OBSERVATION AND DISTINCTION
REPRESENTING INFORMATION IN INFINITE GAMES

DIETMAR BERWANGER AND LAURENT DOYEN

LSV, CNRS & ENS Paris-Saclay, France

Abstract. We compare two approaches for modelling imperfect information in infinite games by using finite-state automata. The first, more standard approach views information as the result of an observation process driven by a sequential Mealy machine. In contrast, the second approach features indistinguishability relations described by synchronous two-tape automata.

The indistinguishability-relation model turns out to be strictly more expressive than the one based on observations. We present a characterisation of the indistinguishability relations that admit a representation as a finite-state observation function. We show that the characterisation is decidable, and give a procedure to construct a corresponding Mealy machine whenever one exists.

1. Introduction

Uncertainty is a main concern in strategic interaction. Decisions of agents are based on their knowledge about the system state, and that is often limited. The challenge grows in dynamical systems, where the state changes over time, and it becomes severe, when the dynamics unravels over infinitely many stages. In this context, one fundamental question is how to model knowledge and the way it changes as information is acquired along the stages of the system run.

Finite-state automata offer a solid framework for the analysis of systems with infinite runs. They allow to reason about infinite state spaces in terms of finite ones — of course, with a certain loss. The connection has proved to be extraordinarily successful in the study of infinite games on finite graphs, in the particular setting of *perfect information* assuming that players are informed about every move in the play history, which determines the actual state of the system. One key insight is that winning strategies, in this setting, can be synthesized effectively [5, 22]: for every game described by finite automata, one can describe the set of winning strategies by an automaton (over infinite trees) and, moreover, construct an automaton (a finite-state Moore machine) that implements a winning strategy.

In this paper, we discuss two approaches for modelling *imperfect information*, where, in contrast to the perfect-information setting, it is no longer assumed that the decision maker is informed about the moves that occurred previously in the play history.

The first, more standard approach corresponds to viewing information as a result of an observation *process* that may be imperfect in the sense that different moves can yield the same observation in a stage of the game. Here, we propose a second approach, which corresponds to representing information as a *state* of knowledge, by describing which histories are indistinguishable to the decision maker.

E-mail address: `dwb@lsv.fr, doyen@lsv.fr`.

Date: January 30, 2020; extended version of a contribution to the Proceeding of STACS 2020.

Concretely, we assume a setting of synchronous games with perfect recall in a partitional information model. Plays proceed in infinitely many stages, each of which results in one move from a finite range. Histories and plays are thus determined as finite or infinite sequences of moves, respectively.

To represent information partitions, we consider two models based on finite-state automata. In the observation-based model, which corresponds to the standard approach in computing science and non-cooperative game theory, the automaton is a sequential Mealy machine that inputs moves and outputs observations from a finite alphabet. The machine thus describes an observation function, which maps any history of moves to a sequence of observations that represents its information set. In the indistinguishability-based model, we use two-tape automata to describe which pairs of histories belong to the same information set.

As an immediate insight, we point out that, in the finite-state setting, the standard model based on observation functions is less expressive than the one based on indistinguishability relations. Intuitively, this is because observation functions can only yield a bounded amount of information in each round — limited by the size of the observation alphabet, whereas indistinguishability relations can describe situations where the amount of information received per round grows unboundedly as the play proceeds.

We investigate the question whether an information partition represented as (an indistinguishability relation given by) a two-tape automaton admits a representation as (an observation function given by) a Mealy machine. We show that this question is decidable, using results from the theory of word-automatic structures. We also present a procedure for constructing a Mealy machine that represents a given indistinguishability relation as an observation function, whenever this is possible.

2. Basic Notions

2.1. **Finite automata.** To represent components of infinite games as finite objects, finite-state automata offer a versatile framework (see [12], for a survey). Here, we use automata of two different types, which we introduce following the notation of [21, Chapter 2].

As a common underlying model, a *semi-automaton* is a tuple $\mathcal{A} = (Q, \Gamma, q_\varepsilon, \delta)$ consisting of a finite set Q of *states*, a finite *input alphabet* Γ, a designated *initial state* $q_\varepsilon \in Q$, and a *transition function* $\delta \colon Q \times \Gamma \to Q$. We define the size $|\mathcal{A}|$ of $\mathcal{A}$ to be the number of its transitions, that is $|Q| \cdot |\Gamma|$. To describe the internal behaviour of the semi-automaton we extend the transition function from letters to input words: the extended transition function $\delta \colon Q \times \Gamma^* \to Q$ is defined by setting, for every state $q \in Q$,

- $\delta(q, \varepsilon) := q$ for the empty word ε, and
- $\delta(q, \tau c) := \delta(\delta(q, \tau), c)$, for any word obtained by the concatenation of a word $\tau \in \Gamma^*$ and a letter $c \in \Gamma$.

On the one hand, we use automata as acceptors of finite words. A *deterministic finite automaton* (for short, DFA) is a tuple $\mathcal{A} = (Q, \Gamma, q_\varepsilon, \delta, F)$ expanding a semi-automaton by a designated subset $F \subseteq Q$ of *accepting states*. We say that a finite input word $\tau \in \Gamma^*$ is *accepted* by $\mathcal{A}$ from a state q if $\delta(q, \tau) \in F$. The set of words in Γ^* that are accepted by $\mathcal{A}$ from the initial state q_ε forms its *language*, denoted $L(\mathcal{A}) \subseteq \Gamma^*$.

Thus, a DFA recognises a set of words. By considering input alphabets over pairs of letters from a basis alphabet Γ, the model can be used to recognise synchronous relations over Γ, that is, relations between words of the same length. We refer to a DFA over an input alphabet $\Gamma \times \Gamma$ as a *two-tape* DFA. The relation recognised by such an automaton consists of all pairs of words $c_1 c_2 \dots c_\ell, c'_1 c'_2 \dots c'_\ell \in \Gamma^*$ such that $(c_1, c'_1)(c_2, c'_2) \dots (c_\ell, c'_\ell) \in L(\mathcal{A})$. With a slight abuse of notation, we also denote this relation by $L(\mathcal{A})$. We say that a synchronous relation is regular if it is recognised by a DFA.

On the other hand, we consider automata with output. A *Mealy* automaton is a tuple $(Q, \Gamma, \Sigma, q_\varepsilon, \delta, \lambda)$ where $(Q, \Gamma, q_\varepsilon, \delta)$ is a semi-automaton, Σ is a finite *output alphabet*, and $\lambda\colon Q \times \Gamma \to \Sigma$ is an output function. To describe the external behaviour of such an automaton, we define the extended output function $\lambda\colon \Gamma^* \times \Gamma \to \Sigma$ by setting $\lambda(\tau, c) := \lambda(\delta(q_\varepsilon, \tau), c)$ for every word $\tau \in \Gamma^*$ and every letter $c \in \Gamma$. Thus, the external behaviour of a Mealy automaton defines a function from the set $\Gamma^+ := \Gamma^* \setminus \{\varepsilon\}$ of nonempty words to Σ. We say that a function on Γ^+ is *regular*, if there exists a Mealy automaton that defines it.

To build new automata from given ones, we will use two types of product constructions. The *synchronised product* of two semi-automata $\mathcal{A}^1 = (Q^1, \Gamma, q^1_\varepsilon, \delta^1)$ and $\mathcal{A}^2 = (Q^2, \Gamma, q^2_\varepsilon, \delta^2)$, over the same alphabet Γ, is the semi-automaton $\mathcal{A}^1 \times \mathcal{A}^2 = (Q^\times, \Gamma, q^\times_\varepsilon, \delta^\times)$ with:

- $Q^\times = Q^1 \times Q^2$,
- $q^\times_\varepsilon = (q^1_\varepsilon, q^2_\varepsilon)$, and
- $\delta^\times((q^1, q^2), c) = (\delta^1(q^1, c), \delta^2(q^2, c))$ for all $q^1 \in Q^1$, $q^2 \in Q^2$, and $c \in \Gamma$.

In the second type of product construction, the two automata run in parallel on separate input tapes, one for each automaton. There is no synchronisation other than the number of processed input symbols, which is always the same in the two automata. The *parallel product* of two semi-automata $\mathcal{A}^1 = (Q^1, \Gamma^1, q^1_\varepsilon, \delta^1)$ and $\mathcal{A}^2 = (Q^2, \Gamma^2, q^2_\varepsilon, \delta^2)$ is the semi-automaton $\mathcal{A}^1 \parallel \mathcal{A}^2 = (Q^\parallel, \Gamma^1 \times \Gamma^2, q^\parallel_I, \delta^\parallel)$ where:

- $Q^\parallel = Q^1 \times Q^2$,
- $q^\parallel_\varepsilon = (q^1_\varepsilon, q^2_\varepsilon)$, and
- $\delta^\parallel((q^1, q^2), (c^1, c^2)) = (\delta^1(q^1, c^1), \delta^2(q^2, c^2))$ for all $q^i \in Q^i$ and $c^i \in \Gamma^i$ (with $i = 1, 2$).

2.2. **Repeated games with imperfect information.** In our general setup, we consider games played in an infinite sequence of stages. In each stage, every player chooses an action from a given set of alternatives, independently and simultaneously. As a consequence, this determines a move that is recorded in the play history. Then, the game proceeds to the next stage. The outcome of the play is thus an infinite sequence of moves.

Decisions of a player are based on the available information, which we model by a partition of the set of play histories into information sets: at the beginning of each stage game, the player is informed of the information set to which the actual play history belongs (in the partition associated to the player). Accordingly, a strategy for a player is a function from information sets to actions. Every strategy profile (that is, a collection of strategies, one for each player) determines a play.

Basic questions in this setup concern strategies of an individual player to enforce an outcome in a designated set of winning plays or to maximise the value of a given payoff function, regardless of the strategy of other players. More advanced issues target joint strategies of coalitions among players towards coordinating on a common objective, or equilibrium profiles. Scenarios where the available actions depend on the history, or where the play might end after finitely many stages, can be captured by adjusting the information partition together with the payoff or winning condition.

