Different Strokes in Randomised Strategies: Revisiting Kuhn's Theorem under Finite-Memory Assumptions[⋆]

James C. A. Main and Mickael Randour

F.R.S.-FNRS & UMONS – Université de Mons, Belgium

Abstract. Two-player (antagonistic) games on (possibly stochastic) graphs are a prevalent model in theoretical computer science, notably as a framework for reactive synthesis.

Optimal strategies may require randomisation when dealing with inherently probabilistic goals, balancing multiple objectives, or in contexts of partial information. There is no unique way to define randomised strategies. For instance, one can use so-called *mixed* strategies or *behavioural* ones. In the most general setting, these two classes do not share the same expressiveness. A seminal result in game theory — *Kuhn's theorem* — asserts their equivalence in games of perfect recall.

This result crucially relies on the possibility for strategies to use *infinite memory*, i.e., unlimited knowledge of all past observations. However, computer systems are finite in practice. Hence it is pertinent to restrict our attention to *finite-memory* strategies, defined as automata with outputs. Randomisation can be implemented in these in different ways: the *initialisation*, *outputs* or *transitions* can be randomised or deterministic respectively. Depending on which aspects are randomised, the expressiveness of the corresponding class of finite-memory strategies differs.

In this work, we study two-player concurrent stochastic games and provide a complete taxonomy of the classes of finite-memory strategies obtained by varying which of the three aforementioned components are randomised. Our taxonomy holds in games of perfect and imperfect information with perfect recall, and in games with more than two players. We also provide an adapted taxonomy for games with imperfect recall.

Keywords: two-player games on graphs · stochastic games · Markov decision processes · finite-memory strategies · randomised strategies

1 Introduction

Games on graphs. Games on (possibly stochastic) graphs have been studied for decades, both for their own interest (e.g., [1, 2, 3]) and for their value as a framework for *reactive synthesis* (e.g., [4, 5, 6, 7]). The core problem is almost always to find *optimal strategies* for the players: strategies that guarantee winning for Boolean winning conditions (e.g., [8, 9, 10, 11]), or strategies that achieve the best possible payoff in quantitative contexts (e.g., [1, 12, 13]). In multi-objective settings, one is interested in *Pareto-optimal* strategies (e.g., [14, 15, 16, 17]), but the bottom line is the same: players are looking for strategies that guarantee the best possible results.

In reactive synthesis, we model the interaction between a system and its uncontrollable environment as a two-player antagonistic game, and we represent the specification to ensure as a winning objective. An optimal strategy for the system in this game then constitutes a formal blueprint for a *controller* to implement in the real world [7].

Randomness in strategies. In essence, a *pure strategy* is simply a function mapping histories (i.e., the past and present of a play) to an action deterministically.

Optimal strategies may require *randomisation* when dealing with inherently probabilistic goals, balancing multiple objectives, or in contexts of partial information: see, e.g., [18, 16, 19, 17]. There are different ways of randomising strategies. For instance, a *mixed* strategy is essentially a probability distribution over a set of pure strategies. That is, the player randomly selects a pure strategy at the beginning of the game and then follows it for the entirety of the play without resorting to randomness ever again. By contrast, a *behavioural* strategy randomly selects an action at each step: it thus maps histories to probability distributions over actions.

[⋆] James C. A. Main is an F.R.S.-FNRS Research Fellow and Mickael Randour is an F.R.S.-FNRS Research Associate. Both authors are members of the TRAIL institute. This work has been supported by the Fonds de la Recherche Scientifique – FNRS under Grant n° T.0188.23 (PDR ControlleRS).

Kuhn's theorem. In full generality, these two definitions yield different classes of strategies (e.g., [20] or [21, Chapter 11]). Nonetheless, Kuhn's theorem [22] proves their equivalence under a mild hypothesis: in games of *perfect recall*, for any mixed strategy there is an equivalent behavioural strategy and vice-versa. A game is said to be of perfect recall for a given player if said player never forgets their previous knowledge and the actions they have played (i.e., they can observe their own actions). Let us note that perfect recall and *perfect information* are two different notions: perfect information is not required to have perfect recall.

Let us highlight that Kuhn's theorem crucially relies on two elements. First, mixed strategies can be distributions over an *infinite* set of pure strategies. Second, strategies can use *infinite memory*, i.e., they are able to remember the past completely, however long it might be. Indeed, consider a game in which a player can choose one of two actions in each round. One could define a (memoryless) behavioural strategy that selects one of the two actions by flipping a coin each round. This strategy generates infinitely many sequences of actions, therefore any equivalent mixed strategy needs the ability to randomise between infinitely many different sequences, and thus, infinitely many pure strategies. Moreover, infinitely many of these sequences require infinite memory to be generated (due to their non-regularity).

Finite-memory strategies. From the point of view of reactive synthesis, infinite-memory strategies, along with randomised ones relying on infinite supports, are undesirable for implementation. This is why a plethora of recent advances has focused on *finite-memory* strategies, usually represented as (a variation on) Mealy machines, i.e., finite automata with outputs. See, e.g., [3, 14, 23, 17, 24, 25]. Randomisation can be implemented in these finite-memory strategies in different ways: the *initialisation*, *outputs* or *transitions* can be randomised or deterministic respectively.

Depending on which aspects are randomised, the expressiveness of the corresponding class of finite-memory strategies differs: in a nutshell, *Kuhn's theorem crumbles when restricting ourselves to finite memory.* For instance, we show that some finite-memory strategies with only randomised outputs (i.e., the natural equivalent of behavioural strategies) cannot be emulated by finite-memory strategies with only randomised initialisation (i.e., the natural equivalent of mixed strategies) — see Lemma 5.3. Similarly, it is known that some finite-memory strategies that are encoded by Mealy machines using randomisation in all three components admit no equivalent using randomisation only in outputs [26, 20].

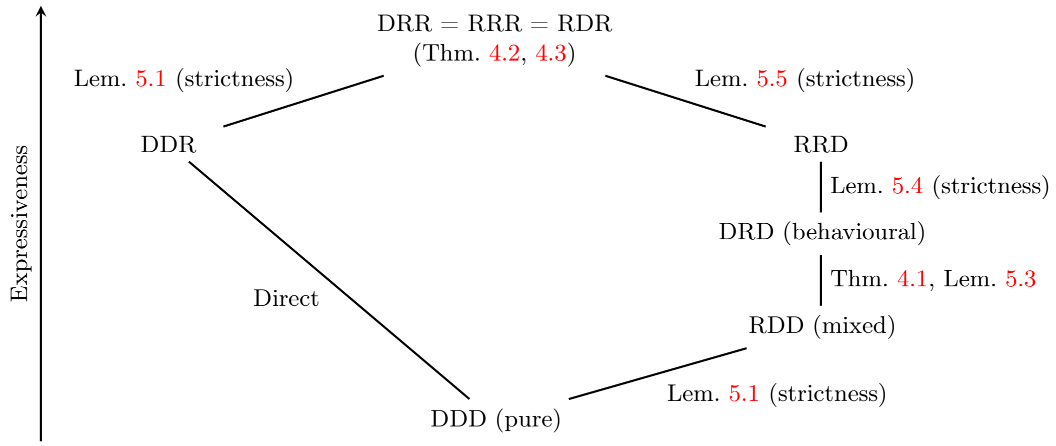


Fig. 1.1: Lattice of strategy classes in terms of expressible probability distributions over plays against all strategies of the other player. In the three-letter acronyms, the letters, in order, refer to the initialisation, outputs and updates of the Mealy machines: D and R respectively denote deterministic and randomised components. Each line in the figure indicates that the class above is strictly more expressive than the class below.

Our contributions. We consider *two-player zero-sum concurrent stochastic games of perfect information* (e.g., [27, 28]), encompassing two-player turn-based (deterministic) games and Markov decision processes as particular subcases. We establish a *Kuhn-like taxonomy* of the classes of finite-memory strategies obtained by varying which of the three aforementioned Mealy machine components are randomised: we illustrate it in Figure 1.1, and describe it fully in Section 3.

Let us highlight a few elements. Naturally, the least expressive model corresponds to pure strategies. In contrast to what happens with infinite memory, and as noted in the previous paragraph, we see that mixed strategies are strictly less expressive than behavioural ones. We also observe that allowing randomness both in initialisation and in outputs (RRD strategies) yields an even more expressive class — and incomparable to what is obtained by allowing randomness in updates only. Finally, the most expressive class is obviously obtained when allowing randomness in all components; yet it may be dropped in initialisation or in outputs without reducing the expressiveness — but not in both simultaneously.

To compare the expressiveness of strategy classes, we consider *outcome-equivalence*, as defined in Section 2. Intuitively, two strategies are outcome-equivalent if, against any strategy of the opponent, they yield identical probability distributions (i.e., they induce identical Markov chains). Hence we are agnostic with regard to the objective, winning condition, payoff function, or preference relation of the game, and with regard to how they are defined (e.g., colours on actions, states, transitions, etc).

Finally, let us note that in our setting of two-player concurrent stochastic games, the perfect recall hypothesis holds. Most importantly, we assume that actions are visible. Lifting this hypothesis drastically changes the relationships between the different models. While our main presentation considers two-player perfect-information games for the sake of simplicity, we show in Section 6 that *our results hold in games with more than two players* and, in Section 7, that *our results hold in games of imperfect information* too, assuming visible actions. We provide an adapted taxonomy for *games in which actions are not visible* in Section 7.

Related work. We discuss several axes of research related to our work.

The first one deals with the *various types of randomness* one can inject in strategies and their consequences. Obviously, Kuhn's theorem [22] is a major inspiration, as well as the examples of differences between strategy models presented in [20]. On a different but related note, [29] studies when randomness is not helpful in games nor strategies (as it can be simulated by other means or does not intervene).

A second direction focuses on trying to characterise the *power of finite-memory strategies*, with or without randomness. One can notably cite [3] for memoryless strategies, and [30, 24], [25], and [31] for finite-memory ones in deterministic, stochastic, and infinite-arena games respectively.

The power of strategies also depends on the information they are allowed to register to update their memory: colours, as in the papers of the previous paragraph, or the sequence of states [32, 33, 34], observations [35] or sequences of actions or labels [33].

The last axis concentrates on the use of *randomness as a means to simplify strategies* and/or reduce their memory requirements. Examples of this endeavour can be found in [36, 37, 38, 14, 39]. These are further motivations to understand randomised strategies even in contexts where randomness is not needed a priori to play optimally.

Outline. Section 2 summarises all preliminary notions. In Section 3, we present the taxonomy illustrated in Figure 1.1 and comment on it. We divide its proofs into two sections: Section 4 establishes the inclusions, and Section 5 establishes the separation of distinct strategy classes. Finally, Sections 6 and 7 present how we transfer our results to the richer settings of multi-player games and of games of imperfect information respectively. We conclude in Section 8. Appendix A is a technical appendix dedicated to the details of an equation introduced in Section 2.

A preliminary version of this work has been previously published as a conference paper [40]. This version presents in detail the contributions of the conference paper with full proofs and extends the results of the conference paper by considering a broader class of games; only turn-based games are considered in [40], whereas we consider *concurrent games* here. The separation of strategy classes presented in Section 5 has been enriched with examples derived from specifications, to complement the examples provided on a one-player games with one state and two actions (this is arguably the simplest possible setting in which we can consider non-trivial strategies). Finally, the generalisation to games of imperfect information presented in Section 7 has been extended to consider the case of games with imperfect recall.

2 Preliminaries

Set-theoretic notation. We let $\mathbb{N}$ and $\mathbb{Q}$ denote the sets of natural and rational numbers respectively. Given sets A and $B' \subseteq B$, and a function $f\colon A \to B$, we let $f^{-1}(B') = \{a \in A \mid f(a) \in B'\}$ denote the inverse image of B' by f. For the inverse image of singleton sets, we write $f^{-1}(b)$ instead of $f^{-1}(\{b\})$ for any $b \in B$.

Probability. Given any countable set A, we write $\mathcal{D}(A)$ for the set of probability distributions over A, i.e., the set of functions $\mu\colon A \to [0,1]$ such that $\sum_{a\in A}\mu(a) = 1$. Given such a probability distribution μ, we let $\mathsf{supp}(\mu) = \{a \in A \mid \mu(a) > 0\}$ be the support of μ.

Given a set A and a σ-algebra $\mathcal{F}$ over A, we denote by $\mathcal{D}(A, \mathcal{F})$ the set of probability distributions over the measurable space $(A, \mathcal{F})$.

Games. We consider two-player concurrent stochastic games of perfect information played on graphs. We denote the two players by $\mathcal{P}_1$ and $\mathcal{P}_2$. At the start of a play, a pebble is placed on some initial state. In each round, both players simultaneously select an action available in said state and the next state is chosen randomly following a distribution depending on the current state and the actions chosen by the players. The game proceeds for an infinite number of rounds, yielding an infinite play.

Formally, a two-player *concurrent stochastic game of perfect information*, or simply a *game*, is a tuple $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ where S is a non-empty finite set of states, $A^{(1)}$ and $A^{(2)}$ are finite sets of actions for each player and $\delta\colon S \times A^{(1)} \times A^{(2)} \to \mathcal{D}(S)$ is a (partial) probabilistic transition function. We write $\bar{A} = A^{(1)} \times A^{(2)}$ in the following. Elements of $\bar{A}$ are denoted with a bar to emphasise that they are pairs of actions. Given $\bar{a} \in \bar{A}$, we adopt the convention that $\bar{a}$ is given by the pair $(a^{(1)}, a^{(2)})$.

For any state $s \in S$, we let $\bar{A}(s) = \{\bar{a} \in \bar{A} \mid \delta(s, \bar{a}) \text{ is defined}\}$ and require that $\bar{A}(s)$ is of the form $A^{(1)}(s) \times A^{(2)}(s)$ for some subsets $A^{(i)}(s)$ of $A^{(i)}$, i.e., the actions available to a player in a state are not constrained by the choices of the other. We assume that for all $s \in S$, $\bar{A}(s)$ is non-empty, i.e., there are no deadlocks in the game.

A *play* of $\mathcal{G}$ is an infinite sequence $s_0\bar{a}_0s_1 \ldots \in (S\bar{A})^\omega$ such that for all $k \in \mathbb{N}$, $\delta(s_k, \bar{a}_k)(s_{k+1}) > 0$. A *history* is a finite prefix of a play ending in a state. Given a play $\pi = s_0\bar{a}_0s_1\bar{a}_1 \ldots$ and $k \in \mathbb{N}$, we write $\pi_{|k}$ for the history $s_0\bar{a}_0 \ldots \bar{a}_{k-1}s_k$. For any history $h = s_0\bar{a}_0 \ldots \bar{a}_{k-1}s_k$, we let $\mathsf{last}(h) = s_k$. We write $\mathsf{Plays}(\mathcal{G})$ to denote the set of plays of $\mathcal{G}$, $\mathsf{Hist}(\mathcal{G})$ to denote the set of histories of $\mathcal{G}$. Given some initial state $s_{\mathsf{init}} \in S$, we write $\mathsf{Hist}(\mathcal{G}, s_{\mathsf{init}})$ for the set of histories starting in state s_{init}.

There exist several classes of games that have been studied in their own right. A game is *turn-based* if at each round, only one player can influence the next transition. In other words, $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ is turn-based if for all states $s \in S$, there exists $i \in \{1, 2\}$ such that $|A^{(i)}(s)| = 1$ (in which case $\mathcal{P}_{3-i}$ controls s). Turn-based games are traditionally described via a partition of the state space into states controlled by $\mathcal{P}_1$ and states controlled by $\mathcal{P}_2$. A game is *deterministic* if its transitions are not subject to randomness; a game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ is a deterministic game if for all $s \in S$ and $\bar{a} \in \bar{A}(s)$, $\delta(s, \bar{a})$ is a Dirac distribution.

An interesting subclass of turn-based games is that of one-player games. A game is a *one-player game* if only one player controls all transitions. A game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ is a one-player game if there exists $i \in \{1, 2\}$ such that for all $s \in S$, $|A^{(i)}(s)| = 1$. A one-player game in the sense above is the equivalent of a *Markov decision process* (MDP) in our context, and will be referred to as such. When dealing with MDPs, we lighten notation and drop information related to the inactive player. We view MDPs as tuples (S, A, δ) where S is a finite set of states, A is a finite set of actions and $\delta\colon S \times A \to \mathcal{D}(S)$ is the transition function. Notions defined for two-player concurrent games can be directly adapted to MDPs, e.g., a play is a sequence in $(SA)^\omega$ instead of a sequence in $(S\bar{A})^\omega$.

We fix a game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ for the remainder of the section.

Strategies and outcomes. A strategy is a function that describes how a player should act based on a history. Players need not act in a deterministic fashion: they can use randomisation to select an action. Formally, a (behavioural) *strategy* of $\mathcal{P}_i$ is a function $\sigma_i\colon \mathsf{Hist}(\mathcal{G}) \to \mathcal{D}(A^{(i)})$ such that for all histories $h \in \mathsf{Hist}(\mathcal{G})$, $\mathsf{supp}(\sigma_i(h)) \subseteq A^{(i)}(\mathsf{last}(h))$. In other words, a strategy assigns to any history a distribution over the actions available to $\mathcal{P}_i$ in this state.

When both players fix a strategy and an initial state is decided, we obtain a purely stochastic process, i.e., a Markov chain. Let us recall the relevant σ-algebra for the definition of probabilities over plays. For any history $h \in \mathsf{Hist}(\mathcal{G})$, we define $\mathsf{Cyl}(h) = \{\pi \in \mathsf{Plays}(\mathcal{G}) \mid h \text{ is a prefix of } \pi\}$, the *cylinder* of h, consisting of plays that extend h. Let us denote by $\mathcal{F}_{\mathcal{G}}$ the σ-algebra generated by all cylinder sets.

Let σ_1 and σ_2 be strategies of $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively and $s_{\mathsf{init}} \in S$ be an initial state. We define the probability measure (over $(\mathsf{Plays}(\mathcal{G}), \mathcal{F}_{\mathcal{G}})$) induced by playing σ_1 and σ_2 from s_{init} in $\mathcal{G}$, written $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}$, as follows. For any history $h = s_0\bar{a}_0 \ldots s_n \in \mathsf{Hist}(\mathcal{G}, s_{\mathsf{init}})$, the probability $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h))$ assigned to $\mathsf{Cyl}(h)$ is given by the product

$$\prod_{k=0}^{n-1} \sigma_1(s_0\bar{a}_0 \ldots s_k)(a_k^{(1)}) \cdot \sigma_2(s_0\bar{a}_0 \ldots s_k)(a_k^{(2)}) \cdot \delta(s_k, \bar{a}_k)(s_{k+1}).$$

For any history $h \in \mathsf{Hist}(\mathcal{G}) \setminus \mathsf{Hist}(\mathcal{G}, s_{\mathsf{init}})$, we set $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) = 0$. By Carathéodory's extension theorem [41, Theorem A.1.3], the measure described above can be extended in a unique fashion to $(\mathsf{Plays}(\mathcal{G}), \mathcal{F}_{\mathcal{G}})$. For MDPs, we drop the strategy of the absent player in the notation of this distribution and write $\mathbb{P}^{\sigma_1}_{s_{\mathsf{init}}}$.

Let σ_i be a strategy of $\mathcal{P}_i$. A play or play prefix $s_0\bar{a}_0s_1\dots$ is said to be *consistent* with σ_i if for all action indices k, it holds that $\sigma_i(s_0\bar{a}_0\dots s_k)(a^{(i)}_k) > 0$.[1]

Outcome-equivalence of strategies. In later sections, we study the expressiveness of finite-memory strategy models depending on the type of randomisation allowed. Two strategies may yield the same outcomes despite being different: the actions suggested by a strategy in an inconsistent history can be changed without affecting which probability distributions are induced by the strategy. Therefore, instead of using the equality of strategies as a measure of equivalence, we consider some weaker notion of equivalence, referred to as outcome-equivalence.

We say that two strategies σ_1 and τ_1 of $\mathcal{P}_1$ are *outcome-equivalent* if for any strategy σ_2 of $\mathcal{P}_2$ and for any initial state s_{init}, the probability distributions $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}$ and $\mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}$ coincide.

We now provide the criterion used in our proofs to establish the outcome-equivalence of strategies. This criterion does not invoke the probability distributions induced by strategies directly. The idea is that when comparing two strategies of a player, we need only be concerned with the suggestions these strategies provide in histories that are consistent with them. In other words, any deviation in unreachable histories does not affect the outcome. Hence, one could reformulate outcome-equivalence as having to suggest the same distributions over actions in histories that are consistent with (one of) the strategies. In the sequel, we prove that this reformulation is indeed equivalent to the definition of outcome-equivalence. We rely on this reformulation to prove the outcome-equivalence of two strategies.

Lemma 2.1 (Strategic criterion for outcome-equivalence). *Let σ_i and τ_i be two strategies of $\mathcal{P}_i$. These two strategies are outcome-equivalent if and only if for all histories $h \in \mathsf{Hist}(\mathcal{G})$, h consistent with σ_i implies $\sigma_i(h) = \tau_i(h)$.*

Proof. To aid with notation, we assume that $i = 1$; the proof of the other case is done by exchanging the players below. First, we assume that σ_1 and τ_1 are outcome-equivalent. Let $h \in \mathsf{Hist}(\mathcal{G})$ be a history that is consistent with σ_1. Let s_{init} denote the first state of h and let σ_2 be a strategy of $\mathcal{P}_2$ consistent with h. Let $a^{(1)} \in A^{(1)}(s)$ and $a^{(2)} \in \mathsf{supp}(\sigma_2(h))$, and write $\bar{a} = (a^{(1)}, a^{(2)})$. Let $s \in \mathsf{supp}(\delta(\mathsf{last}(h)), \bar{a}))$. By definition of the probability of a cylinder set and consistency of h with both σ_1 and σ_2, we have

$$\sigma_1(h)(a^{(1)}) = \frac{\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h\bar{a}s))}{\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) \cdot \sigma_2(h)(a^{(2)}) \cdot \delta(\mathsf{last}(h), \bar{a})(s)}.$$

Furthermore, the outcome-equivalence of σ_1 and τ_1 implies that $\mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) = \mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) > 0$. Therefore, we have

$$\tau_1(h)(a^{(1)}) = \frac{\mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h\bar{a}s))}{\mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) \cdot \sigma_2(h)(a^{(2)}) \cdot \delta(\mathsf{last}(h), \bar{a})(s)}.$$

It follows from the equations above and the outcome-equivalence of σ_1 and τ_1 that $\sigma_1(h)(a^{(1)}) = \tau_1(h)(a^{(1)})$. We have shown that $\sigma_1(h) = \tau_1(h)$, which ends the proof of the first direction.

Let us now assume that σ_1 and τ_1 coincide over histories consistent with σ_1. Let σ_2 be a strategy of $\mathcal{P}_2$ and $s_{\mathsf{init}} \in S$ be an initial state. It suffices to study the probability of cylinder sets. Let $h \in \mathsf{Hist}(\mathcal{G})$ be a history starting in s_{init}. If h is consistent with σ_1, then all prefixes of h also are, therefore the definition of the probability of a cylinder ensures that $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) = \mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h))$. Otherwise, if h is not consistent with σ_1, then h is necessarily of the form $h'\bar{a}h''$ with h' consistent with σ_1 and $\sigma_1(h')(a^{(1)}) = 0$. It follows that $\tau_1(h')(a^{(1)}) = 0$, thus $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(h) = \mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}(h) = 0$. This shows that σ_1 and τ_1 are outcome-equivalent, ending the proof. □

Subclasses of strategies. A strategy is called *pure* if it does not use randomisation; a pure strategy of $\mathcal{P}_i$ can be viewed as a function $\mathsf{Hist}(\mathcal{G}) \to A^{(i)}$. A strategy that only uses information on the current state of the play is called *memoryless*: a strategy σ_i of $\mathcal{P}_i$ is memoryless if for all histories $h, h' \in \mathsf{Hist}(\mathcal{G})$, $\mathsf{last}(h) = \mathsf{last}(h')$ implies $\sigma_i(h) = \sigma_i(h')$. Memoryless strategies can be viewed as functions $S \to \mathcal{D}(A^{(i)})$. Strategies that are both memoryless and pure can be viewed as functions $S \to A^{(i)}$.

