such that $g\langle u\rangle \in \beta(h\langle U\rangle)$ and $\vec{v} = \vec{v}'$.*

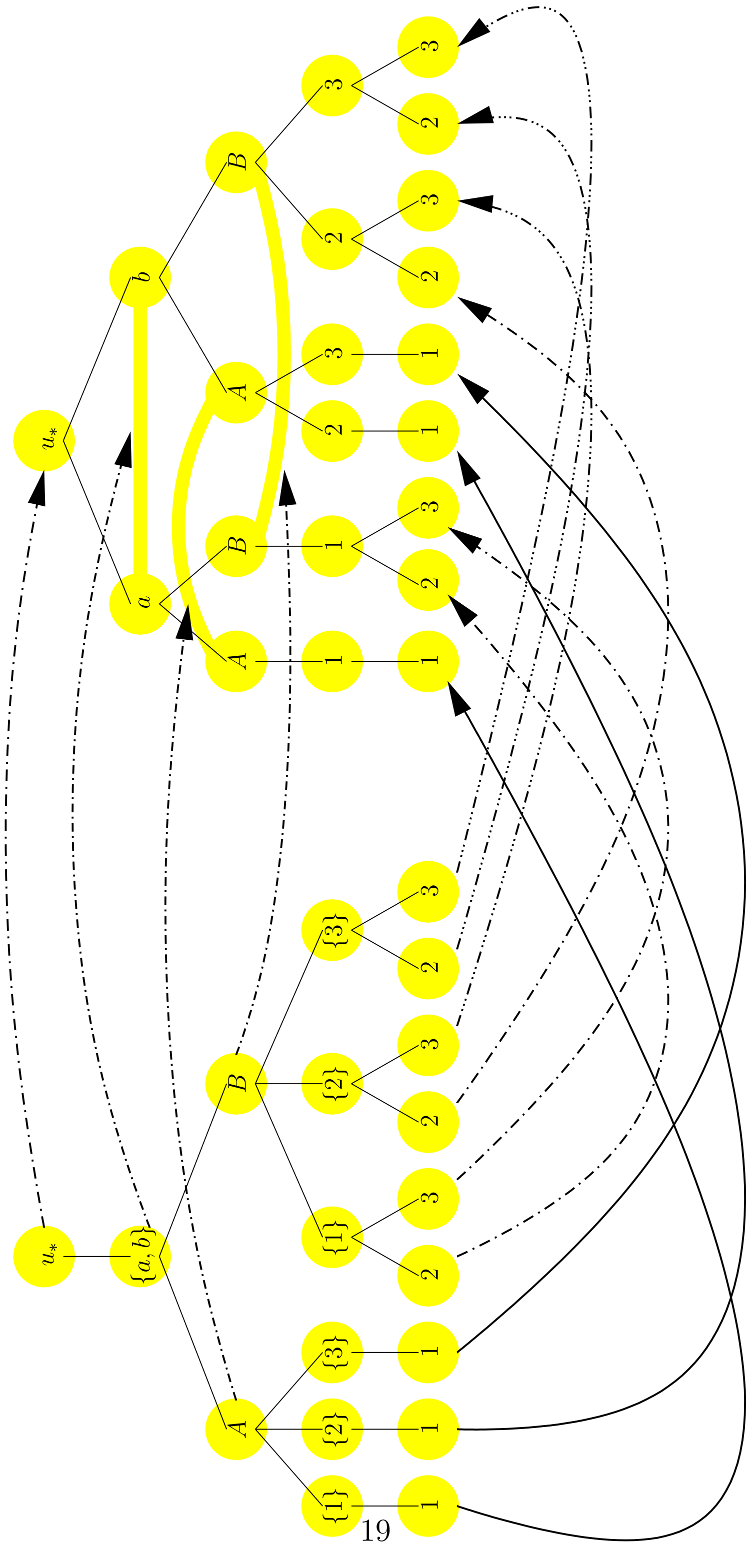


Figure 5: A partial depiction of the bijection β from histories in SY-PI$(sy^{\times})$ to sets of histories from $SY(sy^{\times})$. β is displayed using several kinds of arrows to enhance readability. Note that these different arrows do not reflect any conceptual difference. Sets of histories in the range of β (found in the right-hand structure) turn out to be information cells, cf. Lemma 9.

For β to be bijection between H_{PI} and $\mathcal{H}$, it ought to be the case that β has range $\mathcal{H}$ rather than $\wp(H)$. I lay down the following result.

Lemma 8 *β is a function of type $H_{\mathrm{PI}} \to \mathcal{H}$.*

The latter lemma is strengthened in the following lemma.

Lemma 9 *β is a bijection between H_{PI} and $\mathcal{H}$.*

The isomorphism result follows from tying together the previous statements.

Lemma 10 *The structures $\langle H_{\mathrm{PI}}, \succ_{\mathrm{PI}} \rangle$ and $\langle \mathcal{H}, \succ \rangle$ are isomorphic.*

Proof. Lemma 9 showed that β is a bijection between H_{PI} and $\mathcal{H}$. It remains to be shown that β preserves structure, that is, for every pair of histories $h, h' \in H_{\mathrm{PI}}$, it is the case that $h \succ_{\mathrm{PI}} h'$ iff $\beta(h) \succ \beta(h')$. Recall that for $C' \in \mathcal{H}$ to be the immediate successor of $C \in \mathcal{H}$, there must exists two histories g, g' from C, C', respectively, such that $g \succ g'$. The claim is proved by a straightforward inductive argument on the length of the histories in H_{PI}. I shall omit spelling out the details of the proof, only mentioning the Propositions on which it relies:

From left to right. Follows from Proposition 6.3 and Proposition 6.4.

From right to left. Follows from Propositions 7.1 and 7.2. □

The claim that Scotland Yard and its perfect information variant are highly similar is justified by pointing at the structures (game trees) that the games give rise to and the latter lemma, saying that they are isomorphic.

3.2 Backwards induction algorithms

The structures $\langle H_{\mathrm{PI}}, \succ_{\mathrm{PI}} \rangle$ and $\langle \mathcal{H}, \succ \rangle$ are not only isomorphic, they also preserve the property of being winnable for the cops. Traditionally it is backwards induction algorithms that compute whether the cops win, but such algorithms are only defined on games with perfect information. As a consequence, the backwards induction algorithm applies readily to the game tree of any perfect information SY-PI(sy). Matters are not so straightforward in case of $SY(sy)$. But as I shall prove, a backwards induction algorithm can be developed in this case as well. To this end, I define the notion of a winning strategy for games with imperfect information, and introduce a backwards induction algorithm that computes whether a structure allows for a winning strategy, for the Scotland Yard game with imperfect information.

I claim that the way of modelling a Scotland Yard game with imperfect information – be it the standard way using information sets $\langle \mathcal{I}_i \rangle_{i \in N}$ or as in Definition 2 – is immaterial, when it comes to $\exists$ having a winning strategy.

Definition 11 *Let $SY(sy) = \langle N, H, P, \sim, U \rangle$ be a Scotland Yard game constituted by sy. Then, we call the structure $\langle \mathcal{S}, \succ' \rangle$ a* plan of action for $\exists$ in $SY(sy)$*, if*

- $\mathcal{S} \subseteq \mathcal{H}$
- $\{\langle u_*, \vec{v}_* \rangle\} \in \mathcal{S}$
- $\succ' = \succ \cap (\mathcal{S} \times \mathcal{S})$
- *for every $C \in \mathcal{S}$ such that $P(C) = \exists$, there exists exactly one $C' \in \mathcal{S}$ such that $C \succ' C'$*
- *for every $C \in \mathcal{S}$ such that $P(C) = \forall$ and every $C' \in \mathcal{H}$ such that $C \succ C'$, it is the case that $C' \in \mathcal{S}$.*

Call $\langle \mathcal{S}, \succ' \rangle$ a winning plan of action for $\exists$ in $SY(sy)$*, if $\langle \mathcal{S}, \succ' \rangle$ is a plan of action and every terminal cell $C \in \mathcal{S}$ only contains histories h such that $U(h) = win$.*

Before I show that the backwards induction algorithm applied to $SY(sy)$ gives the same result as the standard one applied to SY-PI(sy), for every sy, it needs to be shown that the way of modelling $SY(sy)$ using the extended notion of $\sim$, does not have an influence. That is, had I modelled the Scotland Yard game for instance sy as a customary $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ then it would allow for a winning strategy for the cops if, and only if, my backwards inductions algorithm accepts the input $SY(sy) = \langle N, H, P, \sim, U \rangle$.

First let me recite the necessary vocabulary in the notation used in this chapter for Scotland Yard games modelled in the customary way. A *strategy in* $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ is defined, cf. [16], as a function S mapping every information partition $I \in \mathcal{I}_\exists$ to an action $A(h)$, for some $h \in C$.[4] Let S be a

[4]Recall that for every pair of histories h and h' belonging to $\exists$, if h and h' sit in the same information partition I, then $A(h) = A(h')$.

strategy in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ and let $h = \langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \ldots \langle u_i, \vec{v}_i \rangle \in H$ be a history, then call h *in accordance with* S, if for every $1 \leq j \leq i$, $S(I_j) = \vec{v}_j$, where I_j is the information partition in $\mathcal{I}_\exists$ containing $\langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \ldots \langle u_j \rangle$. A strategy S in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ is called *winning for* $\exists$ if every terminal history h in H that is in accordance with S is won for $\exists$: $U(h) = \mathit{win}$.

Proposition 12 *Let sy be a Scotland Yard instance and let $G(sy)$ equal the tuple $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ be the extensive game with imperfect information modelling the Scotland Yard game constituted by sy in the customary way. Then, $\exists$ has a winning plan of action in $SY(sy)$ iff she has a winning strategy in $G(sy)$.*

Proof. It is not so hard to see that every winning strategy S in the extensive game $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ can be transformed into a winning plan of action $\langle \mathcal{S}, \succ' \rangle$ in $SY(sy)$ and vice versa.

From left to right. Suppose S is a winning strategy in the extensive game $G(sy) = \langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$. Let H_S be all histories in H that are in accordance with S. Let $\mathcal{H}_S$ be the partition of H_S, such that for any two histories $h, h' \in H_S$, if $h \sim h'$, then there is a cell $D \in \mathcal{H}_S$ containing both h and h'. I claim that $T = \langle \mathcal{H}_S, \succ \cap (\mathcal{H}_S \times \mathcal{H}_S) \rangle$ is a winning plan of action in $SY(sy)$. To this end, it needs proof that (i) $\mathcal{H}_S \subseteq \mathcal{H}$, that (ii) T is a winning plan of action.

To prove (i), one needs to show that $\mathcal{H}_S$ is a set of information cells. To this end, it suffices to show that H_S is closed under $\sim$: if $h \in H_S$ and $h \sim h'$, then $h' \in H_S$. I do so by means of an inductive argument. The base case is trivial. Suppose that $h \in H_S$ and that h' is a history such that $h \sim h'$, where

$$h = h_0 \langle u_0, \vec{v}_0 \rangle \langle u_1, \vec{v}_1 \rangle \quad \text{and} \quad h' = h'_0 \langle u'_0, \vec{v}'_0 \rangle \langle u'_1, \vec{v}'_1 \rangle.$$

The other case is trivial and therefore omitted. It needs to be shown that h' is in accordance with S as well, that is, $h' \in H_S$. Since h is in accordance with S, $S(I) = \vec{v}_1$, where $I \in \mathcal{I}_\exists$ is the information partition holding h. Derive from Proposition 3.3 that $\vec{v}_1 = \vec{v}'_1$ and that $h_0 \langle u_0, \vec{v}_0 \rangle \langle u_1 \rangle \sim h'_0 \langle u'_0, \vec{v}'_0 \rangle \langle u'_1 \rangle$. Since h is in accordance with S, so is $h_0 \langle u_0, \vec{v}_0 \rangle \langle u_1 \rangle$. Applying the inductive hypothesis yields that $h'_0 \langle u'_0, \vec{v}'_0 \rangle \langle u'_1 \rangle$ is in accordance with S. Hence, $S(I) = \vec{v}_1 = \vec{v}'_1$, implying that h' is in accordance with S.