For our formal treatment of information structures, we use the model of abstract infinite games as introduced by Thomas in his seminal paper on strategy synthesis [25]; the relevant questions for more elaborate settings, such as infinite games on finite graphs or concurrent game structures can be reduced easily to this abstraction. The underlying model is consistent with the classical definition of extensive games with information partitions and perfect recall due to von Neumann and Morgenstern [27], in the formulation of Kuhn [14]. For a more detailed account on partitional information, we refer to Bacharach [1] and Geanakoplos [10].

Our formalisation captures the information structures of repeated games with imperfect monitoring as studied in non-cooperative game theory (see the survey of Gossner and Tomala [11]), and of infinite games with partial observation on finite-state systems as studied in computing science (see Reif [24], Lin and Wonham [17], van der Meyden and Wilke [26], Chatterjee et al. [6], Berwanger et al. [2]). For background on the modelling of knowledge, and the notion of synchronous perfect recall we refer to Chapter 8 in the book of Fagin et al. [8].

2.2.1. *Move and information structure.* As a basic object for describing a game, we fix a finite set Γ of *moves*. A *play* is an infinite sequence of moves $\pi = c_1 c_2 \ldots \in \Gamma^\omega$. A *history* (of length ℓ) is a finite prefix $\tau = c_1 c_2 \ldots c_\ell \in \Gamma^*$ of a play; the empty history ε has length zero. The *move structure* of the game is the set Γ^* of histories equipped with the successor relation, which consists of all pairs $(\tau, \tau c)$ for $\tau \in \Gamma^*$ and $c \in \Gamma$. For convenience, we denote the move structure of a game on Γ simply by Γ^* omitting the (implicitly defined) successor relation.

The information available to a player is modelled abstractly by a partition $\mathcal{U}$ of the set Γ^* of histories; the parts of $\mathcal{U}$ are called *information sets* (of the player). The intended meaning is that if the actual history belongs to an information set U, then the player considers every history in U possible. The particular case where all information sets in the partition are singletons characterises the setting of *perfect information.*

The *information structure* (of the player) is the quotient $\Gamma^*/_{\mathcal{U}}$ of the move structure by the information partition. That is, the first-order structure on the domain consisting of the information sets, with a binary relation connecting two information sets (U, U') whenever there exists a history $\tau \in U$ with a successor history $\tau c \in U'$. Throughout this article, we assume the perspective of just one player, so we simply refer to the information structure of the game.

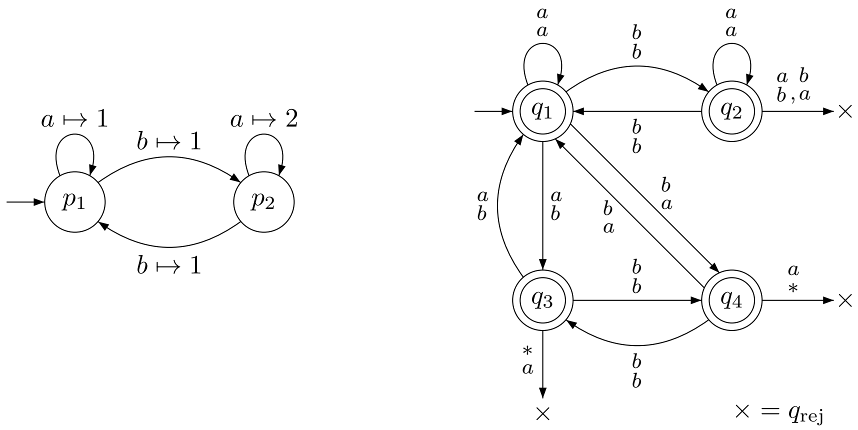
Our information model is *synchronous*, which means, intuitively, that the player always knows how many stages have been played. Formally, this amounts to asserting that all histories in an information set have the same length; in particular the empty history forms a singleton information set. Further, we assume that the player has *perfect recall* — he never forgets what he knew previously. Formally, if an information set contains nonempty histories τc and $\tau' c'$, then the predecessor history τ is in the same information set as τ'. In different terms, an information partition satisfies synchronous perfect recall if, whenever a pair of histories $c_1 \ldots c_\ell$ and $c'_1 \ldots c'_\ell$ belongs to an information set, then for every stage $t \leq \ell$, the prefix histories $c_1 \ldots c_t$ and $c'_1 \ldots c'_t$ belong to the same information set. As a direct consequence, the information structures that arise from such partitions are indeed trees.

Lemma 2.1. *For every information partition $\mathcal{U}$ of perfect synchronous recall, the information structure $\Gamma^*/_{\mathcal{U}}$ is a directed tree.*

We will use the term *information tree* when referring to the information structure associated with an information partition with synchronous perfect recall.

In the following, we discuss two alternative representations of information partitions.

2.2.2. *Observation.* The first alternative consists in describing the information received by the player in each stage. To do so, we specify a set Σ of *observation*



(A) Mealy automaton (B) Two-tape DFA

FIGURE 1. A Mealy automaton and a two-tape DFA over alphabet $\Gamma = \{a, b\}$ describing the same information partition (the symbol $*$ stands for $\{a, b\}$)

symbols and an *observation function* $\beta\colon \Gamma^+ \to \Sigma$. Intuitively, the player observes at every nonempty history τ the symbol $\beta(\tau)$; under the assumption of perfect recall, the information available to the player at history $\tau = c_1c_2\dots c_\ell$ is thus represented by the sequence of observations $\beta(c_1)\beta(c_1c_2)\dots\beta(c_1\dots c_\ell)$, which we call *observation history* (at τ); let us denote by $\hat{\beta}\colon \Gamma^* \to \Sigma^*$ the function that returns, for each play history, the corresponding observation history.

The information partition $\mathcal{U}_\beta$ represented by an observation function β is the collection of sets $U_\eta := \{\tau \in \Gamma^* \mid \hat{\beta}(\tau) = \eta\}$ indexed by observation histories $\eta \in \hat{\beta}(\Gamma^*)$. Clearly, information partitions described in this way verify the conditions of synchronous perfect recall: each information set U_η consists of histories of the same length (as η), and for every pair τ, τ' of histories with different observations $\hat{\beta}(\tau) \neq \hat{\beta}(\tau')$, and every pair of moves $c, c' \in \Gamma$, the observation history of the successors τc and $\tau' c'$ will also differ $\hat{\beta}(\tau c) \neq \hat{\beta}(\tau' c')$.

To describe observation functions by a finite-state automaton, we fix a *finite* set Σ of observations and specify a Mealy automaton $\mathcal{M} = (Q, \Gamma, \Sigma, q_\varepsilon, \delta, \lambda)$, with moves from Γ as input and observations from Σ as output. Then, we consider the extended output function of $\mathcal{M}$ as an observation function $\beta_{\mathcal{M}}\colon \Gamma^+ \to \Sigma$.

To illustrate, Figure 1a shows a Mealy automaton defining an observation function. The input alphabet is the set $\Gamma = \{a, b\}$ of moves, and the output alphabet is the set $\{1, 2\}$ of observations. For example, the histories abb and bba map to the same observation sequence, namely 111, thus they belong to the same information set; the information partition on histories of length 2 is $\{aa, ab, bb\}, \{ba\}$.

This formalism captures the standard approach for describing information in finite-state systems (see, e.g., Reif [24], Lin and Wonham [17], Kupferman and Vardi [15], van der Meyden and Wilke [26]).

2.2.3. *Indistinguishability.* As a second alternative, we represent information partitions as equivalence relations between histories, such that the equivalence classes correspond to information sets. Intuitively, a player cannot distinguish between equivalent histories.

We say that an equivalence relation is an *indistinguishability* relation if the represented information partition satisfies the conditions of synchronous perfect recall.

The following characterisation simply rephrases the relevant conditions for partitions in terms of equivalence relations.

Lemma 2.2. *An equivalence relation $R \subseteq \Gamma^* \times \Gamma^*$ is an indistinguishability relation if, and only if, it satisfies the following properties:*

(1) *For every pair $(\tau, \tau') \in R$, the histories τ, τ' are of the same length.*
(2) *For every pair of histories $\tau, \tau' \in R$ of length ℓ, every pair (ρ, ρ') of histories of length $t \leq \ell$ that occur as prefixes of τ, τ', respectively, is also related by $(\rho, \rho') \in R$.*

As a finite-state representation, we will consider indistinguishability relations recognised by two-tape automata. To illustrate, Figure 1b shows a two-tape automaton that defines the same information partition as the Mealy automaton of Figure 1a. Here and throughout the paper, the state q_{rej} represents a rejecting sink state. For example, the pair of words τ_1, τ_2 where $\tau_1 = abb$ and $\tau_2 = bba$ is accepted by the automaton (the state q_1 is accepting), meaning that the two words are indistinguishable.

Given a two-tape automaton $\mathcal{A} = (Q, \Gamma \times \Gamma, q_\varepsilon, \delta, F)$, the recognised relation $L(\mathcal{A})$ is, by definition, synchronous and hence satisfies condition (1) of Lemma 2.2. To decide whether $\mathcal{A}$ indeed represents an indistinguishability relation, we can use standard automata-theoretic techniques to verify that $L(\mathcal{A})$ is an equivalence relation, and that it satisfies the perfect-recall condition (2) of Lemma 2.2.

Lemma 2.3. *The question whether a given two-tape automaton recognises an indistinguishability relation with perfect recall is decidable in polynomial (actually, cubic) time.*

The idea of using finite-state automata to describe information constraints of players in infinite games has been advanced in a series of work by Maubert and different coauthors [19, 20, 4, 7], with the aim of extending the classical framework of temporal logic and automata for perfect-information games to more expressive structures. In the general setup, the formalism features binary relations between histories that can be asynchronous and may not satisfy perfect recall. The setting of synchronous perfect recall is adressed as a particular case described by a one-state automaton that compares observation sequences rather than move histories. This allows to capture indistinguishability relations that actually correspond to regular observation functions in our setup.