[1] We use the terminology of consistency not only for plays and histories, but also for prefixes of plays that end with an action pair.

A strategy is said to be *finite-memory* (FM) if it can be encoded by a Mealy machine, i.e., an automaton with outputs along its edges. We can include randomisation in the initialisation, outputs and updates (i.e., transitions) of the Mealy machine. Formally, a *stochastic Mealy machine* of $\mathcal{P}_i$ is a tuple $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$, where M is a finite set of memory states, $\mu_{\mathsf{init}} \in \mathcal{D}(M)$ is an initial distribution, $\alpha_{\mathsf{nxt}} \colon M \times S \to \mathcal{D}(A^{(i)})$ is the (stochastic) next-move function and $\alpha_{\mathsf{up}} \colon M \times S \times A^{(i)} \to \mathcal{D}(M)$ is the (stochastic) update function.

Before we explain how to define the strategy induced by a Mealy machine, let us first describe how these machines work. Fix a Mealy machine $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$. Let $s_0 \in S$. At the start of a play, an initial memory state m_0 is selected randomly following μ_{init}. Then, at each step of the play, the action $a_k^{(i)}$ of $\mathcal{P}_i$ is chosen following the distribution $\alpha_{\mathsf{nxt}}(m_k, s_k)$, whereas the action $a^{(3-i)}$ is independently chosen according to the strategy of $\mathcal{P}_{3-i}$. The memory state m_{k+1} is then randomly updated following the distribution $\alpha_{\mathsf{up}}(m_k, s_k, \bar{a}_k)$ and the game state s_{k+1} is chosen following the distribution $\delta(s_k, \bar{a}_k)$, both choices being made independently.

Let us now explain how a strategy can be derived from a Mealy machine. As explained previously, when in a certain memory state $m \in M$ and game state $s \in S$, the probability of an action $a^{(i)} \in A^{(i)}(s)$ being chosen is given by $\alpha_{\mathsf{nxt}}(m, s)(a^{(i)})$. Therefore, the probability of choosing the action $a^{(i)} \in A^{(i)}$ after some history $h = ws$ (where $w \in (S\bar{A})^*$ and $s = \mathsf{last}(h)$) is given by the sum, for each memory state $m \in M$, of the probability that m was reached after w has taken place (i.e., after $\mathcal{M}$ processes w), multiplied by $\alpha_{\mathsf{nxt}}(m, s)(a^{(i)})$.

To provide a formal definition of the strategy induced by $\mathcal{M}$, we first describe the distribution over memory states of $\mathcal{M}$ after elements of $(S\bar{A})^*$ take place (under the strategy induced by $\mathcal{M}$). We formally define this distribution inductively. Details for the derivation of the inductive formula, which rely on conditional probabilities, are deferred to Appendix A.

The distribution μ_ε over memory states after the empty word ε (i.e., nothing) has taken place is by definition μ_{init}. Assume inductively that we know the distribution μ_w for $w = s_0\bar{a}_0 \dots s_{k-1}\bar{a}_{k-1}$. We explain how to derive $\mu_{ws_k\bar{a}_k}$ from μ_w for any state $s_k \in \mathsf{supp}(\delta(s_{k-1}, \bar{a}_{k-1}))$ and for any pair of actions $\bar{a}_k \in \bar{A}(s_k)$.

In general, the choice of an action by $\mathcal{P}_i$ conditions what the predecessor memory states could be. First, we note that if $\alpha_{\mathsf{nxt}}(m', s_k)(a_k^{(i)}) = 0$ holds for all memory states $m' \in \mathsf{supp}(\mu_w)$, then the action $a_k^{(i)}$ is actually never chosen. We leave this case undefined (the related conditional probabilities are ill-defined) and assume $a_k^{(i)} \in \mathsf{supp}(\alpha_{\mathsf{nxt}}(m', s_k))$ for some $m' \in \mathsf{supp}(\mu_w)$. The equation for $\mu_{ws_k\bar{a}_k}$ uses the likelihood of being in a memory state knowing that the action $a_k^{(i)}$ was chosen, and not μ_w directly. We have, for any memory state $m \in M$,

$$\mu_{ws_k\bar{a}_k}(m) = \frac{\sum_{m' \in M} \mu_w(m') \cdot \alpha_{\mathsf{up}}(m', s_k, \bar{a}_k)(m) \cdot \alpha_{\mathsf{nxt}}(m', s_k)(a_k^{(i)})}{\sum_{m' \in M} \mu_w(m') \cdot \alpha_{\mathsf{nxt}}(m', s_k)(a_k^{(i)})}.$$

We remark that this quotient is not well-defined whenever for all $m' \in \mathsf{supp}(\mu_w)$, $\alpha_{\mathsf{nxt}}(m', s_k)(a_k^{(i)}) = 0$, further justifying the distinction above.

Using these distributions, we formally define the (partial) strategy $\sigma_i^{\mathcal{M}}$ induced by the Mealy machine $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ as the strategy $\sigma_i^{\mathcal{M}} \colon \mathsf{Hist}(\mathcal{G}) \to \mathcal{D}(A^{(i)})$ such that for all histories $h = ws$, for all actions $a^{(i)} \in A^{(i)}(s)$,

$$\sigma_i^{\mathcal{M}}(h)(a^{(i)}) = \sum_{m \in M} \mu_w(m) \cdot \alpha_{\mathsf{nxt}}(m, s)(a^{(i)}).$$

This strategy is only partially defined because distributions μ_w are not defined for all $w \in (S\bar{A})^*$. All histories for which $\sigma_i^{\mathcal{M}}$ is undefined can be shown to be of the form $h\bar{a}h'$ such that $\sigma_i^{\mathcal{M}}$ is defined for h and $\sigma_i^{\mathcal{M}}(h)(a^{(i)}) = 0$. Therefore, no matter how the partial definition of $\sigma_i^{\mathcal{M}}$ given above is extended, it does not influence the induced probability distribution over plays involving this strategy.

Classifying finite-memory strategies. In the sequel, we investigate the relationships between different classes of finite-memory strategies with respect to expressive power. We classify finite-memory strategies following the type of stochastic Mealy machines that can induce them. We introduce a concise notation for each class: we use three-letter acronyms of the form XXX with X ∈ {D,R}, where the letters, in order, refer to the initialisation, outputs and updates of the Mealy machines, with D and R respectively denoting deterministic and randomised components. For instance, we will write RRD to denote the class of Mealy machines that have randomised initialisation and outputs, but deterministic updates. We also apply this terminology to FM strategies: we will say that an FM strategy is in the class XXX — i.e., it is an XXX strategy — if it is induced by an XXX Mealy machine.

Moreover, in the remainder of the paper, we will abusively identify Mealy machines and their induced FM strategies. For instance, we will say that $\mathcal{M}$ is an XXX strategy to mean that $\mathcal{M}$ is an XXX Mealy machine (thus inducing an XXX strategy). As a by-product of this identification, we apply the terminology introduced previously for strategies to Mealy machines, without explicitly referring to the strategy they induce. For instance, we may say a history is consistent with some Mealy machine, or that two Mealy machines are outcome-equivalent. Let us note however that we will not use a Mealy machine in lieu of its induced strategy whenever we are interested in the strategy itself as a function. This choice lightens notations; the strategy induced by a Mealy machine need not be introduced unless it is required as a function.

We close this section by commenting on some of the classes, and discuss previous appearances in the literature, under different names. Pure strategies use no randomisation: hence, the class DDD corresponds to pure FM strategies, which can be represented by Mealy machines that do not rely on randomisation.

Strategies in the class DRD have been referred to as *behavioural* FM strategies in [20]. The name comes from the randomised outputs, reminiscent of behavioural strategies that output a distribution over actions after a history. We note that stochastic Mealy machines that induce DRD strategies are such that their distributions over memory states are Dirac due to the deterministic initialisation and updates.

Similarly, RDD strategies have been referred to as *mixed* FM strategies [20]. The general definition of a mixed strategy is a distribution over pure strategies: under a mixed strategy, a player randomly selects a pure strategy at the start of a play and plays according to it for the whole play. RDD strategies are similar in the way that the random initialisation can be viewed as randomly selecting some DDD strategy (i.e., a pure FM strategy) among a *finite* selection of such strategies.

The elements of RRR, the broadest class of FM strategies, have been referred to as general FM strategies [20] and stochastic-update FM strategies [42, 43]. The latter name highlights the random nature of updates and insists on the difference with models that rely on deterministic updates, more common in the literature.

3 Taxonomy of finite-memory strategies

In this section, we comment on the relationships between the classes of finite-memory strategies in terms of expressiveness. We say that a class $\mathcal{C}_1$ of FM strategies is no less expressive than a class $\mathcal{C}_2$ if for all games $\mathcal{G}$, for all FM strategies $\mathcal{M} \in \mathcal{C}_2$ in $\mathcal{G}$, one can find some FM strategy $\mathcal{M}' \in \mathcal{C}_1$ of $\mathcal{G}$ such that $\mathcal{M}$ and $\mathcal{M}'$ are outcome-equivalent strategies. For the sake of brevity, we will say that $\mathcal{C}_2$ is included in $\mathcal{C}_1$, and write $\mathcal{C}_2 \subseteq \mathcal{C}_1$.

Figure 1.1 summarises our results. Each line representing an inclusion is decorated with a reference to the relevant results. The strictness results hold in one-player deterministic games. In particular, there are no collapses in the diagram in the turn-based setting, which subsumes two-player deterministic turn-based games and Markov decision processes.

Some inclusions follow directly from some classes having more randomisation power than others: a deterministic component can be emulated using Dirac distributions. For instance, the inclusion DRD $\subseteq$ RRD follows from the fact that RRD Mealy machines have both randomised initialisation and outputs whereas DRD ones only have randomised outputs. The inclusions RDD $\subseteq$ DRD, RRR $\subseteq$ DRR and RRR $\subseteq$ RDR, which do not follow from such arguments, are covered in Section 4.

Pure strategies are strictly less expressive than any other class of FM strategies; pure strategies cannot induce any non-Dirac distributions on plays in deterministic one-player games. Other arguments for the separation of classes of strategies are provided in Section 5.

We close this section by comparing our results with Kuhn's theorem. Kuhn's theorem asserts that the classes of behavioural strategies and mixed strategies in games of perfect recall share the same expressiveness. Games of perfect recall have two traits: players never forget the sequence of histories controlled by them that have taken place and they can see their own actions. In particular, stochastic games of perfect information are a special case of games of perfect recall. Recall that mixed strategies are distributions over pure strategies. We comment briefly on the techniques used in the proof of Kuhn's theorem, and compare them with the finite-memory setting. Let us fix a game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$.

On the one hand, the emulation of mixed strategies with behavioural strategies is performed as follows. Let p_i be a mixed strategy of $\mathcal{P}_i$, i.e., a distribution over pure strategies of $\mathcal{G}$. An outcome-equivalent behavioural strategy σ_i is constructed such that, for all histories $h \in \mathsf{Hist}(\mathcal{G})$ and actions $a^{(i)} \in A^{(i)}(\mathsf{last}(h))$, the probability $\sigma_i(h)(a^{(i)})$ is defined as

$$\frac{p_i(\{\tau_i \text{ pure strategy} \mid \tau_i \text{ consistent with } h \text{ and } \tau_i(h) = a^{(i)}\})}{p_i(\{\tau_i \text{ pure strategy} \mid \tau_i \text{ consistent with } h\})}.$$

In the finite-memory case, similar ideas can be used to show that RDD $\subseteq$ DRD. In the proof of Theorem 4.1, from some RDD strategy (i.e., a so-called mixed FM strategy), we construct a DRD strategy (i.e., a so-called behavioural FM strategy) that keeps track of the finitely many pure FM strategies that the RDD strategy mixes and that are consistent with the current history. An adaption of the quotient above is used in the next-move function of the DRD strategy.

On the other hand, the emulation of behavioural strategies by mixed strategies exploits the fact that mixed strategies may randomise over *infinite* sets. In a finite-memory setting, the same techniques cannot be applied. As a consequence, the class of RDD strategies is strictly included in the class of DRD strategies. In a certain sense, one could say that Kuhn's theorem only partially holds in the case of FM strategies.

4 Non-trivial inclusions

This section covers the non-trivial inclusions that are asserted in the lattice of Figure 1.1. The structure of this section is as follows. Section 4.1 covers the inclusion RDD $\subseteq$ DRD. The inclusion RRR $\subseteq$ DRR is presented in Section 4.2. Finally, we close this section by proving the inclusion RRR $\subseteq$ RDR in Section 4.3.

4.1 Simulating RDD strategies with DRD ones

In this section, we focus on the classes RDD and DRD. We prove that for all RDD strategies in any game, one can find some outcome-equivalent DRD strategy (Theorem 4.1). Let us note that the converse inclusion is not true, and this discussion is relegated to Section 5.2. The construction provided in the proof of Theorem 4.1 yields a DRD strategy that has a state space of size exponential in the size of the state space of the original RDD strategy. We complement Theorem 4.1 by proving that there are some RDD strategies for which this exponential blow-up in the number of states is necessary for any outcome-equivalent DRD strategy (Lemma 4.1). We show that this blow-up is unavoidable in both deterministic turn-based two-player games and MDPs.

Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game. Fix an RDD strategy $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ of $\mathcal{P}_i$. Let us sketch how to emulate $\mathcal{M}$ with a DRD strategy $\mathcal{N} = (N, n_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ built with a subset construction-like approach. The memory states of $\mathcal{N}$ are functions $f\colon \mathsf{supp}(\mu_{\mathsf{init}}) \to M \cup \{\bot\}$. A memory state f is interpreted as follows. For all initial memory states $m_0 \in \mathsf{supp}(\mu_{\mathsf{init}})$, we have $f(m_0) = \bot$ if the history seen up to now is not consistent with the pure FM strategy $(M, m_0, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$, and otherwise $f(m_0)$ is the memory state reached in the same pure FM strategy after processing the current history. Updates are naturally derived from these semantics.

Using this state space and update scheme, we can compute the likelihood of each memory state of the mixed FM strategy $\mathcal{M}$ after some sequence $w \in (S\bar{A})^*$ has taken place. Indeed, we keep track of each initial memory state from which it was possible to be consistent with w, and, for each such initial memory state m_0, the memory state reached after w was processed starting in m_0. Therefore, this likelihood can be inferred from μ_{init}; the probability of $\mathcal{M}$ being in $m \in M$ after w has been processed is given by the (normalised) sum of the probability of each initial memory state $m_0 \in \mathsf{supp}(\mu_{\mathsf{init}})$ such that $f(m_0) = m$.

The definition of the next-move function of $\mathcal{N}$ is directly based on the distribution over states of $\mathcal{M}$ described in the previous paragraph, and ensures that the two strategies select actions with the same probabilities at any given state. For any action $a^{(i)} \in A^{(i)}(s)$, the probability of $a^{(i)}$ being chosen in game state s and in memory state f is determined by the probability of $\mathcal{M}$ being in some memory state m such that $\alpha_{\mathsf{nxt}}(m, s) = a^{(i)}$, where this probability is inferred from f.

Intuitively, we postpone the initial randomisation and instead randomise at each step in an attempt of replicating the initial distribution in the long run. In the sequel, we formalise the DRD strategy outlined above and prove its outcome-equivalence with the RDD strategy it is based on.

Theorem 4.1. *Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game. Let $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be an RDD strategy of $\mathcal{P}_i$. There exists a DRD strategy $\mathcal{N} = (N, n_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ such that $\mathcal{N}$ and $\mathcal{M}$ are outcome-equivalent.*

Proof. We formalise the strategy described above. Let us write M_0 for the support of the initial distribution μ_{init} of $\mathcal{M}$. We define the set of memory states N to be the set of functions $M_0 \to M \cup \{\bot\}$. The initial memory state of $\mathcal{N}$ is given by the identity function $n_{\mathsf{init}}\colon m_0 \mapsto m_0$ over M_0. The update function

β_{up} is as follows. For any $f \in N$, any $s \in S$ and $\bar{a} \in \bar{A}(s)$, we let $\beta_{\mathsf{up}}(f, s, \bar{a})$ be the function f' such that for all $m_0 \in M_0$, we have

$$f'(m_0) = \begin{cases} \alpha_{\mathsf{up}}(f(m_0), s, \bar{a}) & \text{if } f(m_0) \in M \text{ and } \alpha_{\mathsf{nxt}}(f(m_0), s) = a^{(i)} \\ \perp & \text{otherwise.} \end{cases}$$

Whenever we perform an update of the memory, we refine our knowledge on what the initial memory state could have been according to the actions selected by $\mathcal{P}_i$ prior to the update. This refinement proceeds by mapping to $\perp$ any initial memory states m_0 such that the played action would not have been selected in the memory state $f(m_0) \in M$, effectively removing m_0 from the set of initial memory states from which we could have started.

The next-move function β_{nxt} is defined as follows: for any memory state $f \in N$ and $s \in S$, we let $\beta_{\mathsf{nxt}}(f, s)$ be arbitrary if f maps $\perp$ to all memory states, and otherwise $\beta_{\mathsf{nxt}}(f, s)$ is the distribution over $A^{(i)}$ such that, for all $a^{(i)} \in A^{(i)}(s)$, we have

$$\beta_{\mathsf{nxt}}(f, s)(a^{(i)}) = \sum_{\substack{m_0 \in M_0 \\ \alpha_{\mathsf{nxt}}(f(m_0), s) = a^{(i)}}} \frac{\mu_{\mathsf{init}}(m_0)}{\sum_{m_0' \in f^{-1}(M)} \mu_{\mathsf{init}}(m_0')}.$$

We note that the memory state $f \in N$ mapping $\perp$ to all initial memory states is only reached whenever a history inconsistent with $\mathcal{M}$ has taken place under $\mathcal{M}$. Thanks to Lemma 2.1, we need not take in account histories inconsistent with $\mathcal{M}$ to establish the outcome-equivalence of $\mathcal{M}$ and $\mathcal{N}$. This explains why the next-move function is left arbitrary in that case.

We now show that $\mathcal{M}$ and $\mathcal{N}$ are outcome-equivalent via Lemma 2.1. To this end, we first show a relation, for each $w \in (S\bar{A})^*$ consistent with $\mathcal{M}$, between the distribution $\mu_w \in \mathcal{D}(M)$ over the memory states of $\mathcal{M}$ after processing w and the function f_w reached after $\mathcal{N}$ reads w (recall that for a DRD strategy, the distribution over its states after processing w is a Dirac distribution). Formally, this relation is as follows: for any $w \in (S\bar{A})^*$ consistent with $\mathcal{M}$ and any memory state $m \in M$, we have

$$\mu_w(m) = \frac{\sum_{m_0 \in f_w^{-1}(m)} \mu_{\mathsf{init}}(m_0)}{\sum_{m_0 \in f_w^{-1}(M)} \mu_{\mathsf{init}}(m_0)}. \tag{4.1}$$

In the above, $f_w^{-1}(M)$ is the set of potential initial $m_0 \in M_0$ of $\mathcal{M}$ that are compatible with w taking place. This equation intuitively expresses that $\mathcal{N}$ accurately keeps track of the current distribution over memory states of $\mathcal{M}$ along a play. A corollary of the above is that whenever we follow histories consistent with $\mathcal{M}$, we are assured to never reach the memory state of $\mathcal{N}$ that assigns $\perp$ to all states in M_0.

We prove Equation (4.1) with an inductive argument. The case of $w = \varepsilon$ is trivial: by definition $\mu_\varepsilon = \mu_{\mathsf{init}}$ and f_ε is the identity function over M_0. Now, let us assume that Equation (4.1) holds for $w' \in (S\bar{A})^*$ consistent with $\mathcal{M}$, and let us prove it for $w = w's\bar{a}$ consistent with $\mathcal{M}$.

When writing relations between $\mu_{w'}$ and μ_w in the remainder of the proof, we adopt notation slightly different to Section 2. In this case, the update function α_{up} and next-move α_{nxt} of $\mathcal{M}$ are deterministic. Thus, instead considering sums weighted by Dirac distributions, we only sum over relevant states for clarity.

First, we remark that it may be the case that $f_w^{-1}(M) \neq f_{w'}^{-1}(M)$. In light of this, we must take care not to have $f_w^{-1}(M) = \emptyset$, in which case the denominator of the right-hand side of Equation (4.1) evaluates to zero. From the definition of β_{up}, it follows that $f_w^{-1}(M)$ is formed of the memory elements $m_0 \in f_{w'}^{-1}(M)$ such that $\alpha_{\mathsf{nxt}}(f_{w'}(m_0), s) = a^{(i)}$. We know that $w = w's\bar{a}$ is consistent with $\mathcal{M}$. This implies there is some $m \in M$ such that $\alpha_{\mathsf{nxt}}(m, s) = a^{(i)}$ and $\mu_{w'}(m) > 0$. From the inductive hypothesis (Equation (4.1) with w'), we obtain that there is some $m_0 \in f_{w'}^{-1}(M)$ such that $f_{w'}(m_0) = m$, otherwise the right-hand side of the equation would evaluate to zero. The equality $f_{w'}(m_0) = m$ implies $m_0 \in f_w^{-1}(M)$, thus we have shown that $M_0(m)$ is non-empty.

Now that we have shown that Equation (4.1) is well-defined for w, we move on to its proof. Let us write $\alpha_{\mathsf{nxt}}(\cdot, s)^{-1}(a^{(i)})$ for the set $\{m \in M \mid \alpha_{\mathsf{nxt}}(m, s) = a^{(i)}\}$. By definition, we have

$$\mu_w(m) = \frac{\sum_{\substack{m' \in \alpha_{\mathsf{nxt}}(\cdot, s)^{-1}(a^{(i)}) \\ \alpha_{\mathsf{up}}(m', s, \bar{a}) = m}} \mu_{w'}(m')}{\sum_{m' \in \alpha_{\mathsf{nxt}}(\cdot, s)^{-1}(a^{(i)})} \mu_{w'}(m')}.$$

For the numerator, we obtain from the inductive hypothesis that

$$\sum_{\substack{m' \in \alpha_{\mathsf{nxt}}(\cdot,s)^{-1}(a^{(i)})\\ \alpha_{\mathsf{up}}(m',s,\bar{a})=m}} \mu_{w'}(m') = \sum_{\substack{m' \in \alpha_{\mathsf{nxt}}(\cdot,s)^{-1}(a^{(i)})\\ \alpha_{\mathsf{up}}(m',s,\bar{a})=m}} \sum_{m_0 \in f_{w'}^{-1}(m')} \frac{\mu_{\mathsf{init}}(m_0)}{\sum_{m_0' \in f_{w'}^{-1}(M)} \mu_{\mathsf{init}}(m_0')}$$

$$= \sum_{m_0 \in f_w^{-1}(m)} \frac{\mu_{\mathsf{init}}(m_0)}{\sum_{m_0' \in f_{w'}^{-1}(M)} \mu_{\mathsf{init}}(m_0')}.$$

To derive the simple sum from the double sum, we rely on the fact that $f_w(m_0) = m$ holds if and only if $\alpha_{\mathsf{up}}(f_{w'}(m_0), s, \bar{a}) = m$ and $\alpha_{\mathsf{nxt}}(f_{w'}(m_0), s) = a^{(i)}$, by definition of β_{up}.