As for (ii), it needs to be shown that T is closed under taking actions and preserves winning. But this follows easily from S's being a winning strategy.

From right to left. Suppose $\langle \mathcal{S}, \succ' \rangle$ is a winning plan of action in $SY(sy)$. Then, for every $C \in \mathcal{S}$ belonging to $\exists$ there is one C' such that $C \succ C'$. Essentially similar to Proposition 6.4 one proves that $C' = \{h\langle u, \vec{v}_C \rangle \mid h\langle u \rangle \in C\}$, for some $\vec{v}_C \in V^n$. Put $S(C) = \vec{v}_C$ and for every information partition C not present in $\mathcal{S}$, put $S(C) = \vec{v}$ for an arbitrary vector of vertices $\vec{v}$ that properly extends every history in C. It is readily observes that S is a winning strategy in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$. □

Definition 13 *Let $SY(sy) = \langle N, H, P, \sim, U \rangle$ be a Scotland Yard game constituted by sy. The algorithm B-Ind effectively labels every cell $C \in \mathcal{H}$ with B-Ind$(C) \in \{win, lose\}$ and proceeds as follows:*

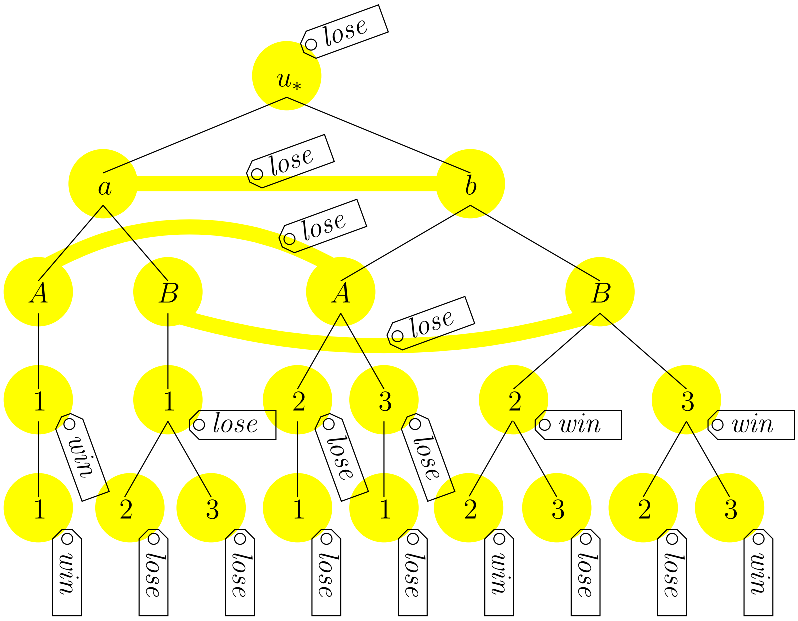
- *every $h \in Z$ is painted color$(h) \in \{white, limegreen\}$ in such a way that color$(h) = white$ iff $U(h) = win$*
- *every terminal information cell $C \in \mathcal{H}$ is given the label B-Ind$(C) = win$ iff color$(h) = white$, for every $h \in C$*
- *until every cell has been labelled, apply the following routine to every cell $C \in \mathcal{H}$ that has no label, but all of whose successors have:*
    - · *If $P(C) = \exists$ and there exists a successor C' of C that has been labelled B-Ind$(C') = win$, then C gets the label B-Ind$(C) = win$; otherwise, C gets the label B-Ind$(C) = lose$.*
    - · *If $P(C) = \forall$ and there exists a successor h' of C that has been labelled B-Ind$(C') = lose$, then C gets the label B-Ind$(C) = lose$; otherwise, C gets the label B-Ind$(C) = win$.*

Write B-Ind(sy) to denote B-Ind$(\langle u_*, \vec{v}_* \rangle)$.

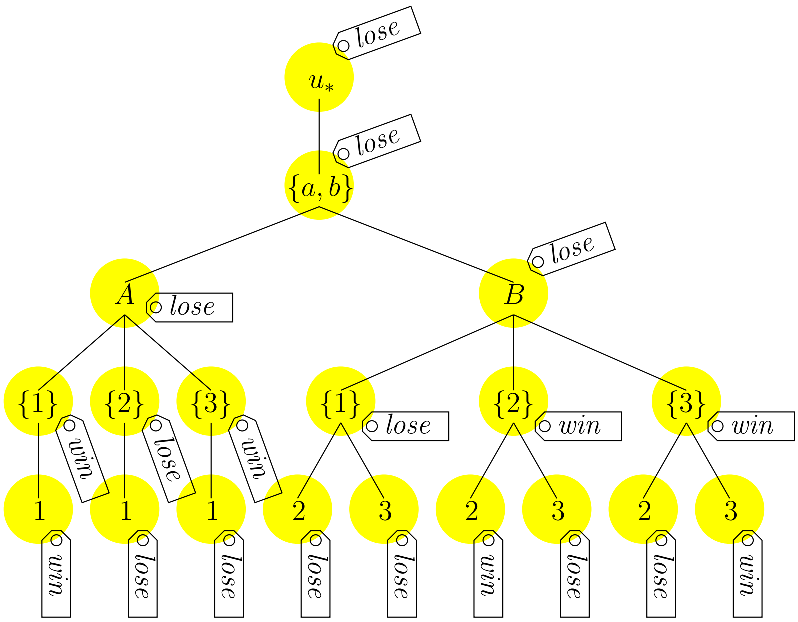
Proposition 14 *Let $SY(sy)$ be a Scotland Yard game constituted by sy. Then, $\exists$ has a winning strategy in $SY(sy)$ iff B-Ind$(sy) = win$.*

Proof. Trivial. □

The analogous definitions and proposition for Scotland Yard-PI follow below.



(a)



(b)

Figure 6: The histories in $SY(sy^{\times})$ and SY-PI$(sy^{\times})$ labelled by the backwards induction algorithm B-Ind and B-Ind_{PI} depicted in (a) and (b), respectively.

Definition 15 *Let* $SY\text{-PI}(sy) = \langle N_{\text{PI}}, H_{\text{PI}}, P_{\text{PI}}, U_{\text{PI}} \rangle$ *be a Scotland Yard-PI game constituted by* sy*. Then, we call the structure* $\langle S_{\text{PI}}, \succ'_{\text{PI}} \rangle$ *a* plan of action for $\exists$ in $SY\text{-PI}(sy)$*, if*

- $S_{\text{PI}} \subseteq H_{\text{PI}}$
- $\langle u_*, \vec{v}_* \rangle \in S_{\text{PI}}$
- $\succ'_{\text{PI}} = \succ_{\text{PI}} \cap (S_{\text{PI}} \times S_{\text{PI}})$
- *for every* $h \in S_{\text{PI}}$ *such that* $P_{\text{PI}}(h) = \exists$*, there exists exactly one* $h' \in S_{\text{PI}}$ *such that* $h \succ'_{\text{PI}} h'$
- *for every* $h \in S_{\text{PI}}$ *such that* $P_{\text{PI}}(h) = \forall$ *and every* $h' \in H_{\text{PI}}$ *such that* $h \succ_{\text{PI}} h'$*, it is the case that* $h' \in S_{\text{PI}}$*.*

Call $\langle S_{\text{PI}}, \succ'_{\text{PI}} \rangle$ *a* winning plan of action for $\exists$ in $SY\text{-PI}(sy)$*, if* $\langle S_{\text{PI}}, \succ'_{\text{PI}} \rangle$ *is a strategy and every terminal history* $h \in S_{\text{PI}}$ *has it that* $U_{\text{PI}}(h) = win$*.*

Definition 16 *Let* $SY\text{-PI}(sy) = \langle N_{\text{PI}}, H_{\text{PI}}, P_{\text{PI}}, U_{\text{PI}} \rangle$ *be a Scotland Yard-PI game constituted by* sy*. The algorithm* $B\text{-}Ind_{\text{PI}}$ *effectively labels every history* h *in* H_{PI} *with* $B\text{-}Ind_{\text{PI}}(h) \in \{win, lose\}$*, proceeding as follows:*

- *Every* $h \in Z_{\text{PI}}$ *is labelled* $B\text{-}Ind_{\text{PI}}(h) = U_{\text{PI}}(h)$*.*
- *Until every history has been labelled, apply the following routine to every history* $h \in H_{\text{PI}}$ *that has no label, but all whose successors have:*
  - · *If* $P(h) = \exists$ *and there exists a successor* h' *of* h *that has been labelled* $B\text{-}Ind_{\text{PI}}(h') = win$*, then* h *gets the label* $B\text{-}Ind_{\text{PI}}(h) = win$*; otherwise,* h *gets the label* $B\text{-}Ind_{\text{PI}}(h) = lose$*.*
  - · *If* $P(h) = \forall$ *and there exists a successor* h' *of* h *that has been labelled* $B\text{-}Ind_{\text{PI}}(h') = lose$*, then* h *gets the label* $B\text{-}Ind_{\text{PI}}(h) = lose$*; otherwise,* h *gets the label* $B\text{-}Ind_{\text{PI}}(h) = win$*.*

Write $B\text{-}Ind_{\text{PI}}(sy)$ *to denote* $B\text{-}Ind_{\text{PI}}(\langle u_*, \vec{v}_* \rangle)$*.*

It is easy to see that every winning strategy constitutes a winning plan of action. Conversely, it is an easy exercise to show that every winning plan of action gives rise to a winning strategy, simply by extending it so as to apply to every information partition. What action is prescribed precisely is immaterial.

Proposition 17 *Let SY-PI(sy) be a Scotland Yard-PI game constituted by sy. Then, $\exists$ has a winning strategy in SY-PI(sy) iff $B\text{-}Ind_{\mathrm{PI}}(sy) = win$.*

Proof. Trivial. □

For an example of the two backwards induction algorithms at work, see Figure 6.

The upcoming Lemma states that β is a bijection respecting the labels of the respective objects.

Lemma 18 *For every $h \in H_{\mathrm{PI}}$, $B\text{-}Ind_{\mathrm{PI}}(h) = B\text{-}Ind(\beta(h))$.*

Proof. I prove by induction on the histories $h \in H_{\mathrm{PI}}$. The most interesting case is the base step.

Suppose $h\langle U, \vec{v}\rangle \in Z_{\mathrm{PI}}$. Suppose $B\text{-}Ind_{\mathrm{PI}}(h\langle U, \vec{v}\rangle) = win$. By definition of $B\text{-}Ind_{\mathrm{PI}}$ applied to terminal histories it follows that $(U - \{\vec{v}\}) = \varnothing$. Put differently, every $u \in U$ is an element of $\{\vec{v}\}$. Now consider an arbitrary history $g\langle u', \vec{v}'\rangle$ from $\beta(h\langle U, \vec{v}\rangle)$. By definition of β, it follows that $u' \in U$ and that $\vec{v}' = \vec{v}$. But then, $u' \in \{\vec{v}'\}$ and consequently $U(g\langle u', \vec{v}'\rangle) = win$. Thus the backwards induction for Scotland Yard paints $g\langle u', \vec{v}'\rangle$ with the color *white*. Since $g\langle u', \vec{v}'\rangle$ was chosen arbitrarily, conclude that every history in $\beta(h\langle U, \vec{v}\rangle)$ is painted *white*, whence $B\text{-}Ind(\beta(h\langle U, \vec{v}\rangle)) = win$.