Another approach of relating game histories via automata has been proposed recently by Fournier and Lhote [9]. The authors extend our framework to arbitrary synchronous relations, which are not necessarily prefix closed — and thus do not satisfy perfect recall.

2.2.4. *Equivalent representations.* In general, any partition of a set X can be represented either as an equivalence relation on X — equating the elements of each part — or as a (complete) invariant function, that is a function $f\colon X \to Z$ such that $f(x) = f(y)$ if, and only if, x, y belong to the same part. Thus equivalence relations and invariant functions represent different faces of the same mathematical object. The correspondence is witnessed by the following canonical maps.

For every function $f\colon X \to Z$, the *kernel* relation

$$\ker f := \{(x, y) \in X \times X \mid f(x) = f(y)\}$$

is an equivalence. Given an equivalence relation $\sim \subseteq X \times X$, the *quotient map* $[\,\cdot\,]_\sim\colon X \to 2^X$, which sends each element $x \in X$ to its equivalence class $[x]_\sim := \{y \in X \mid y \sim x\}$, is a complete invariant function for $\sim$. Notice that the kernel of the quotient map is just $\sim$.

For the case of information partitions with synchronous perfect recall, the above correspondence relates indistinguishability relations and observation-history functions.

Lemma 2.4. *If $\beta \colon \Gamma^* \to \Sigma$ is an observation function, then $\ker \hat{\beta}$ is an indistinguishability relation that describes the same information partition. Conversely, if $\sim$ is an indistinguishability relation, then the quotient map is an observation function that describes the same information partition.*

Accordingly, every information partition given by an indistinguishability relation can be alternatively represented by an observation function, and vice versa. However, if we restrict to finite-state representations, the correspondence might not be preserved. In particular, as the quotient map of any indistinguishability relation on Γ^* has infinite range (histories of different length are always distinguishable), it is not definable by a Mealy automaton, which has finite output alphabet.

3. Observation is Weaker than Distinction

Firstly, we shall see that, for every regular observation function, the corresponding indistinguishability relation is also regular.

Proposition 3.1. *For every observation function β given by a Mealy automaton of size m, we can construct a two-tape* DFA *of size $O(m^2)$ that defines the corresponding indistinguishability relation $\ker \hat{\beta}$.*

Proof. To construct such a two-tape automaton, we run the given Mealy automaton on the two input tapes simultaneously, and send it into a rejecting sink state whenever the observation output on the first tape differs from the output on the second tape. Accordingly, the automaton accepts a pair $(\tau, \tau') \in (\Gamma \times \Gamma)^*$ of histories, if and only if, their observation histories are equal $\hat{\beta}(\tau) = \hat{\beta}(\tau')$. □

The statement of Proposition 3.1 is illustrated in Figure 1 where the structure of the two-tape DFA of Figure 1b is obtained as a parallel product of two copies of the Mealy automaton in Figure 1a, where $q_1 = (p_1, p_1)$, $q_2 = (p_2, p_2)$, $q_3 = (p_1, p_2)$, and $q_4 = (p_2, p_1)$.

For the converse direction, however, the model of imperfect information described by regular indistinguishability relations is strictly more expressive than the one based on regular observation functions.

Lemma 3.2. *There exists a regular indistinguishability relation that does not correspond to any regular observation function.*

Proof. As an example, consider a move alphabet with three letters $\Gamma := \{a, b, c\}$, and let $\sim \in \Gamma^* \times \Gamma^*$ relate two histories τ, τ' whenever they are equal or none of them contains the letter c. This is an indistinguishability relation, and it is recognised by the two-tape automaton of Figure 2.

We argue that the induced information tree has unbounded branching. All histories of the same length n that do not contain c are indistinguishable, hence $U_n = \{a, b\}^n$ is an information set. However, for every history $w \in U_n$ the history wc forms a singleton information set. Therefore U_n has at least 2^n successors in the information tree, for every n.

However, for any observation function, the degree of the induced information tree is bounded by the size of the observation alphabet. Hence, the information partition described by $\sim$ cannot be represented by an observation function of finite range and so, a fortiori, not by any regular observation function. □

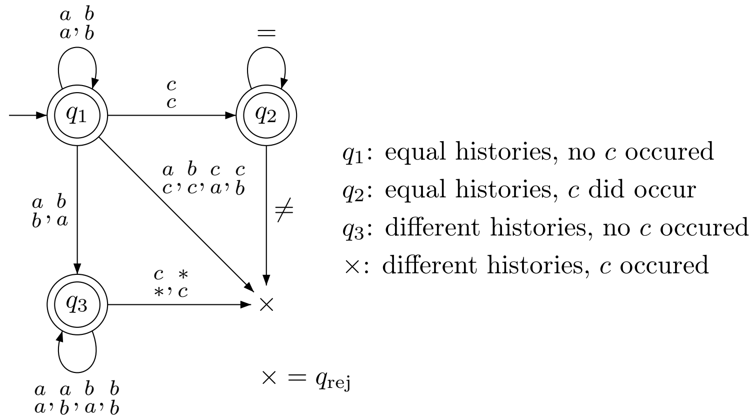


Figure 2. A two-tape DFA defining an indistinguishability relation that does not correspond to any regular observation function (the symbol $=$ stands for $\{{a \atop a}, {b \atop b}, {c \atop c}\}$, the symbol $\neq$ stands for $\{{x \atop y} \in \Gamma \times \Gamma \mid x \neq y\}$, and the symbol $*$ stands for $\{a, b, c\}$)

4. Which Distinctions Correspond to Observations

We have just seen, as a necessary condition for an indistinguishability relation to be representable by a regular observation function, that the information tree needs to be of bounded branching. In the following, we show that this condition is actually sufficient.

Theorem 4.1. *Let Γ be a finite set of moves. A regular indistinguishability relation $\sim$ admits a representation as a regular observation function if, and only if, the information tree $\Gamma^*/_\sim$ is of bounded branching.*

Proof. The *only-if*-direction is immediate. If for an indistinguishability relation $\sim$, there exists an observation function $\beta\colon \Gamma^+ \to \Sigma$ with finite range (not necessarily regular) such that $\sim = \ker \hat{\beta}$, then the maximal degree of the information tree $\Gamma^*/_\sim$ is at most $|\Sigma|$. Indeed, the observation-history function $\hat{\beta}$ is a strong homomorphism from the move tree Γ^* to the tree of observation histories $\hat{\beta}(\Gamma^*) \subseteq \Sigma^*$: it maps every pair $(\tau, \tau c)$ of successive move histories to the pair of successive observation histories $(\hat{\beta}(\tau), \hat{\beta}(\tau)\beta(\tau c))$, and conversely, for every pair of successive observation histories, there exists a pair of successive move histories that map to it. By the Homomorphism Theorem (in the general formulation of Mal'cev [18]), it follows that the information tree $\Gamma^*/_\sim = \Gamma^*/_{\ker \hat{\beta}}$ is isomorphic to the image $\hat{\beta}(\Gamma^*)$, which, as a subtree Σ^*, has degree at most $|\Sigma|$.

To verify the *if*-direction, consider an indistinguishability relation $\sim$ over Γ^*, given by a DFA $\mathcal{R}$, such that the information tree $\Gamma^*/_\sim$ has branching degree at most $n \in \mathbb{N}$.

Let us fix an arbitrary linear ordering $\preceq$ of Γ. First, we pick as a representative for each information set, its least element with respect to the lexicographical order $<_{\mathrm{lex}}$ induced by $\preceq$. Then, we order the information sets in $\Gamma^*/_\sim$ according to the lexicographical order of their representatives. Next, we define the *rank* of any nonempty history $\tau c \in \Gamma^*$ to be the index of its information set $[\tau c]_\sim$ in this order, restricted to successors of $[\tau]_\sim$ — this index is bounded by n. Let us consider the observation function β that associates to every history its rank. We claim that (1) it describes the same information partition as $\sim$ and (2) it is a regular function.

To prove the first claim, we show that whenever two histories are indistinguishable $\tau \sim \tau'$, they yield the same observation sequence $\hat{\beta}(\tau) = \hat{\beta}(\tau')$. The rank of a history is determined by its information set. Since $\tau \sim \tau'$, every pair (ρ, ρ') of prefix histories of the same length are also indistinguishable, and therefore yield the same rank $\beta(\rho) = \beta(\rho')$. By definition of $\hat{\beta}$, it follows that $\hat{\beta}(\tau) = \hat{\beta}(\tau')$. Conversely, to verify that $\hat{\beta}(\tau) = \hat{\beta}(\tau')$ implies $\tau \sim \tau'$, we proceed by induction on the length of histories. The basis concerns only the empty history and thus holds trivially. For the induction step, suppose $\hat{\beta}(\tau c) = \hat{\beta}(\tau' c')$. By definition of $\hat{\beta}$, we have in particular $\hat{\beta}(\tau) = \hat{\beta}(\tau')$, which by induction hypothesis implies $\tau \sim \tau'$. Hence, the information sets of the continuations τc and $\tau' c'$ are successors of the same information set $[\tau]_\sim = [\tau']_\sim$ in the information tree $\Gamma^*/\sim$. As we assumed that the histories τc and $\tau' c'$ have the same rank, it follows that they indeed belong to the same information set, that is $\tau c \sim \tau' c'$.

To verify the second claim on the regularity of the observation function β, we first notice that the following languages are regular:

- the (synchronous) lexicographical order $\{(\tau, \tau') \in (\Gamma \times \Gamma)^* \mid \tau \leq_{\text{lex}} \tau'\}$,
- the set of representatives $\{\tau \in \Gamma^* \mid \tau \leq_{\text{lex}} \tau' \text{ for all } \tau' \sim \tau\}$, and
- the representation relation $\{(\tau, \tau') \in \sim \mid \tau' \text{ is a representative}\}$.