For the denominator, we obtain from the inductive hypothesis,

$$\sum_{m' \in \alpha_{\mathsf{nxt}}(\cdot,s)^{-1}(a^{(i)})} \mu_{w'}(m') = \sum_{m' \in \alpha_{\mathsf{nxt}}(\cdot,s)^{-1}(a^{(i)})} \sum_{m_0 \in f_{w'}^{-1}(m')} \frac{\mu_{\mathsf{init}}(m_0)}{\sum_{m_0' \in f_{w'}^{-1}(M)} \mu_{\mathsf{init}}(m_0')}$$

$$= \sum_{m_0 \in f_w^{-1}(M)} \frac{\mu_{\mathsf{init}}(m_0)}{\sum_{m_0' \in f_{w'}^{-1}(M)} \mu_{\mathsf{init}}(m_0')}.$$

The last equality is a consequence of the definition of β_{up}: recall that $f_w^{-1}(M)$ consists of the elements m_0 of $M_0(w')$ such that $\alpha_{\mathsf{nxt}}(f_{w'}(m_0), s) = a^{(i)}$. By combining the two equations above, we immediately obtain Equation (4.1), ending the inductive argument.

We now establish the outcome-equivalence of $\mathcal{M}$ and $\mathcal{N}$. Let $h = ws \in \mathsf{Hist}(\mathcal{G})$ be a history of $\mathcal{G}$ consistent with $\mathcal{M}$. Let $a^{(i)} \in A^{(i)}(s)$ be an action enabled in s. The probability of $a^{(i)}$ being played after h under $\mathcal{M}$ is given by the weighted sum $\sum_{m \in \alpha_{\mathsf{nxt}}(\cdot,s)^{-1}(a^{(i)})} \mu_w(m)$. Under $\mathcal{N}$, the probability of $a^{(i)}$ being played is $\beta_{\mathsf{nxt}}(f_w, s)(a^{(i)})$. It follows from Equation (4.1) that these two probabilities coincide. We have shown the outcome-equivalence of strategies $\mathcal{M}$ and $\mathcal{N}$, ending the proof. □

The construction of a DRD strategy provided in the proof of Theorem 4.1 leads to an exponential blow-up of the memory state space. For an RDD strategy $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$, we have constructed an outcome-equivalent DRD strategy with a state space consisting of functions $\mathsf{supp}(\mu_{\mathsf{init}}) \to M \cup \{\bot\}$, therefore with a state space of size $(|M|+1)^{|\mathsf{supp}(\mu_{\mathsf{init}})|}$. In the upcoming lemma, we state that an exponential blow-up in the number of initial memory states cannot be avoided in general, even in the turn-based setting.

Lemma 4.1. *Let $k \in \mathbb{N}_0$. There exist a two-player turn-based deterministic game (respectively an MDP) $\mathcal{G}_k$ with $k+2$ states, $4k+2$ transitions, $k+2$ actions, and an RDD strategy $\mathcal{M}_k$ of $\mathcal{P}_1$ with k states such that any outcome-equivalent DRD strategy must have at least $2^k - 1$ states.*

Proof. We construct a two-player turn-based deterministic game $\mathcal{G}_k = (S_k, A_k^{(1)}, A_k^{(2)}, \delta_k)$ as follows. We let $S_k = \{s_j \mid 1 \leq j \leq k\} \cup \{t, s^\star\}$. The sets of actions, common to the two players, are $A_k := A_k^{(1)} = A_k^{(2)} = \{a_i \mid 1 \leq i \leq k\} \cup \{b, \bot\}$. All states besides t are controlled by $\mathcal{P}_1$ in the following sense. For all $1 \leq j \leq k$, we let $A_k^{(1)}(s_j) = \{a_j, b\}$ and $A_k^{(2)}(s_j) = \{\bot\}$. Next, we let $A_k^{(1)}(s^\star) = \{a_j \mid 1 \leq j \leq k\}$ and $A_k^{(2)}(s^\star) = \{\bot\}$. Finally, for state t, we have $A_k^{(1)}(t) = \{\bot\}$ and $A_k^{(1)}(t) = A_k \setminus \{\bot\}$.

We define the deterministic transition function δ_k as a function $S_k \times \bar{A}_k \to S_k$ (instead of dealing with Dirac distributions over successor states). For each $j \in \{1, \dots, k\}$, all transitions from s_j move back to t, i.e., $\delta_k(s_j, a_j, \bot) = \delta_k(s_j, b, \bot) = t$. In state t, we set for all $j \in \{1, \dots, k\}$, $\delta_k(t, \bot, a_j) = s_j$ and $\delta_k(t, \bot, b) = s^\star$. In state $s^\star$, for all $j \in \{1, \dots, k\}$, the action a_j labels a self-loop, i.e., we have $\delta_k(s^\star, a_j, \bot) = s^\star$. We illustrate the game $\mathcal{G}_3$ in Figure 4.1. We omit $\bot$ actions from edge labels to lighten the figure.

We define an RDD strategy $\mathcal{M}_k = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ of $\mathcal{P}_1$ as follows. We let $M = \{1, \dots, k\}$, and μ_{init} is taken to be the uniform distribution over M. The memory update function is taken to be trivial: we set $\alpha_{\mathsf{up}}(m, s, \bar{a}) = m$ for all $m \in M$, $s \in S_k$ and $\bar{a} \in \bar{A}_k$. For each memory state $m \in M$, we let $\alpha_{\mathsf{nxt}}(m, s_m) = \alpha_{\mathsf{nxt}}(m, s^\star) = a_m$ and, for all $j \neq m$, we let $\alpha_{\mathsf{nxt}}(m, s_j) = b$, and we let $\alpha_{\mathsf{nxt}}(m, t) = \bot$. In $\mathcal{M}$, once the initial state is decided, it no longer changes. In the memory state $m \in M$, the strategy prescribes action a_m in the states s_m and $s^\star$, and in states s_j with $j \neq m$, the strategy prescribes action b.

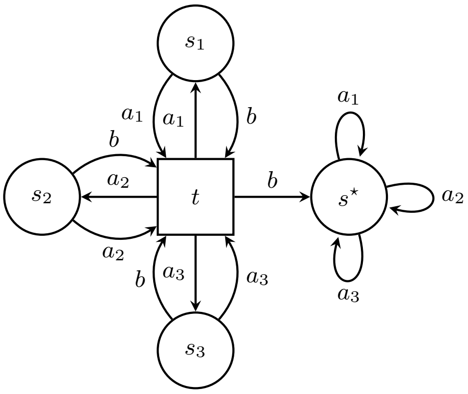


Fig. 4.1: The game $\mathcal{G}_3$ from the proof of Lemma 4.1. Circles and squares respectively represent states controlled by $\mathcal{P}_1$ and $\mathcal{P}_2$.

We now establish that all DRD strategies that are outcome-equivalent to $\mathcal{M}$ must have at least $2^k - 1$ memory states. Let $\mathcal{N} = (N, n_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ be one such FM strategy. We give a lower bound on $|N|$ by showing that there must be at least $2^k - 1$ distinct distributions of the form $\beta_{\mathsf{nxt}}(\cdot, s^\star)$.

Let $E = \{j_1, \ldots, j_\ell\} \subsetneq M$ be a proper subset of M. Consider the history ($\perp$ actions are omitted and parentheses are provided to improve readability) $h_E = (t\, a_{j_1}\, s_{j_1}\, b)(t\, a_{j_2}\, s_{j_2}\, b) \ldots (t\, a_{j_\ell}\, s_{j_\ell}\, b) t\, b\, s^\star$. Let $m \in E$. We see that along the history h_E, the action b is used in state s_m. Therefore, h_E is not consistent with the pure FM strategy $(M, m, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ derived from $\mathcal{M}$ by setting its initial state to m. Similarly, we see that for $m \notin E$, the history h_E is consistent with the pure FM strategy $(M, m, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$. Thus, the set of actions that can be played after h_E when following $\mathcal{M}_n$ is exactly the set $\{a_m \mid m \in M \setminus E\} \neq \emptyset$. Due to the deterministic initialisation and updates of DRD strategies, there must be some $n_E \in N$ such that $\mathsf{supp}(\beta_{\mathsf{nxt}}(n_E, s^\star)) = \{a_m \mid m \in M \setminus E\}$. Necessarily, we must have $\mathsf{supp}(\beta_{\mathsf{nxt}}(n_E, s^\star)) \neq \mathsf{supp}(\beta_{\mathsf{nxt}}(n_{E'}, s^\star))$ whenever $E \neq E'$, hence $n_E \neq n_{E'}$. Consequently, we must have at least one memory state in $\mathcal{N}$ per proper subset of M, i.e., $|N| \geq 2^k - 1$.

The proof of the existence of a suitable MDP remains. We explain how to adapt the deterministic game $\mathcal{G}_k$. To change $\mathcal{G}_k$ to a suitable MDP $\mathcal{G}'_k$, keep the same state space and remove all actions of $\mathcal{P}_2$. All transitions are left unchanged except the transitions from state t, which are altered as follows. When using $\perp$ in t, we let there be a uniform probability of reaching a state other than t in $\mathcal{G}'_k$. The only (formal) change to be made to $\mathcal{M}_k$ to obtain a suitable RDD strategy $\mathcal{M}'_k$ of $\mathcal{G}'_k$ is to remove the actions of $\mathcal{P}_2$ from updates.

By performing these changes, we can reuse the argument above for the two-player case to conclude that any DRD strategy that is outcome-equivalent to $\mathcal{M}'_k$ in $\mathcal{G}'_k$ requires at least $2^k - 1$ memory states. This concludes our explanation of how to adapt the game and strategy above to the context of MDPs. □

4.2 Simulating RRR strategies with DRR ones

In this section, we establish that DRR strategies are as expressive as RRR strategies, i.e., randomness in the initialisation can be removed. We outline the ideas behind the construction of a DRR strategy that is outcome-equivalent to a given RRR strategy. The general idea is to simulate the behaviour of the RRR strategy at the start of the play using a new initial memory state and then move back into the RRR strategy we simulate.

We substitute the random selection of an initial memory element in two stages. To ensure the first action is selected in the same way under both the supplied strategy and the strategy we construct, we rely on randomised outputs. The probability of selecting an action $a^{(i)}$ in a given state s of the game in our new initial memory state is given as the sum of selecting action $a^{(i)}$ in state s in each memory state m weighed by the initial probability of m.

We then leverage the stochastic updates to simulate that we had been using the original RRR strategy from the start. To achieve this, we base the update function of the constructed Mealy machine on the equations for the update of the distribution over memory states after a some sequence in $w \in (S\bar{A})^*$ takes place (denoted by μ_w in Section 2).

We now state our expressiveness result and formalise the construction outlined above.

Theorem 4.2. *Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game. Let $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be an RRR strategy of $\mathcal{P}_i$. There exists a DRR strategy $\mathcal{N} = (N, n_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ such that $\mathcal{N}$ and $\mathcal{M}$ are outcome-equivalent, and such that $|N| = |M| + 1$.*

Proof. Let us define $\mathcal{N} = (N, n_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ as follows. Let n_{init} be such that $n_{\mathsf{init}} \notin M$. We set $N = M \cup \{n_{\mathsf{init}}\}$. We let β_{up} and β_{nxt} coincide with α_{up} and α_{nxt} over $M \times S \times \bar{A}$ and $M \times S$ respectively (for the update function, we identify distributions over M to distributions over N that assign probability zero to n_{init}). It remains to define these two functions over $\{n_{\mathsf{init}}\} \times S \times \bar{A}$ and $\{n_{\mathsf{init}}\} \times S$ respectively.

First, we complete the definition of the memory update function β_{up}. Let $s \in S$ and $\bar{a} \in \bar{A}$. We let $\beta_{\mathsf{up}}(n_{\mathsf{init}}, s, \bar{a})(n_{\mathsf{init}}) = 0$. We assume that there exists some $m_0 \in M$ such that $\mu_{\mathsf{init}}(m_0) > 0$ and $\alpha_{\mathsf{nxt}}(m_0, s)(a^{(i)}) > 0$ (i.e., the action $a^{(i)}$ has a positive probability of being played in s at the start of a play under the strategy $\mathcal{M}$). We set, for all $m \in M$,

$$\beta_{\mathsf{up}}(n_{\mathsf{init}}, s, \bar{a})(m) = \frac{\sum_{m' \in M} \mu_{\mathsf{init}}(m') \cdot \alpha_{\mathsf{up}}(m', s, \bar{a})(m) \cdot \alpha_{\mathsf{nxt}}(m', s)(a^{(i)})}{\sum_{m' \in M} \mu_{\mathsf{init}}(m') \cdot \alpha_{\mathsf{nxt}}(m', s)(a^{(i)})}.$$

Whenever we have $\alpha_{\mathsf{nxt}}(m_0, s)(a^{(i)}) = 0$ for all $m_0 \in M$ such that $\mu_{\mathsf{init}}(m_0) > 0$, we let $\beta_{\mathsf{up}}(n_{\mathsf{init}}, s, \bar{a})$ be arbitrary.

For the next-move function β_{nxt}, we define, for all states $s \in S$ and actions $a^{(i)} \in A^{(i)}(s)$,

$$\beta_{\mathsf{nxt}}(n_{\mathsf{init}}, s)(a^{(i)}) = \sum_{m \in M} \mu_{\mathsf{init}}(m) \cdot \alpha_{\mathsf{nxt}}(m, s)(a^{(i)}).$$

It remains to prove that $\mathcal{M}$ and $\mathcal{N}$ are outcome-equivalent. By Lemma 2.1, it suffices to show that both strategies suggest the same distributions over actions along histories consistent with $\mathcal{M}$. We provide a proof in two steps. First, we consider histories with a single state. Second, we show that the distributions over memory states coincide in both Mealy machines after any $w \in S\bar{A}$ that is consistent with $\mathcal{M}$ takes place. We conclude from this and the construction of $\mathcal{N}$ that $\mathcal{M}$ and $\mathcal{N}$ map all histories that are consistent with $\mathcal{M}$ and have more than one state to the same distribution over actions of $\mathcal{P}_i$, ending the proof.

We show the first claim above. Let $s \in S$ and $a^{(i)} \in A^{(i)}(s)$. On the one hand, the probability of the action $a^{(i)}$ being played after the history s under $\mathcal{M}$ is given by

$$\sum_{m \in M} \mu_{\mathsf{init}}(m) \cdot \alpha_{\mathsf{nxt}}(m, s)(a^{(i)}).$$

On the other hand, the probability of this same action $a^{(i)}$ being played after the history s under $\mathcal{N}$ is given by $\beta_{\mathsf{nxt}}(n_{\mathsf{init}}, s)(a^{(i)})$. These two probabilities coincide by construction.

Second, let $w = s\bar{a} \in S\bar{A}$ be consistent with $\mathcal{M}$. Let μ_w and ν_w denote the distribution over memory states after w takes place under $\mathcal{M}$ and $\mathcal{N}$ respectively. Fix some $m \in M$, and let us prove that $\mu_w(m) = \nu_w(m)$. On the one hand, we have

$$\begin{aligned} \mu_w(m) &= \frac{\sum_{m' \in M} \mu_{\mathsf{init}}(m') \cdot \alpha_{\mathsf{up}}(m', s, \bar{a})(m) \cdot \alpha_{\mathsf{nxt}}(m', s)(a^{(i)})}{\sum_{m' \in M} \mu_{\mathsf{init}}(m') \cdot \alpha_{\mathsf{nxt}}(m', s)(a^{(i)})} \\ &= \beta_{\mathsf{up}}(n_{\mathsf{init}}, s, a^{(i)})(m), \end{aligned}$$

and on the other hand, we have (because n_{init} is the sole initial state of $\mathcal{N}$),

$$\nu_w(m) = \frac{\beta_{\mathsf{up}}(n_{\mathsf{init}}, s, \bar{a})(m) \cdot \beta_{\mathsf{nxt}}(n_{\mathsf{init}}, s)(a^{(i)})}{\beta_{\mathsf{nxt}}(n_{\mathsf{init}}, s)(a^{(i)})} = \beta_{\mathsf{up}}(n_{\mathsf{init}}, s, \bar{a})(m).$$

We have shown that $\mu_w = \nu_w$. Furthermore, because α_{nxt} and β_{nxt} agree over $M \times S$, and that α_{up} and β_{up} agree over $M \times S \times \bar{A}$, this equality generalises to all $w \in (S\bar{A})^+$ that are consistent with $\mathcal{M}$. It follows that for any history $h \in (S\bar{A})^+S$ that is consistent with $\mathcal{M}$, the images of h by the strategies induced by $\mathcal{M}$ and $\mathcal{N}$ match. We conclude that $\mathcal{M}$ and $\mathcal{N}$ are outcome-equivalent by Lemma 2.1. □

4.3 Simulating RRR strategies with RDR ones

We are concerned in this section with the simulation of RRR strategies by RDR strategies, i.e., with substituting randomised outputs with deterministic outputs. The idea behind the removal of randomisation in outputs is to simulate said randomisation by means of both stochastic initialisation and updates. These are used to preemptively perform the random selection of an action, simultaneously with the selection of an initial or successor memory state.

Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game and let $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be an RRR strategy of $\mathcal{P}_i$. We construct an RDR strategy $\mathcal{N} = (N, \nu_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ that is outcome-equivalent to $\mathcal{M}$ and such that $|N| \leq |M| \cdot |S| \cdot |A^{(i)}|$. The state space of $\mathcal{N}$ consists of pairs (m, σ_i) where $m \in M$ and $\sigma_i \colon S \to A^{(i)}$ is a pure memoryless strategy of $\mathcal{P}_i$. To achieve our bound on the size of N, we cannot consider all pure memoryless strategies of $\mathcal{P}_i$, as there are exponentially many. We illustrate how we select pure memoryless strategies to achieve the aforementioned bound through the following example. We apply the upcoming construction on a DRD strategy (which is a special case of RRR strategies) with a single memory state (i.e., a memoryless randomised strategy).

Example 4.1. We consider a game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ where $S = \{s_1, s_2, s_3\}$, $A^{(1)} = \{a_1, a_2, a_3\}$ and all actions are enabled in all states. We need not specify $A^{(2)}$ and δ for this example. For our construction, we fix an order on the actions of $\mathcal{G}$: $a_1 < a_2 < a_3$.

Let $\mathcal{M} = (\{m\}, m, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be the DRD strategy such that $\alpha_{\mathsf{nxt}}(m, s_1)$ and $\alpha_{\mathsf{nxt}}(m, s_2)$ are uniform distributions over $\{a_1, a_2\}$ and $A^{(1)}$ respectively, and $\alpha_{\mathsf{nxt}}(m, s_3)$ is defined by $\alpha_{\mathsf{nxt}}(m, s_3)(a_1) = \frac{1}{3}$, $\alpha_{\mathsf{nxt}}(m, s_3)(a_2) = \frac{1}{6}$ and $\alpha_{\mathsf{nxt}}(m, s_3)(a_3) = \frac{1}{2}$.

Figure 4.2 illustrates the probability of each action being chosen in each state as the length of a segment. Let us write $0 = x_1 < x_2 < x_3 < x_4 < x_5 = 1$ for all of the endpoints of the segments appearing in the illustration. For each index $k \in \{1, \dots, 4\}$, we define a pure memoryless strategy σ_k that assigns to each state the action lying in the segment above it in the figure. For instance, σ_2 is such that $\sigma_2(s_1) = a_1$ and $\sigma_2(s_2) = \sigma_2(s_3) = a_2$. Furthermore, for all $k \in \{1, \dots, 4\}$, the length $x_{k+1} - x_k$ of its corresponding interval denotes the probability of the strategy being chosen during stochastic updates.

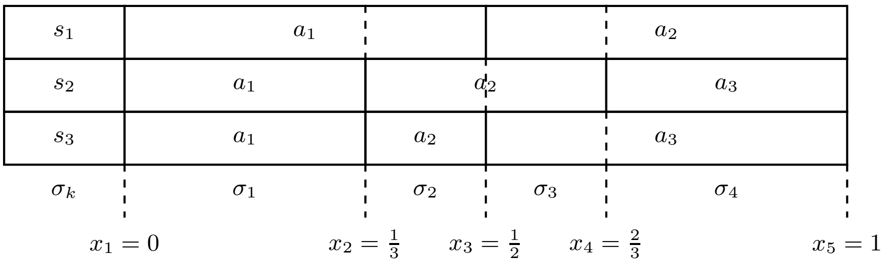


Fig. 4.2: Representation of cumulative probability of actions under strategy $\mathcal{M}$ and derived memoryless strategies.

We construct an RDR strategy $\mathcal{N} = (N, \nu_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ that is outcome-equivalent to $\mathcal{M}$ in the following way. We let $N = \{m\} \times \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. The initial distribution is given by $\nu_{\mathsf{init}}(m, \sigma_k) = x_{k+1} - x_k$, i.e., the probability of σ_k in the illustration. We set, for any $j, k \in \{1, \dots, 4\}$, $s \in S$ and $a^{(1)} \in A^{(1)}$, $\beta_{\mathsf{up}}((m, \sigma_k), s, a^{(1)})((m, \sigma_j)) = x_{j+1} - x_j$. Finally, we let $\beta_{\mathsf{nxt}}((m, \sigma_k), s) = \sigma_k(s)$ for all $k \in \{1, \dots, 4\}$ and $s \in S$.

The argument for the outcome-equivalence of $\mathcal{N}$ and $\mathcal{M}$ is the following; for any state $s \in S$, the probability of moving into a memory state (m, σ_k) such that $\sigma_k(s) = a$ is by construction the probability $\alpha_{\mathsf{nxt}}(m, s)$. ◁

In the previous example, we had a unique memory state m and we defined some memoryless strategies from the next-move function partially evaluated in this state (i.e., from $\alpha_{\mathsf{nxt}}(m, \cdot)$). In general, each memory state may have a different partially evaluated next-move function, and therefore we must define some memoryless strategies for each individual memory state. For each memory state, we can bound the number of derived memoryless strategies by $|S| \cdot |A^{(i)}|$; we look at cumulative probabilities over actions (of which there are at most $|A^{(i)}|$) for each state. This explains our announced bound on $|N|$.

Furthermore, in general, the memory update function is not trivial. Generalising the construction above can be done in a straightforward manner to handle updates. Intuitively, the probability to move to some memory state of the form (m, σ_i) is given by the probability of moving into m multiplied by the probability of σ (in the sense of Figure 4.2).

We now formally state our result in the general setting and provide its proof. The Mealy machine we construct has updates that do not depend on the actions of the player who owns it; this property is useful when we study games of imperfect information in Section 7.

Theorem 4.3. *Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game. Let $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be an RRR strategy of $\mathcal{P}_i$. There exists an RDR strategy $\mathcal{N} = (N, \nu_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ such that $\mathcal{N}$ and $\mathcal{M}$ are outcome-equivalent, and such that $|N| \leq |M| \cdot (|S| \cdot (|A^{(i)}| - 1) + 1)$. Furthermore, the updates of $\mathcal{N}$ do not depend on the actions of $\mathcal{P}_i$.*

Proof. Let us fix a linear order on the set of actions $A^{(i)}$, denoted by $<$. Fix some $m \in M$. We let $x_1^m < \ldots < x_{\ell(m)}^m$ denote the elements of the set

$$\left\{ \sum_{b^{(i)} < a^{(i)}} \alpha_{\mathsf{nxt}}(m, s)(b^{(i)}) \mid s \in S,\, a^{(i)} \in A^{(i)} \right\}$$

that are strictly inferior to 1, and let $x_{\ell(m)+1}^m = 1$. These x_j^m represent the cumulative probability provided by $\alpha_{\mathsf{nxt}}(m, \cdot)$ over actions of $\mathcal{P}_i$ taken in order, for each state of $\mathcal{G}$. For each $j \in \{1, \ldots, \ell(m)\}$, we define a memoryless strategy $\sigma_j^m \colon S \to A^{(i)}$ as follows: we have $\sigma_j^m(s) = a^{(i)}$ if $\sum_{b^{(i)} < a^{(i)}} \alpha_{\mathsf{nxt}}(m, s)(b^{(i)}) \leq x_j^m < \sum_{b^{(i)} \leq a^{(i)}} \alpha_{\mathsf{nxt}}(m, s)(b^{(i)})$. In other words, for any state $s \in S$, we have $\sigma_j^m(s) = a^{(i)}$ whenever x_j^m is at least the cumulative probability of actions strictly inferior to $a^{(i)}$ in $\alpha_{\mathsf{nxt}}(m, s)$ and at most the cumulative probability of actions up to action $a^{(i)}$ included. Refer to Figure 4.2 of Example 4.1 for an explicit illustration. We refer to $x_{j+1}^m - x_j^m$ as the probability of σ_j^m in the sequel.