Conversely, suppose $B\text{-}Ind_{\mathrm{PI}}(h\langle U, \vec{v}\rangle) = lose$. By definition of $B\text{-}Ind_{\mathrm{PI}}$ applied to terminal histories it follows that $(U - \{\vec{v}\})$ contains at least one object, call it u. By Proposition 6.3 derive that there exists a history $g\langle u\rangle \in \beta(h\langle U\rangle)$. From Proposition 6.4 it follows that $g\langle u, \vec{v}\rangle$ is a successor of $g\langle u\rangle$, since $h\langle U\rangle$ is a successor of $h\langle U, \vec{v}\rangle$. Furthermore, $g\langle u, \vec{v}\rangle$ is an element of $\beta(h\langle U, \vec{v}\rangle)$. Since u does not sit in $\{\vec{v}\}$ the history $g\langle u, \vec{v}\rangle$ is painted *limegreen*, by the backwards induction algorithm of Scotland Yard. Since one of its elements is painted *limegreen*, it is the case that $B\text{-}Ind(\beta(h\langle U, \vec{v}\rangle)) = lose$.

Suppose $h\langle U\rangle$ is non-terminal. Suppose that $B\text{-}Ind_{\mathrm{PI}}(h\langle U\rangle) = win$, therefore, $h\langle U\rangle$ has a successor $h\langle U, \vec{v}\rangle$, such that $B\text{-}Ind_{\mathrm{PI}}(h\langle U, \vec{v}\rangle) = win$. Applying the inductive hypothesis to $h\langle U, \vec{v}\rangle$ yields that $B\text{-}Ind(\beta(h\langle U, \vec{v}\rangle)) = win$. Lemma 10 established that β is a order preserving bijection. Hence, $\beta(h\langle U\rangle) \succ \beta(h\langle U, \vec{v}\rangle)$ and therefore we may conclude that $B\text{-}Ind(\beta(h\langle U\rangle)) = win$. The converse case runs along the same line.

Suppose $h\langle U, \vec{v}\rangle$ is non-terminal. Analogous to the previous case. □

Tying together these results brings us to the desired conclusion:

Theorem 19 *Let sy be a Scotland Yard instance. Then, $\exists$ has a winning strategy in $SY(sy)$ iff she has a winning strategy in SY-*PI*(sy).*

Proof. Follows immediately from Propositions 14 and 17 and Lemma 18, since $\exists$ has a winning strategy in $SY(sy)$ iff $B\text{-}Ind(sy) = win$ iff $B\text{-}Ind_{\mathrm{PI}}(sy) = win$ iff $\exists$ has a winning strategy in SY-PI(sy). $\square$

4 Scotland Yard is PSPACE-complete

Let Scotland Yard be the set of all Scotland Yard instances sy such that $\exists$ has a winning strategy in $SY(sy)$. As a special case let the set of Scotland Yard instances Scotland Yard$_{\clubsuit}$ equal

$$\{\langle G, \langle u_*, \vec{v}_* \rangle, f\rangle \in \text{Scotland Yard} \mid f \text{ has range } \{\clubsuit\}\},$$

where $\clubsuit \in \{show, hide\}$.

Scotland Yard and Scotland Yard$_{show}$ both have PSPACE-complete complexity, as I show in this section. From this one may conclude that the imperfect information in Scotland Yard does not have a computational impact. Surprisingly, if $\exists$ cannot see the whereabouts of $\forall$ at any stage of the game, the decidability problem ends up being NP-complete. That is, Scotland Yard$_{hide}$ is complete for NP. The latter claim is treated in Section 5.

Lemma 20 Scotland Yard $\in$ PSPACE*.*

Proof. Required is a PSPACE algorithm that for arbitrary Scotland Yard instances sy decides whether $\exists$ has a winning strategy in $SY(sy)$. By Theorem 19, it is sufficient to provide a PSPACE algorithm that decides the same problem with respect to SY-PI(sy). This equivalence comes in useful, because SY-PI(sy) is a game of perfect information and can for this reason be dealt with by means of the traditional machinery. In fact, the very same machinery provided to us by Papadimitriou cited in Section 1. Papadimitriou, namely, observed that deciding the value of a game with perfect information can be done in PSPACE if the following requirements are met:

- the length of any legal sequence of moves is bounded by a polynomial in the size of the input;

- given a 'board position' of the game there is a polynomial-space algorithm that constructs all possible next actions and board positions; or, if there is none, decides whether the board position is a win for either player.

It is easy to show that SY-PI(sy) meets those conditions. As to the first one, namely, the length of the description of any history is polynomially bounded in the number of rounds k of the game. By assumption $k \leq \|V\| \leq \|sy\|$, whence the description of every history is polynomial in the size of the input. As to (2), if $h\langle U, \vec{v}\rangle$ is a non-terminal history, then its successors are either (depending on f) only $h\langle U, \vec{v}\rangle\langle\{w_1, \ldots, w_m\}\rangle$ or all of $h\langle U, \vec{v}\rangle\langle\{w_1\}\rangle, \ldots, h\langle U, \vec{v}\rangle\langle\{w_m\}\rangle$, where $E(U - \{\vec{v}\}) = \{w_1, \ldots, w_m\}$. These can clearly be constructed in PSPACE.

In the worst case, for an arbitrary history $h\langle U, \vec{v}\rangle\langle U'\rangle$ in which $\exists$ is to move there are $\|V\|^n$ many vectors $\vec{v}'$ such that $E(\vec{v}, \vec{v}')$, where n is the number of $\exists$'s pawns on the game board. This number is clearly exponential in the size of the input. Nevertheless, every vector $\vec{v}'$ in $V^n = \{v_1, \ldots, v_{\|V\|}\}^n$ can be constructed in polynomial space, simply by writing down the vector $\langle v_1, \ldots, v_1\rangle \in V^n$ that comes first in the lexicographical ordering and successively constructing the remaining vectors that follow it up in the same ordering. □

Hardness is shown by reduction from QBF. To introduce the problem properly, let me introduce some standard terminology from propositional logic. A *literal* is a propositional variable or a negated propositional variable. A *clause* is a disjunction of literals. A boolean formula is in *conjunctive normal form* (CNF), if it is a conjunction of clauses. A boolean formula is said to be in *3-CNF*, if it is in CNF and all its clauses contain exactly three literals. The problem of QBF has *quantified boolean formulas* $\forall x_1 \exists y_1 \ldots \forall x_n \exists y_n \ \phi$ as instances, in which ϕ is a boolean formula in 3-CNF. QBF questions whether it is the case that for every truth value for x_1, there is a truth value for y_1, ..., such that the boolean formula $\phi(\vec{x}, \vec{y})$ is satisfied by the resulting truth assignment. Put formally, QBF is the set such that

$$\psi \in \text{QBF} \quad \text{iff} \quad \{\mathit{true}, \mathit{false}\} \models \psi,$$

where ψ is a QBF instance.

Lemma 21 SCOTLAND YARD$_{\mathit{show}}$ *is* PSPACE-*hard.*

Proof. Given a QBF instance $\psi = \forall x_1 \exists y_1 \ldots \forall x_n \exists y_n\ \phi$, where $\phi = C_1 \wedge \ldots \wedge C_m$ is a boolean formula in 3-CNF, it suffices to construct a Scotland Yard instance sy_ψ, such that $\psi \in$ QBF if and only if $sy_\psi \in$ SCOTLAND YARD$_{show}$. To this end, let me construct the initial position of the game constituted by sy_ψ. The formal specification of sy_ψ follows from this construction directly.

Set $i = 0$. For $i \leq n + 1$, do as follows:

- If $i = 0$, lay down the *opening-gadget*, that is schematically depicted in Figure 7.a. Moreover, distribute the pawns from

$$\{\exists_{x_1}, \ldots, \exists_{x_n}, \exists_{y_1}, \ldots, \exists_{y_n} \exists_d, \forall\}$$

  over the vertices of the opening-gadget as indicated in its depiction.

- If $1 \leq i \leq n$, first put the x_i*-gadget* at the bottom of the already constructed game board. Next, put the y_i*-gadget* below the justly introduced x_i-gadget. Figures 7.b and 7.c give a schematic account of the x_i-gadget and y_i-gadget, respectively. (Note that as a result of these actions, every vertex in the board game is connected to at least one other vertex, except for the ones on the top row of the opening-gadget and the ones on the bottom row of the y_i-gadget.)

- If $i = n + 1$, put the *clause-gadget* (see Figure 7.d) at the bottom of the already constructed board game. This gadget requires a little tinkering before the construction terminates, in order to encode the boolean formula ϕ by adding edges to the clause-gadgets (not present in the depiction), as follows:

  · For every variable $z \in \{\vec{x}, \vec{y}\}$ and clause C in ϕ: If z occurs as a literal in C, then join the vertices named "$+z$" and "C" by an edge. If $\neg z$ occurs as a literal in C, then join the vertices named "$-z$" and "C" by an edge.

- Set $i = i + 1$.

Note that the board game can be considered *layered*, indicate by the horizontal, dotted lines. These layers are numbered $-2, -1, \ldots, 4n+5$, enabling us to refer to these layers when we describe strategies. Note that the division in layers is not complete: in between the $4(i-1)+3$rd and $4(i-1)+4$th layer of the x_i-gadget we find two vertices, not laying on any layer.

As I pointed out, the above construction procedure yields a game board, rather than a Scotland Yard instance. But sy_ψ is easily derived from the board game, in the sense that the graph is completely spelt out and so are the initial positions of the pawns. Therefore, sy_ψ is fully specified after putting $f : \{1, \ldots, 4n+5\} \to \mathit{show}$. Hence, sy_ψ is an instance of $\textsc{Scotland Yard}_{\mathit{show}}$.

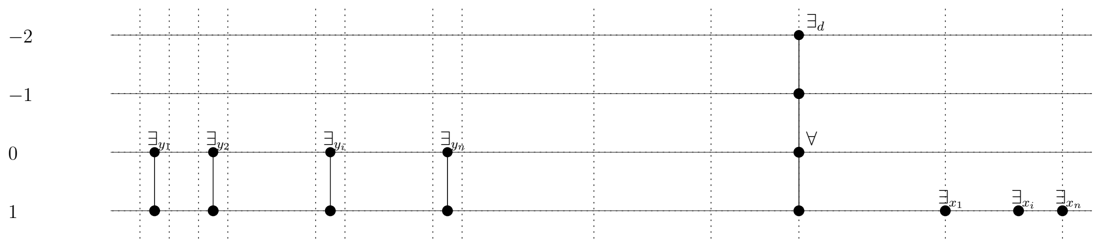
It remains to be shown that $\psi \in$ QBF iff $sy_\psi \in \textsc{Scotland Yard}_{\mathit{show}}$.

From left to right. Suppose that $\{\mathit{true}, \mathit{false}\} \models \psi$, then there is a *way of picking* truth values for the existentially quantified variables that renders ψ's boolean part ϕ true, no matter what truth values were assigned to the universally quantified variables. $\exists$'s winning strategy in $SY(sy_\psi)$ (being witness of the fact that $sy_\psi \in \textsc{Scotland Yard}_{\mathit{show}}$) can be read off from the aforementioned way of picking. We do so by interpreting moves in $SY(sy_\psi)$ as assigning truth values to variables and vice versa: Actions performed by $\forall$ from layer $4(i-1)+1$ to layer $4(i-1)+2$ will be interpreted as assigning a truth value to the universally quantified variable x_i. In particular, a move by $\forall$ to the vertex named "$+x_i$" ("$-x_i$") will be interpreted as assigning to x_i the value *true* (*false*). Conversely, if $\exists$'s way of picking prescribes assigning *true* (*false*) to y_i this will be reflected in $SY(sy_\psi)$ by moving $\exists_{y_i}$ to the vertex named "$+y_i$" ("$-y_i$") on layer $4(i-1)+5$.