Given automata recognising these languages, we can then construct, for each $k \leq n$, an automaton $\mathcal{A}_k$ that recognises the set of histories of rank at least k: together with the representative of the input history, guess the $k-1$ representatives that are below in the lexicographical order. Finally, we take the synchronous product of the automata $\mathcal{A}_1 \dots \mathcal{A}_k$ and equip it with an output function as follows: for every transition in the product automaton all components of the target state, up to some index k, are accepting — we define the output of the transition to be just this index k. This yields a Mealy automaton that outputs the rank of the input history, as desired. □

For further use, we estimate the size of the Mealy automaton defining the rank function as outlined in the proof. Suppose that an indistinguishability relation $\sim \subseteq (\Gamma \times \Gamma)^*$ given by a two-tape DFA $\mathcal{R}$ of size m gives rise to an information tree $\Gamma^*/_{L(\mathcal{R})}$ of degree n. The lexicographical order is recognisable by a two-tape DFA of size $O(|\Gamma|^2)$, bounded by $O(m)$; to recognise the set of representatives we take the synchronous product of this automaton with $\mathcal{R}$, and apply a projection and a complementation, obtaining a DFA of size bounded by $2^{O(m^2)}$); for the representation relation, we take a synchronous product of this automaton with $\mathcal{R}$ and obtain a two-tape DFA of size still bounded by $2^{O(m^2)}$. For every index $k \leq n$, the automaton $\mathcal{A}_k$ can be constructed via projection from a synchronous product of n such automata, hence its size is bounded by $2^{2^{O(nm^2)}}$. The Mealy automaton for defining the rank runs all these n automata synchronously, so it is of the same order of magnitude $2^{2^{O(nm^2)}}$.

To decide whether the information tree represented by a regular indistinguishability relation has bounded degree, we use a result from the theory of word-automatic structures [13, 3]. For the purpose of our presentation, we define an automatic presentation of a tree $T = (V, E)$ as a triple $(\mathcal{A}_V, \mathcal{A}_=, \mathcal{A}_E)$ of automata with input alphabet Γ, together with a surjective naming map $h\colon L \to V$ defined on a set of words $L \subseteq \Gamma^*$ such that

- $L(\mathcal{A}_V) = L$,
- $L(\mathcal{A}_=) = \ker h$, and

- $L(\mathcal{A}_E) = \{(u, v) \in L \times L \mid (h(u), h(v)) \in E\}$.

In this case, h is an isomorphism between $T = (V, E)$ and the quotient $(L, L(\mathcal{A}_E))/_{L(\mathcal{A}_=)}$. The size of such an automatic presentation is the added size of the three component automata. A tree is automatic if it has an automatic presentation.

For an information partition given by a indistinguishability relation $\sim$ defined by a two-tape-DFA $\mathcal{R}$ on a move alphabet Γ, the information tree $\Gamma^*/_\sim$ admits an automatic presentation with the naming map that sends every history τ to its information set $[\tau]_\sim$, and

- as domain automaton A_V, the one-state automaton accepting all of Γ^* (of size Γ);
- as the equality automaton $\mathcal{A}_=$, the two-tape DFA $\mathcal{R}$, and
- for the edge relation, a two-tape DFA $\mathcal{A}_E$ that recognises the relation

$$\{(\tau, \tau' c) \in \Gamma^* \times \Gamma^* \mid (\tau, \tau') \in L(\mathcal{R})\}.$$

The latter automaton is obtained from $\mathcal{R}$ by adding transitions from each accepting state, with any move symbol on the first tape and the padding symbol on the second tape, to a unique fresh accepting state from which all outgoing transitions lead to the rejecting sink q_{rej}. Overall, the size of the presentation will thus be bounded by $O(|\mathcal{R}|)$.

Now, we can apply the following result of Kuske and Lohrey.

Proposition 4.2. *([16, Propositions 2.14–2.15]) The question whether an automatic structure has bounded degree is decidable in exponential time. If the degree of an automatic structure is bounded, then it is bounded by $2^{2^{m^{O(1)}}}$ in the size m of the presentation.*

This allows to conclude that the criterion of Theorem 4.1 characterising regular indistinguishability relations that are representable by regular observation functions is effectively decidable. By following the construction for the rank function outlined in the proof of the theorem, we obtain a fourfold exponential upper bound for the size of a Mealy automaton defining an observation function.

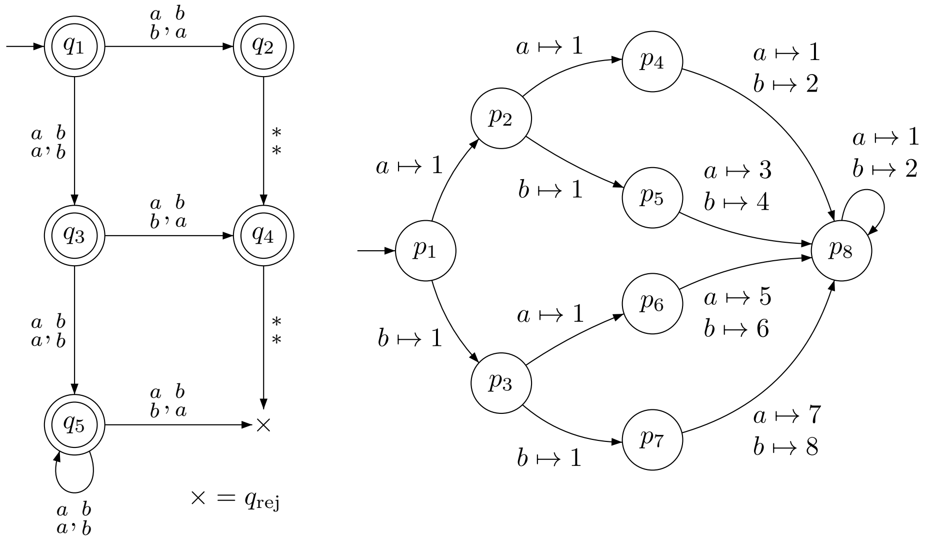
Theorem 4.3.
(i) *The question whether an indistinguishability relation given as a two-tape* DFA *admits a representation as a regular observation function is decidable in exponential time (with respect to the size of the* DFA*).*
(ii) *Whenever this is the case, we can construct a Mealy automaton of fourfold-exponential size and with at most doubly exponentially many output symbols that defines a corresponding observation function.*

5. Improving the Construction of Observation Automata

Theorem 4.3 establishes only a crude upper bound on the size of a Mealy automaton corresponding to a given indistinguishability DFA. In this section, we present a more detailed analysis that allows to improve the construction by one exponential.

Firstly, let us point out that an exponential blowup is generally unavoidable, for the size of the automaton and for its observation alphabet.

Example 5.1. Figure 3a shows a two-tape DFA that compares histories over a move alphabet $\{a, b\}$ with an embargo period of length k. Every pair of histories of length less than k is accepted, whereas history pairs of length k and onwards are rejected if, and only if, they are different (the picture illustrates the case for $k = 3$). A Mealy automaton that describes this indistinguishability relation needs to produce, for every different prefix of length k, a different observation symbol. To do so, it has



(A) Two-tape DFA (B) Mealy automaton

Figure 3. A synchronous two-tape automaton with $2k$ states (here $k = 3$) for which an equivalent observation Mealy automaton requires exponential number of states (2^k)

to store the first k symbols, which requires 2^k states and 2^k observation symbols (see Figure 3b).

We will first identify some structural properties of indistinguishability relations and their DFA, and then present the concrete construction.

5.1. **Structural properties of regular indistinguishability relations.** For the following, let us fix a move alphabet Γ and a two-tape DFA $\mathcal{R} = (Q, \Gamma \times \Gamma, q_\varepsilon, \delta, F)$ defining an indistinguishability relation $L(\mathcal{R}) = \sim$. For convenience, we will usually write $\delta(q, \frac{\tau}{\tau'})$ for $\delta(q, (\tau, \tau'))$.

We assume that the automaton $\mathcal{R}$ is minimal, in the usual sense that all states are reachable from the initial state, and the languages accepted from two different states are different. Note that, due to the property that whenever two histories are distinguishable, their continuations are also distinguishable, minimality of $\mathcal{R}$ also implies that all its states are accepting, except for the single sink state q_{rej}, that is, $F = Q \setminus \{q_{\text{rej}}\}$.

First, we classify the states according to the behaviour of the automaton when reading the same input words on both tapes. On the one hand, we consider the states reachable from the initial state on such inputs, which we call *reflexive* states:

$$\mathsf{Ref} = \{q \in Q \mid \exists \tau \in \Gamma^* : \delta(q_\varepsilon, \tfrac{\tau}{\tau}) = q\}.$$

On the other hand, we consider the states from which it is possible to reach the rejecting sink by reading the same input word on both tapes, which we call *ambiguous* states,

$$\mathsf{Amb} = \{q \in Q \mid \exists \tau \in \Gamma^* : \delta(q, \tfrac{\tau}{\tau}) = q_{\text{rej}}\}.$$

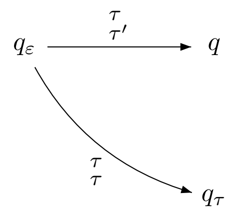
For instance, in the running example of Figure 1, the reflexive states are $\mathsf{Ref} = \{q_1, q_2\}$ and the ambiguous states are $\mathsf{Amb} = \{q_3, q_4, q_{\text{rej}}\}$.

Since indistinguishability relations are reflexive, all the reflexive states are accepting and by reading any pair of identical words from a reflexive state, we always reach an accepting state. Therefore, a reflexive state cannot be ambiguous. Perhaps less obviously, the converse also holds: a non-reflexive state must be ambiguous.

Lemma 5.1 (Partition Lemma)**.** $Q \setminus \mathsf{Ref} = \mathsf{Amb}$.