Let $m \in M$, $s \in S$ and $a^{(i)} \in A^{(i)}(s)$. We show that we can relate $\alpha_{\mathsf{nxt}}(m, s)(a^{(i)})$ and the sum of the probabilities of each σ_j^m such that $\sigma_j^m(s) = a^{(i)}$ as follows. First, we introduce some notation. Let $I(m, s, a^{(i)})$ denote the set of indices j such that $\sigma_j^m(s) = a^{(i)}$, i.e., the indices such that the jth strategy related to m prescribes action $a^{(i)}$ in s. It holds that

$$\sum_{j \in I(m,s,a^{(i)})} (x_{j+1}^m - x_j^m) = \alpha_{\mathsf{nxt}}(m, s)(a^{(i)}). \tag{4.2}$$

Let $s \in S$ and $a^{(i)} \in A^{(i)}(s)$. Equation (4.2) can be proven as follows. First, note that all indices j appearing in the sum are consecutive by construction. Therefore, the sum above is telescoping and is equal to $x_{j^+ +1}^m - x_{j^-}^m$, where j^+ and j^- denote the largest and smallest indices in the sum respectively. By construction, we have $x_{j^-}^m = \sum_{b^{(i)} < a^{(i)}} \alpha_{\mathsf{nxt}}(m, s)(b^{(i)})$ and $x_{j^+ +1}^m = \sum_{b^{(i)} \leq a^{(i)}} \alpha_{\mathsf{nxt}}(m, s)(b^{(i)})$. We conclude that $x_{j^+ +1}^m - x_{j^-}^m = \alpha_{\mathsf{nxt}}(m, s)(a^{(i)})$, proving Equation (4.2). This equation is used to establish the outcome-equivalence of $\mathcal{M}$ with the strategy defined below.

We now define an RDR strategy $\mathcal{N} = (N, \nu_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$. We define

$$N = \{(m, \sigma_j^m) \mid m \in M,\, 1 \leq j \leq \ell(m)\}.$$

The initial distribution and update function of $\mathcal{N}$ are derived from those of $\mathcal{M}$ multiplied with the probability of the memoryless strategy that appears in the second component of the memory state of $\mathcal{N}$ into which we move. The initial distribution ν_{init} is defined as $\nu_{\mathsf{init}}((m, \sigma_j^m)) = \mu_{\mathsf{init}}(m) \cdot (x_{j+1}^m - x_j^m)$ for all $(m, \sigma_j^m) \in N$. The update function is defined as $\beta_{\mathsf{up}}((m, \sigma_j^m), s, \bar{a})((m', \sigma_k^{m'})) = \alpha_{\mathsf{up}}(m, s, \bar{b})(m') \cdot (x_{k+1}^{m'} - x_k^{m'})$, where $\bar{b} = (\sigma_j^m(s), a^{(2)})$ if $i = 1$ (respectively $\bar{b} = (a^{(1)}, \sigma_j^m(s))$ if $i = 2$), for all $(m, \sigma_j^m), (m', \sigma_k^{m'}) \in N$, $s \in S$ and $\bar{a} \in \bar{A}$. We remark that this update function does not depend on the action of $\mathcal{P}_i$ given as input. Finally, the deterministic next-move function of $\mathcal{N}$ is defined as $\beta_{\mathsf{nxt}}((m, \sigma_j^m), s) = \sigma_j^m(s)$ for all $(m, \sigma_j^m) \in N$ and all $s \in S$.

We now prove the outcome-equivalence of $\mathcal{M}$ and $\mathcal{N}$. For any $w \in (S\bar{A})^*$, let μ_w (resp. ν_w) denote the distribution over M (resp. N) after w has occurred under strategy $\mathcal{M}$ (resp. $\mathcal{N}$). It follows from Lemma 2.1 and the definition of strategies derived from FM strategies that it suffices to establish, for all histories $h = ws$ consistent with $\mathcal{M}$, that the following holds:

$$\sum_{m \in M} \mu_w(m) \cdot \alpha_{\mathsf{nxt}}(m, s)(a^{(i)}) = \sum_{m \in M} \sum_{j \in I(m,s,a^{(i)})} \nu_w((m, \sigma_j^m)). \tag{4.3}$$

To prove Equation (4.3), we rely on the following property: for any $w \in (S\bar{A})^*$ consistent with $\mathcal{M}$, $\mu_w(m)$ is proportional to $\nu_w((m, \sigma_j^m))$. To be precise, for any $w \in (S\bar{A})^*$ consistent with $\mathcal{M}$, we have

$$\nu_w((m, \sigma_j^m)) = (x_{j+1}^m - x_j^m) \cdot \mu_w(m). \tag{4.4}$$

To show Equation (4.4), we proceed by induction. Consider the empty word $w = \varepsilon$. Because $\mu_{\mathsf{init}} = \mu_\varepsilon$ and $\nu_{\mathsf{init}} = \nu_\varepsilon$, Equation (4.4) follows from the definition of ν_{init}. Let us now assume inductively that for $w' \in (S\bar{A})^*$ consistent with $\mathcal{M}$, we have Equation (4.4) and let us prove it for $w = w's\bar{a}$ consistent with $\mathcal{M}$. Fix $(m, \sigma_j^m) \in N$.

To invoke the inductive relation between ν_w and $\nu_{w'}$, we must have that w is consistent with $\mathcal{N}$. There exist $m' \in \mathsf{supp}(\mu_{w'})$ such that $\alpha_{\mathsf{nxt}}(m', s)(a^{(i)}) > 0$ and $k \in I(m', s, a^{(i)})$ (this set is non-empty due to $\alpha_{\mathsf{nxt}}(m', s)(a^{(i)}) > 0$). By the induction hypothesis, we obtain $\nu_{w'}((m', \sigma_k^{m'})) > 0$, which is sufficient to conclude that w is consistent with $\mathcal{N}$. We thus obtain, from the equation relating ν_w and $\nu_{w'}$,

$$\nu_w((m, \sigma_j^m)) = \frac{\sum_{m' \in M} \sum_{k \in I(m', s, a^{(i)})} \nu_{w'}((m', \sigma_k^{m'})) \cdot \beta_{\mathsf{up}}((m', \sigma_k^{m'}), s, \bar{a})((m, \sigma_j^m))}{\sum_{m' \in M} \sum_{k \in I(m', s, a^{(i)})} \nu_{w'}((m', \sigma_k^{m'}))}.$$

The numerator of the above can be rewritten as follows, by successively using the definition of β_{up} followed by the inductive hypothesis and Equation (4.2):

$$\begin{aligned}
&\sum_{m' \in M} \sum_{k \in I(m', s, a^{(i)})} \nu_{w'}((m', \sigma_k^{m'})) \cdot \alpha_{\mathsf{up}}(m', s, \bar{a})(m) \cdot (x_{j+1}^m - x_j^m) \\
&= (x_{j+1}^m - x_j^m) \cdot \sum_{m' \in M} \left(\alpha_{\mathsf{up}}(m', s, \bar{a})(m) \cdot \mu_{w'}(m') \cdot \sum_{k \in I(m', s, a^{(i)})} (x_{k+1}^{m'} - x_k^{m'}) \right) \\
&= (x_{j+1}^m - x_j^m) \cdot \sum_{m' \in M} \alpha_{\mathsf{up}}(m', s, \bar{a})(m) \cdot \mu_{w'}(m') \cdot \alpha_{\mathsf{nxt}}(m', s)(a^{(i)}).
\end{aligned}$$

Following the same reasoning, the denominator can be rewritten as

$$\sum_{m' \in M} \mu_{w'}(m') \cdot \alpha_{\mathsf{nxt}}(m', s)(a^{(i)}).$$

By combining the equations above and the formula for the update of μ_w, we obtain $\nu_w((m, \sigma_j^m)) = (x_{j+1}^m - x_j^m) \cdot \mu_w(m)$, ending the proof of Equation (4.4).

We show that Equation (4.4) implies Equation (4.3), which will prove that $\mathcal{M}$ and $\mathcal{N}$ are outcome-equivalent. Let $h = ws \in \mathsf{Hist}(\mathcal{G})$ be a history consistent with $\mathcal{M}$. Let $a^{(i)} \in A^{(i)}(s)$. The probability that the action $a^{(i)}$ is chosen after history h under $\mathcal{M}$ is given by $\sum_{m \in M} \mu_w(m) \cdot \alpha_{\mathsf{nxt}}(m, s)(a^{(i)})$. The probability that $a^{(i)}$ is selected after h under $\mathcal{N}$, on the other hand, is given by

$$\begin{aligned}
\sum_{m \in M} \sum_{j \in I(m, s, a^{(i)})} \nu_w((m, \sigma_j^m)) &= \sum_{m \in M} \left(\mu_w(m) \cdot \sum_{j \in I(m, s, a^{(i)})} (x_{j+1}^m - x_j^m) \right) \\
&= \sum_{m \in M} \mu_w(m) \cdot \alpha_{\mathsf{nxt}}(m, s)(a^{(i)}).
\end{aligned}$$

In the above, the first equation is obtained from Equation (4.4) and the second equation follows from Equation (4.2). This concludes the argument for the outcome-equivalence of our two FM strategies.

To end the proof of this theorem, we prove the upper bound on $|N|$ given in the statement of the result. For any memory state $m \in M$, $\ell(m)$ is bounded by $|S| \cdot (|A^{(i)}| - 1) + 1$: by definition of the numbers x_j^m, we see that we must have $\ell(m) \leq |S| \cdot |A^{(i)}|$. To obtain the aforementioned bound, observe that for all $s \in S$, we have $\sum_{b^{(i)} < \min A^{(i)}} \alpha_{\mathsf{nxt}}(m, s)(b^{(i)}) = 0$, i.e., 0 admits (at least) $|S|$ different writings in the set of the x_j^ms, hence $\ell(m) \leq |S| \cdot |A^{(i)}| - (|S| - 1) = |S| \cdot (|A^{(i)}| - 1) + 1$. Therefore, we have at most $|S| \cdot (|A^{(i)}| - 1) + 1$ pairs of the form (m, σ_j^m) per memory state $m \in M$. It follows that $|N| \leq |M| \cdot (|S| \cdot (|A^{(i)}| - 1) + 1)$. □

Remark 4.1. The choice of the order on the set of actions fixed at the start of the previous proof influences the size of the constructed strategy. It is not necessary to use the same ordering of actions for all memory states. The order is used to define all memoryless strategies of the form σ_j^m, which do not interact with strategies associated to other memory states. For this reason, it is possible to use different orderings on actions depending on the memory state m that is considered. ◁

Remark 4.2. The upper bound on the number of memory states given in the statement of Theorem 4.3 can be slightly improved in a turn-based setting. In general, we can replace the term $|S|$ in the bound by the number of states that $\mathcal{P}_i$ controls (more precisely, by the number of $\mathcal{P}_i$-controlled states with at least two enabled actions). ◁

5 Separating classes of strategies

We now discuss the separation of strategies given in the lattice of Figure 1.1, and in particular we consider the strictness of inclusions. All separation results hold in one-player deterministic games with a single state and two actions. This is one of the simplest possible settings to show that strategy classes are distinct. Indeed, in a game with a single state and a single action, the only strategy is to always play the unique action, and therefore all strategy classes collapse into one. For the entirety of this section, we let $\mathcal{G}_{a,b}$ denote the game depicted in Figure 5.1, and we provide strategies of $\mathcal{G}_{a,b}$ to show that strategy classes differ. We complement the separating strategies from $\mathcal{G}_{a,b}$ with problem instances from the literature for which strategies from some class suffice whereas strategies from the compared class do not.

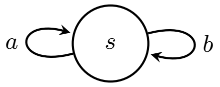
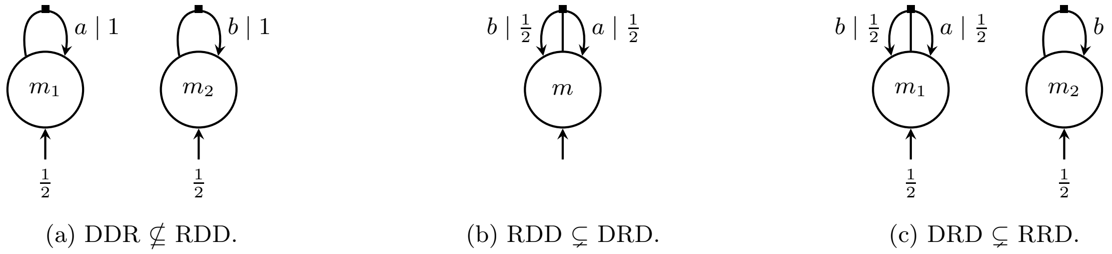


Fig. 5.1: The MDP $\mathcal{G}_{a,b}$ with a single state and two actions.

We illustrate FM strategies witnessing non-inclusions asserted in the lattice of Figure 1.1 in Figures 5.2 and 5.4. The Mealy machines are interpreted as follows. Edges that exit memory states read a game state (omitted in these figures due to s being the sole involved game state) and split into edges labelled by an action and a probability of this action being played, e.g., for $c \in \{a, b\}$ and $p \in [0, 1]$, the notation $c \mid p$ indicates that the probability of playing action c in the current memory state is p. In Figure 5.4, the edges are further split after the choice of an action for randomised updates. The edge labels following this second split represent the probabilities of stochastic updates. This second split is omitted whenever an update is deterministic.



(a) DDR $\not\subseteq$ RDD.

(b) RDD $\subsetneq$ DRD.

(c) DRD $\subsetneq$ RRD.

Fig. 5.2: Depictions of Mealy machines witnessing the strictness of three inclusions asserted in Figure 1.1. For the sake of readability, we do not label transitions by s as it is the sole state the Mealy machines can read in $\mathcal{G}_{a,b}$, and the only state with a choice in the games of Figure 5.3.

The rest of the section is structured as follows. We discuss the strict inclusion of DDD in RDD and show that RDD is not included in DDR in Section 5.1. Section 5.2 complements the previous Section 4.1 and presents a DRD strategy that has no outcome-equivalent RDD counterpart. The strict inclusion of the class DRD in the class of RRD strategies is covered in Section 5.3. Finally, we prove that DDR is not included in RRD in Section 5.4, which implies that DDR is incomparable to the strategy classes RDD, DRD and RRD.

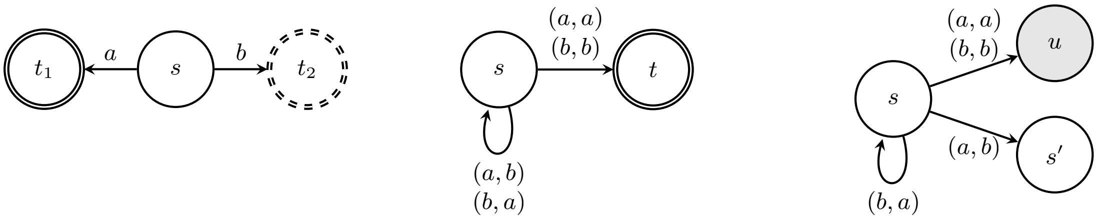
5.1 DDD strategies are strictly less expressive than RDD ones

Pure FM strategies are less powerful than RDD strategies. The latter class of strategies can induce non-Dirac distributions over the plays of $\mathcal{G}_{a,b}$, whereas the former cannot. We illustrate a strategy that has no outcome-equivalent DDD strategy in Figure 5.2a. Furthermore, there is no DDR strategy that is outcome equivalent to the strategy depicted in Figure 5.2a: DDR strategies lack the ability to provide a randomised action at the first step of a game. We obtain the following result.

Lemma 5.1. *There exists an RDD strategy of $\mathcal{P}_1$ in $\mathcal{G}_{a,b}$ such that there is no outcome-equivalent DDR strategy (in particular, there is no outcome-equivalent DDD strategy).*

We now provide a setting and example from [44, 16] in which RDD strategies can satisfy a specification that pure strategies cannot. We consider MDPs with several reachability objectives with absorbing targets.

Let $\mathcal{G} = (S, A, \delta)$ be an MDP and let $k \geq 1$. Given $T \subseteq S$, we let $\mathsf{Reach}(T)$ denote the set of plays in which a state of T occurs; this set of plays is the *reachability objective* for target T. For all $1 \leq j \leq k$, we let $T_j \subseteq S$ be a set of *absorbing states*, i.e., for all $s \in T_j$ and all $a \in A(s)$, $\delta(s, a)(s) = 1$. Given a vector $q = (q_j)_{1 \leq j \leq k} \in ([0, 1] \cap \mathbb{Q})^k$ and an initial state $s_{\mathsf{init}} \in S$, we consider the problem of determining the existence of a strategy σ_1 such that for all $1 \leq j \leq k$, $\mathbb{P}^{\sigma_1}_{s_{\mathsf{init}}}(\mathsf{Reach}(T_j)) \geq q_j$. If there exists such a strategy, we say that q is *achievable* (from s_{init}). We give an instance of the problem that illustrates that pure strategies do not suffice below.



(a) An MDP with two reachability objectives.

(b) A concurrent reachability game.

(c) A concurrent safety game (snowball game [26]).

Fig. 5.3: Games we use to further illustrate the separation of classes of strategies via example specifications. States depicted without outgoing transitions have outgoing self-loops that are omitted to lighten figures.

Example 5.1. Consider the MDP depicted in Figure 5.3a and let s be the initial state. We consider the two targets $T_1 = \{t_1\}$ and $T_2 = \{t_2\}$ and the vector $q = (\frac{1}{2}, \frac{1}{2})$. It is clear that no pure strategy witnesses the achievability of q; any pure strategy achieves the vector $(1, 0)$ or $(0, 1)$ if it chooses action a or b in s respectively. However, there is an RDD strategy that witnesses the achievability of q; any extension of the strategy depicted in Figure 5.2a that accounts for the new game states t_1 and t_2 achieves q. ◁

It turns out that RDD strategies suffice to witness that a vector is achievable no matter the considered instance of the problem. We provide a short proof of this statement below.

Lemma 5.2. *Let q be an achievable vector in the MDP $\mathcal{G}$ with respect to the reachability objectives* $\mathsf{Reach}(T_1)$, …, $\mathsf{Reach}(T_k)$ *for absorbing targets T_1, …, $T_k \subseteq S$. There exists an RDD strategy witnessing the achievability of q.*

Proof. It is shown in [44] that the set of achievable vectors is a polyhedral set. Furthermore, the vertices of this set are achievable by pure memoryless strategies. It follows that any achievable vector is dominated by a convex combination of vectors achievable by pure memoryless strategies. We conclude that RDD strategies suffice to achieve q. □

5.2 RDD strategies are strictly less expressive than DRD ones

The goal of this section is to show that there exists a DRD strategy that cannot be emulated by any RDD strategy. Let us first explain some intuition behind this statement. Intuitively, an RDD strategy can only randomise once at the start between a finite number of pure FM (DDD) strategies. After this initial randomisation, the sequence of actions prescribed by the RDD strategy is fixed relative to the play in progress. Any DRD strategy that chooses an action randomly at each step, such as the strategy depicted in Figure 5.2b, i.e., the strategy playing actions a and b with uniform probability at each step in $\mathcal{G}_{a,b}$, cannot be reproduced by an RDD strategy. Indeed, this randomisation generates an infinite number of patterns of actions. These patterns cannot all be captured by an RDD strategy due to the fact that its initial randomisation is over a finite set.

Lemma 5.3. *There exists a DRD strategy of $\mathcal{P}_1$ in $\mathcal{G}_{a,b}$ such that there is no outcome-equivalent RDD strategy.*

Proof. Let $\sigma_1 \colon \{s\} \to \mathcal{D}(\{a, b\})$ be the memoryless strategy in $\mathcal{G}_{a,b}$ induced by the Mealy machine depicted in Figure 5.2b. The distribution $\sigma_1(s)$ is the uniform distribution over $\{a, b\}$. The strategy σ_1 induces a probability distribution over plays of $\mathcal{G}_{a,b}$ such that all plays have a probability of zero. Indeed, let π be a play of $\mathcal{G}_{a,b}$. One can view the singleton $\{\pi\}$ as the decreasing intersection $\bigcap_{k\in\mathbb{N}} \mathsf{Cyl}(\pi_{|k})$. Hence, the probability of $\{\pi\}$ is the limit of the probability of $\mathsf{Cyl}(\pi_{|k})$ when k goes to infinity. One can easily show that the probability under σ_1 of $\mathsf{Cyl}(\pi_{|k})$ is $\frac{1}{2^k}$. It follows that the probability of $\{\pi\}$ is zero.

We now establish that there is no outcome-equivalent RDD strategy. First, let us recall that any RDD strategy can be presented as a distribution over a finite number of pure FM strategies. Given that there are no probabilities on the transitions of $\mathcal{G}_{a,b}$, for any pure strategy σ_1^{pure}, there is a single outcome under σ_1^{pure}. We can infer that, for any RDD strategy of $\mathcal{G}_{a,b}$, there must be at least one play that has a non-zero probability, and therefore this strategy cannot be outcome-equivalent to σ_1, ending the proof. □

We present a setting in which RDD strategies do not suffice, whereas DRD strategies suffice. We study concurrent reachability games. Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game, $s_{\mathsf{init}} \in S$ be an initial state and $T \subseteq S$ be a set of target vertices. We consider the reachability objective $\mathsf{Reach}(T)$ again. In a concurrent zero-sum reachability game, the goal of $\mathcal{P}_1$ is to maximise the worst-case probability of $\mathsf{Reach}(T)$. Formally, we say that a strategy σ_1 of $\mathcal{P}_1$ ensures the threshold $q \in [0, 1]$ from s_{init} if $\inf_{\sigma_2} \mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Reach}(T)) \geq q$, where σ_2 ranges over strategies of $\mathcal{P}_2$. The goal of $\mathcal{P}_1$ is to ensure the greatest possible threshold.

The supremum of the thresholds that can be ensured from s_{init} is called the *value* of s_{init}. A strategy is *optimal* from s_{init} if it ensures the value of s_{init}. If there exists an optimal strategy from a state s_{init} of value 1, we say that $\mathcal{P}_1$ wins almost-surely from s_{init}.

We illustrate in the following example that RDD strategies may be unable to ensure thresholds that DRD strategies can in concurrent reachability games.

Example 5.2. Consider the game depicted in Figure 5.3b and let s be the initial state. Let $T = \{t\}$ be the target.

We first claim that there are no RDD strategies of $\mathcal{P}_1$ that win almost-surely from s. We fix an RDD Mealy machine $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ of $\mathcal{P}_1$ and let $\sigma_1^{\mathcal{M}}$ denote the strategy it induces. For all $m_{\mathsf{init}} \in \mathsf{supp}(\mu_{\mathsf{init}})$, we consider the pure FM strategy $\sigma_1^{m_{\mathsf{init}}}$ induced by $(M, m_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$. We fix $m_{\mathsf{init}} \in \mathsf{supp}(\mu_{\mathsf{init}})$ and a pure strategy σ_2 of $\mathcal{P}_2$ such that for all histories h ending in s, $\sigma_2(h) \neq \sigma_1^{m_{\mathsf{init}}}(h)$. It follows that $\mathbb{P}^{\sigma_1^{m_{\mathsf{init}}},\sigma_2}_{s}(\mathsf{Reach}(T)) = 0$. This implies that $\mathcal{M}$ is not almost-surely winning from s because, by the law of total probability, we have

$$\mathbb{P}^{\sigma_1^{\mathcal{M}},\sigma_2}_{s}(\mathsf{Reach}(T)) = \sum_{m\in M} \mu_{\mathsf{init}}(m) \cdot \mathbb{P}^{\sigma_1^{m},\sigma_2}_{s}(\mathsf{Reach}(T)).$$

On the other hand, the memoryless randomised strategy depicted in Figure 5.2b is almost-surely winning. At each round prior to a visit of t, no matter the choices of $\mathcal{P}_2$, this strategy ensures a probability of $\frac{1}{2}$ of matching the action of $\mathcal{P}_2$. It follows that this strategy is almost-surely winning. ◁

In full generality, there need not exist optimal strategies in concurrent reachability games [26]. Nonetheless, memoryless randomised strategies (which are a restricted class of DRD strategies) can be used to ensure any possible threshold in these games. In particular, if there exists an optimal strategy, there always exists one that is memoryless. We summarise these results in the following theorem.