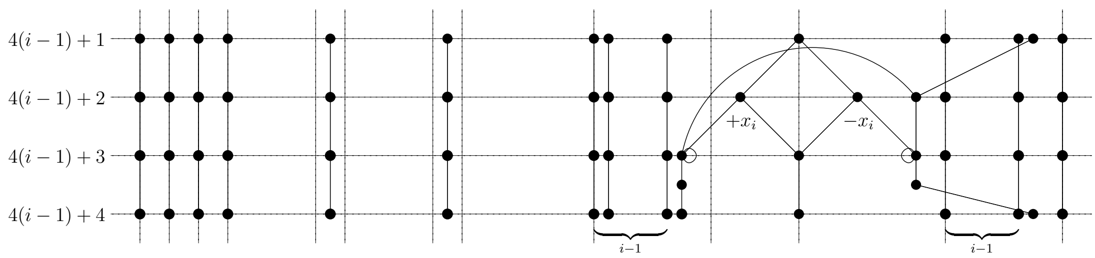
Roughly speaking, $\exists$ goes about as follows: when she is to chose between moving $\exists_{y_i}$ to the vertex named "$+y_i$" or "$-y_i$" she interprets $\forall$'s previous moves as a truth assignment and observes which truth value is prescribed by the way of picking. Next, she interprets this truth value as a move in $SY(sy_\psi)$ as above and moves $\exists_{y_i}$ to the according vertex. This intuition underlies the full specification of $\exists$'s strategy, described below:

For $0 \leq i \leq n+1$ let $\exists$'s strategy be as follows:

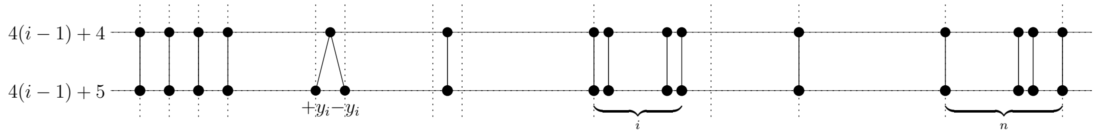
- Above all: If any pawn can capture $\forall$, do so!
- For every pawn that stands on a vertex on layer j that is connected to exactly one vertex on layer $j+1$, move it to this vertex. If the pawn at stake is actually $\exists_{x_i}$ standing on a vertex on layer $4(i-1)+3$, it cannot move to the vertex on layer $4(i-1)+4$, because there is a vertex v in between. In this case, move $\exists_{x_i}$ to v and in the subsequent round of the game move it downwards to layer $4(i-1)+4$.



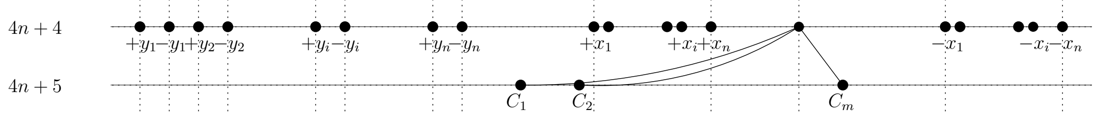
(a) Opening-gadget



(b) x_i-gadget



(c) y_i-gadget



(d) Clause-gadget

Figure 7: The gadgets that make up the initial position of the board game constituted by $SY(sy_\psi)$. The dotted lines are merely 'decoration' of the game board, to enhance readability. The horizontal, dotted lines are referred to as 'layers'.

- If $\exists_{x_i}$ stands on a vertex on layer $4(i-1)+2$, then move it to the vertex on layer $4(i-1)+3$ that is connected to the vertex where $\forall$ is positioned.

- If $\exists_{y_i}$ stands on a vertex on layer $4(i-1)+4$, and the way of picking prescribes assigning *true* (*false*) to y_i, then move it to the vertex on layer $4(i-1)+5$ that is named "$+y_i$" ("$-y_i$").

- If $\exists_d$ stands on a vertex on layer j that has two successors on layer $j+1$, then move it along the left-hand (right-hand) edge, if j is even (odd).

- If $\exists_z$ (for $z \in \{\vec{x}, \vec{y}\}$) stands on a vertex on layer $4n+4$ and this vertex is not connected to a vertex on which $\forall$ is positioned, move it along an arbitrary edge (possibly upwards).

As to $\forall$'s behavior I claim without rigorous proof that after $4n+4$ rounds of the game (that is, without being captured at an earlier stage of the game) he has traversed a path leading through exactly one of the vertices named "$+x_i$" and "$-x_i$", for every $x_i \in \{\vec{x}\}$, ending up in a vertex named "C", for some clause C in ϕ. To see that this must be the case: moving $\forall$ upwards at any stage of the game results in an immediate capturing by $\exists_d$. (In fact, $\exists_d$'s sole purpose in life is capturing $\forall$, when he moves upward.) If $\forall$ is moved to one of the reflexive vertices on layer $4(i-1)+3$ he is captured by $\exists_{x_i}$ who moves along the reflexive edge.

Suppose the $4n+4$th round of the game is over and $\exists$ played the entire game according to the above strategy, then she the above strategy prescribes a move that makes her win. Concludingly, $\exists$'s strategy is in fact a *winning strategy.*

Upon arriving at layer $4n+4$, pawn $\exists_z$ (for $z \in \{\vec{x}, \vec{y}\}$) stands on a vertex named "$+z$" or "$-z$", reflecting that z was assigned *true* or *false*, respectively. By assumption on the successfulness of the way of picking, that guided $\exists$ through $SY(sy_\psi)$, we have that the truth assignment that is associated with the positions of the pawns $\exists_{x_1}, \ldots, \exists_{x_n}, \exists_{y_1}, \ldots, \exists_{y_n}$ makes ϕ true. That is, under that truth assignment, for every clause C in ϕ there is a literal L that is made true. Now, if $L = z$, then $\exists_z$ stands on the vertex named "$+z$" and this vertex and the vertex named "C" are joined by an edge; and if $L = \neg z$, then $\exists_z$ stands on the vertex named "$-z$" and this vertex and the vertex named "C" are joined by an edge. So no matter to which vertex named

"C" pawn $\forall$ moves during his $4n+5$th move, for at least one $z \in \{\vec{x}, \vec{y}\}$ it is the case that $\exists_z$ can move to this vertex named "C" and capture him there.

From right to left. Suppose that $\{true, false\} \not\models \psi$, then there is a way of picking truth values for the universally quantified variables that renders the boolean part false, no matter what truth values were subsequently assigned to the existentially quantified variables. I leave out the argumentation that this way of picking constitutes a winning strategy for $\forall$ in $SY(sy_\psi)$, as it is similar to the argumentation in the converse direction. But let us note one crucial property of $\forall$'s winning strategy: it moves pawn $\forall$ downwards, during every round in the game. Therefore, the only round in which it can be captured is the last one: on a vertex on layer $4n+5$.

We need to pay close attention, though, to $\exists$'s behavior. that is, to see that $\exists$ cannot change her sad destiny (losing) by deviating from the behavior specified in the rules below. The gist of this behavior is that it results in pawn $\exists_{x_i}$ 'remembering' $\forall$'s moves on layer $4(i-1)+1$ and that after $4n+4$ rounds the pawns $\exists_{x_1}, \ldots, \exists_{x_n}, \exists_{y_1}, \ldots, \exists_{y_n}$ all stand on a vertex on layer $4n+4$. The gist is that just as above the positions of these pawns on vertices on layer $4n+4$ reflect a truth assignment, that falsify the boolean part.

The rules are as follows:

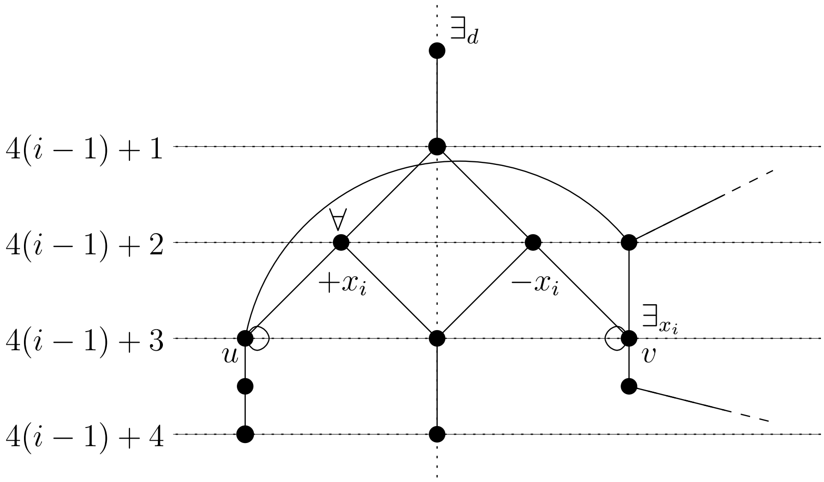
(1) If $\exists_{x_i}$ stands on a vertex on layer $4(i-1)+2$, then move it to the vertex on layer $4(i-1)+3$ that is connected to the vertex where $\forall$ is positioned.

(2) If $\exists_d$ stands on a vertex on layer j that is connected to two vertices below, then move it along the left-hand (right-hand) edge, if j is even (odd).

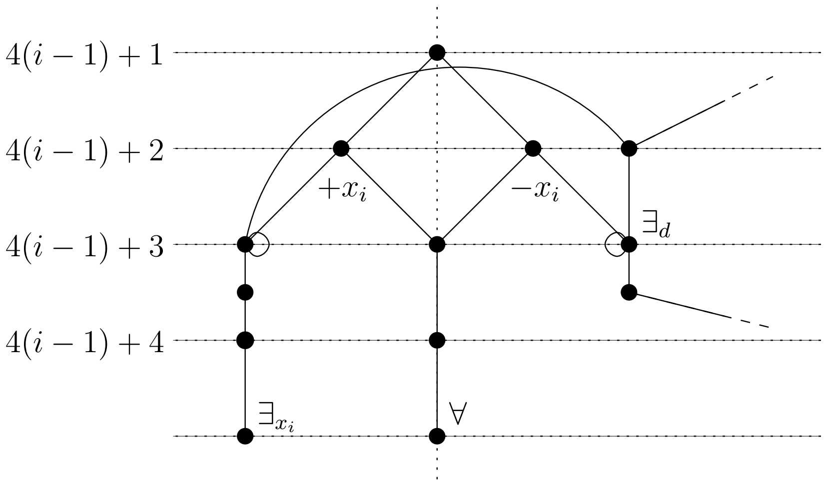
(3) For every pawn that stands on a vertex on layer j that is connected to exactly one vertex on layer $j+1$, move it to this vertex. (With the same exception as before with regard to $\exists_{x_i}$ standing on a vertex on layer $4(i-1)+3$.)

Now, I argue that not behaving in correspondence with (1)-(3) will also result in a loss for $\exists$:

(1) Suppose $\exists_{x_i}$ stands on the vertex on layer $4(i-1)+2$, having two options: u and v. Let u be the vertex on layer $4(i-1)+3$ that is



(a)



(b)

Figure 8: Positions on the game board that may occur if $\exists$ does not play according to rule (1) and (2), depicted in (a) and (b), respectively.

connected to the vertex where $\forall$ is positioned (see Figure 8.a). For the sake of the argument let us suppose that $\exists_{x_i}$ is moved to v. In that case, $\forall$ may safely move to u. If $\forall$ continues the game by moving its pawn downwards it wins automatically, since after the final round (round $4n + 5$) its pawn stands on a vertex on layer $4n + 4$, due to the extra vertex sitting in between layers $4(i - 1) + 3$ and $4(i - 1) + 4$, without there being any opportunity for $\exists$ to capture him. As such, the pawn $\exists_{x_i}$ is forced to *remember* what vertex $\forall$ visited on layer $4(i - 1) + 2$: the one named "$+x_i$" or "$-x_i$"?

(2) Suppose $\exists_d$ stands on a vertex on layer $4(i - 1) + 1$ and from there moves along the right-hand edge twice (see Figure 8.b). $\forall$ can exploit this move by moving as he would do normally, except for round $4n+5$, during which he moves upwards. This behavior results in a guaranteed win for $\forall$, since none of $\exists$'s pawns is pursuing $\forall$ closely enough to capture it, after moving upwards.

(3) Suppose any pawn controlled by $\exists$ moves upwards instead of downwards. We see that this can never result in a win for $\exists$, because $\forall$ (behaving as he does) can only be captured in the last round of the game, on a vertex on layer $4n + 5$. In particular, any pawn $\exists_z$, for $z \in \{\vec{x}, \vec{y}\}$, the shortest path to a vertex on layer $4n + 5$ is of length $4n + 5$. Now, if $\exists_z$ is moved upwards, it cannot (during the last round of the game) capture $\forall$.