Proof. The inclusion $\mathsf{Amb} \subseteq Q \setminus \mathsf{Ref}$ (or, equivalently, that Amb and Ref are disjoint) follows from the definitions and the fact that $\sim$ is a reflexive relation, and thus $\delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\tau}) \neq q_{\mathrm{rej}}$ for all histories τ.

To show that $Q \setminus \mathsf{Ref} \subseteq \mathsf{Amb}$, let us consider an arbitrary state $q \in Q \setminus \mathsf{Ref}$. By minimality of $\mathcal{R}$, the state q is reachable from q_ε: there exist histories τ, τ' such that $\delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\tau'}) = q$. Let $q_\tau = \delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\tau})$ be the state reached after reading $\genfrac{}{}{0pt}{}{\tau}{\tau}$ (see figure). Thus, $q_\tau \in \mathsf{Ref}$ and in particular $q_\tau \neq q$. Again by minimality of $\mathcal{R}$, the languages accepted from q and q_τ are different. Hence, there exist histories π, π' such that $\genfrac{}{}{0pt}{}{\pi}{\pi'}$ is accepted from q and rejected from q_τ, or the other way round. In the former case, we have that $\tau\pi \sim \tau'\pi'$ and $\tau\pi \not\sim \tau\pi'$, which by transitivity of $\sim$, implies $\tau\pi' \not\sim \tau'\pi'$. This means that from state q reading $\genfrac{}{}{0pt}{}{\pi'}{\pi'}$ leads to q_{rej}, showing that $q \in \mathsf{Amb}$, which we wanted to prove. In the latter case, the argument is analogous.



□

We say that a pair of histories accepted by $\mathcal{R}$ is *ambiguous*, if, upon reading them, the automaton $\mathcal{R}$ reaches an ambiguous state other than q_{rej}. Histories τ, τ' that form an ambiguous pair are thus indistinguishable, so they must map to the same observation. However, there exists a suffix π such that the extensions $\tau \cdot \pi$ and $\tau' \cdot \pi$ become distinguishable. Therefore, any observation automaton for $\mathcal{R}$ has to reach two different states after reading τ and τ' since otherwise, the extensions by the suffix π would produce the same observation sequence, making $\tau \cdot \pi$ and $\tau' \cdot \pi$ wrongly indistinguishable. The argument generalises immediately to collections of more than two histories. We call a set of histories that are pairwise ambiguous an *ambiguous clique*.

We shall see later, in the proof of Lemma 5.5, that if the size of ambiguous cliques is unbounded, then the information tree $\Gamma^*/_{L(\mathcal{R})}$ has unbounded branching, and therefore there exists no Mealy automaton corresponding to $\mathcal{R}$. Now, we show conversely that whenever the size of the ambiguous cliques is bounded, we can construct such a Mealy automaton.

We say that two histories $\tau, \tau' \in \Gamma^*$ of the same length are *interchangeable*, denoted by $\tau \approx \tau'$, if $\delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\pi}) = \delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau'}{\pi})$, for all $\pi \in \Gamma^*$. Note that $\approx$ is an equivalence relation and that $\tau \approx \tau'$ implies $\delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\tau'}) \in \mathsf{Ref}$. The converse also holds.

Lemma 5.2. *For all histories $\tau, \tau' \in \Gamma^*$, we have $\delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\tau'}) \in \mathsf{Ref}$ if, and only if, $\tau \approx \tau'$.*

Proof. One direction, that $\tau \approx \tau'$ implies $\delta(q_\varepsilon, \genfrac{}{}{0pt}{}{\tau}{\tau'}) \in \mathsf{Ref}$), follows immediately from the definitions (take $\pi = \tau'$ in the definition of interchangeable histories).

For the reverse direction, let us suppose that $\delta(q_\varepsilon, \frac{\tau}{\tau'}) \in \mathsf{Ref}$. We will show that, for all histories τ'', the states $q_1 = \delta(q_\varepsilon, \frac{\tau}{\tau''})$ and $q_2 = \delta(q_\varepsilon, \frac{\tau'}{\tau''})$ accept the same language. Towards this, let π_1, π_2 be an arbitrary pair of histories such that $\frac{\pi_1}{\pi_2}$ is accepted from q_1. Then,

- $\tau\pi_1 \sim \tau'\pi_1$, because $\delta(q_\varepsilon, \frac{\tau}{\tau'}) \in \mathsf{Ref}$, and from a reflexive state reading $\frac{\pi_1}{\pi_1}$ does not lead to q_{rej} (by Lemma 5.1).
- $\tau\pi_1 \sim \tau''\pi_2$, because $\delta(q_\varepsilon, \frac{\tau}{\tau''}) = q_1$ and $\frac{\pi_1}{\pi_2}$ is accepted from q_1.

By transitivity of $\sim$, it follows that $\tau'\pi_1 \sim \tau''\pi_2$, hence $\frac{\pi_1}{\pi_2}$ is accepted from $q_2 = \delta(q_\varepsilon, \frac{\tau'}{\tau''})$. Accordingly, the language accepted from q_1 is included in the language accepted from q_2; the converse inclusion holds by a symmetric argument. Since the states q_1 and q_2 accept the same languages, and because the automaton $\mathcal{R}$ is minimal, it follows that $q_1 = q_2$, which means that τ and τ' are interchangeable. □

According to Lemma 5.2 and because $q_{\text{rej}} \notin \mathsf{Ref}$, all pairs of interchangeable histories are also indistinguishable. In other words, the interchangeability relation $\approx$ refines the indistinguishability relation $\sim$, and thus $[\tau]_\approx \subseteq [\tau]_\sim$ for all histories $\tau \in \Gamma^*$. In the running example (Figure 1), the sets $\{aa, ab, bb\}$ and $\{ba\}$ are $\sim$-equivalence classes, and the sets $\{aa, bb\}$, $\{ab\}$, and $\{ba\}$ are $\approx$-equivalence classes.

Let us lift the lexicographical order $\leq_{\mathsf{lex}}$ to sets of histories of the same length by comparing the smallest word of each set: we write $S \leq S'$ if $\min S \leq_{\mathsf{lex}} \min S'$. This allows us to rank the $\approx$-equivalence classes contained in a $\sim$-equivalence class, in increasing order. In the running example, if we consider the $\sim$-equivalence class $\{aa, ab, bb\}$, $\{aa, bb\}$ gets rank 1, and $\{ab\}$ gets rank 2 because $\{aa, bb\} \leq \{ab\}$. On the other hand, the $\sim$-equivalence class $\{ba\}$, as a singleton, gets rank 1.

Now, we denote by $\mathsf{idx}(\tau)$ the rank of the $\approx$-equivalence class containing τ. For example, $\mathsf{idx}(bb) = 1$ and $\mathsf{idx}(ab) = 2$. Further, we denote by $\mathsf{mat}(\tau)$ the square matrix of dimension $n = \max_{\tau' \in [\tau]_\sim} \mathsf{idx}(\tau')$ where we associate to each coordinate $i = 1, \ldots, n$ the i-th $\approx$-equivalence class C_i contained in $[\tau]_\sim$. The (i, j)-entry of $\mathsf{mat}(\tau)$ is the state $q_{ij} = \delta(q_\varepsilon, \frac{\tau_i}{\tau_j})$ where $\tau_i \in C_i$ and $\tau_j \in C_j$. Thanks to interchangeability, the state q_{ij} is well defined being independent of the choice of τ_i and τ_j.

Example 5.2. In the running example, we have $a \sim b$ thus $\mathsf{mat}(a) = \mathsf{mat}(b)$:

$$\mathsf{mat}(a) = \mathsf{mat}(b) = \begin{matrix} & \{a\} & \{b\} \\ \{a\} & q_1 & q_3 \\ \{b\} & q_4 & q_2 \end{matrix}.$$

Moreover $[aa]_\approx = \{aa, bb\}$, and $[ab]_\approx = \{ab\}$, and $[ba]_\approx = \{ba\}$, and thus:

$$\mathsf{mat}(aa) = \mathsf{mat}(ab) = \mathsf{mat}(bb) = \begin{pmatrix} q_1 & q_3 \\ q_4 & q_2 \end{pmatrix} \text{ and } \mathsf{mat}(ba) = \begin{pmatrix} q_2 \end{pmatrix}.$$

Note that the non-diagonal entries q_3 and q_4 are ambiguous states. This is true in general.

It is easy to see that diagonal entries in such matrices are reflexive states (Lemma 5.2). We can show conversely that non-diagonal entries are ambiguous states.

Lemma 5.3. *For all histories τ, the non-diagonal entries in* $\mathsf{mat}(\tau)$ *are ambiguous states.*

Proof. Non-diagonal entries in $\mathsf{mat}(\tau)$ correspond to pair of histories that are not $\approx$-equivalent, therefore those entries are not reflexive states (Lemma 5.2), hence they must be ambiguous states (Lemma 5.1). □

Next, we show how to construct, given $\mathsf{idx}(\tau)$ and $\mathsf{mat}(\tau)$, for some history τ, and a move $a \in \Gamma$, the index and matrix $\mathsf{idx}(\tau a)$ and $\mathsf{mat}(\tau a)$. The construction is independent of τ.

First, given a $n \times n$ matrix M with entries in Q, we define $\mathsf{next}(M)$ to be the $n \cdot |\Gamma| \times n \cdot |\Gamma|$ matrix obtained by substituting each entry q_{ij} in M with the $|\Gamma| \times |\Gamma|$ matrix where every (a, b)-entry is $\delta(q_\varepsilon, {a \atop b})$, as illustrated in the following example.