Theorem 5.1 ([26, 45]). *In all concurrent reachability games, if a threshold q can be ensured by $\mathcal{P}_1$, then there exists a memoryless strategy that ensures q.*

5.3 DRD strategies are strictly less expressive than RRD ones

In this section, we show that there exists an RRD strategy that has no outcome-equivalent DRD strategy. The example we provide is based on existing results for concurrent safety games, i.e., games where the goal is the complement of a reachability objective. Given a game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$, we let $T \subseteq S$ be a set of states and let $\mathsf{Safe}(T)$ be the *safety objective*, which is the set of plays that do not traverse T. A strategy σ_1 of $\mathcal{P}_1$ in $\mathcal{G}$ is said to be *positively winning* for the safety objective $\mathsf{Safe}(T)$ from an initial state s_{init} if for all strategies σ_2 of $\mathcal{P}_2$, $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Safe}(T)) > 0$.

Consider the game depicted in Figure 5.3c with the safety objective $\mathsf{Safe}(\{u\})$ and consider s to be its initial state. It is shown in [26] that $\mathcal{P}_1$ does not have a positively winning DRD strategy in this game. The authors of [20] show however there exists a positively winning RRD strategy. The Mealy machine of Figure 5.2c matches their positively winning RRD Mealy machine.

The main idea underlying the strategy induced by this Mealy machine is the following. It attempts the action a at all steps with a positive probability due to memory state m_1. It also has a positive probability of never playing a due to memory state m_2. Therefore, a is played after a history $s(bs)^k$ with a probability that decreases to zero as k increases, as otherwise a would eventually occur almost-surely.

This behaviour cannot be achieved with a DRD strategy. The distribution over memory states of a DRD strategy following a history is a Dirac distribution due to the deterministic initialisation and deterministic updates. It follows that DRD strategies suggest actions with probabilities given directly by the next-move function, i.e., the image of a DRD strategy is finite. It follows that there is no DRD strategy that is outcome-equivalent to the strategy depicted in Figure 5.2c. We formalise this argument in the proof of the following lemma.

Lemma 5.4. *There exists an RRD strategy of $\mathcal{P}_1$ in $\mathcal{G}_{a,b}$ such that there is no outcome-equivalent DRD strategy.*

Proof. We consider the RRD strategy σ_1 induced by the Mealy machine $\mathcal{M} = (M, m_1, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ depicted in Figure 5.2c. For any $w \in (\{s\}\{a,b\})^*$, let μ_w denote the distribution over M after w as taken place under $\mathcal{M}$. It can be shown by induction that for any $k \in \mathbb{N}$, $\mu_{(sb)^k}(m_1) = 1 - \mu_{(sb)^k}(m_2) = \frac{1}{2^k+1}$ and for any $w \in (\{s\}\{a,b\})^*$ with at least one occurrence of a, $\mu_w(m_1) = 1$. It follows that for any $k \in \mathbb{N}$, $\sigma_1((sb)^k s)(a) = \frac{1}{2(2^k+1)}$ and $\sigma_1((sb)^k s)(b) = \frac{2^{k+1}+1}{2(2^k+1)}$, and for any history h containing an occurrence of a, $\sigma_1(h)(a) = \sigma_1(h)(b) = \frac{1}{2}$. We obtain that σ_1 plays the action a with positive probabilities that can be arbitrarily small and that all histories of $\mathcal{G}_{a,b}$ are consistent with σ_1.

We now show that no DRD strategy is outcome-equivalent to σ_1. Let $\mathcal{N} = (N, n_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ denote a DRD strategy and let τ_1 denote its induced strategy. By Lemma 2.1, τ_1 is outcome-equivalent to σ_1 if and only if both strategies are equal, as all histories are consistent with σ_1. For all $h \in \mathsf{Hist}(\mathcal{G}_{a,b})$, due to the deterministic initialisation and updates of $\mathcal{N}$, we have $\tau_1(h) = \beta_{\mathsf{nxt}}(n, \mathsf{last}(h))$ for some $n \in N$. In particular, τ_1 cannot play the action a with arbitrarily small positive probabilities as it can only assign finitely many distributions to histories. We conclude that $\tau_1 \neq \sigma_1$, which ends the proof. □

We return to positively winning strategies in concurrent safety games. It is argued in [20] that RRR strategies are sufficient to win positively in any concurrent safety game. We build on their argument to show that RRD strategies suffice to win positively in any concurrent safety game.

Each state in a concurrent safety game can be assigned a rank. States of highest rank are those from which $\mathcal{P}_2$ wins almost-surely for their dual reachability objective. States of minimal rank, if they are not simultaneously of maximal rank, are those from which $\mathcal{P}_1$ can surely enforce the safety objective no matter the strategy of $\mathcal{P}_2$, i.e., $\mathcal{P}_1$ has a (memoryless) strategy such that all plays consistent with this strategy that start from a state of minimal rank satisfy the safety objective.

Let $s \in S$ be a state that is positively winning. There exists an action of $\mathcal{P}_1$, which we will call a *sound action*, and a set $A_\star^{(2)}(s) \subseteq A^{(2)}(s)$ of actions of $\mathcal{P}_2$ such that the sound action surely prevents moving to states of higher rank against all actions in $A_\star^{(2)}(s)$. Furthermore, for actions of $\mathcal{P}_2$ outside of $A_\star^{(2)}(s)$, there is an action of $\mathcal{P}_1$ that moves to a state of strictly lower rank with positive probability. For instance, in the snowball game depicted in Figure 5.3c, for state s, the action b is a sound action for s with respect to $A_\star^{(2)}(s) = \{a\}$.

The property we require on our strategy to win positively is to use a strategy much like that of Figure 5.2c. On the one hand, it must have a positive probability of only using sound actions from any point: this way, the safety objective is ensured whenever $\mathcal{P}_2$ only uses actions in the sets of the form $A_\star^{(2)}(s)$ in the remainder of the play. On the other hand, to account for the possibility of $\mathcal{P}_2$ taking an action outside of $A_\star^{(2)}(s)$ in state s, all actions should have a positive probability of occurring in all rounds, so a vertex of lower rank can be reached with positive probability in this case.

Because the state space is finite, one of two cases occurs. If $\mathcal{P}_2$ only resorts to actions compatible with sound actions from some point on, then the safety objective is satisfied with positive probability because sound actions are guaranteed to be always played from some point on with positive probability. Otherwise, states of minimal ranks are reached with positive probability, from which $\mathcal{P}_1$ can surely avoid T.

The idea of the RRR strategy proposed in [20] to obtain the behaviour described above is to rely on pairs of memory states. In a pair, one memory state only proposes sound actions and the other memory state suggests all actions uniformly at random. When initialising the Mealy machine and each time there is a change in the rank of states, to ensure the resulting strategy has the property above, a stochastic memory update is used to give a uniform probability over such a pair of states.

We show that it suffices to randomise once at the start, for each rank (besides the maximum and minimum one), whether only sound actions should be suggested or whether we should play uniformly at random. This allows us to avoid stochastic updates and obtain an RRD strategy.

Theorem 5.2. *Let $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ be a game and $T \subseteq S$ be a set of states. There exists an RRD strategy $\mathcal{M}$ such that, for all $s_{\mathsf{init}} \in S$, if there exists a positively winning strategy from s_{init} for the objective $\mathsf{Safe}(T)$, then $\mathcal{M}$ is positively winning from s_{init}.*

Proof. We assume that there exists at least some state from which $\mathcal{P}_1$ wins positively, otherwise the result is immediate. We use properties of [26, Algorithm 3], which computes the set of almost-surely winning states in a concurrent reachability game, i.e., the complement of the set of positively winning states for the player with a safety objective. Each iteration of this algorithm computes two sets of states that are positively winning for $\mathcal{P}_1$ and (essentially) removes them from the state space. Therefore, it yields a non-increasing sequence $S = U_0 \supseteq U_1 \ldots \supseteq U_k$ of sets of states ($k + 2$ being double the number of iterations of the algorithm) such that $S \setminus U_k$ is the set of positively winning states for $\mathcal{P}_1$. In particular, note that $T \subseteq U_k$. Let, for all $s \in S$, $\mathsf{rk}(s)$ be the greatest j such that $s \in U_j$.

The sequence of sets $(U_j)_{1 \leq j \leq k}$ has the following property. For all states $s \in S$ such that $\mathsf{rk}(s) < k$, there exists a sound action $a^{(1)}_{\mathsf{sd}}(s) \in A^{(1)}(s)$ and a subset $A^{(2)}_\star(s) \subseteq A^{(2)}(s)$ such that (i) for all $a^{(2)} \in A^{(2)}_\star(s)$ and all $s' \in \mathsf{supp}(\delta(s, a^{(1)}_{\mathsf{sd}}(s), a^{(2)}))$, $\mathsf{rk}(s') \leq \mathsf{rk}(s)$, and (ii) for all $a^{(2)} \in A^{(2)}(s) \setminus A^{(2)}_\star(s)$, there exists an action $a^{(1)} \in A^{(1)}(s)$ and a state $s' \in \mathsf{supp}(\delta(s, a^{(1)}, a^{(2)}))$ such that $\mathsf{rk}(s') < \mathsf{rk}(s)$. These conditions follow from the structure of the algorithm. In particular, the pure memoryless strategy of $\mathcal{P}_1$ that only plays sound actions, when played from states of rank 0, is such that all of its outcomes satisfy $\mathsf{Safe}(T)$ (i.e., states of rank 0 are surely winning for $\mathcal{P}_1$).

We now define an RRD strategy. Let $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ such that $M = \{\mathsf{sd}, \mathsf{un}\}^{k-1}$ (sd and un respectively stand for sound and uniform). We let μ_{init} be a uniform distribution over M. Let $m = (m_j)_{1 \leq j \leq k-1} \in M$ and $s \in S$. If $\mathsf{rk}(s) = k$, we let $\alpha_{\mathsf{nxt}}(m, s)$ be arbitrary. Otherwise, if $\mathsf{rk}(s) = 0$ or $m_{\mathsf{rk}(s)} = \mathsf{sd}$, we let $\alpha_{\mathsf{nxt}}(m, s)$ be a Dirac distribution on $a^{(1)}_{\mathsf{sd}}(s)$. Otherwise (if $0 < \mathsf{rk}(s) < k$ and $m_{\mathsf{rk}(s)} = \mathsf{un}$), we let $\alpha_{\mathsf{nxt}}(m, s)$ be a uniform distribution over $A^{(1)}(s)$. The deterministic memory updates are trivial: for all $m \in M$, $s \in S$ and $\bar{a} \in \bar{A}(s)$, we let $\alpha_{\mathsf{up}}(m, s, \bar{a}) = m$. Given $w \in (S\bar{A})^*$, we let μ_w denote the distribution over memory states of $\mathcal{M}$ after w has taken place. For $m \in M$, we let σ_1^m be the strategy induced by the Mealy machine obtained by fixing the initial state of $\mathcal{M}$ to m.

We now prove that $\mathcal{M}$ induces a positively winning strategy from any state from which $\mathcal{P}_1$ has a positively winning strategy. Let s_0 be such a state and let σ_2 be an arbitrary strategy of $\mathcal{P}_2$. We use an inductive argument on histories, starting with the history $h_0 = s_0$. At step j of the induction, we assume that we have some history $h_j = w_j s_j$ consistent with σ_2 such that $\mathsf{rk}(s_j) < k - j$ and $\mathsf{supp}(\mu_{w_j}) = \{\mathsf{sd}, \mathsf{un}\}^{\mathsf{rk}(s_j)} \times M_j$ for some $M_j \subseteq \{\mathsf{sd}, \mathsf{un}\}^{k - \mathsf{rk}(s_j)}$ (this last hypothesis implies that h_j is consistent with $\mathcal{M}$, otherwise μ_{w_j} would not be defined). This induction hypothesis is clearly satisfied at step 0 of the induction (positively winning states have rank at most $k - 1$).

We consider two cases. First, we assume that, for all extensions $w_j h$ of h_j, if they are consistent with σ_2 and only sound actions are used by $\mathcal{P}_1$ in the suffix h, then $\mathsf{supp}(\sigma_2(w_j h)) \subseteq A^{(2)}_\star(\mathsf{last}(h))$. We remark that if $\mathsf{rk}(s_j) = 0$, we are necessarily in this case. We claim that for all extensions $w_j h$ of h_j consistent with σ_2 in which only sound $\mathcal{P}_1$ actions occur in h, it holds that all states in h have rank at most $\mathsf{rk}(s_j)$. This follows by a straightforward induction using the definition of sound actions and actions in sets $A^{(2)}_\star(s')$ (informally, the rank of states cannot increase at each step in this setting).

By the induction hypothesis, there exists some $m \in \mathsf{supp}(\mu_{w_j})$ such that $m_\ell = \mathsf{sd}$ for all $\ell \leq \mathsf{rk}(s_j)$. In particular, h_j is consistent with σ_1^m due to the definition of updates in $\mathcal{M}$. It follows from the above that all extensions of h_j that are consistent with both σ_1^m and σ_2 satisfy $\mathsf{Safe}(T)$ (because all targets have rank k). Therefore, only a subset of $\mathsf{Cyl}(h_j)$ of $\mathbb{P}_s^{\sigma_1^m, \sigma_2}$-measure zero is not included in $\mathsf{Safe}(T)$. Therefore, $\mathbb{P}_s^{\sigma_1^m, \sigma_2}(\mathsf{Safe}(T)) \geq \mathbb{P}_s^{\sigma_1^m, \sigma_2}(\mathsf{Cyl}(h_j)) > 0$. We conclude that $\mathbb{P}_s^{\sigma_1, \sigma_2}(\mathsf{Safe}(T)) > 0$ as $\mathbb{P}_s^{\sigma_1^m, \sigma_2}(\mathsf{Safe}(T))$ is the conditional probability of $\mathsf{Safe}(T)$ with respect to $\mathbb{P}_s^{\sigma_1, \sigma_2}$ assuming that the initial memory state is m.

Next, assume that there exists a history $w_j h$ extending h_j that is consistent with σ_2, in which only sound actions are used by $\mathcal{P}_1$ in the suffix h and such that $\mathsf{supp}(\sigma_2(w_j h)) \not\subseteq A^{(2)}_\star(\mathsf{last}(h))$. We assume that $w_j h$ is the shortest such extension of h_j. We fix $a^{(2)} \in \mathsf{supp}(\sigma_2)(w_j h) \setminus A^{(2)}_\star(\mathsf{last}(h))$, and $a^{(1)} \in A^{(1)}(\mathsf{last}(h))$ and $s_{j+1} \in \mathsf{supp}(\delta(\mathsf{last}(h), a^{(1)}, a^{(2)}))$ such that $\mathsf{rk}(s_{j+1}) < \mathsf{rk}(\mathsf{last}(h))$. We let $\bar{a} = (a^{(1)}, a^{(2)})$.

We define $h_{j+1} = w_j h \bar{a} s_{j+1}$ and show that it satisfies the induction hypothesis above. First, by construction, h_{j+1} is consistent with σ_2. Second, it holds that $\mathsf{rk}(\mathsf{last}(h)) \leq \mathsf{rk}(s_j)$. This can be shown by the same argument as in the first case, as only sound actions occur in h and all $\mathcal{P}_2$ actions taken in any state s in h are in $A_\star^{(2)}(s)$. It follows that $\mathsf{rk}(s_{j+1}) < \mathsf{rk}(s_j)$, implying that $\mathsf{rk}(s_{j+1}) < k-(j+1)$. Third, it can be shown by a straightforward induction that $\mathsf{supp}(\mu_w) = \mathsf{supp}(\mu_{w_j})$ for w such that $w_jh = w\mathsf{last}(h)$. The omitted inductive argument is based on the fact that all $\mathcal{P}_1$ actions are sound in h, are taken in states of rank at most $\mathsf{rk}(s_j)$ and $\mathsf{supp}(\mu_{w_j}) = \{\mathsf{sd},\mathsf{un}\}^{\mathsf{rk}(s_j)} \times M_j$. Finally, it holds that $\mathsf{supp}(\mu_{w_jh\bar{a}}) = \{m \in \mathsf{supp}(\mu_{w_j}) \mid m_{\mathsf{rk}(\mathsf{last}(h))} = \mathsf{un}\}$ if $a^{(1)} \neq a^{(1)}_{\mathsf{sd}}(\mathsf{last}(h))$ and $\mathsf{supp}(\mu_{w_jh\bar{a}}) = \mathsf{supp}(\mu_{w_j})$ otherwise. By the inductive hypothesis, we obtain that

$$\mathsf{supp}(\mu_{w_jh\bar{a}}) = \{\mathsf{sd},\mathsf{un}\}^{\mathsf{rk}(\mathsf{last}(h))-1} \times I \times \{\mathsf{sd},\mathsf{un}\}^{\mathsf{rk}(s_j)-\mathsf{rk}(\mathsf{last}(h))} \times M_j,$$

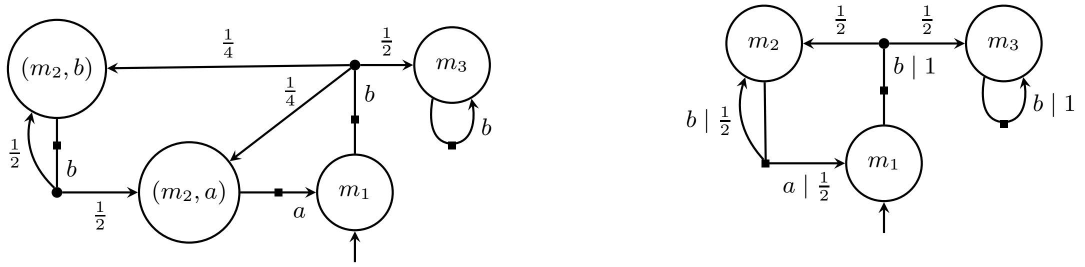
where $I = \{\mathsf{un}\}$ in the first case, and $I = \{\mathsf{sd},\mathsf{un}\}$ otherwise. This shows that we can continue the inductive argument with h_{j+1}.

The second case can occur in the worst case only in the $k-1$ first steps of the induction: at step k, s_k has rank 0, which guarantees we find ourselves in the first case. This concludes the proof that $\mathcal{M}$ is positively winning from s_0. □

5.4 RRD and DDR strategies are incomparable

We prove in this section that the classes RRD and DDR of finite-memory strategies are incomparable. We have previously shown Lemma 5.1, which states that RDD $\not\subseteq$ DDR and therefore implies that DRD $\not\subseteq$ DDR and RRD $\not\subseteq$ DDR. It remains to show that DDR $\not\subseteq$ RRD.

We illustrate a DDR strategy of $\mathcal{G}_{a,b}$ that has no outcome-equivalent RRD strategy in Figure 5.4a. For ease of analysis, we illustrate in Figure 5.4b a DRR strategy with fewer states that is outcome-equivalent to the Mealy machine depicted in Figure 5.4a. The DDR strategy of Figure 5.4a can be obtained by applying the construction of Theorem 4.3 to the Mealy machine of Figure 5.4b.



(a) A DDR strategy witnessing DDR $\not\subseteq$ RRD.

(b) An outcome-equivalent RRR strategy with fewer states.

Fig. 5.4: Outcome-equivalent strategies witnessing the non-inclusion DDR $\not\subseteq$ RRD. For the sake of readability, we do not label transitions by s as it is the sole state the Mealy machines can read in $\mathcal{G}_{a,b}$. We omit the probability of actions in Figure 5.4a as outputs are deterministic.

Intuitively, these strategies have a non-zero probability of never using action a after any history, while they have a positive probability of using action a at any time besides the first round and right after the action a occurs. The behaviour described above cannot be reproduced by an RRD strategy. There are two reasons to this.

First, along any play consistent with an RRD strategy, the support of the distribution over memory states cannot increase in size. Because of deterministic updates, the probability carried by a memory state m can only be transferred to at most one state, and may be lost if the used action cannot be used while in m. This property does not hold for strategies that have stochastic updates, such as those of Figure 5.4.

Second, one can force situations in which the size of the support of the distribution over memory states of an RRD strategy must decrease. If after a given history h, the action a has a positive probability of never being used despite being assigned a positive probability at each round after h, then at some point there must be some memory state of the RRD strategy that has positive probability and that assigns (via the next-move function) probability zero to action a. For instance, this is the case from the start with the RRD strategy depicted in Figure 5.2c. Intuitively, if at all times all memory states in the support of the distribution over memory states after the current history assign a positive probability to action a, the probability of using a at each round after h would be bounded from below by the smallest positive probability assigned to a by the next-move function. Therefore a would eventually be played almost-surely assuming h has taken place, contradicting the fact that there was a positive probability of never using action a after h. By using action a at a point in which some memory state in the support of the distribution over memory states assigns probability zero to a, the size of the support of the memory state distribution decreases.

By design of our DDR strategy, if one assumes the existence of an outcome-equivalent RRD strategy, then it is possible to construct a play along which the size of the support of the distribution over memory states of the RRD strategy decreases infinitely often. Because this size cannot increase along a play, this is not possible, i.e., there is no such RRD strategy. We formalise the sketch above in the proof of the following lemma.

Lemma 5.5. *There exists a DDR strategy of $\mathcal{P}_1$ in $\mathcal{G}_{a,b}$ such that there is no outcome-equivalent RRD strategy.*

Proof. Consider the Mealy machine $\mathcal{M} = (M, m_1, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ depicted in Figure 5.4b. We recall that $\mathcal{M}$ is a DRR Mealy machine that is outcome-equivalent to the DDR strategy illustrated in Figure 5.4a. It therefore suffices to show that there are no RRD strategies that are outcome-equivalent to $\mathcal{M}$ to end this proof.

Let σ_1 denote the strategy induced by $\mathcal{M}$. Intuitively, σ_1 operates as follows. It always uses b in the first round and otherwise has a positive probability of never using action a while always having a positive probability of playing a at any round. Whenever the action a is used, the behaviour of the strategy resets in the following sense: witnessing action a ensures that $\mathcal{M}$ finds itself in memory state m_1 after the update, thus the strategy repeats its behaviour from the initial state of $\mathcal{M}$.

Lemma 2.1 ensures that we need only study plays consistent with σ_1 for matters related to outcome-equivalence. The finite sequences of actions that can be suggested by this strategy can be described by the regular expression $(b^+a)^*b^*$. Therefore, we require only the definition of σ_1 over histories in which the underlying sequence of actions is in this language. For any $w \in (\{s\}\{a, b\})^*$, let μ_w denote the distribution over memory states of $\mathcal{M}$ after w has taken place. It can be shown by induction that for any $w \in ((\{s\}\{b\})^+\{s\}\{a\})^*$ and $k \geq 1$, we have $\mu_w(m_1) = 1$ and $\mu_{w(sb)^k}(m_2) = 1 - \mu_{w(sb)^k}(m_3) = \frac{1}{2^{k-1}+1}$. It follows that for any history h consistent with σ_1 of the form s or $h'as$ and $k \geq 1$, we have $\sigma_1(h)(b) = 1$ and $\sigma_1(h(bs)^k)(a) = 1 - \sigma_1(h(bs)^k)(b) = \frac{1}{2^k+2}$.