This concludes the proof. □

The previous two lemmata are sufficient arguments to settle completeness.

Theorem 22 SCOTLAND YARD *and* SCOTLAND YARD$_{show}$ *are* PSPACE-*complete.*

Proof. Lemma 20 proved that SCOTLAND YARD is solvable in PSPACE. To check whether an instance *sy* has a function f with range $\{show\}$ is trivial, therefore, also SCOTLAND YARD$_{show}$ is solvable in PSPACE.

PSPACE-hardness was proven for SCOTLAND YARD$_{show}$ in Lemma 21. Since the latter problem is a specialization of SCOTLAND YARD, it follows immediately that SCOTLAND YARD is PSPACE-hard as well.

Hence, both problems are complete for PSPACE. □

5 Ignorance is (computational) bless

Intuitively, adding imperfect information makes a game harder. However, if one restricts oneself to Scotland Yard instances in which $\forall$'s whereabouts are only known at the beginning of the game, then *deciding* whether $\exists$ has a winning strategy is NP-complete, cf. Theorem 25. After the proof of this theorem, I show that from a quantitative point of view it is indeed harder for $\exists$ to win an arbitrary Scotland Yard game.

Lemma 23 SCOTLAND YARD$_{hide}$ $\in$ NP.

Proof. I make use of the equivalence between the Scotland Yard game and its perfect information counterpart Scotland Yard-PI. It suffices to give an NP algorithm that decides whether $\exists$ has a winning strategy in an arbitrary SY-PI(sy), where sy's information function has range $\{hide\}$. That is, for every integer i on which f is properly defined, we have that $f(i) = hide$. Let me now repeat the game rule from page 15 that regulates $\forall$'s behavior in the game of Scotland Yard-PI:

2-PI. Let $U' = E(U - \{\vec{v}\})$. If $f(i) = hide$, then set $U = U'$ and $\forall$ positions a $\forall$ pawn on every vertex v iff $v \in U$. If $f(i) = show$, then $\forall$ picks a vertex $u' \in U'$, removes all his pawns from the board, and puts one pawn on u'. Set $U = \{u'\}$.

Since $f(i)$ never equals *show*, by assumption, we can harmlessly replace it by the following rule:

2-PI$'$. Set $U' = E(U - \{\vec{v}\})$ and $\forall$ positions a $\forall$ pawn on every vertex v iff $v \in U$.

But doing so yields a game in which $\forall$ plays no active rôle anymore, in the sense that the set U at any round of the game is completely determined by $\exists$'s past moves. Put differently, any game constituted by an instance of SCOTLAND YARD$_{hide}$ essentially is a one-player game! Having obtained this insight, it is easy to see that the following algorithm decides in non-deterministic polynomial time whether $\exists$ has a winning strategy in the k-round SY-PI(sy):

- Non-deterministically guess a k number of n-dimensional vectors of vertices $\vec{v}_1, \ldots, \vec{v}_k \in V^n$.

- Set $U = \{u_*\}$, $\vec{v} = \vec{v}_*$, and $i = 1$; then for $i \leq k$ proceed as follows:
  - If $E(\vec{v}, \vec{v}_i)$, then set $\vec{v} = \vec{v}_i$; else, reject.
  - If $(U - \{\vec{v}\}) = \varnothing$, then accept; else, set $U = (U - \{\vec{v}\})$.
  - Set $i = i + 1$.
- If after k rounds there are still $\forall$ pawns present on the game board, reject.

This algorithm is obviously correct: $\exists$ has a winning strategy in SY-PI(sy) iff it accepts sy. Hence, Scotland Yard$_{hide}$ is in NP. $\square$

To prove hardness, I reduce from 3-Sat, that takes boolean formulae ϕ as instance that are in 3-CNF. ϕ is in 3-Sat iff it is satisfiable, that is, there exists a truth assignment of its variables that makes ϕ true. Henceforth, I make the assumption that no clause in a 3-Sat instance contains two copies of one propositional variable. This goes obviously without loss of generality.

Lemma 24 Scotland Yard$_{hide}$ *is* NP-*hard.*

Proof. To reduce from 3-Sat, let $\phi = C_1 \wedge \ldots \wedge C_m$ be an instance of 3-Sat over the variables $x_1, \ldots, x_n$. On the basis of ϕ we will construe a Scotland Yard instance sy_ϕ such that ϕ is satisfiable iff $\exists$ has a winning strategy in SY-PI(sy_ϕ). In fact, sy_ϕ will be read off from the initial game board that is put together as follows.

Set $i = 0$; for $i \leq n$ proceed as follows:

- If $i = 0$, lay down the *clause-gadget* from Figure 9.a. The sub-graphs H_i are fully connected graphs with four elements, whose vertices are connected with the vertices w_i.
- If $1 \leq i \leq n$, put the x_i-*gadget* to the right of the already constructed game board, see Figure 9.b. It will be convenient to refer to the vertex q^i by means of $-^i_{m+1}$ and $+^i_{m+1}$.

  For every $0 \leq j \leq m$, do as follows:
  - if x_i occurs as a literal in C_j, add the edges $\langle +^i_j, w_j \rangle$ and $\langle w_j, +^i_{j+1} \rangle$
  - if $\neg x_i$ occurs as a literal in C_j, add the edges $\langle -^i_j, w_j \rangle$ and $\langle w_j, -^i_{j+1} \rangle$
  - add the edges $\langle v_j, -^i_{j+1} \rangle$ and $\langle v_j, +^i_{j+1} \rangle$.

Note that C_0 refers to no clause, and that $-^i_{m+1} = +^i_{m+1} = q^i$.

- Set $i = i + 1$.

The Scotland Yard instance sy_ϕ is derived from the board game: the digraph is completely spelt out and the initial positions are as indicated in the gadgets. Therefore, sy_ϕ is fully specified after putting $f : \{1, \ldots, 2m+2\} \to \{\mathit{hide}\}$. Hence, sy_ϕ is an instance of Scotland Yard$_{\mathit{hide}}$.

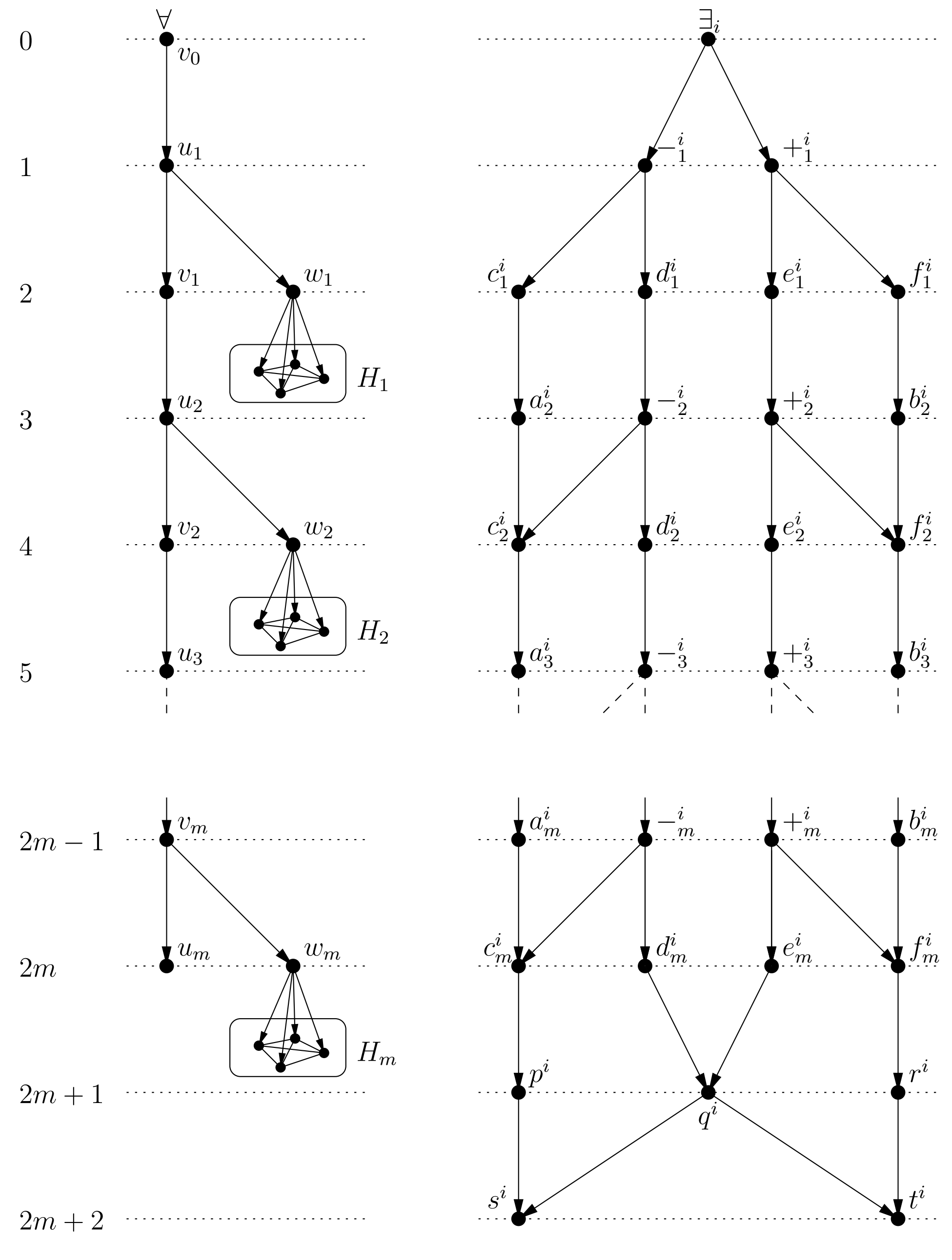
It remains to be shown that $\phi \in$ 3-Sat iff $sy_\phi \in$ Scotland Yard$_{\mathit{hide}}$.

By Theorem 19, it is sufficient to show that $\phi \in$ 3-Sat iff $\exists$ has a winning strategy in SY-PI(sy_ϕ).

From left to right. Suppose ϕ is satisfiable, then there exists a truth assignment $t : \{\vec{x}\} \to \{\mathit{true}, \mathit{false}\}$ such that for every clause C_j in ϕ, there exists at least one literal that is true under t. Let us describe a strategy for $\exists$ that is based on t and argue that it is in fact a winning strategy for her in SY-PI(sy_ϕ):

- If $\exists_i$ stands on the vertex on layer 0 and $t(x_i) = \mathit{true}$ (false), then move it to $+^i_1$ ($-^i_1$) on layer 1.
- If $\exists_i$ stands on $-^i_j$ ($+^i_j$) and $-^i_j$ ($+^i_j$) happens to be connected to w_j, then move it to w_j. If $\exists_i$ stands on $-^i_j$ ($+^i_j$) and $-^i_j$ ($+^i_j$) is not connected with w_j, then move it to d^i_j.
- If $\exists_i$ stands on w_j, move it to $\pm^i_{j+1}$, where $\pm \in \{+, -\}$. Note that this move is deterministic, since there is an edge from w_j to $+^i_{j+1}$ only if x_i occurs as a literal in C_j. By assumption of ϕ being an instance of 3-Sat, it cannot be the case that also $\neg x_i$ occurs as a literal in C_j. Hence, there is no edge from w_j to $-^i_{j+1}$.
- If $\exists_i$ stands on d^i_j (e^i_j) then move it to $-^i_{j+1}$ ($+^i_{j+1}$). If $\exists_i$ stands on d^i_m or e^i_m then move it to q^i.
- If $\exists_i$ stands on q^i, then move it to s^i if $t(x_i) = \mathit{true}$ and to t^i if $t(x_i) = \mathit{false}$.

Observe that if $\exists$ plays according to the above strategy, every pawn $\exists_i$ will eventually traverse either all vertices $-^i_1, \ldots, -^i_m$ or all vertices $+^i_1, \ldots, +^i_m$, given that $t(x_i) = \mathit{false}$ or $t(x_i) = \mathit{true}$, respectively.