Example 5.3. In the running example, the $|\Gamma| \times |\Gamma|$ matrix associated with state q_1 is:

$$q_1 \mapsto \begin{pmatrix} \delta(q_1, {a \atop a}) & \delta(q_1, {a \atop b}) \\ \delta(q_1, {b \atop a}) & \delta(q_1, {b \atop b}) \end{pmatrix} = \begin{pmatrix} q_1 & q_3 \\ q_4 & q_2 \end{pmatrix}.$$

The matrices associated with the other states are (where we denote the q_{rej} state by $\times$):

$$q_2 \mapsto \begin{pmatrix} q_2 & \times \\ \times & q_1 \end{pmatrix} \quad q_3 \mapsto \begin{pmatrix} \times & q_1 \\ \times & q_4 \end{pmatrix} \quad q_4 \mapsto \begin{pmatrix} \times & \times \\ q_1 & q_3 \end{pmatrix}.$$

Hence for $M = \begin{pmatrix} q_1 & q_3 \\ q_4 & q_2 \end{pmatrix}$, we have $\mathsf{next}(M) = \begin{pmatrix} q_1 & q_3 & \times & q_1 \\ q_4 & q_2 & \times & q_4 \\ \times & \times & q_2 & \times \\ q_1 & q_3 & \times & q_1 \end{pmatrix}$.

Second, for every $n \times n$ matrix M with entries in Q, every $i \in \{1, \ldots, n\}$, and every move $a \in \Gamma$, we define $\mathsf{succ}_a(M, i) = (N, j)$, by the following construction:

(*i*) Initialise $N = \mathsf{next}(M)$; consider the (a, a) entry of the $|\Gamma| \times |\Gamma|$ matrix substituting the (i, i)-entry of M in N, and initialise j to be its position on the diagonal of N;

(*ii*) for every $1 \le k \le n \cdot |\Gamma|$, if the (k, j)-entry of N is the q_{rej} state, then remove the k-th row and k-th column (note that the j-th row and j-th column are never removed) and update the index j accordingly;

(*iii*) if two columns of N are identical, then remove the column and the corresponding row at the larger position. If the removed column is at the position j, assign the (smaller) position of the remaining duplicate column to j. Repeat this step until no two columns are identical. Return the final value of the N and j.

Example 5.4. Consider $M = \begin{pmatrix} q_1 & q_3 \\ q_4 & q_2 \end{pmatrix}$ and $i = 2$, which are the matrix and index of the history $\tau = b$ in the running example. In figures, the index i is depicted as a vertical arrow pointing to the ith column of the matrix. We obtain $\mathsf{succ}_a(M, i)$ (the matrix and index of $\tau' = ba$) as follows:

$$\begin{pmatrix} q_1 & \overset{\downarrow}{q_3} \\ q_4 & q_2 \end{pmatrix} \xrightarrow{(i)} \begin{pmatrix} q_1 & q_3 & \overset{\downarrow}{\times} & q_1 \\ q_4 & q_2 & \times & q_4 \\ \times & \times & q_2 & \times \\ q_1 & q_3 & \times & q_1 \end{pmatrix} \xrightarrow{(ii)} \begin{pmatrix} \overset{\downarrow}{q_2} \end{pmatrix} \xrightarrow{(iii)} \begin{pmatrix} \overset{\downarrow}{q_2} \end{pmatrix}$$

and we obtain $\mathsf{succ}_b(M, i)$ (the matrix and index of $\tau' = bb$) as follows:

$$\overset{\downarrow}{\begin{pmatrix} q_1 & q_3 \\ q_4 & q_2 \end{pmatrix}} \xrightarrow{(i)} \overset{\qquad\quad\downarrow}{\begin{pmatrix} q_1 & q_3 & \times & q_1 \\ q_4 & q_2 & \times & q_4 \\ \times & \times & q_2 & \times \\ q_1 & q_3 & \times & q_1 \end{pmatrix}} \xrightarrow{(ii)} \overset{\qquad\quad\downarrow}{\begin{pmatrix} q_1 & q_3 & q_1 \\ q_4 & q_2 & q_4 \\ q_1 & q_3 & q_1 \end{pmatrix}} \xrightarrow{(iii)} \overset{\downarrow}{\begin{pmatrix} q_1 & q_3 \\ q_4 & q_2 \end{pmatrix}}.$$

With the successor function along moves defined in this way, we obtain an homomorphic image of Γ^* on matrix-index pairs.

Lemma 5.4. *For all histories $\tau \in \Gamma^*$ and moves $c \in \Gamma$, if $(M, i) = (\mathsf{mat}(\tau), \mathsf{idx}(\tau))$, then $\mathsf{succ}_c(M, i) = (\mathsf{mat}(\tau c), \mathsf{idx}(\tau c))$.*

Proof. The result follows from the following remarks:

- In step (i), since $M = \mathsf{mat}(\tau)$ we can associate to each row/column of M an $\approx$-equivalence class (contained in $[\tau]_\sim$), say $C_1, C_2, \ldots, C_n$. For $b \in \Gamma$, and C an $\approx$-equivalence class, let $Cb = [wb]_\approx$ for $w \in C$ (which is independent of the choice of w and thus well-defined - it is easy to prove that $w \approx w'$ implies $wb \approx w'b$). We can associate to the rows/columns of $\mathsf{next}(M)$ the $\approx$-equivalence classes $C_j b$ (for each $1 \leq j \leq n$ and $b \in \Gamma$) in lexicographic order. The stored index is the index of the $\approx$-equivalence class of τa.
- In step (ii), we remove the rows/columns associated with an $\approx$-equivalence class that is not contained in $[\tau a]_\sim$ The stored index (pointing to the $\approx$-equivalence class containing τa) is updated accordingly.
- In step (iii), we merge identical rows/columns which correspond to identical $\approx$-equivalence classes. Keeping the leftmost class ensures the lexicographic order between $\approx$-equivalence classes is preserved. At the end, each $\approx$-equivalence class contained in $[\tau a]_\sim$ is indeed associated to some row/column, and the resulting matrix is $\mathsf{mat}(\tau a)$ with the correct index $\mathsf{idx}(\tau a)$. □

5.2. **Constructing the observation automaton.** For the remainder of the paper, let us fix an alphabet Γ and a two-tape DFA $\mathcal{R} = (Q, \Gamma \times \Gamma, \delta, q_\varepsilon, F)$ such that the branching degree of the information tree $\Gamma^*/_{L(\mathcal{R})}$ is bounded. Let m be the size of $\mathcal{R}$.

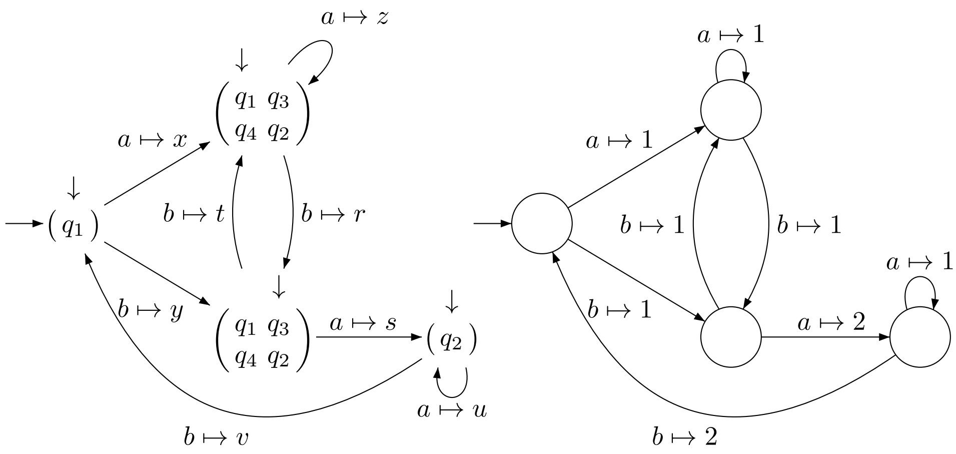
We define a Mealy automaton $\mathcal{F} = (P, \Gamma, \Sigma, p_\varepsilon, \delta, \lambda)$ over the input alphabet Γ and an output alphabet Σ in two phases: first, we define the semi-automaton $\mathcal{F}_0 = (P, \Gamma, p_\varepsilon, \delta)$ and then we construct the output alphabet Σ and the output function λ. To define the semi-automaton $\mathcal{F}_0$, we set:

- $P := \{(M, i) \mid M = \mathsf{mat}(\tau) \text{ and } i = \mathsf{idx}(\tau) \text{ for some history } \tau\}$,
- $p_\varepsilon := (q_\varepsilon, 1)$,
- for every state $(M, i) \in P$ and every move $c \in \Gamma$, let $\delta((M, i), c) = \mathsf{succ}_c(M, i)$.

According to Lemma 5.4, the state space P is the closure of $\{p_\varepsilon\}$ under the c-successor operation, for all $c \in \Gamma$. It remains to show that P is finite. The key is to bound the dimension of the largest matrix in P, which is the size of the largest ambiguous clique.

Lemma 5.5. *If the branching degree of the information tree $\Gamma^*/_{L(\mathcal{R})}$ is bounded, then the largest ambiguous clique contains at most a doubly-exponential number of histories (with respect to the size of $\mathcal{R}$).*

Proof. First we show by contradiction that the size of the ambiguous cliques is bounded. Since the number of ambiguous states in $\mathcal{R}$ is finite, if there exists an



(A) Transition structure (B) Instantiated observations

FIGURE 4. Construction of the Mealy automaton from the two-tape DFA of Figure 1b

arbitrarily large ambiguous clique, then by Ramsey's theorem [23], there exists an arbitrarily large set $\{\tau_1, \tau_2, \ldots, \tau_k\}$ of histories and a state $q \in \mathsf{Amb} \setminus \{q_{\text{rej}}\}$ such that $\delta(q_\varepsilon, {\tau_i \atop \tau_j}) = q$ for all $1 \le i < j \le k$. By definition of Amb, there exists a nonempty history τc such that $\delta(q, {\tau c \atop \tau c}) = q_{\text{rej}}$. Consider such a history τc of minimal length. The histories $\tau_i \tau$ $(i = 1, \ldots, k)$ are in the same $\sim$-equivalence class, but the equivalence classes $[\tau_i \tau c]_\sim$ are pairwise distinct. Therefore, the number of successors of $[\tau_i \tau]_\sim$ is at least k, thus arbitrarily large, in contradiction with the assumption that the branching degree the information tree $\Gamma^*/_{L(\mathcal{R})}$ is bounded.