We show that for any history h consistent with σ_1 in which the last used action is a, it holds that $\mathbb{P}_s^{\sigma_1}(\{h(bs)^\omega\}) > 0$, i.e., there is a positive probability of a never being played again after any occurrence of a. Let h be one such history.

We first show that $\mathbb{P}_s^{\sigma_1}(\{h(bs)^\omega\}) = \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(h)) \cdot \mathbb{P}_s^{\sigma_1}(\{(sb)^\omega\})$. We have, for any $k \in \mathbb{N}$, $\sigma_1(h(bs)^k)(b) = \sigma_1(s(bs)^k)(b)$ by definition of σ_1. Furthermore, the sequences $(\mathsf{Cyl}(s(bs)^k))_{k\in\mathbb{N}}$ and $(\mathsf{Cyl}(h(bs)^k))_{k\in\mathbb{N}}$ respectively decrease when taking their intersections to the singletons $\{(sb)^\omega\}$ and $\{h(bs)^\omega\}$. We obtain the following equations from the definition of $\mathbb{P}_s^{\sigma_1}$:

$$\begin{aligned}
\mathbb{P}_s^{\sigma_1}(\{h(bs)^\omega\}) &= \lim_{k\to\infty} \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(h(bs)^k)) \\
&= \lim_{k\to\infty} \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(h)) \cdot \prod_{\ell=0}^{k-1} \sigma_1(h(bs)^\ell)(b) \\
&= \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(h)) \cdot \lim_{k\to\infty} \cdot \prod_{\ell=0}^{k-1} \sigma_1(s(bs)^\ell)(b) \\
&= \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(h)) \cdot \lim_{k\to\infty} \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(s(bs)^k)) \\
&= \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(h)) \cdot \mathbb{P}_s^{\sigma_1}(\{(sb)^\omega)\}).
\end{aligned}$$

In light of the above, to show that $\mathbb{P}_s^{\sigma_1}(\{h(bs)^\omega\}) > 0$, it suffices to establish that $\mathbb{P}_s^{\sigma_1}(\{(sb)^\omega\}) > 0$ because h is assumed to be consistent with σ_1. It can be shown that $\mathbb{P}_s^{\sigma_1}(\{(sb)^\omega\}) = \frac{1}{2}$ as follows:

$$\begin{aligned}\mathbb{P}_s^{\sigma_1}(\{(sb)^\omega\}) &= \lim_{k\to\infty} \mathbb{P}_s^{\sigma_1}(\mathsf{Cyl}(s(bs)^k)) \\ &= \lim_{k\to\infty} 1 \cdot \prod_{j=1}^{k-1} \frac{2^j+1}{2^j+2} \\ &= \lim_{k\to\infty} \frac{1}{2^{k-1}} \cdot \prod_{j=1}^{k-1} \frac{2^j+1}{2^{j-1}+1} \\ &= \lim_{k\to\infty} \frac{1}{2^{k-1}} \cdot \frac{2^{k-1}+1}{2^{1-1}+1} = \frac{1}{2};\end{aligned}$$

the product of the probabilities of b being played in each round is simplified using the fact that the denominator of a term is double the numerator of the previous one. This closes the proof of our claimed inequality.

We now show that no RRD strategy is outcome-equivalent to σ_1. Let $\mathcal{N} = (N, \nu_{\mathsf{init}}, \beta_{\mathsf{nxt}}, \beta_{\mathsf{up}})$ be an RRD Mealy machine and let τ_1 be the strategy it induces. For any $w \in (\{s\}\{a,b\})^*$, let ν_w denote the distribution over memory states in N after w has taken place under $\mathcal{N}$.

The remainder of the proof is structured as follows; we prove two properties of RRD strategies and use them to show that τ_1 cannot be outcome-equivalent to σ_1. The first claim if that for any history $h = ws$ consistent with τ_1 and action $c \in \{a, b\}$ such that $\tau_1(h)(c) > 0$, we have $|\mathsf{supp}(\nu_w)| \geq |\mathsf{supp}(\nu_{wsc})|$, i.e., the size of the support of the distribution over memory states of $\mathcal{N}$ does not increase as the play progresses. The second claim is that for any history h consistent with τ_1, if the probability of a never appearing again after h is non-zero, i.e., $\mathbb{P}_s^{\tau_1}(\{h(bs)^\omega\}) > 0$, and for any $k \in \mathbb{N}$, we have $\tau_1(h(bs)^k)(a) > 0$, then there exists some $k_0 \in \mathbb{N}$ such that $|\mathsf{supp}(\nu_{h(bs)^{k_0}b})| > |\mathsf{supp}(\nu_{h(bs)^{k_0+1}a})|$.

Let us first prove the first claim. It follows from a careful inspection of how the distribution over memory states is updated from one step to the next. Let $h = ws$ be consistent with σ_1 and $c \in \{a, b\}$ such that $\tau_1(h)(c) > 0$. For any memory state $n \in N$, recall that

$$\nu_{wsc}(n) = \frac{\sum_{n' \in N} \nu_w(n') \cdot \beta_{\mathsf{up}}(n', s, c)(n) \cdot \beta_{\mathsf{nxt}}(n', s)(c)}{\sum_{m' \in M} \nu_w(n') \cdot \beta_{\mathsf{nxt}}(n', s)(c)}.$$

Because updates are deterministic, for any given $n' \in N$, there is a unique $n \in N$ such that $\beta_{\mathsf{up}}(n', s, c)(n) = 1$. Therefore any element in $\mathsf{supp}(\nu_w)$ transfers its probability to at most one memory state when deriving ν_{wsc}. This ends the proof of the first claim. We note (for the proof of the second claim) that if $n' \in \mathsf{supp}(\nu_w)$ is such that $\beta_{\mathsf{nxt}}(n', s)(c) = 0$, then n' does not transfer its probability to any state, and in this case, we have $|\mathsf{supp}(\nu_w)| > |\mathsf{supp}(\nu_{wsc})|$.

We now move on to the second claim. Let h be consistent with τ_1 and assume that $\mathbb{P}_s^{\tau_1}(\{h(bs)^\omega\}) > 0$, and for any $k \in \mathbb{N}$, we have $\tau_1(h(bs)^k)(a) > 0$. In light of the comment above regarding the second claim, it suffices to show that for some $k_0 \in \mathbb{N}$, we have some $n \in \mathsf{supp}(\nu_{h(bs)^{k_0}b})$ such that $\beta_{\mathsf{nxt}}(n, s)(a) = 0$. Assume towards a contradiction that this is not the case, i.e., for all $k \in \mathbb{N}$ and all $n \in \mathsf{supp}(\nu_{h(bs)^k b})$, we have $\beta_{\mathsf{nxt}}(n, s)(a) > 0$. Let $k \in \mathbb{N}$. We show that the probability $\tau_1(h(bs)^{k+1})(a)$ is bounded below by a positive constant independent of k. This follows from the assumption that $\beta_{\mathsf{nxt}}(n, s)(a) > 0$ for all $n \in \mathsf{supp}(\nu_{h(bs)^k b})$ via the relations

$$\begin{aligned}\tau_1(h(bs)^{k+1})(a) &= \sum_{n \in N} \nu_{h(bs)^k b}(n) \cdot \beta_{\mathsf{nxt}}(n, s)(a) \\ &\geq \sum_{n \in N} \nu_{h(bs)^k b}(n) \cdot \min_{n' \in N^{a>0}} \beta_{\mathsf{nxt}}(n', s)(a) \\ &= \min_{n' \in N^{a>0}} \beta_{\mathsf{nxt}}(n', s)(a) > 0,\end{aligned}$$

where $N^{a>0} = \{n \in N \mid \beta_{\mathsf{nxt}}(n, s)(a) > 0\}$. It follows that the action a must be used almost-surely assuming h has taken place, contradicting the fact that $\mathbb{P}_s^{\tau_1}(\{h(bs)^\omega\}) > 0$. This ends the proof of the second claim.

We now show that τ_1 cannot be outcome-equivalent to σ_1 by contradiction. Assume τ_1 is outcome-equivalent to σ_1. Due to the properties of σ_1 shown above, we can repeatedly use the two claims above to construct a sequence of non-zero natural numbers $(k_\ell)_{\ell \in \mathbb{N}}$ such that $(|\mathsf{supp}(\nu_{w_\ell})|)_{\ell \in \mathbb{N}}$ is an infinite

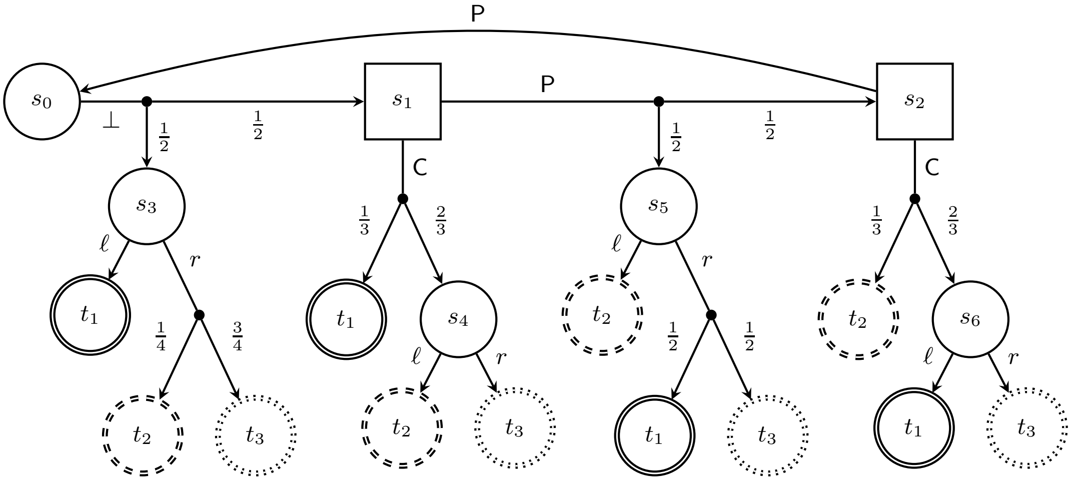


Fig. 5.5: A turn-based stochastic game with multiple reachability objectives [46]. Circles and squares respectively represent states controlled by $\mathcal{P}_1$ and $\mathcal{P}_2$. The only action enabled for players who do not control a state is $\perp$. States t_1, t_2 and t_3 are drawn repeatedly for clarity (duplicates all represent the same state). Actions P and C of $\mathcal{P}_2$ stand for proceed and check respectively.

decreasing sequence of natural numbers, where $w_0 = \varepsilon$ and for all $\ell \in \mathbb{N}$, $w_{\ell+1} = w_\ell(sb)^{k_\ell}sa$. This contradicts the well-order of $\mathbb{N}$. This shows that there are no RRD strategies that are outcome-equivalent to σ_1. □

As in the previous sections, we provide a game and a specification that cannot be accomplished using an RRD strategy, but can be accomplished using a DDR strategy. In the following example, we consider a two-player turn-based game with several reachability objectives with absorbing targets. The goal is to construct, if it exists, a strategy that ensures given thresholds for several reachability objectives at once.

Example 5.3. We consider the two-player turn-based game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ depicted in Figure 5.5 (ownership of vertices is distinguished by their shape), originating from [46]. As $\mathcal{G}$ is turn-based, we lighten the notation of histories and plays by only indicating the action of the player in control of the state. We also simplify notations for updates of Mealy machines by only taking in account the actions we keep in plays. We let $A = A^{(1)} \cup A^{(2)}$ denote the set of actions. We consider three targets: $T_j = \{t_j\}$ for $j \in \{1, 2, 3\}$.

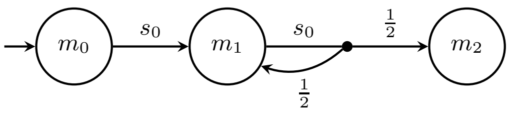


Fig. 5.6: A Mealy machine update scheme for the game of Figure 5.5. Updates that do not change the memory state are not depicted.

In [46], it is shown that there is no DRD strategy σ_1 of $\mathcal{P}_1$ such that for all strategies σ_2 of $\mathcal{P}_2$, $\mathbb{P}^{\sigma_1,\sigma_2}_{s_0}(\mathsf{Reach}(T_j)) \geq \frac{1}{3}$ for all $j \in \{1, 2, 3\}$, despite there existing an infinite-memory one. We prove that (i) there is no RRD strategy that satisfies this specification and (ii) there exists a DDR strategy that does.

We let, for $k \in \mathbb{N}$, $h_k = s_0(\perp s_1 \mathsf{P} s_2 \mathsf{P} s_0)^k$. A description of satisfactory strategies is provided in the technical report [47, Appendix B]. A strategy σ_1 of $\mathcal{P}_1$ ensures that all targets are each visited with probability $\frac{1}{3}$ if for all $k \in \mathbb{N}$, $\sigma_1(h_k \perp s_3)(\ell) = 1 - \frac{1}{3 \cdot 2^{k-1}}$, $\sigma_1(h_k \perp s_1 \mathsf{C} s_4)(\ell) = 1 - \frac{1}{2^{k+2}}$, $\sigma_1(h_k \perp s_1 \mathsf{P} s_5)(\ell) = 1 - \frac{1}{3 \cdot 2^k}$ and $\sigma_1(h_k \perp s_1 \mathsf{C} s_6)(\ell) = 1 - \frac{1}{2^{k+2}}$, and for all $k \in \mathbb{N}$, the first two equations are necessary to comply with the specification.

Let $\mathcal{M}$ be an RRD strategy and let $\tau_1^{\mathcal{M}}$ be its induced strategy. We show that $\tau_1^{\mathcal{M}}$ cannot satisfy the multi-objective query by showing that the set of distributions $\{\tau_1^{\mathcal{M}}(h_k \perp s_3) \mid k \in \mathbb{N}\}$ must be a finite set, which is incompatible with the requirements given above.

Let μ_w denote the distribution over memory states after $w \in (SA)^*$ has taken place under $\mathcal{M}$. For all $k \in \mathbb{N}$ and $m \in M$, it holds that $\mu_{h_k \bot s_3}(m) = \sum_{m' \in M'} \mu_{\mathsf{init}}(m')$ for some $M' \subseteq M$ (which depends on both k and m). This follows from the equations for the updates of the distributions μ_w. In all states along $h_k\bot$, $\mathcal{P}_1$ only has a single action. Furthermore, $\mathcal{M}$ has deterministic updates. Therefore, if w and wsa are prefixes of $h_k\bot$, for all memory states $m \in M$, we obtain $\mu_{wsa}(m)$ is the sum of $\mu_w(m')$ for all memory states m' such that $\alpha_{\mathsf{up}}(m', s, a) = m$. In particular, this implies that the set of distributions $\{\mu_{h_k\bot} \mid k \in \mathbb{N}\}$ is finite, which shows that $\{\tau_1^{\mathcal{M}}(h_k \bot s_3) \mid k \in \mathbb{N}\}$ is a finite set by definition of the strategy induced by a Mealy machine.

We now describe a Mealy machine $\mathcal{N}$ that induces a strategy that coincides with σ_1 over $\mathsf{Cyl}(s_0)$, i.e., that ensures a probability of $\frac{1}{3}$ for all three reachability objectives. Once more, we provide a DRR strategy that can be transformed into an outcome-equivalent DDR strategy via the construction underlying Theorem 4.3. We depict the relevant update scheme in Figure 5.6; updates that do not change the current memory state are omitted from the figure. Let ν_w denote the distribution over memory states of $\mathcal{N}$ after $w \in (SA)^*$ has taken place under $\mathcal{N}$. Let $k \in \mathbb{N}$. Below, we are interested in the distribution over memory states only for $w_k \in \{h_k\bot, h_k\bot s_1\mathsf{C}, h_k\bot s_1\mathsf{P}, h_k\bot s_1\mathsf{P}s_2\mathsf{C}\}$: it can be shown by a straightforward induction that we have $\nu_{w_k}(m_1) = 1 - \nu_{w_k}(m_2) = \frac{1}{2^k}$.

We now specify the next-move function of $\mathcal{N}$ and describe the strategy $\sigma_1^{\mathcal{N}}$ induced by $\mathcal{N}$. We let $\alpha_{\mathsf{nxt}}(m_0, s)$ be an arbitrary Dirac distribution for all states $s \in \{s_3, s_4, s_5, s_6\}$ (we require Dirac distributions so our Mealy machine has an outcome-equivalent DDR strategy). For s_3, we let $\alpha_{\mathsf{nxt}}(m_1, s_3)(r) = \frac{2}{3}$ and $\alpha_{\mathsf{nxt}}(m_2, s_3)(\ell) = 1$. It follows that for all $k \in \mathbb{N}$, we have $\sigma_1^{\mathcal{N}}(h_k\bot s_3)(r) = \frac{2}{3\cdot 2^k} = \frac{1}{3\cdot 2^{k-1}}$. For s_4, we let $\alpha_{\mathsf{nxt}}(m_1, s_4)(r) = \frac{1}{4}$ and $\alpha_{\mathsf{nxt}}(m_2, s_4)(\ell) = 1$. We obtain that for all $k \in \mathbb{N}$, we have $\sigma_1^{\mathcal{N}}(h_k\bot s_2\mathsf{C}s_4)(r) = \frac{1}{4\cdot 2^k} = \frac{1}{2^{k+2}}$. For s_5, we let $\alpha_{\mathsf{nxt}}(m_1, s_5)(r) = \frac{1}{3}$ and $\alpha_{\mathsf{nxt}}(m_2, s_5)(\ell) = 1$. For all $k \in \mathbb{N}$, it holds that $\sigma_1^{\mathcal{N}}(h_k\bot s_2\mathsf{P}s_5)(r) = \frac{1}{3\cdot 2^k}$. Finally, for s_6, we let $\alpha_{\mathsf{nxt}}(m_1, s_6)(r) = \frac{1}{4}$ and $\alpha_{\mathsf{nxt}}(m_2, s_6)(\ell) = 1$. We conclude that for all $k \in \mathbb{N}$, $\sigma_1^{\mathcal{N}}(h_k\bot s_2\mathsf{P}s_2\mathsf{C}s_6)(r) = \frac{1}{4\cdot 2^k} = \frac{1}{2^{k+2}}$. This shows that $\sigma_1^{\mathcal{N}}$ ensures all reachability objectives are satisfied with probability at least $\frac{1}{3}$. ◁

Consider a turn-based stochastic game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$ and targets $T_1, \ldots, T_k \subseteq S$. The general form of the problem treated in the example above is to decide, given an initial state $s_{\mathsf{init}} \in S$ and a threshold vector $q \in ([0,1] \cap \mathbb{Q})^k$ whether there exists a strategy σ_1 of $\mathcal{P}_1$ such that for all strategies σ_2 of $\mathcal{P}_2$, we have $\mathbb{P}_{s_{\mathsf{init}}}^{\sigma_1,\sigma_2}(\mathsf{Reach}(T_j)) \geq q_j$ for all $j \in \{1, \ldots, k\}$. It is not known whether RRR strategies of $\mathcal{P}_1$ suffice to provide a positive answer whenever possible in general. However, finite-memory strategies suffice to approximate any vector for which the problem has a positive answer. More precisely, if $\mathcal{P}_1$ can ensure q from $s_{\mathsf{init}} \in S$, then for all $\varepsilon > 0$, $\mathcal{P}_1$ has an DRD strategy such that for all strategies σ_2 of $\mathcal{P}_2$ and all $j \in \{1, \ldots, k\}$, it holds that $\mathbb{P}_{s_{\mathsf{init}}}^{\sigma_1,\sigma_2}(\mathsf{Reach}(T_j)) \geq q_j - \varepsilon$ [46, 48].

6 Extension: multiplayer games

In the previous sections, we have only considered two-player games. We show that the lattice of Figure 1.1 extends to games with more than two players.

Let $n \geq 1$ be a number of players. Formally, an n-player concurrent stochastic game is a tuple $\mathcal{G} = (S, (A^{(i)})_{1 \leq i \leq n}, \delta)$ where S is a non-empty finite set of states, $A^{(i)}$ is a finite set of actions for each player and $\delta \colon S \times A^{(1)} \times \ldots \times A^{(n)} \to \mathcal{D}(S)$ is a probabilistic transition function. We reuse the notation $\bar{A} = A^{(1)} \times \ldots \times A^{(n)}$. We impose the same constraints as in the two-player case regarding actions enabled in states, i.e., whether an action is available to a player is independent of the choices of others. Plays, histories, strategies, Mealy machines and probability distributions over plays induced by strategies are defined in a similar way as in the two-player setting.

The definition of outcome-equivalence can be naturally extended to multi-player games. Instead of quantifying universally over strategies of the other player as is done in the two-player setting, one quantifies universally over strategies of all other players in the definition of outcome-equivalence. Formally, two strategies σ_1 and τ_1 of $\mathcal{P}_1$ are outcome-equivalent if for all strategies σ_i of $\mathcal{P}_i$ for $2 \leq i \leq n$ and all $s \in S$, $\mathbb{P}_s^{\sigma_1,\sigma_2,\ldots,\sigma_n} = \mathbb{P}_s^{\tau_1,\sigma_2,\ldots,\sigma_n}$.

A single (fictitious) player derived from a coalition of players has access to more behaviours than the coalition, as the single player can randomise over action profiles whereas individual players can only randomise over their own set of actions. This implies that all probability distributions over action profiles that can be induced by strategies of the players of the coalition playing individually can be simulated by the fictitious player, but the inverse is not true. This is the crux of the argument showing that our results carry over to the multi-player setting.

Theorem 6.1. *The taxonomy of Figure 1.1 established in two-player games extends to multiplayer games.*

Proof. All results that witness that two classes in the lattice of Figure 1.1 are separated (i.e., Lemmas 5.1, 5.3, 5.4 and 5.5) hold in one-player games, which are a subclass of multiplayer games.

We now prove that the inclusion results extend to this setting. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two classes of finite-memory strategies referred to in Figure 1.1 such that the lattice asserts that $\mathcal{C}_1 \subseteq \mathcal{C}_2$. Let $\mathcal{G} = (S, (A^{(i)})_{1\leq i\leq n}, \delta)$ be an n-player game. In the following argument, we only consider strategies of $\mathcal{P}_1$ to simplify notation. We let $\mathcal{G}' = (S, A^{(1)}, \prod_{2\leq i\leq n} A^{(i)}, \delta)$ be the two-player (coalition) game in which the players other than $\mathcal{P}_1$ are grouped together. Although the sets of histories and plays of $\mathcal{G}$ and $\mathcal{G}'$ differ syntactically (due to the nature of action tuples), there is a natural bijection between these sets. For this reason, we identify them. Therefore, all strategies of $\mathcal{P}_1$ in $\mathcal{G}$ are strategies of $\mathcal{P}_1$ in $\mathcal{G}'$ and vice-versa.