(a) Clause-gadget

(b) x_i-gadget

Figure 9: The gadgets that make up the initial position of SY-PI(sy_ϕ). The sub-graph H_j is a fully connected graph with 4 elements, all of whose vertices are connected with the vertex w_j.

To show that this strategy is indeed winning against any of $\forall$'s strategies, consider the sets of vertices U_j^i that $\forall$ occupies on the clause-gadget and the x_i-gadget, after round $1 \leq j \leq 2m+2$ of the game in which $\exists$ moved as described above. Initially, $\forall$ has one pawn on v_0; thus, $U_0^i = \{v_0\}$. Let us suppose without loss of generality that $t(x_i) = \mathit{true}$. Then, $U_1^i = \{u_1, -_1^i\}$ as the $\forall$ pawn put on $+_1^i$ is captured by $\exists_i$ moving thereto in accordance with the strategy $\exists$ plays. I leave it to the reader to check that for $1 \leq j \leq m-1$, it is the case that

$$\begin{aligned} U_{2j}^i &= \{v_j, c_j^i, d_j^i\} \\ U_{2j+1}^i &= \{u_{j+1}, a_{j+1}^i, -_{j+1}^i\}. \end{aligned}$$

The crucial insight being that the $\forall$ pawn put on w_j can be captured iff there exists at least one literal in C_j that is made true by t. Since t was assumed to be a satisfying assignment, there must be at least one $\exists$ pawn that captures the universal pawn on w_j. It is prescribed by the above strategy that $\exists_i$ is moved to any w_j-vertex, if possible. Furthermore, it is required to return to the x_i-gadget in the next round of the game, capturing the $\forall$ pawn that was positioned on $+_j^i$, from v_j.

After round $2m-1$, $\forall$ cannot continue safely walking on $v_0, u_1, \ldots, v_m$; indeed, he only has pawns on the x_i-gadgets: $U_{2m}^i = \{c_m^i, d_m^i\}$. The pawn put on q is captured by $\exists$, so we get that $U_{2m+1}^i = \{p^i\}$. Following the strategy above, $\exists$ moves $\exists_i$ from q to s, whence $U_{2m+2}^i = \varnothing$. Since i was chosen arbitrarily, we have that $\forall$ has not pawns left on any x_i-gadget and therefore has lost after exactly $2m+2$ rounds of playing.

From right to left. Suppose ϕ is not satisfiable, then for every truth assignment t to the variables in ϕ, there exists a clause C_j in ϕ, that is made false. In the converse direction of this proof, we saw that every $\exists_i$ traverses one of the paths $-_1^i, \ldots, -_m^i, q^i$ and $+_1^i, \ldots, +_m^i, q^i$, depending on $t(x_i)$. This behavior I call *in accordance with the truth value $t(x_i)$ assigned to x_i*; if this behavior is displayed with respect to every $1 \leq i \leq n$, then I say that it is *in accordance with the truth assignment t*.

For now, assume that $\exists$ plays in accordance with some truth assignment t. Since ϕ is not satisfiable, it is not satisfied by t either. Therefore, there is a clause C_j that is not satisfied by t. This is reflected during the playing of the game by the fact that after round $2j$ there is a $\forall$ pawn positioned on w_j that cannot be captured by any $\exists_i$. This state of affairs will result in a win for $\forall$, as he moves pawns to every vertex in H_j during round $2j+1$. By construction, H_j is a connected graph on which he can keep on putting pawns indefinitely.

Remains to be shown that $\exists$ cannot avoid losing by deviating from playing in accordance with some truth assignment. I make the following claims: (A) If after round $1 \leq 2j-1 \leq 2m+1$ there is an i such that no $\exists$ pawn is positioned on $-^i_j$ or $+^i_j$, then $\exists$ loses. (B) If after round $2 \leq 2j \leq 2m$ there is an $\exists$ pawn positioned on c^i_j or f^i_j, then $\exists$ loses. I prove by induction. While proving these claims, I take the easily derived fact for granted that during round $2j-1$ of the game $\forall$ has a pawn on u_j and that during round $2j$ of the game $\forall$ has a pawn on v_j.

Base step. (A) Suppose after round $2m+1$ no $\exists$ pawn is on q^i (recall that $-^i_{m+1} = +^i_{m+1} = q^i$). Then, there is a $\forall$ pawn on q^i, since by construction of the game board, u_m is connected to q^i. During the next round, $\forall$ has pawns on both s^i and t^i, none of which can be captured by $\exists$, as she has no pawns on the x_i-gadget.

(B) Suppose after round $2m$ there is an $\exists$ pawn positioned on c^i_m, say. Let us make case a distinction here regarding the state of affairs after round $2m+1$: (i) there is an $\exists$ pawn on q^i. Obviously, this pawn cannot be the one we had on c^i_m after round $2m$, since c^i_m and q^i are not connected. Therefore there are two of $\exists$'s pawns on the x_i-gadget. As there are exactly n pawns at $\exists$'s disposal, during round $2m+1$ there is a x_h-gadget avoid of $\exists$ pawns. In particular, there is no $\exists$ pawn on q^h. Applying clause (A) proved, yields that $\exists$ cannot win from this position. (ii) there is no $\exists$ pawn on q^i. Then, there is a $\forall$ pawn on q^i after round $2m+1$, coming from u_m. Therefore, after round $2m+2$ there is a $\forall$ pawn on t^i; coming from c^i_m, $\exists$ can only capture $\forall$'s pawn at s^i.

Induction step. (A) Suppose after round $1 \leq 2j-1 \leq 2m-1$ there is an i such that no $\exists$ pawn is positioned on $-^i_j$ or $+^i_j$. Since $\forall$ has a pawn on v_{j-1} after round $2j-2$, he has pawns on both $-^i_j$ and $+^i_j$ after round $2j-1$. If after the next round $\forall$ occupies the vertices c^i_j, d^i_j, e^i_j or d^i_j, e^i_j, f^i_j this implies that one of $\exists$'s pawns is on c^i_j or f^i_j, respectively. But then she loses in virtue of the inductive hypothesis of (B). So, suppose that after round $2j$ $\forall$ occupies all the vertices $c^i_j, d^i_j, e^i_j, f^i_j$. Then, for the x_i-gadget to be cleansed of $\forall$ pawns it is required that during some later round of the game there are at least two $\exists$ pawns on this gadget. But then during this round the inductive hypothesis of (A) kicks in, yielding that $\exists$ loses.

(B) Suppose after round $2 \leq 2j \leq 2m-2$ there is an $\exists$ pawn positioned on c^i_j, say. Then, after round $2j+2$ the same pawn is positioned on c^i_{j+1}. Applying the inductive hypothesis of (B) teaches that $\exists$ loses.

I leave it to the reader to check that if $\exists$ plays in such a way that the premises of (A) and (B) do not apply, during any of the rounds of the game, then she plays in accordance with some truth assignment. However, playing according to any truth assignment is bound to be a losing way of playing, as I argued earlier. This concludes the proof. $\square$

Tying together the latter two theorems yields NP-completeness for the specialization of Scotland Yard in which $\forall$ does not give any information.

Theorem 25 SCOTLAND YARD$_{hide}$ *is* NP-*complete.*

Proof. Immediate from Lemmata 23 and 24. $\square$

We saw that from a computational point of view, it is easier to solve the decision problem SCOTLAND YARD, when $\forall$ does not reveal himself during the game. Yet, in a quantitative sense it becomes harder for $\exists$ to play this game, in that there are games in which $\exists$ has no winning strategy if $\forall$ does not reveal himself at all, but she would have had a winning strategy if $\forall$ was to reveal himself once. To make this claim precise fix two functions g and h, where

$$g : \{1, \ldots, k\} \to \{\mathit{hide}\} \quad \text{and} \quad h : \{1, \ldots, k\} \to \{\mathit{hide}, \mathit{show}\}$$

such that $h(j) = \mathit{show}$, for some j. Then, for every digraph G and initial positions $\langle u_*, \vec{v}_* \rangle$ on this digraph it is the case that if $\exists$ has a winning strategy in $SY(G, \langle u_*, \vec{v}_* \rangle, g)$ then she has one in $SY(G, \langle u_*, \vec{v}_* \rangle, h)$. The converse does not hold, however. Consider the graph $F = \langle V_j, E_j \rangle$, such that

$$\begin{array}{rcl} V_j & = & \{-2, -1, \ldots, j-3, j-2\} \cup \{i_t \mid j-1 \leq i \leq k-1 \text{ and } t \in \{0,1\}\} \\ E_j & = & \text{the symmetric closure of } E'_j, \text{ where} \\ E'_j & = & \{\langle -2, -1 \rangle, \ldots, \langle j-3, j-2 \rangle, \langle j-2, (j-1)_0 \rangle, \langle j-2, (j-1)_1 \rangle\} \cup \\ & & \{\langle i_t, (i+1)_t \rangle \mid j-1 \leq i \leq k-2 \text{ and } t \in \{0,1\}\}. \end{array}$$

F can be depicted as a fork. Furthermore, let $\exists$ have one pawn that is initially positioned on node -2 and let $\forall$'s pawn be initially positioned on

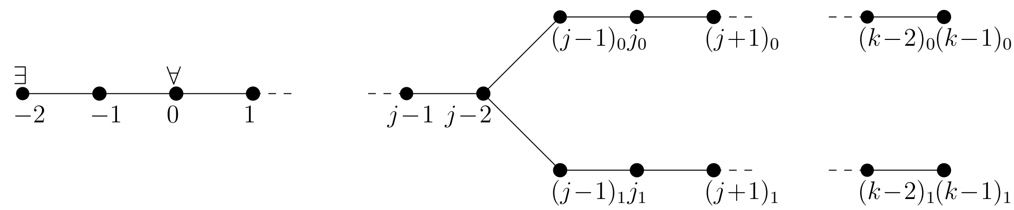


Figure 10: The forked graph F. $\exists$ has a winning strategy iff she knows $\forall$'s position during round j.

node 0. Thus, $u_* = -2$ and $v_* = 0$. For a graphical representation of the initial position of the game board constituted by F, see Figure 10.

I claim that $\exists$ has a winning strategy in $SY(G, \langle u_*, v_* \rangle, h)$. This fact is most easily established by observing the perfect information variant game SY-PI$(G, \langle u_*, v_* \rangle, h)$. Let $\exists$'s winning strategy consist of moving to the right, in case she has only one vertex to her right. If she is to move from vertex $j-2$ (in round j), she moves to $(j-1)_t$, depending on the vertex j_t on which $\forall$ is at. This t is unique, since by assumption $h(j) = \mathit{show}$.

It is easy to see that after the jth round of the game, $\forall$ has only one pawn on one of the fork's blades. In particular, at the beginning of round k, $\forall$ has one pawn on node $(k-1)_t$ and $\exists$ is at $(k-3)_t$. The only successor of $(k-1)_t$ is $(k-2)_t$, the vertex on which $\exists$ arrives during the last round of the game. Therefore, the above description constitutes a winning strategy for $\exists$ in SY-PI$(G, \langle u_*, v_* \rangle, h)$.

Obviously, $\exists$ does not have a winning strategy in SY-PI$(G, \langle u_*, v_* \rangle, g)$, as $\forall$ splits his troops among the two blades of the fork. In k round $\exists$ can only capture the $\forall$ pawns on one of the blades and loses.

6 Acknowledgements

I wish to thank the organizers of the seventh August de Morgan workshop for inviting me to present my research. I gratefully acknowledge Peter van Emde Boas for carefully proofreading an earlier version, and Ulle Endriss for his remarks that lead to Section 5 in this writing.

Finally, let me acknowledge Victor de Boer, Berend ter Borg, and Sicco Kuijper with whom I played many games of Scotland Yard on Sunday afternoons, already ten years ago.

A The boring bits of Section 2

Proposition 6 For every history $h' \in H_{\mathrm{PI}}$, the following hold:

1. If $h' = h\langle U\rangle$ and $f(\ell(h\langle U\rangle)) = \mathit{hide}$, then it is the case that $U = \{u \mid g\langle u\rangle \in H, \text{ for some } g \in \beta(h)\}$.