Note that the size of the largest ambiguous clique corresponds to the maximum number of $\approx$-equivalence classes contained in an $\sim$-equivalence class (Lemma 5.3). We show that this number is at most doubly-exponential. Similarly to the proof of Theorem 4.1, we notice that the set of $\approx$*-representatives* defined by $\{\tau \in \Gamma^* \mid \tau \le_{\text{lex}} \tau' \text{ for all } \tau' \approx \tau\}$ is regular, and therefore the representation relation $\{(\tau, \tau') \in \sim \mid \tau' \text{ is a } \approx\text{-representative}\}$ is also regular. Using a result of Weber [28, Theorem 2.1], there is a bound on the number of $\approx$-representatives that a history can have that is exponential in the size ℓ of the two-tape DFA recognising the representation relation, namely $O(\ell)^\ell$, and ℓ is bounded by $2^{O(m^2)}$ by the same argument as in the proof of Theorem 4.1 (where m is the size of $\mathcal{R}$). This provides a doubly-exponential bound $2^{2^{O(m^2)}}$ on the size of the ambiguous cliques. □

According to Lemma 5.5, the dimension k of the largest matrix in P is at most doubly exponential in $|\mathcal{R}|$. The number of matrices of a fixed dimension d is at most $|Q|^{d^2}$. Overall the number of matrices that appear in P is therefore bounded by $k \cdot |Q|^{k^2}$, and as the index is at most k, it follows that the number of states in P is bounded by $k^2 \cdot |Q|^{k^2}$, that is exponential in k and triply exponential in the size of $\mathcal{R}$.

The construction of the Mealy automaton for the two-tape DFA of Figure 1b is shown in Figure 4a. The variables x, y, z, r, s, t, u, v represent the (currently) unknown observation values of the output function. We will build a system of constraints over these variables by considering pairs of histories in the automaton, and in the Mealy automaton. For example, for $\tau = a$ and $\tau' = b$, we have $\tau \sim \tau'$

(according to the automaton), and therefore we derive the constraint $x = y$ in the Mealy automaton.

We are now ready to define the output function. Towards this, we associate to each state $p \in P$ and letter $a \in \Gamma$, a variable $x_{p,a}$ intended to represent the output value $\lambda(p, a)$. We gather all constraints that these variables should satisfy to describe a valid output function, and we show that the constraints are satisfiable.

For the semi-automaton $\mathcal{F}_0$ defined so far, consider the parallel product $\mathcal{F}_0 \parallel \mathcal{F}_0$ (which is a semi-automaton over the alphabet $\Gamma \times \Gamma$), and the synchronised product of $\mathcal{F}_0 \parallel \mathcal{F}_0$ with $\mathcal{R}$ (thus again a semi-automaton over alphabet $\Gamma \times \Gamma$).

Our constraints are either equality or disequality between variables. We construct a set Φ of constraints by looking at the synchronised product $(\mathcal{F}_0 \parallel \mathcal{F}_0) \times R$: for every reachable state $((p_1, p_2), q)$ with $q \neq q_{\text{rej}}$ and all letters $a, b \in \Gamma$ (possibly $a = b$), if $\delta(q, {a \atop b}) \neq q_{\text{rej}}$, then add the constraint $x_{p_1,a} = x_{p_2,b}$ to Φ, otherwise add the constraint $x_{p_1,a} \neq x_{p_2,b}$ to Φ.

Example 5.5. We obtain the following set of constraints for the Mealy automaton of Figure 4 (we omit trivial constraints such as $x = x$):

$x = y$	witnessed by $a \sim b$	$s \neq t$ witnessed by $ba \not\sim bb$
$t = z$	witnessed by $aa \sim bb$	$u \neq v$ witnessed by $baa \not\sim bab$
$r = t$	witnessed by $ab \sim bb$	$z \neq s$ witnessed by $aa \not\sim ba$
$z = r$	witnessed by $aa \sim ab$	$r \neq s$ witnessed by $ab \not\sim ba$

which is equivalent to the set of constraints $\{x = y, z = r = t, t \neq s, u \neq v\}$ and is satisfiable, e.g., with the following assignment (see Figure 4b):

$x = y = 1$	$s = 2$	$u = 1$
$z = r = t = 1$		$v = 2$

Lemma 5.6.
- *The set Φ of constraints is satisfiable (over any infinite domain).*
- *Every satisfying assignment for Φ describes an output function $\lambda \colon P \times \Gamma \to \Sigma$ such that $(P, \Gamma, \Sigma, p_\varepsilon, \delta, \lambda)$ is an observation automaton equivalent to $\mathcal{R}$.*

Proof. For the first point, it is sufficient to show that no contradiction occurs in Φ, namely that the following situations are impossible: Φ contains the constraint $x_1 \neq x_k$ and a chain of equalities between variables $x_1 = x_2$, $x_2 = x_3, \ldots, x_{k-1} = x_k$. Towards a contradiction, suppose that such a situation occurs — with $k = 3$ for simplicity of presentation, the argument generalises straightforwardly to every finite k — and assume $x_{p,a} = x_{r,b} = x_{s,\gamma}$ and $x_{p,a} \neq x_{s,\gamma}$ are constraints in Φ. It follows that:

(1) there exist histories u_1, u_2 such that
- $p = \delta(p_\varepsilon, u_1)$,
- $r = \delta(p_\varepsilon, u_2)$,
- $u_1 a \sim u_2 b$;

(2) there exist histories v_2, v_3 such that
- $r = \delta(p_\varepsilon, v_2)$,
- $s = \delta(p_\varepsilon, v_3)$,
- $v_2 b \sim v_3 \gamma$;

(3) there exist histories w_1, w_3 such that $w_1 \sim w_3$ and
- $p = \delta(p_\varepsilon, w_1)$,
- $s = \delta(p_\varepsilon, w_3)$,
- $w_1 a \not\sim w_3 \gamma$.

Note that the states p and r differ only by their index, not by their matrix (by Lemma 5.4 because $u_1 \sim u_2$, and thus $\mathsf{mat}(u_1) = \mathsf{mat}(u_2)$), analogously for states r and s. Hence, for some matrix M we can write $p = (M, m_1)$, $r = (M, m_2)$, and

$s = (M, m_3)$. Then it follows from Lemma 5.4 and the definitions of $\mathsf{mat}(\cdot)$ and $\mathsf{idx}(\cdot)$ that (denoting by $M(i,j)$ the (i,j)-entry of M):

- $M(m_1, m_2) = \delta(q_\varepsilon, {u_1 \atop u_2})$,
- $M(m_2, m_3) = \delta(q_\varepsilon, {v_2 \atop v_3})$,
- $M(m_1, m_3) = \delta(q_\varepsilon, {w_1 \atop w_3})$.

Now consider, in the $\sim$-equivalence class $[u_1]_\sim$ of u_1, the m_3-th $\approx$-equivalence class C, and a word $u_3 \in C$. Then $\mathsf{mat}(u_3) = M$ and $\mathsf{idx}(u_3) = m_3$, thus $s = (M, m_3) = \delta(p_\varepsilon, u_3)$. It follows that:

- $M(m_2, m_3) = \delta(q_\varepsilon, {u_2 \atop u_3})$,
- $M(m_1, m_3) = \delta(q_\varepsilon, {u_1 \atop u_3})$,

and therefore $u_2 b \sim u_3\gamma$ and $u_1 a \not\sim u_3\gamma$, which together with $u_1 a \sim u_2 b$ contradicts the transitivity of $\sim$. Hence, we can conclude that the constraint set Φ is satisfiable.

For the second point, fix a satisfying assignment for the constraints in Φ. Take the set of values assigned to the variables as the (finite) output alphabet Σ, and define the output function by $\lambda(p, a) = x_{p,a}$.

We show by induction on the length of histories that the indistinguishability relation induced by the Mealy automaton is the same as the one defined by $\mathcal{R}$. The base case is trivial. For the induction step, let us consider an arbitrary pair τ, τ' of histories of the same length, under the induction hypothesis, $\hat{\lambda}(\tau) = \hat{\lambda}(\tau')$ if, and only if, $\tau \sim \tau'$ (according to the automaton $\mathcal{R}$). For any pair $a, b \in \Gamma$ of letters, if $\tau \not\sim \tau'$, then $\tau a \not\sim \tau' b$ and $\hat{\lambda}(\tau a) \neq \hat{\lambda}(\tau' b)$. Else, if $\tau \sim \tau'$, let $p = \delta(p_\varepsilon, \tau)$ and $p' = \delta(p_\varepsilon, \tau')$ be the states reached in the semi-automaton $\mathcal{F}_0$ after reading τ and τ', and let $q = \delta(q_\varepsilon, {\tau \atop \tau'})$ be the state reached in the automaton $\mathcal{R}$ after reading the pair (τ, τ'). It follows that the state $((p, p'), q)$ is reachable in the synchronised product $(\mathcal{F}_0 \parallel \mathcal{F}_0) \times R$. Here, we distinguish two cases:

- if $\tau a \sim \tau' b$, then the constraint $x_{p,a} = x_{p',b}$ is in Φ, and therefore the observation of a in state p is the same as the observation of b in state p' $(\lambda(p, a) = \lambda(p', b))$.
- if $\tau a \not\sim \tau' b$, then the constraint $x_{p,a} \neq x_{p',b}$ is in Φ, and therefore the observation of a in state p is different from the observation of b in state p' $(\lambda(p, a) \neq \lambda(p', b))$.