Let $\sigma_1 \in \mathcal{C}_1$ be a strategy of $\mathcal{P}_1$. Because $\mathcal{C}_1 \subseteq \mathcal{C}_2$ holds for two-player games, there exists a strategy $\tau_1 \in \mathcal{C}_2$ such that σ_1 and τ_1 are outcome-equivalent in $\mathcal{G}'$. We claim that σ_1 and τ_1 are outcome-equivalent in $\mathcal{G}$. Let $\sigma_2, \ldots, \sigma_n$ be strategies of players other than $\mathcal{P}_1$ and $s \in S$ be an initial state. Consider the strategy τ_2 of the second player in $\mathcal{G}'$ defined by $\tau_2(h)(a^{(2)}, \ldots, a^{(n)}) = \prod_{2\leq i\leq n} \sigma_i(h)(a^{(i)})$ for all $h \in \mathsf{Hist}(\mathcal{G})$ and all $a^{(i)} \in A^{(i)}$ for $2 \leq i \leq n$. By definition of distributions induced by plays and outcome-equivalence of σ_1 and τ_1 in $\mathcal{G}'$, we obtain $\mathbb{P}^{\sigma_1,\sigma_2,\ldots,\sigma_n}_{\mathcal{G},s} = \mathbb{P}^{\sigma_1,\tau_2}_{\mathcal{G}',s} = \mathbb{P}^{\tau_1,\tau_2}_{\mathcal{G}',s} = \mathbb{P}^{\tau_1,\sigma_2,\ldots,\sigma_n}_{\mathcal{G},s}$ (where the subscript also indicates the relevant game), ending the proof. □

7 Extension: imperfect information

This section discusses games of imperfect information, and how our results transfer to this setting. In Section 7.1, we introduce definitions and terminology for games of imperfect information. We discuss finite-memory strategies in this setting in Section 7.2. Finally, we close with Section 7.3, in which we argue that the lattice of Figure 1.1 transfers to games with perfect recall and provide an adaptation for games of imperfect recall.

7.1 Games of imperfect information

We consider two-player stochastic games of imperfect information played on graphs. Unlike games of perfect information, the players are not fully informed of the current state of the play and the actions that are used along the play. Instead, they perceive an *observation* for each state and action, and this observation may be shared between different states and actions, making them indistinguishable. These observations are not shared between the players; each player perceives the ongoing play differently.

We formalise this game model. A *concurrent stochastic game of imperfect information* is defined as a tuple $\Gamma = (S, A^{(1)}, A^{(2)}, \delta, \mathcal{Z}_1, \mathsf{Obs}_1, \mathcal{Z}_2, \mathsf{Obs}_2)$ where $(S, A^{(1)}, A^{(2)}, \delta)$ is a game of perfect information, $\mathcal{Z}_i$ is a finite set of observations of $\mathcal{P}_i$ for $i \in \{1, 2\}$ and $\mathsf{Obs}_i\colon S \cup A^{(1)} \cup A^{(2)} \to \mathcal{Z}_i$ is the observation function of $\mathcal{P}_i$, which assigns an observation to each state and action. We require that for any $i \in \{1, 2\}$, for any two states $s, s' \in S$, $\mathsf{Obs}_i(s) = \mathsf{Obs}_i(s')$ implies $\bar{A}(s) = \bar{A}(s')$, i.e., in two indistinguishable states, the same actions are available to both players. We fix Γ for the remainder of the section and let $\mathcal{G}$ denote the underlying game of perfect information.

Plays and histories of Γ are respectively defined as plays and histories of $\mathcal{G}$. We reuse the notations $\mathsf{Plays}(\Gamma)$ and $\mathsf{Hist}(\Gamma)$ for the sets of plays of Γ and histories of Γ respectively. We extend the observation functions to pairs of actions and to histories. For any $\bar{a} = (a^{(1)}, a^{(2)}) \in \bar{A}$, we let $\mathsf{Obs}_i(\bar{a}) = (\mathsf{Obs}_i(a^{(1)}), \mathsf{Obs}_i(a^{(2)}))$ and for all histories $h = s_0\bar{a}_0 \ldots s_k$ of Γ, we let $\mathsf{Obs}_i(h) = \mathsf{Obs}_i(s_0)\mathsf{Obs}_i(\bar{a}_0) \ldots \mathsf{Obs}_i(s_k)$. This extension is used to define strategies in games of imperfect information.

In our setting, $\mathcal{P}_i$ has *perfect recall* if $\mathcal{P}_i$ can distinguish their own actions. Formally, $\mathcal{P}_i$ has perfect recall if the set of actions $A^{(i)}$ is included in the set $\mathcal{Z}_i$ and that for all $a^{(i)} \in A^{(i)}$ and $x \in S \cup A^{(1)} \cup A^{(2)}$, $\mathsf{Obs}_i(x) = a^{(i)}$ if and only if $x = a^{(i)}$.

In Γ, players can only rely on the observations they perceive to select actions. A pure *(observation-based) strategy* of Γ is a function $\sigma_i\colon \mathsf{Obs}_i(\mathsf{Hist}(\Gamma)) \to A^{(i)}$. Randomised strategies can be defined as mixed strategies (i.e., distributions over pure observation-based strategies) or behavioural strategies. Specifically, an *observation-based behavioural strategy* is a function $\sigma_i\colon \mathsf{Obs}_i(\mathsf{Hist}(\Gamma)) \to \mathcal{D}(A^{(i)})$. We will refer to (behavioural) strategies of the underlying game of perfect information $\mathcal{G}$ as *history-based strategies* to distinguish them from observation-based ones.

In contrast to the perfect information setting, if we do not assume perfect recall, there need not be an equivalence between behavioural and mixed strategies. Thankfully, randomised strategies (be they mixed or behavioural) of Γ are a subclass of history-based strategies. This allows us to directly reuse notions previously defined for history-based strategies. For instance, the probability distributions over plays of Γ induced by a pair of strategies from an initial state is the corresponding distribution in $\mathcal{G}$. Furthermore, we avoid the need to consider mixed strategies explicitly this way.

We remark that the equivalent definition of outcome-equivalence for two strategies of $\mathcal{P}_1$ formulated in Lemma 2.1 also extends to the imperfect information setting. On the one hand, $\mathcal{P}_2$ has access to fewer strategies, therefore the condition given in the lemma implies outcome-equivalence (the proof establishes a stronger statement). On the other hand, the other direction requires strategies of $\mathcal{P}_2$ that are consistent with the histories considered in the proof; it suffices to consider a strategy of $\mathcal{P}_2$ that selects all available actions at random at all times for the argument to work.

7.2 Finite-memory strategies

A strategy is *finite-memory* if it is induced by a (stochastic) Mealy machine that reads observations instead of states and actions. Formally, we define an *observation-based Mealy machine* of $\mathcal{P}_i$ as a tuple $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ where M is a finite set of memory states, μ_{init} is an initial distribution over M, $\alpha_{\mathsf{up}} \colon M \times \mathcal{Z}_i^3 \to \mathcal{D}(M)$ is the update function and $\alpha_{\mathsf{nxt}} \colon M \times \mathcal{Z}_i \to \mathcal{D}(A^{(i)})$ is the next-move function.

An observation-based Mealy machine is a special case of a Mealy machine whose updates and outputs must coincide given inputs with the same observations. We can thus derive a history-based strategy from an observation-based Mealy machine in the same way as in the perfect information setting.

To transfer our results on games of perfect information to games of imperfect information, we reuse the same classification of Mealy machines with three-letter acronyms for observation-based Mealy machines. As was the case in the earlier sections, we will abusively say, e.g., $\mathcal{M}$ is an RRR observation-based strategy to mean that $\mathcal{M}$ is an observation-based Mealy machine with stochastic initialisation, outputs and updates, and avoid referring to the observation-based strategy it induces in this way.

In general, an observation-based Mealy machine may not induce a behavioural strategy of Γ. This can be illustrated with a simple RDD strategy.

Example 7.1. We build a one-player game of imperfect information $\Gamma_{a,b}$ from the game $\mathcal{G}_{a,b}$ of Figure 5.1. We assign to s, a and b a shared observation o. We consider the Mealy machine depicted in Figure 5.2a; note that its updates only depend on the memory state and not on the input and outputs. The strategy it induces, which we will denote by σ_1, has a uniform probability of only playing a or only playing b.

No observation-based behavioural strategy is outcome-equivalent to σ_1. Let $\tau_1 \colon \{o\}(\{o\}^2)^* \to \mathcal{D}(\{a, b\})$ be a behavioural strategy. For it to be outcome-equivalent to σ_1, τ_1 has differentiate between the histories sas and sbs and play action a and b respectively following these histories. However, because both strategies share the same sequence of observations, τ_1 cannot be outcome-equivalent to σ_1. ◁

We provide two sufficient conditions that ensure that observation-based Mealy machine induce a behavioural strategy. The first one we present introduces a restriction on the games. The second one involves no assumptions on games, but instead considers a restricted class of Mealy machines.

First, we show that all finite-memory strategies are behavioural in games with perfect recall. Intuitively, the distribution over memory states depends heavily on the sequence of actions used by the considered player; the choice of actions conditions the distribution over memory states at each time it is updated. The visibility of actions makes it so the distribution over memory states depends only on the observations fed to the Mealy machine.

Lemma 7.1. *Let $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be an observation-based Mealy machine of $\mathcal{P}_i$. Assume that $\mathcal{P}_i$ has perfect recall in Γ. Then the strategy induced by $\mathcal{M}$ is an observation-based behavioural strategy.*

Proof. Let μ_w denote the distribution over memory states of $\mathcal{M}$ after w has taken place, for $w \in (S\bar{A})^*$. By definition of the strategy induced by a Mealy machine, it suffices to show the following: for all $w, v \in (S\bar{A})^*$ that are mapped to the same sequence of observations, we have $\mu_w = \mu_v$.

Let $w, v \in (S\bar{A})^*$ such that w and v are mapped to the same sequence of observations. We proceed by induction on the length of the considered sequence w (which matches that of v). At the start of a play, an initial memory state is drawn following μ_{init}. Hence the distribution over memory states after the empty word ε is $\mu_\varepsilon = \mu_{\mathsf{init}}$. In this case, there is nothing to show for the uniformity argument.

We now assume the following by induction: for $w = s_0\bar{a}_0 \dots s_k\bar{a}_k$, the distribution μ_w over M is well-defined and coincides with μ_v for $v = t_0\bar{b}_0 \dots t_k\bar{b}_k$ that can be mapped to the same sequence of observations as w. We consider $w' = ws_{k+1}\bar{a}_{k+1}$ and $v' = vt_{k+1}\bar{b}_{k+1}$ that share the same sequence of observations. We describe $\mu_{w'}$, then infer that $\mu_{w'} = \mu_{v'}$.

Due to the visibility of actions, we have $a_{k+1}^{(i)} = \mathsf{Obs}_i(a_{k+1}^{(i)}) = b_{k+1}^{(i)}$. We distinguish two cases: $\mu_{w'}$ is well-defined or it is not. First, if for all $m \in \mathsf{supp}(\mu_w)$, we have $\alpha_{\mathsf{nxt}}(m, \mathsf{Obs}_i(s_{k+1}))(a_{k+1}^{(i)}) = 0$, then $\mu_{w'}$ and $\mu_{v'}$ are both undefined (i.e., w' and v' are inconsistent with $\mathcal{M}$). Therefore, we assume that there is $m \in \mathsf{supp}(\mu_w)$ such that $\alpha_{\mathsf{nxt}}(m, \mathsf{Obs}_i(s_{k+1}))(a_{k+1}^{(i)}) > 0$. In this case, we have, for any memory state $m \in M$,

$$\mu_{w'}(m) = \frac{\sum_{m' \in M} \mu_w(m') \cdot \alpha_{\mathsf{up}}(m', \mathsf{Obs}_i(s_{k+1}), \mathsf{Obs}_i(\bar{a}_{k+1}))(m) \cdot \alpha_{\mathsf{nxt}}(m', \mathsf{Obs}_i(s_{k+1}))(a_{k+1}^{(i)})}{\sum_{m' \in M} \mu_w(m') \cdot \alpha_{\mathsf{nxt}}(m, \mathsf{Obs}_i(s_{k+1}))(a_{k+1}^{(i)})}.$$

The equation for $\mu_{v'}$ is the same as above, except s_{k+1} and $a_{k+1}^{(3-i)}$ are respectively replaced with t_{k+1} and $b_{k+1}^{(3-i)}$. It follows immediately from $\mathsf{Obs}_i(s_{k+1}) = \mathsf{Obs}_i(t_{k+1})$ and $\mathsf{Obs}_i(a_{k+1}^{(3-i)}) = \mathsf{Obs}_i(b_{k+1}^{(3-i)})$ that $\mu_{w'} = \mu_{v'}$. This ends the inductive argument and the proof. □

We have seen through Example 7.1 that when lifting the perfect recall hypothesis, Mealy machines with randomised initialisation need not induce behavioural strategies. A similar claim can be shown for strategies with randomised updates, e.g., by adapting the RDD example so the randomised initialisation is emulated by a stochastic memory update after the first round of the game. On the other hand, DRD strategies always induce behavioural strategies.

Lemma 7.2. *Let $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be a DRD strategy of $\mathcal{P}_i$ in $\mathit{\Gamma}$. Then the strategy induced by $\mathcal{M}$ is a behavioural strategy.*

Proof. For a DRD strategy, the distribution over memory states at any point is a Dirac distribution. More precisely, the memory state m_w reached after $w \in (S\bar{A})^*$ is defined by induction. We have $m_\varepsilon = m_{\mathsf{init}}$ and for $ws\bar{a} \in (S\bar{A})^+$, we have $m_{ws\bar{a}} = \alpha_{\mathsf{up}}(m_w, \mathsf{Obs}_i(s), \mathsf{Obs}_i(\bar{a}))$. It is easy to see that m_w depends only on the observations assigned to w, which is sufficient to end the proof. □

7.3 Transferring our taxonomy to imperfect information

We are now concerned with transferring our taxonomy of finite-memory strategies in games of perfect information to games of imperfect information. We remark that all non-inclusions witnessed in the perfect information case hold in the imperfect information case.

On the one hand, if a player cannot perceive their own actions, some inclusions of the lattice in Figure 1.1 fail. This is already suggested by Example 7.1 and Lemma 7.2, which imply together that RDD ⊆ DRD does not hold without perfect recall. On the other hand, it can be argued that the lattice of Figure 1.1 stays unchanged in games where a player can see their actions.

Imperfect recall. We illustrate the lattice for general games of imperfect information in Figure 7.1. We first discuss the non-trivial inclusion that is preserved in this broader setting, then we explain why the others fail.

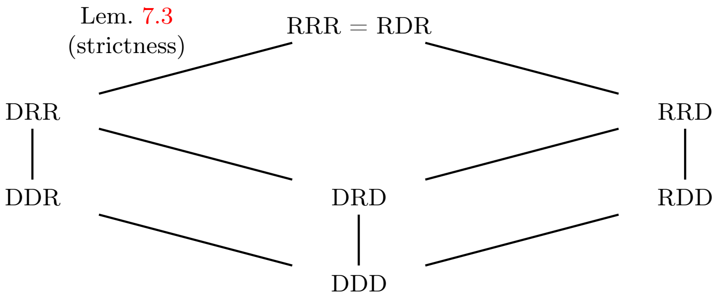


Fig. 7.1: Lattice of finite-memory strategy classes in games of imperfect information with imperfect recall. We decorate edges with relevant results introduced in this section.

Of the three non-trivial inclusions shown in Section 4, only RRR ⊆ RDR still holds in this setting. The idea is that Theorem 4.3, unlike the other two inclusion theorems, does not rely on the visibility of actions in the construction of the Mealy machine. It even provides a Mealy machine that is agnostic to a player's own actions. Because states and the actions of the other player only serve the role of inputs (i.e., their nature does not matter), we can adapt the proof of the theorem directly to obtain its direct reformulation in games of imperfect information.

The other two non-trivial inclusions, RDD $\subseteq$ RDR and RRR $\subseteq$ DRR fail in this setting. As explained previously, Example 7.1 and Lemma 7.2 show that the first inclusion cannot hold. Furthermore, we obtain that DRR and RDD (and RRD) are incomparable. To illustrate this, we prove that the strategy introduced in Example 7.1 has no DRR equivalent.

Lemma 7.3. *Let $\mathit{\Gamma}_{a,b}$ denote the game of imperfect information derived from $\mathcal{G}_{a,b}$ (Figure 5.1) by assigning observation o to everything. There exists an RDD strategy in $\mathit{\Gamma}_{a,b}$ such that there is no outcome-equivalent DRR strategy.*

Proof. We consider the Mealy machine depicted in Figure 5.2a as in Example 7.1 and let σ_1 denote the history-based strategy it induces. Let $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ be a DRR strategy of $\mathit{\Gamma}_{a,b}$ and let $\tau_1^{\mathcal{M}}$ be the history-based strategy it induces. We assume towards a contradiction that $\tau_1^{\mathcal{M}}$ and σ_1 are outcome-equivalent.

We have $\alpha_{\mathsf{nxt}}(m_{\mathsf{init}}, o)(a) = \tau_1^{\mathcal{M}}(s)(a) = \sigma_1(s)(a) = \frac{1}{2}$. It follows that the distributions μ_{sa} and μ_{sb} over M after sa and sb have respectively occurred are, by definition, for all $m \in M$,

$$\mu_{sa}(m) = \frac{\alpha_{\mathsf{up}}(m_{\mathsf{init}}, o, o)(m) \cdot \frac{1}{2}}{\frac{1}{2}} = \mu_{sb}(m).$$

We conclude that $\tau_1^{\mathcal{M}}(sas) = \tau_1^{\mathcal{M}}(sbs)$. However, the outcome-equivalence of σ_1 and $\tau_1^{\mathcal{M}}$ implies that $\tau_1^{\mathcal{M}}(sas)(a) = \sigma_1(sas)(a) = 1$ and $\tau_1^{\mathcal{M}}(sbs)(b) = \sigma_1(sbs)(b) = 1$, which constitutes a contradiction. □

Perfect recall. We now consider games where the player we study can see their own actions. In this case, we have the following theorem.

Theorem 7.1. *The taxonomy of Figure 1.1 for $\mathcal{P}_i$ established in games of perfect information extends to games with imperfect information with perfect recall.*

Proof. We have previously explained that Theorem 4.3 holds even without perfect recall. Therefore, we need only generalise the statements of Theorems 4.1 and 4.2 to games with imperfect information and perfect recall. As we did with Theorem 4.3, we briefly argument how to adapt their proofs when replacing states and actions with observations in a setting with perfect recall.

In Theorem 4.1, we simulate RDD strategies by means of DRD strategies. We keep track of a finite set of pure FM strategies and remove one whenever we perceive an action that is inconsistent with it. The visibility of actions makes this approach viable in games of imperfect information. Furthermore, the RDD strategy that is simulated and all of the pure FM strategies encoded in the simulating DRD strategy all use the exact same observation-based update scheme. Therefore, any RDD strategy has an outcome-equivalent DRD counterpart in games of imperfect information with perfect recall.

Theorem 4.2 claims that any RRR strategy admits some outcome-equivalent DRR strategy. The approach consists in adding a new initial memory state, and then leverage stochastic updates to enter the supplied RRR strategy from the second step of the game and proceed as though we had been using it from the start. We designed the updates from the new initial memory state so that, from the second step in the game, the distribution over memory states was the same in the RRR strategy and the constructed DRR one. More precisely, the update probability distribution from the new initial state is defined as the probability over the memory states of the RRR strategy after one step. The main argument of the proof of Lemma 7.1 ensures that this distribution is robust to the passage to imperfect information and justifies that the proof approach generalises to this setting. □

8 Conclusion

We have provided a complete classification of randomised finite-memory strategies based on the notion of outcome-equivalence in concurrent games of perfect and imperfect information. We have shown that all inclusions of the studied strategy classes can be witnessed by effective constructions. Regarding the separation of strategy classes, we have provided examples on the simplest possible game and, additionally, illustrated the separation of classes on games that use specifications from the literature.

Outcome-equivalence is a specification-agnostic means of comparing strategies; two strategies that are outcome-equivalent provide the same performance against any specification no matter the strategy of the other player (or players in a multiplayer setting). In particular, the established inclusions are universal in a sense, as they hold no matter the means of comparing the behaviour of strategies. Nonetheless, outcome-equivalence is a very strong criterion for the comparison of strategies. Given some specification and a strategy in a class, even if there is no outcome-equivalent strategy in another class, there may be a strategy of the second class that performs just as well, or even better with respect to the specification. This suggests further work, where, given a family of games or specifications, we use some alternative means of comparing strategies and attempt to provide a similar taxonomy in this setting, or to attempt to understand the simplest strategies required to satisfy relevant families of specifications.

References

1. A. Ehrenfeucht, J. Mycielski, Positional strategies for mean payoff games, International Journal of Game Theory 8 (2) (1979) 109–113.
2. A. Condon, The complexity of stochastic games, Information and Computation 96 (2) (1992) 203–224. doi:10.1016/0890-5401(92)90048-K.
3. H. Gimbert, W. Zielonka, Games where you can play optimally without any memory, in: M. Abadi, L. de Alfaro (Eds.), Proceedings of the 16th International Conference on Concurrency Theory, CONCUR 2005, San Francisco, CA, USA, August 23–26, 2005, Vol. 3653 of Lecture Notes in Computer Science, Springer, 2005, pp. 428–442. doi:10.1007/11539452_33.
4. E. Grädel, W. Thomas, T. Wilke (Eds.), Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001], Vol. 2500 of Lecture Notes in Computer Science, Springer, 2002. doi:10.1007/3-540-36387-4.
5. M. Randour, Automated synthesis of reliable and efficient systems through game theory: A case study, in: Proc. of ECCS 2012, Springer Proceedings in Complexity XVII, Springer, 2013, pp. 731–738. doi:10.1007/978-3-319-00395-5_90.
6. R. Brenguier, L. Clemente, P. Hunter, G. A. Pérez, M. Randour, J. Raskin, O. Sankur, M. Sassolas, Non-zero sum games for reactive synthesis, in: A. Dediu, J. Janousek, C. Martín-Vide, B. Truthe (Eds.), Proceedings of the 10th International Conference on Language and Automata Theory and Applications, LATA 2016, Prague, Czech Republic, March 14–18, 2016, Vol. 9618 of Lecture Notes in Computer Science, Springer, 2016, pp. 3–23. doi:10.1007/978-3-319-30000-9_1.
7. R. Bloem, K. Chatterjee, B. Jobstmann, Graph games and reactive synthesis, in: E. M. Clarke, T. A. Henzinger, H. Veith, R. Bloem (Eds.), Handbook of Model Checking, Springer, 2018, pp. 921–962. doi:10.1007/978-3-319-10575-8_27.
8. E. A. Emerson, C. S. Jutla, The complexity of tree automata and logics of programs (extended abstract), in: Proceedings of the 29th Annual Symposium on Foundations of Computer Science, FOCS 1988, White Plains, New York, USA, October 24–26, 1988, IEEE Computer Society, 1988, pp. 328–337. doi:10.1109/SFCS.1988.21949.
9. W. Zielonka, Infinite games on finitely coloured graphs with applications to automata on infinite trees, Theoretical Computer Science 200 (1-2) (1998) 135–183.
10. V. Bruyère, Q. Hautem, M. Randour, Window parity games: an alternative approach toward parity games with time bounds, in: D. Cantone, G. Delzanno (Eds.), Proceedings of the 7th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2016, Catania, Italy, September 14–16, 2016, Vol. 226 of EPTCS, 2016, pp. 135–148. doi:10.4204/EPTCS.226.10.
11. T. Brihaye, F. Delgrange, Y. Oualhadj, M. Randour, Life is random, time is not: Markov decision processes with window objectives, in: Fokkink and van Glabbeek [49], pp. 8:1–8:18. doi:10.4230/LIPIcs.CONCUR.2019.8.
12. P. Bouyer, N. Markey, M. Randour, K. G. Larsen, S. Laursen, Average-energy games, Acta Informatica 55 (2) (2018) 91–127. doi:10.1007/s00236-016-0274-1.
13. V. Bruyère, Q. Hautem, M. Randour, J. Raskin, Energy mean-payoff games, in: Fokkink and van Glabbeek [49], pp. 21:1–21:17. doi:10.4230/LIPIcs.CONCUR.2019.21.
14. K. Chatterjee, M. Randour, J. Raskin, Strategy synthesis for multi-dimensional quantitative objectives, Acta Informatica 51 (3-4) (2014) 129–163. doi:10.1007/s00236-013-0182-6.
15. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, J. Raskin, The complexity of multi-mean-payoff and multi-energy games, Information and Computation 241 (2015) 177–196. doi:10.1016/j.ic.2015.03.001.
16. M. Randour, J. Raskin, O. Sankur, Percentile queries in multi-dimensional Markov decision processes, Formal methods in system design 50 (2-3) (2017) 207–248. doi:10.1007/s10703-016-0262-7.
17. F. Delgrange, J. Katoen, T. Quatmann, M. Randour, Simple strategies in multi-objective MDPs, in: A. Biere, D. Parker (Eds.), Proceedings (Part I) of the 26th International Conference on Tools and Algorithms

for the Construction and Analysis of Systems, TACAS 2020, Held as Part of ETAPS 2020, Dublin, Ireland, April 25–30, 2020, Vol. 12078 of Lecture Notes in Computer Science, Springer, 2020, pp. 346–364. doi:10.1007/978-3-030-45190-5_19.