2. If $P(h') = \forall$ and $f(\ell(h')+1) = \mathit{show}$, then it is the case that $\{u \mid h' \succ h'\langle\{u\}\rangle, \text{ for some } h'\langle\{u\}\rangle \in H_{\mathrm{PI}}\} = \{u \mid g\langle u\rangle \in H, \text{ for some } g \in \beta(h')\}$.

3. If $h' = h\langle U\rangle \in H_{\mathrm{PI}}$ and $u \in U$, then there exists a history $g \in \beta(h)$ such that $g\langle u\rangle \in H$.

4. If $h' = h\langle U, \vec{v}\rangle \in H_{\mathrm{PI}}$, then it is the case that $\beta(h\langle U, \vec{v}\rangle) = \{g\langle u, \vec{v}\rangle \mid g\langle u\rangle \in \beta(h\langle U\rangle)\}$.

Proof. The proof hinges on one big inductive argument on the length of the histories. I warn the reader that the proof of the one item may use the inductive hypothesis of the other item.

The base case in which $\ell(h) = 0$ is trivial and therefore omitted.

1. Fix $h\langle U, \vec{v}\rangle\langle U'\rangle \in H_{\mathrm{PI}}$ such that $f(\ell(h\langle U, \vec{v}\rangle\langle U'\rangle)) = \mathit{hide}$.

   From left to right. Suppose $u' \in U'$, then it suffices to show that there exists a history $g\langle u, \vec{v}\rangle\langle u'\rangle \in H$, such that $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$. To this end, we first observe that $U - \{\vec{v}\} \neq \varnothing$ and that for some $u \in (U - \{\vec{v}\})$ it is the case that $E(u, u')$. Apply the inductive hypothesis of item 3 of this proposition to $h\langle U\rangle$, yielding that there exists a history $g\langle u\rangle \in \beta(h\langle U\rangle)$, since $u \in U$. By the inductive hypothesis of item 4 of this proposition we get that $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$. Since $u \in (U - \{\vec{v}\})$, it certainly does not sit in $\{\vec{v}\}$. Hence, $g\langle u, \vec{v}\rangle\langle u'\rangle$ is a history in H, as $E(u, u')$ and $U - \{\vec{v}\} \neq \varnothing$.

   From right to left. Suppose $g\langle u, \vec{v}\rangle\langle u'\rangle \in H$, where $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$, then it suffices to show that $u' \in U'$. By definition of β it is the case that $g\langle u\rangle \in \beta(h\langle U\rangle)$ and that $u \in U$. Since $g\langle u, \vec{v}\rangle\langle u'\rangle$ is a history, $g\langle u, \vec{v}\rangle$ cannot be a terminal history, whence $u \notin \{\vec{v}\}$ and $E(u, u')$. Consequently, $u' \in U'$, as required.

2. Fix $h\langle U, \vec{v}\rangle \in H_{\mathrm{PI}}$ such that $P(h) = \forall$ and $f(\ell(h\langle U, \vec{v}\rangle) + 1) = \mathit{show}$.

   From left to right. Suppose $h\langle U, \vec{v}\rangle \succ h\langle U, \vec{v}\rangle\langle\{u'\}\rangle$, then it suffices to show that there exists a history $g\langle u, \vec{v}\rangle\langle u'\rangle \in H$, such that $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$. To this end, firstly observe that for some $u \in (U - \{\vec{v}\})$ it must be the case that $E(u, u')$. We apply the inductive hypothesis of item 3 of this proposition to $h\langle U\rangle$, yielding that there exists a history $g\langle u\rangle \in \beta(h\langle U\rangle)$, since $u \in U$. By the inductive hypothesis of item 4 of this proposition we get that $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$. Since $u \in (U - \{\vec{v}\})$, it certainly does not sit in $\{\vec{v}\}$. Hence, $g\langle u, \vec{v}\rangle\langle u'\rangle$ is a history in H, as $E(u, u')$.

   From right to left. Suppose $g\langle u, \vec{v}\rangle\langle u'\rangle \in H$, where $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$, then it suffices to show that $h\langle U, \vec{v}\rangle \succ h\langle U, \vec{v}\rangle\langle\{u'\}\rangle$. By definition of β it is the case that $g\langle u\rangle \in \beta(h\langle U\rangle)$ and that $u \in U$. Since $g\langle u, \vec{v}\rangle\langle u'\rangle$ is a history, $g\langle u, \vec{v}\rangle$ cannot be a terminal history, whence $u \notin \{\vec{v}\}$ and $E(u, u')$. Hence, $h\langle U, \vec{v}\rangle\langle\{u'\}\rangle$ is a history in H_{PI} succeeding $h\langle U, \vec{v}\rangle$.

3. Follows immediately from items 1 and 2 of this proposition.

4. *From left to right.* Follows from the definition.

   From right to left. Fix $h\langle U, \vec{v}\rangle\langle U', \vec{v}'\rangle \in H_{\mathrm{PI}}$. It suffices to show that if $g\langle u, \vec{v}\rangle\langle u'\rangle \in \beta(h\langle U, \vec{v}\rangle\langle U'\rangle)$, then $g\langle u, \vec{v}\rangle\langle u', \vec{v}'\rangle$ is a history. By the inductive hypothesis of this proposition, $\beta(h\langle U, \vec{v}\rangle) = \{g\langle u, \vec{v}\rangle \mid g\langle u\rangle \in \beta(h\langle U\rangle)\}$. It is readily observed from the definition of β that for every history $g\langle u, \vec{w}\rangle\langle u'\rangle \in \beta(h\langle U, \vec{v}\rangle\langle U'\rangle)$ it is the case that $\vec{v} = \vec{w}$. By definition of H, it follows that every history $g\langle u, \vec{w}\rangle\langle u'\rangle \in \beta(h\langle U, \vec{v}\rangle\langle U'\rangle)$ has $g\langle u, \vec{w}\rangle\langle u', \vec{v}'\rangle$ as a successor history, since $E(\vec{v}, \vec{v}')$. Hence, the claim follows.

This concludes the proof. □

Proposition 7 is the converse of the previous Proposition, as it links up histories in H with histories in H_{PI}.

Proposition 7 For every $g' \in H$, the following hold:

1. If $g' = g\langle u\rangle \in H$, then there exists a $h\langle U\rangle \in H_{\mathrm{PI}}$ such that $g \in \beta(h)$ and $u \in U$.

2. If $g' = g\langle u, \vec{v}\rangle \in H$, then there exists a $h\langle U, \vec{v}'\rangle \in H_{\mathrm{PI}}$ such that $g\langle u\rangle \in \beta(h\langle U\rangle)$ and $\vec{v} = \vec{v}'$.

Proof. Again, the proof is one big inductive argument on the length of the histories.

The base case in which $\ell(h) = 0$ is trivial and therefore omitted.

1. Fix $g\langle u, \vec{v}\rangle\langle u'\rangle \in H$. Clearly, $g\langle u, \vec{v}\rangle$ is no terminal history and therefore $u \notin \{\vec{v}\}$ and $E(u, u')$. By the inductive hypothesis of item 2 of this proposition, it follows that there exists a $h\langle U, \vec{v}\rangle \in H_{\mathrm{PI}}$, such that $g\langle u\rangle \in \beta(h\langle U\rangle)$. By definition of β, we derive that $u \in U$. Consequently, $U - \{\vec{v}\}$ contains at least one object, namely u. This implies that $h\langle U, \vec{v}\rangle$ is not a terminal history. Since $E(u, u')$, there must exist a history $h\langle U, \vec{v}\rangle\langle U'\rangle$ such that $u' \in U'$.

2. Fix $g\langle u, \vec{v}\rangle\langle u', \vec{v}'\rangle \in H$. Clearly, $g\langle u, \vec{v}\rangle$ is no terminal history, whence $u \notin \{\vec{v}\}$ and furthermore $E(\vec{v}, \vec{v}')$. By the inductive hypothesis of item 1 of this proposition, it follows that $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$, for some $h\langle U, \vec{v}\rangle \in H_{\mathrm{PI}}$ such that $u \in U$. Since $u \notin \{\vec{v}\}$, $U - \{\vec{v}\}$ is not empty. Consequently, the history $h\langle U, \vec{v}\rangle\langle E(U - \{\vec{v}\})\rangle$ exists and by definition of β, $g\langle u, \vec{v}\rangle\langle u'\rangle \in \beta(h\langle U, \vec{v}\rangle\langle E(U - \{\vec{v}\})\rangle)$. Since $E(\vec{v}, \vec{v}')$, it follows that $h\langle U, \vec{v}\rangle\langle E(U - \{\vec{v}\}), \vec{v}'\rangle$ is a history in H_{PI} as well.

This concludes the proof. □

Lemma 8 β is a function of type $H_{\mathrm{PI}} \to \mathcal{H}$.

Proof. I prove by induction on the structure of histories $h' \in H_{\mathrm{PI}}$. I omit the base step.

Suppose $h' = h\langle U\rangle$. By definition $\beta(h\langle U\rangle) = \{g\langle u\rangle \in H \mid g \in \beta(h) \text{ and } u \in U\}$. It is easily derived from Proposition 6.3 that $\beta(h\langle U\rangle)$ is non-empty. By the inductive hypothesis, we have that $\beta(h) = \{g_1, \ldots, g_m\} \in \mathcal{H}$, whence $g_1 \sim \ldots \sim g_m$. I show that for every $g\langle u\rangle, g'\langle u'\rangle \in \beta(h\langle U\rangle)$, $g\langle u\rangle \sim g'\langle u'\rangle$. To this end I make a case distinction:

Suppose $f(\ell(h\langle U\rangle)) = \mathit{hide}$. This case follows directly from the definition of $\sim$, since $g \succ g\langle u\rangle$ and $g' \succ g'\langle u'\rangle$.

Suppose $f(\ell(h\langle U\rangle)) = \mathit{show}$. Since $\forall$ has to reveal his position, $U = \{v\}$ is a singleton. But then, if $g\langle u\rangle, g'\langle u'\rangle$ are both histories in $\beta(h\langle\{v\}\rangle)$, then

$u = u' = v$. Consequently, it follows from the definition of $\sim$ that $g\langle u\rangle \sim g'\langle u'\rangle$.

Remains to be shown that there exists no superset of $\beta(h\langle U\rangle)$ that is closed under $\sim$ as well. For the sake of contradiction, let $g^+\langle u^+\rangle$ be such that $g^+\langle u^+\rangle \sim g\langle u\rangle$, for every $g\langle u\rangle \in \beta(h\langle U\rangle)$, but $g^+\langle u^+\rangle \notin \beta(h\langle U\rangle)$. From the latter we derive that either (A) $g^+ \notin \beta(h)$ or (B) $u^+ \notin U$. For the sake of contradiction assume (A), that is, $g^+ \notin \beta(h)$. Therefore, $g^+ \not\sim g$, for any $g \in \beta(h)$. From Proposition 3.4 it follows immediately that $g^+\langle u^+\rangle \not\sim g_1\langle u_1\rangle$, since $g_1\langle u_1\rangle \in \beta(h\langle U\rangle)$. This contradicts the assumption and therefore $g^+ \in \beta(h)$. To derive that (B) cannot hold as well, observe that it follows from Proposition 3.2 that $u^+ = u_1$, since $g^+\langle u^+\rangle \sim g_1\langle u_1\rangle$ and $f(\ell(g\langle u\rangle)) = f(\ell(h\langle U\rangle)) = \mathit{show}$. Since $g_1\langle u_1\rangle \in \beta(h\langle U\rangle)$, $u^+ = u_1 \in U$. Hence, (B) is not true.

Therefore, $\beta(h\langle U\rangle$ is a greatest subset of H closed under $\sim$ and as such sits in $\mathcal{H}$.