In either case, we thus have $\hat{\lambda}(\tau a) = \hat{\lambda}(\tau' b)$ if, and only if, $\tau a \sim \tau' b$, which concludes the proof. □

Lemma 5.6 establishes the correctness of the constructed Mealy automaton $\mathcal{F}$. Since the size of $\mathcal{F}$ is exponential in the size k of the largest ambiguous clique, and k is at most doubly-exponential (Lemma 5.5), we get the following result.

Theorem 5.7. *For every indistinguishability relation given by a two-tape* DFA $\mathcal{R}$ *such that the information tree* $\Gamma^*/_{L(\mathcal{R})}$ *is of bounded branching, we can construct a Mealy automaton of triply exponential size (with respect to the size of* $\mathcal{R}$*) that defines a corresponding observation function.*

6. Conclusion

The question of how to model information in infinite games is fundamental to defining their strategy space. As the decisions of each player are based on the available information, strategies are functions from information sets to actions. Accordingly, the information structure of a player in a game defines the support of her strategy space.

The assumption of synchronous perfect recall gives rise to trees as information structures (Lemma 2.1). In the case of observation functions with a finite range Σ, these trees are subtrees of the complete $|\Sigma|$-branching tree Σ^* — on which ω-tree automata can work (see [25, 12] for surveys on such techniques). Concretely, every strategy based on observations can be represented as a labelling of the tree Σ^* with actions; the set of all strategies for a given game forms a regular (that is, automata-recognisable) set of trees. Moreover, when considering winning conditions that are also regular, Rabin's Theorem [22] allows to conclude that winning strategies also form a regular set. Indeed, we can construct effectively a tree automaton that recognises the set of strategies (for an individual player) that enforce a regular condition and, if this set is non-empty, we can also synthesise a Mealy automaton that defines one of these strategies. In summary, the interpretation of strategies as observation-directed trees allows us to search the set of all strategies systematically for winning ones using tree-automatic methods.

In contrast, when setting out with indistinguishability relations, we obtain more complicated tree structures that do not offer a direct grip to classical tree-automata techniques. As the example of Lemma 3.2 shows, there are cases where the information tree of a game is not regular, and so the set of all strategies is not recognisable by a tree automaton. Accordingly, the automata-theoretic approach to strategy synthesis via Rabin's Theorem cannot be applied to solve, for instance, the basic problem of constructing a finite-state strategy for one player to enforce a given regular winning condition.

On the other hand, modelling information with indistinguishability relations allows for significantly more expressiveness than observation functions. This covers notably settings where a player can receive an unbounded amount of information in one round. For instance, models with causal memory where one player may communicate his entire observation history to another player in one round can be captured with regular indistinguishability relation, but not with observation functions of any finite range. Even when an information partition that can be represented by finite-state observation functions, the representation by an indistinguishability relation may be considerably more succinct. For instance, a player that observes the move history perfectly, but with a delay of d rounds can be described by a two-tape DFA with $O(d)$ many states, whereas any Mealy automaton would require exponentially more states to define the corresponding observation function.

At the bottom line, as a finite-state model of information, indistinguishability relations are strictly more expressive and can be (at least exponentially) more succinct than observation functions. In exchange, the observation-based model is directly accessible to automata-theoretic methods, whereas the indistinguishability-based model is not. Our result in Theorem 4.3 allows to identify effectively the instances of indistinguishability relations for which this gap can be bridged. That is, we may take advantage of the expressiveness and succinctness of indistinguishability relations to describe a game problem and use the procedure to obtain, whenever possible, a reformulation in terms of observation functions towards solving the initial problem with automata-theoretic methods.

This initial study opens several exciting research directions. One immediate question is whether the fundamental finite-state methods on strategy synthesis for games with imperfect information can be extended from the observation-based model to the one based on indistinguishability relations. Is it decidable, given a game for one player with a regular winning condition against Nature, whether there exist a winning strategy ? Can the set of all winning strategies be described by finite-state automata ? In case this set is non-empty, does it contain a strategy defined by a finite-state automaton ?

Another, more technical, question concerns the automata-theoretic foundations of games. The standard models are laid out for representations of games and strategies as trees of a fixed branching degree. How can these automata models be extended to trees with unbounded branching towards capturing strategies constrained by indistinguishability relations ? Likewise, the automatic structures that arise as information quotients of indistinguishability relations form a particular class of trees, where both the successor and the descendant relation (that is, the transitive closure) are regular. On the one hand, this particularity may allow to decide properties about games (viz. their information trees) that are undecidable when considering general automatic trees, notably regarding bisimulation or other forms of game equivalence.

Finally, in a more application-oriented perspective, it will be worthwhile to explore indistinguishability relations as a model for games where players can communicate via messages of arbitrary length. In particular this will allow to extend the framework of infinite games on finite graphs to systems with causal memory considered in the area of distributed computing.

References

[1] Michael Bacharach. Some extensions of a claim of Aumann in an axiomatic model of knowledge. *Journal of Economic Theory*, 37(1):167–190, 1985. doi:10.1016/0022-0531(85)90035-3.

[2] Dietmar Berwanger, Lukasz Kaiser, and Bernd Puchala. A perfect-information construction for coordination in games. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*, volume 13 of *LIPIcs*, pages 387–398. Leibniz-Zentrum fuer Informatik, 2011. doi:10.4230/LIPIcs.FSTTCS.2011.387.

[3] Achim Blumensath and Erich Grädel. Automatic structures. In *Logic in Computer Science (LICS 2000)*, pages 51–62. IEEE Comput. Soc, 2000. doi:10.1109/LICS.2000.855755.

[4] Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Uniform strategies, rational relations and jumping automata. *Information and Computation*, 242:80–107, June 2015. URL: http://www.sciencedirect.com/science/article/pii/S0890540115000279, doi:10.1016/j.ic.2015.03.012.

[5] Julius R. Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969. doi:10.2307/1994916.

[6] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-Francois Raskin. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, Volume 3, Issue 3, 2007. doi:DOI:10.2168/LMCS-3(3:4)2007.

[7] Catalin Dima, Bastien Maubert, and Sophie Pinchinat. Relating Paths in Transition Systems: The Fall of the Modal Mu-Calculus. *ACM Transactions on Computational Logic (TOCL)*, 19(3):23:1–23:33, September 2018. doi:10.1145/3231596.

[8] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about knowledge*. MIT Press, Cambridge, Mass., 2003.

[9] Paulin Fournier and Nathan Lhote. Equivalence kernels of sequential functions and sequential observation synthesis. *CoRR*, abs/1910.06019, 2019. URL: http://arxiv.org/abs/1910.06019.

[10] John Geanakoplos. Common Knowledge. *Journal of Economic Perspectives*, 6(4):53–82, 1992. doi:10.1257/jep.6.4.53.

[11] Olivier Gossner and Tristan Tomala. Repeated games with complete information. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 7616–7630. Springer New York, New York, NY, 2009. doi:10.1007/978-0-387-30440-3_451.

[12] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, logics, and infinite games*. Number 2500 in Lecture notes in computer science. Springer, 2002.

[13] Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, and Daniel Leivant, editors, *Logic and Computational Complexity*, volume 960, pages 367–392. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995. doi:10.1007/3-540-60178-3_93.

[14] Harold W. Kuhn. Extensive games and the problem of information, Contributions to the theory of games II. *Annals of Mathematics Studies*, 28:193–216, 1953.

[15] Orna Kupferman and Moshe Y. Vardi. Synthesis with Incomplete Informatio. In Howard Barringer, Michael Fisher, Dov Gabbay, and Graham Gough, editors, *Advances in Temporal Logic*, Applied Logic Series, pages 109–127. Springer Netherlands, Dordrecht, 2000. doi:10.1007/978-94-015-9586-5_6.

[16] Dietrich Kuske and Markus Lohrey. Automatic structures of bounded degree revisited. *The Journal of Symbolic Logic*, 76(04):1352–1380, 2011. doi:10.2178/jsl/1318338854.

[17] Feng Lin and Walter M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, April 1988. URL: http://www.sciencedirect.com/science/article/pii/0020025588900011, doi:10.1016/0020-0255(88)90001-1.

[18] Anatoly I. Mal'cev. *Algebraic Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1973. doi:10.1007/978-3-642-65374-2.

[19] Bastien Maubert. *Logical foundations of games with imperfect information : uniform strategies. (Fondations logiques des jeux à information imparfaite : stratégies uniformes)*. PhD thesis, University of Rennes 1, France, 2014. URL: https://tel.archives-ouvertes.fr/tel-00980490.

[20] Bastien Maubert and Sophie Pinchinat. A General Notion of Uniform Strategies. *International Game Theory Review*, 16(01):1440004, March 2014. URL: https://www.worldscientific.com/doi/abs/10.1142/S0219198914400040, doi:10.1142/S0219198914400040.

[21] Boleslaw Mikolajczak. *Algebraic and structural automata theory*. Annals of Discrete Mathematics. North-Holland, Amsterdam, 1991.

[22] Michael Oser Rabin. *Automata on Infinite Objects and Church's Problem*. American Mathematical Society, Boston, MA, USA, 1972.

[23] Frank P. Ramsey. On a problem in formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.

[24] John H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984. doi:10.1016/0022-0000(84)90034-5.

[25] Wolfgang Thomas. On the synthesis of strategies in infinite games. In *Symposium on Theoretical Aspects of Computer Science (STACS 1995)*, volume 900, pages 1–13. Springer, 1995. doi:10.1007/3-540-59042-0_57.

[26] Ron van der Meyden and Thomas Wilke. Synthesis of Distributed Systems from Knowledge-Based Specifications. In Martín Abadi and Luca de Alfaro, editors, *CONCUR 2005 – Concurrency Theory*, volume 3653, pages 562–576, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. URL: http://link.springer.com/10.1007/11539452_42, doi:10.1007/11539452_42.

[27] John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.

[28] Andreas Weber. On the valuedness of finite transducers. *Acta Inf.*, 27(8):749–780, 1990. doi:10.1007/BF00264285.