Suppose $h' = h\langle U, \vec{v}\rangle$. From the inductive hypothesis it follows that $\beta(h\langle U\rangle) \in \mathcal{H}$. Put $\beta(h\langle U\rangle) = \{g_1\langle u_1\rangle, \ldots, g_m\langle u_m\rangle\}$. It is easily derived from Proposition 6.3 that $\beta(h\langle U\rangle)$ is non-empty ($m > 0$) and that any two histories from $\beta(h\langle U\rangle)$ are $\sim$-related. It follows from Proposition 6.4 that $\beta(h\langle U, \vec{v}\rangle) = \{g_1\langle u_1, \vec{v}\rangle, \ldots, g_m\langle u_m, \vec{v}\rangle\}$. Furthermore, it follows directly from the definition of $\sim$, that any two histories from $\beta(h\langle U, \vec{v}\rangle)$ are $\sim$-related.

Remains to be shown that there exists no superset of $\beta(h\langle U, \vec{v}\rangle)$ that is also closed under $\sim$. For the sake of contradiction, let us suppose there exists a history $g^+\langle u^+, \vec{v}^+\rangle$, such that $g^+\langle u^+, \vec{v}^+\rangle \sim g\langle u, \vec{v}\rangle$, for every $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$, but $g^+\langle u^+, \vec{v}^+\rangle \notin \beta(h\langle U, \vec{v}\rangle)$. From the latter and the definition of β we derive that either (A) $\vec{v}^+ \neq \vec{v}$ or (B) $g^+\langle u^+\rangle \notin \beta(h\langle U\rangle)$. But actually both (A) and (B) lead to a contradiction: For both (A) and (B) contradict the assumption that $g^+\langle u^+, \vec{v}^+\rangle \sim g\langle u, \vec{v}\rangle$, for every $g\langle u, \vec{v}\rangle \in \beta(h\langle U, \vec{v}\rangle)$, in virtue of Propositions 3.3 and 3.4, respectively.

Therefore, $\beta(h\langle U, \vec{v}\rangle)$ is a greatest subset of H closed under $\sim$ and as such sits in $\mathcal{H}$. □

Lemma 9 β is a bijection between H_{PI} and $\mathcal{H}$.

Proof. It suffices to show that β is surjective and injective.

Surjection. We need to prove that for every $C' \in \mathcal{H}$, there exists a history $h \in H_{\text{PI}}$, such that $C' = \beta(h)$. I do so by induction on the structure of the histories in $C' \in \mathcal{H}$.

Suppose $C' = \{g_1\langle u_1, \vec{v}_1\rangle, \ldots, g_m\langle u_m, \vec{v}_m\rangle\}$. Since $C' \in \mathcal{H}$, it is closed under $\sim$, that is, $g_1\langle u_1, \vec{v}_1\rangle \sim \ldots \sim g_m\langle u_m, \vec{v}_m\rangle$. I derive from Proposition 3.3 that $\vec{v}_1 = \ldots = \vec{v}_m = \vec{v}$ and also that $g_1\langle u_1\rangle \sim \ldots \sim g_m\langle u_m\rangle$. Therefore, there must be one cell $C \in \mathcal{H}$ that contains $g_1\langle u_1\rangle \sim \ldots \sim g_m\langle u_m\rangle$. By the inductive hypothesis, we know that there exists a history $h\langle U\rangle \in H_{\text{PI}}$, such that $\beta(h\langle U\rangle) = C$. By Proposition 7.2 we get that $h\langle U, \vec{v}\rangle \in H_{\text{PI}}$ and by Proposition 6.4 we have that $\beta(h\langle U, \vec{v}\rangle) = \{g\langle u, \vec{v}\rangle \mid g\langle u\rangle \in \beta(h\langle U\rangle) \text{ and } u \in U\} = C'$.

Suppose $C' = \{g_1\langle u_1\rangle, \ldots, g_m\langle u_m\rangle\}$ *and* $f(\ell(g_1\langle u_1\rangle)) = \textit{show}$. Since $C' \in \mathcal{H}$, it is closed under $\sim$, that is, $g_1\langle u_1\rangle \sim \ldots \sim g_m\langle u_m\rangle$. I derive from Proposition 3.2 that $u_1 = \ldots = u_m = u$ and also that $g_1 \sim \ldots \sim g_m$. Therefore, there must be one cell $C \in \mathcal{H}$ that contains $g_1, \ldots, g_m$. By the inductive hypothesis, we know that there exists a history $h \in H_{\text{PI}}$, such that $\beta(h) = C$.

By Proposition 7.1 we derive that there is a set U containing u such that $h\langle U\rangle$ is a successor of h, since $g_1\langle u\rangle$ is a successor of g_1 and $g_1 \in \beta(h)$. Since $f(\ell(g_1\langle u_1\rangle)) = \textit{show}$, U must in fact be a singleton, whence $U = \{u\}$. By definition of β we have that $\beta(h\langle\{u\}\rangle) = \{g_1\langle u\rangle, \ldots, g_m\langle u\rangle\}$, which is simply C'.

Suppose $C' = \{g_1\langle u_1\rangle, \ldots, g_m\langle u_m\rangle\}$ *and* $f(\ell(g_1\langle u_1\rangle)) = \textit{hide}$. Since $C' \in \mathcal{H}$, it is closed under $\sim$, that is, $g_1\langle u_1\rangle \sim \ldots \sim g_m\langle u_m\rangle$. I derive from Proposition 3.1 that $g_1 \sim \ldots \sim g_m$. Therefore, there must be one cell $C \in \mathcal{H}$ that contains $g_1, \ldots, g_m$. By the inductive hypothesis, we know that there exists a history $h \in H_{\text{PI}}$, such that $\beta(h) = C$.

By Proposition 6.1 we derive that $h\langle U\rangle$ is a successor of h, where $U = \{u \mid g\langle u\rangle \in H, \text{ for some } g \in \beta(h)\}$. Since $g_1 \in C = \beta(h)$, it follows that $u_1 \in U$. By definition of β it is immediate that $g_1\langle u_1\rangle \in \beta(h\langle U\rangle)$. In consequence, all of $g_1\langle u_1\rangle, \ldots, g_m\langle u_m\rangle$ sit in $\beta(h\langle U\rangle)$, since they are all $\sim$-related. Hence, $C' = \beta(h\langle U\rangle)$.

Injection. We need to prove that for any pair of histories $h, h' \in H_{\text{PI}}$, if $h \neq h'$ then $\beta(h) \neq \beta(h')$. We do so by induction on the structure of the histories in H.

Suppose $h\langle U\rangle \neq h'\langle U'\rangle$. I distinguish two cases. (i) $h \neq h'$. h and h' give rise to $\beta(h)$ and $\beta(h')$ which are present in $\mathcal{H}$, by Lemma 8. By the inductive hypothesis $\beta(h) \neq \beta(h')$. Since $\beta(h), \beta(h') \in \mathcal{H}$, we derive that $\beta(h) \cap \beta(h') = \varnothing$. From Proposition 6.3 it follows that for every $u \in U$ there exists a $g \in \beta(h)$ such that $g\langle u\rangle \in \beta(h\langle U\rangle)$ and that for every $u' \in U'$ there exists a $g' \in \beta(h')$ such that $g'\langle u'\rangle \in \beta(h'\langle U'\rangle)$. Since the intersection of $\beta(h)$ and $\beta(h')$ is empty, it is the case that $g \neq g'$. Hence, $g\langle u\rangle \neq g'\langle u'\rangle$ and therefore $\beta(h\langle u\rangle) \neq \beta(h'\langle u'\rangle)$. (ii) $h = h'$ and $U \neq U'$. Obviously (without loss of generality), there exists a $u \in U$ that does not sit in U'. From Proposition 6.3 it follows that there exists a $g \in \beta(h)$ such that $g\langle u\rangle \in \beta(h\langle U\rangle)$. By definition of β and the fact that $u \notin U'$, $g\langle u\rangle$ is not an element of $\beta(h'\langle U'\rangle)$ and therefore $\beta(h\langle U\rangle) \neq \beta(h'\langle U'\rangle)$.

Suppose $h\langle U, \vec{v}\rangle \neq h'\langle U', \vec{v}'\rangle$. I distinguish two cases. (i) $h\langle U\rangle \neq h'\langle U\rangle$. By the inductive hypothesis, it follows that $\beta(h\langle U\rangle) \neq \beta(h'\langle U'\rangle)$. Proposition 6.4 has it that $\beta(h\langle U, \vec{v}\rangle) = \{g\langle u, \vec{v}\rangle \mid g\langle u\rangle \in \beta(h\langle U\rangle)\}$ and $\beta(h\langle U', \vec{v}'\rangle) = \{g'\langle u', \vec{v}'\rangle \mid g'\langle u'\rangle \in \beta(h'\langle U'\rangle)\}$. Hence, $\beta(h\langle U, \vec{v}\rangle) \neq \beta(h'\langle U', \vec{v}'\rangle)$. (ii) $h\langle U\rangle = h'\langle U\rangle$ and $\vec{v} \neq \vec{v}'$. This case follows trivially from Proposition 6.4. □

References

[1] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning ways*. Academic Press, London, 1982.

[2] T. C. Biedl, E. D. Demaine, M. L. Demaine, R. Fleischer, L. Jacobsen, and J. I. Munro. The complexity of clickomania. In R. J. Nowakowski, editor, *More games of no chance*, volume 42 of *MSRI Publications*, pages 389–404. Cambridge University Press, 2002.

[3] E. Binz-Blanke. Game review – Scotland Yard. *http://www.io.com/∼beckerdo/games/reviews/ScotlandYard-Review.html*, 2006.

[4] X. Caicedo and M. Krynicki. Quantifiers for reasoning with imperfect information and Σ^1_1-logic. In Walter A. Carnielli and Itala M.L. Ottaviano, editors, *Contemporary Mathematics*, volume 235, pages 17–31. American Mathematical Society, 1999.

[5] P. J. Cameron and W. Hodges. Some combinatorics of imperfect information. *Journal of Symbolic Logic*, 66(2):673–684, 2001.

[6] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28:114–133, 1981.

[7] A. S. Fraenkel. Combinatorial games: selected bibliography with a succinct gourmet introduction. In R. J. Nowakowski, editor, *More games of no chance*, volume 42 of *MSRI Publications*, pages 475–535. Cambridge University Press, 2002.

[8] A. S. Fraenkel, M. R. Garey, D. S. Johnson, T. Schäfer, and Y. Yesha. The complexity of checkers on an $N \times N$ board – preliminary report. In *Proceedings of the 19th Annual Symposium on the foundations of Computer Science*, pages 55–64. IEEE Computer Society, 1978.

[9] A. S. Fraenkel and D. Lichtenstein. Computing a perfect strategy for $n \times n$ chess requires time exponential in n. *Journal of Combinatorial Theory*, Series A 31:199–214, 1981.

[10] A. S. Goldstein and E. M. Reingold. The complexity of pursuit on a graph. *Theoretical computer science*, 143:93–112, 1995.

[11] W. Hodges. Compositional semantics for a language of imperfect information. *Logic Journal of the IGPL*, 5(4):539–563, 1997.

[12] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[13] R. Kaye. Minesweeper is NP-complete. *Mathematical Intelligencer*, 22(2):9–15, 2000.

[14] D. Koller and N. Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and economic behavior*, 4:528–552, 1992.

[15] D. Lichtenstein and M. Sipser. GO is polynomial-space hard. *Journal of the Association for Computing Machinery*, 27:393–401, 1980.

[16] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[17] C. H. Papadimitriou. *Computational complexity.* Addison-Wesley, Reading, Massachusetts, 1994.

[18] G. Peterson, S. Azhar, and J. H. Reif. Lower bounds for multiplayer noncooperative games of incomplete information. *Computers and Mathematics with Applications*, 41:957–992, 2001.

[19] J. H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Science*, 29:274–301, 1984.

[20] T. J. Schäfer. Complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16:185–225, 1978.

[21] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *Proceedings of the 5th ACM Symposium on the Theory of Computing (STOCS 73)*, pages 1–9, 1973.

[22] J. Väänänen. On the semantics of informational independence. *Logic Journal of the Interest Group in Pure and Applied Logics*, 10(3):337–350, 2002.