18. K. Chatterjee, L. Doyen, Partial-observation stochastic games: How to win when belief fails, in: Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25–28, 2012, IEEE Computer Society, 2012, pp. 175–184. doi:10.1109/LICS.2012.28.
19. R. Berthon, M. Randour, J. Raskin, Threshold constraints with guarantees for parity objectives in Markov decision processes, in: I. Chatzigiannakis, P. Indyk, F. Kuhn, A. Muscholl (Eds.), Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, Warsaw, Poland, July 10–14, 2017, Vol. 80 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 121:1–121:15. doi:10.4230/LIPIcs.ICALP.2017.121.
20. J. Cristau, C. David, F. Horn, How do we remember the past in randomised strategies?, in: A. Montanari, M. Napoli, M. Parente (Eds.), Proceedings of the First Symposium on Games, Automata, Logic, and Formal Verification, GANDALF 2010, Minori (Amalfi Coast), Italy, June 17–18, 2010, Vol. 25 of EPTCS, 2010, pp. 30–39. doi:10.4204/EPTCS.25.7.
21. M. J. Osborne, A. Rubinstein, A course in game theory, The MIT Press, 1994.
22. R. J. . Aumann, Mixed and behavior strategies in infinite extensive games, in: M. Dresher, L. S. Shapley, A. W. Tucker (Eds.), Advances in Game Theory. (AM-52), Volume 52, Princeton University Press, 1964, pp. 627–650. doi:10.1515/9781400882014-029.
23. V. Bruyère, E. Filiot, M. Randour, J. Raskin, Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games, Information and Computation 254 (2017) 259–295. doi:10.1016/j.ic.2016.10.011.
24. P. Bouyer, S. Le Roux, Y. Oualhadj, M. Randour, P. Vandenhove, Games where you can play optimally with arena-independent finite memory, Logical Methods in Computer Science 18 (1) (2022). doi:10.46298/lmcs-18(1:11)2022.
25. P. Bouyer, Y. Oualhadj, M. Randour, P. Vandenhove, Arena-independent finite-memory determinacy in stochastic games, in: S. Haddad, D. Varacca (Eds.), Proceedings of the 32nd International Conference on Concurrency Theory, CONCUR 2021, Virtual Conference, August 24–27, 2021, Vol. 203 of LIPIcs, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, pp. 26:1–26:18. doi:10.4230/LIPIcs.CONCUR.2021.26.
26. L. de Alfaro, T. A. Henzinger, O. Kupferman, Concurrent reachability games, Theoretical Computer Science 386 (3) (2007) 188–217. doi:10.1016/j.tcs.2007.07.008.
27. L. S. Shapley, Stochastic games, Proceedings of the National Academy of Sciences 39 (10) (1953) 1095–1100. arXiv:https://www.pnas.org/content/39/10/1095.full.pdf, doi:10.1073/pnas.39.10.1095. URL https://www.pnas.org/content/39/10/1095
28. A. Maitra, W. Sudderth, Stochastic games with Borel payoffs, in: A. Neyman, S. Sorin (Eds.), Stochastic Games and Applications, Springer Netherlands, Dordrecht, 2003, pp. 367–373.
29. K. Chatterjee, L. Doyen, H. Gimbert, T. A. Henzinger, Randomness for free, Information and Computation 245 (2015) 3–16. doi:10.1016/j.ic.2015.06.003.
30. S. Le Roux, A. Pauly, M. Randour, Extending finite-memory determinacy by Boolean combination of winning conditions, in: S. Ganguly, P. K. Pandya (Eds.), Proceedings of the 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, Ahmedabad, India, December 11–13, 2018, Vol. 122 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 38:1–38:20. doi:10.4230/LIPIcs.FSTTCS.2018.38.
31. P. Bouyer, M. Randour, P. Vandenhove, Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs, in: P. Berenbrink, B. Monmege (Eds.), Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, Marseille, France, March 15–18, 2022, Vol. 219 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 16:1–16:16. doi:10.4230/LIPIcs.STACS.2022.16.
32. E. Kopczyński, Half-positional determinacy of infinite games, Ph.D. thesis, Warsaw University (2008).
33. J. Gutierrez, P. Harrenstein, G. Perelli, M. J. Wooldridge, Nash equilibrium and bisimulation invariance, Logical Methods in Computer Science 15 (3) (2019). doi:10.23638/LMCS-15(3:32)2019. URL https://doi.org/10.23638/LMCS-15(3:32)2019
34. A. Casares, P. Ohlmann, Characterising memory in infinite games, in: K. Etessami, U. Feige, G. Puppis (Eds.), Proceedings of the 50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10–14, 2023, Paderborn, Germany, Vol. 261 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 122:1–122:18. doi:10.4230/LIPICS.ICALP.2023.122. URL https://doi.org/10.4230/LIPIcs.ICALP.2023.122
35. N. Bertrand, B. Genest, H. Gimbert, Qualitative determinacy and decidability of stochastic games with signals, Journal of the ACM 64 (5) (2017) 33:1–33:48. doi:10.1145/3107926.
36. K. Chatterjee, L. de Alfaro, T. A. Henzinger, Trading memory for randomness, in: Proceedings of the 1st International Conference on Quantitative Evaluation of Systems, QEST 2004, Enschede, The Netherlands, 27–30 September 2004, IEEE Computer Society, 2004, pp. 206–217. doi:10.1109/QEST.2004.1348035.
37. K. Chatterjee, T. A. Henzinger, V. S. Prabhu, Trading infinite memory for uniform randomness in timed games, in: M. Egerstedt, B. Mishra (Eds.), Proceedings of the 11th International Workshop on Hybrid

Systems: Computation and Control, HSCC 2008, St. Louis, MO, USA, April 22–24, 2008, Vol. 4981 of Lecture Notes in Computer Science, Springer, 2008, pp. 87–100. doi:10.1007/978-3-540-78929-1_7.

38. F. Horn, Random fruits on the Zielonka tree, in: S. Albers, J. Marion (Eds.), Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, Freiburg, Germany, February 26–28, 2009, Vol. 3 of LIPIcs, Schloss Dagstuhl –Leibniz-Zentrum für Informatik, Germany, 2009, pp. 541–552. doi:10.4230/LIPIcs.STACS.2009.1848.
39. B. Monmege, J. Parreaux, P. Reynier, Reaching your goal optimally by playing at random with no memory, in: I. Konnov, L. Kovács (Eds.), Proceedings of the 31st International Conference on Concurrency Theory, CONCUR 2020, Vienna, Austria, September 1–4, 2020, Vol. 171 of LIPIcs, Schloss Dagstuhl –Leibniz-Zentrum für Informatik, 2020, pp. 26:1–26:21. doi:10.4230/LIPIcs.CONCUR.2020.26.
40. J. C. A. Main, M. Randour, Different strokes in randomised strategies: Revisiting Kuhn's theorem under finite-memory assumptions, in: B. Klin, S. Lasota, A. Muscholl (Eds.), Proceedings of the 33rd International Conference on Concurrency Theory, CONCUR 2022, Warsaw, Poland, September 12–16, 2022, Vol. 243 of LIPIcs, Schloss Dagstuhl –Leibniz-Zentrum für Informatik, 2022, pp. 22:1–22:18. doi:10.4230/LIPIcs.CONCUR.2022.22.
41. R. Durrett, Probability: Theory and Examples, 5th Edition, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2019. doi:10.1017/9781108591034.
42. T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, A. Kucera, Markov decision processes with multiple long-run average objectives, Logical Methods in Computer Science 10 (1) (2014). doi:10.2168/LMCS-10(1:13)2014.
43. K. Chatterjee, Z. Kretínská, J. Kretínský, Unifying two views on multiple mean-payoff objectives in Markov decision processes, Logical Methods in Computer Science 13 (2) (2017). doi:10.23638/LMCS-13(2:15)2017.
44. K. Etessami, M. Z. Kwiatkowska, M. Y. Vardi, M. Yannakakis, Multi-objective model checking of Markov decision processes, Logical Methods in Computer Science 4 (4) (2008). doi:10.2168/LMCS-4(4:8)2008.
45. N. Fijalkow, N. Bertrand, P. Bouyer-Decitre, R. Brenguier, A. Carayol, J. Fearnley, H. Gimbert, F. Horn, R. Ibsen-Jensen, N. Markey, B. Monmege, P. Novotný, M. Randour, O. Sankur, S. Schmitz, O. Serre, M. Skomra, Games on graphs, CoRR abs/2305.10546 (2023). doi:10.48550/arXiv.2305.10546.
46. T. Chen, V. Forejt, M. Z. Kwiatkowska, A. Simaitis, C. Wiltsche, On stochastic games with multiple objectives, in: K. Chatterjee, J. Sgall (Eds.), Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science, MFCS 2013, Klosterneuburg, Austria, August 26–30, 2013, Vol. 8087 of Lecture Notes in Computer Science, Springer, 2013, pp. 266–277. doi:10.1007/978-3-642-40313-2_25.
47. T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, C. Wiltsche, On stochastic games with multiple objectives, Tech. Rep. RR-13-06, University of Oxford, Department of Computer Science (2013).
48. P. Ashok, K. Chatterjee, J. Kretínský, M. Weininger, T. Winkler, Approximating values of generalized-reachability stochastic games, in: H. Hermanns, L. Zhang, N. Kobayashi, D. Miller (Eds.), Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2020, Saarbrücken, Germany, July 8–11, 2020, ACM, 2020, pp. 102–115. doi:10.1145/3373718.3394761.
49. W. J. Fokkink, R. van Glabbeek (Eds.), Proceedings of the 30th International Conference on Concurrency Theory, CONCUR 2019, Amsterdam, the Netherlands, August 27–30, 2019, Vol. 140 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CONCUR.2019.

A Probability over memory states in stochastic-update Mealy machines

A.1 Inductive relation for the distribution over memory states

We fix a game $\mathcal{G} = (S, A^{(1)}, A^{(2)}, \delta)$. In this section, we derive the formula for updates of the distribution over memory states of a Mealy machine after a word in $(S\bar{A})^*$ takes place under its induced strategy. We build our reasoning on conditional probabilities. We show the equations for a Mealy machine of $\mathcal{P}_1$; the reasoning for $\mathcal{P}_2$ is analogous. We fix a Mealy machine $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{nxt}}, \alpha_{\mathsf{up}})$ of $\mathcal{P}_1$.

Let $w = s_0\bar{a}_0s_1\bar{a}_1 \dots s_k\bar{a}_k \in (S\bar{A})^*$. We study the distribution over M after w takes place. This distribution is well-defined only under specific assumptions on w. The probability of being in some memory state m after w is formally the conditional probability of being in m at step $k + 1$ given w. We must therefore require that w is of positive probability under $\mathcal{M}$ and (at least) one strategy of $\mathcal{P}_2$, i.e., w must be consistent with $\mathcal{M}$.

We reuse the notation μ_w introduced in Section 2. The main goal of this section is to prove the inductive relation for μ_w recalled below. Assume w is not the empty word and let $w' = s_0\bar{a}_0s_1\bar{a}_1 \dots s_{k-1}\bar{a}_{k-1}$. We prove the equation:

$$\mu_w(m) = \frac{\sum_{m' \in M} \mu_{w'}(m') \cdot \alpha_{\mathsf{up}}(m', s_k, \bar{a}_k)(m) \cdot \alpha_{\mathsf{nxt}}(m', s_k)(a_k^{(1)})}{\sum_{m' \in M} \mu_{w'}(m') \cdot \alpha_{\mathsf{nxt}}(m', s_k)(a_k^{(1)})}. \tag{A.1}$$

We derive this equation by studying the Markov chain induced in $\mathcal{G}$ by $\mathcal{M}$ and a strategy of $\mathcal{P}_2$ from an initial state. We fix a strategy σ_2 of $\mathcal{P}_2$ and an initial state $s_{\mathsf{init}} \in S$. In the sequel, we prove that the equations above hold for any $w \in (S\bar{A})^*$ that starts in s_{init} and is consistent with $\mathcal{M}$ and σ_2. As indicated by Equation (A.1), the choice of σ_2 has no impact on μ_w (this strategy is required so the Markov chain is well-defined).

A.2 Description of the Markov chain

First, we describe the Markov chain induced by playing $\mathcal{M}$ and σ_2 from s_{init} in $\mathcal{G}$. Formally, it is an infinite Markov chain where states are non-empty sequences $(s_0, m_0, \bar{a}_0) \dots (s_k, m_k, \bar{a}_k)$ in $(S \times M \times \bar{A})^*$ where $s_0\bar{a}_0 \dots \bar{a}_{k-1}s_k$ is a history of $\mathcal{G}$ and $\bar{a}_k \in \bar{A}(s_k)$. The initial probability of a state $(s_{\mathsf{init}}, m, \bar{a})$ is given as the product $\mu_{\mathsf{init}}(m) \cdot \alpha_{\mathsf{nxt}}(m, s_{\mathsf{init}})(a^{(1)}) \cdot \sigma_2(s_{\mathsf{init}})(a^{(2)})$; we multiply the probability that m is drawn as the initial memory state, that $a^{(1)}$ is selected in memory state m and that $a^{(2)}$ is selected by σ_2. The initial distribution assigns 0 to any other state of the Markov chain.

Let $t = (s_0, m_0, \bar{a}_0) \dots (s_k, m_k, \bar{a}_k)$ and $t' = t(s_{k+1}, m_{k+1}, \bar{a}_{k+1})$ be two states of the Markov chain. The transition probability from t to t' is defined by the product

$$\delta(s_k, \bar{a}_k)(s_{k+1}) \cdot \alpha_{\mathsf{up}}(m_k, s_k, \bar{a}_k)(m_{k+1}) \cdot \alpha_{\mathsf{nxt}}(m_{k+1}, s_{k+1})(a_{k+1}^{(1)}) \cdot \sigma_2(s_0\bar{a}_0 \dots \bar{a}_k s_{k+1})(a_{k+1}^{(2)}).$$

We define a probability measure over infinite sequences of states of the Markov chain described above in the standard way, using cylinders. Initial infinite sequences of this Markov chain belong in $((S \times M \times \bar{A})^*)^\omega$ and are of the form $t_0(t_0t_1)(t_0t_1t_2)\dots$ where $t_k \in S \times M \times \bar{A}$. We identify these infinite initial sequences to elements of $(S \times M \times \bar{A})^\omega$. We will write $\mathbb{P}$ for the probability distribution over $(S \times M \times \bar{A})^\omega$ obtained this way.

In the sequel, we use random variables defined over $(S \times M \times \bar{A})^\omega$ to refer to components or parts of these sequences and derive Equation (A.1). We introduce some notation. Let B denote a set. For any random variable $X \colon (S \times M \times \bar{A})^\omega \to B$ and $b \in B$, we write $\{X = b\}$ for $X^{-1}(\{b\})$ and omit the braces when evaluating $\mathbb{P}$ over such sets, e.g., we write $\mathbb{P}(X = b)$ for $\mathbb{P}(\{X = b\})$.

We use the following random variables. We denote by S_k (resp. M_k, $\bar{A}_k = (A_k^{(1)}, A_k^{(2)})$) the random variable that describes the state of the game (resp. memory state, pair of actions) at position k of a sequence in $(S \times M \times \bar{A})^\omega$. We write W_k for the random variable describing the sequence $W_k = S_0\bar{A}_0S_1\bar{A}_1 \dots S_{k-1}\bar{A}_{k-1}$ which is the sequence read by $\mathcal{M}$ prior to step k. Similarly, we write H_k (resp. $\overline{M_k}$) for the random variable $H_k = W_kS_k$ (resp. $\overline{M_k} = M_0M_1 \dots M_k$) that describes the history at step k (resp. the sequence of memory states up to step k).

We now list properties of these random variables we refer to in the proof of Equation (A.1). We will be concerned with conditional probabilities, and therefore all upcoming equations will assume that some event has a positive probability. We mainly rely on the properties listed below.

First, memory updates only depend on the latest memory state, game state and pair of actions. Formally, let us take a non-empty sequence $w = s_0\bar{a}_0 \dots s_{k-1}\bar{a}_{k-1} \in (S\bar{A})^+$ such that $\mathbb{P}(W_k = w) > 0$.

For any sequence of memory states $\overline{m} = m_0m_1 \dots m_{k-1} \in M^k$ such that $\mathbb{P}(\overline{M_{k-1}} = \overline{m} \mid W_k = w) > 0$, we have, for every state $m \in M$,

$$\begin{aligned}\mathbb{P}(M_k = m \,|W_k = w \wedge \overline{M_{k-1}} = \overline{m})&\\ &= \mathbb{P}(M_k = m \mid S_{k-1} = s_{k-1} \wedge M_{k-1} = m_{k-1} \wedge \bar{A}_{k-1} = \bar{a}_{k-1})\\ &= \alpha_{\mathsf{up}}(m_{k-1}, s_{k-1}, \bar{a}_{k-1})(m).\end{aligned}$$

Next, memory updates are independent from game state updates. In particular, for any history $h = s_0\bar{a}_0 \dots \bar{a}_{k-1}s_k \in \mathsf{Hist}(\mathcal{G})$ such that $\mathbb{P}(H_k = h) > 0$, we have for any $m \in M$,

$$\mathbb{P}(M_k = m \mid H_k = h) = \mathbb{P}(M_k = m \mid W_k = w),$$

where w denotes $s_0\bar{a}_0 \dots s_{k-1}\bar{a}_{k-1}$.

The last three properties are related to the probability of actions following a history. To formulate these properties, we fix $h = s_0\bar{a}_0 \dots s_k \in \mathsf{Hist}(\mathcal{G})$ such that $\mathbb{P}(H_k = h) > 0$ and a sequence of memory states $\overline{m} = m_0m_1 \dots m_k \in M^{k+1}$ that is compatible with h, i.e., such that $\mathbb{P}(\overline{M_k} = \overline{m} \mid H_k = h) > 0$. First, we note that the actions choices of the players are independent given h and $\overline{m}$, i.e., for all $\bar{a} \in \bar{A}(s_k)$, we have

$$\begin{aligned}\mathbb{P}(\bar{A}_k = \bar{a} \,|H_k = h \wedge \overline{M_k} = \overline{m})&\\ &= \mathbb{P}(A_k^{(1)} = a^{(1)} \mid H_k = h \wedge \overline{M_k} = \overline{m}) \cdot \mathbb{P}(A_k^{(2)} = a^{(2)} \mid H_k = h \wedge \overline{M_k} = \overline{m}).\end{aligned}$$

Second, we remark that the next action of $\mathcal{P}_1$ at any point depends only on the last state of the history and the last memory state. In other words, for any sequence of memory states $\overline{m} = m_0m_1 \dots m_k \in M^{k+1}$ such that $\mathbb{P}(\overline{M_k} = \overline{m} \mid H_k = h) > 0$ (i.e., any sequence of memory states likely to occur by processing h) and action $a^{(1)} \in A^{(1)}(s_k)$,

$$\begin{aligned}\mathbb{P}(A_k^{(1)} = a^{(1)} \mid H_k = h \wedge \overline{M_k} = \overline{m}) &= \mathbb{P}(A_k^{(1)} = a^{(1)} \mid S_k = s_k \wedge M_k = m_k)\\ &= \alpha_{\mathsf{nxt}}(m_k, s_k)(a^{(1)}).\end{aligned}$$

Finally, we remark that the probability of the next action of $\mathcal{P}_2$ is given by $\sigma_2(h)$ and is independent of the sequence of memory states $\overline{m}$. Formally, we have,

$$\mathbb{P}(A_k^{(2)} = a^{(2)} \mid H_k = h \wedge \overline{M_k} = \overline{m}) = \mathbb{P}(A_k^{(2)} = a^{(2)} \mid H_k = h) = \sigma_2(h)(a^{(2)}).$$

A.3 Proving Equation (A.1)

Let $w = s_0\bar{a}_0s_1\bar{a}_1 \dots s_k\bar{a}_k \in (S\bar{A})^*$ such that $\mathbb{P}(W_{k+1} = w) > 0$. For any $m \in M$, the probability $\mu_w(m)$ is formalised by the conditional probability $\mathbb{P}(M_{k+1} = m \mid W_{k+1} = w)$. Henceforth, we assume that w is non-empty. Let $w' = s_0\bar{a}_0 \dots s_{k-1}\bar{a}_{k-1}$ be the prefix of w without its last state and last action. To prove Equation (A.1), we must express μ_w as a function of $\mu_{w'}$.

We fix $m \in M$ for the remainder of the section. The first step in our approach is to consider all possible paths in $\mathcal{M}$ that reach m and have a positive probability of occurring whenever w is processed by $\mathcal{M}$. Considering these paths will allow us to exhibit the term in which α_{up} appears within Equation (A.1). We use the notation Paths_w^m for the set of sequences $m_0m_1 \dots m_k$ such that the path $m_0m_1 \dots m_km$ in $\mathcal{M}$ is compatible with w, i.e., we let

$$\mathsf{Paths}_w^m = \{m_0m_1 \dots m_k \in M^{k+1} \mid \mathbb{P}(\overline{M_{k+1}} = m_0 \dots m_km \mid W_{k+1} = w) > 0\}.$$

We define, for any memory state $m' \in M$, the set $\mathsf{Paths}_{w'}^{m'}$ as a subset of M^k in the same fashion. It follows from the law of total probability (formulated for conditional probabilities), that

$$\begin{aligned}\mu_w(m) &= \mathbb{P}(M_{k+1} = m \mid W_{k+1} = w)\\ &= \sum_{\overline{m}m' \in \mathsf{Paths}_w^m} \mathbb{P}(M_{k+1} = m \mid W_{k+1} = w \wedge \overline{M_k} = \overline{m}m') \cdot \mathbb{P}(\overline{M_k} = \overline{m}m' \mid W_{k+1} = w)\\ &= \sum_{\overline{m}m' \in \mathsf{Paths}_w^m} \alpha_{\mathsf{up}}(m', s_k, \bar{a}_k)(m) \cdot \mathbb{P}(\overline{M_k} = \overline{m}m' \mid W_{k+1} = w)\\ &= \sum_{m' \in M} \sum_{\overline{m} \in \mathsf{Paths}_{w'}^{m'}} \alpha_{\mathsf{up}}(m', s_k, \bar{a}_k)(m) \cdot \mathbb{P}(\overline{M_k} = \overline{m}m' \mid W_{k+1} = w)\\ &= \sum_{m' \in M} \left(\alpha_{\mathsf{up}}(m', s_k, \bar{a}_k)(m) \cdot \sum_{\overline{m} \in \mathsf{Paths}_{w'}^{m'}} \mathbb{P}(\overline{M_k} = \overline{m}m' \mid W_{k+1} = w) \right).\end{aligned}$$

We now shift our focus to the inner sum. Let us fix $m' \in M$. This sum is indexed by all paths in $\mathcal{M}$ that reach m' and have positive probability. Therefore, it follows from the law of total probability that

$$\sum_{\overline{m} \in \mathsf{Paths}_{w'}^{m'}} \mathbb{P}(\overline{M_k} = \overline{m}m' \mid W_{k+1} = w) = \mathbb{P}(M_k = m' \mid W_{k+1} = w).$$

We underscore that this probability is not $\mu_{w'}(m') = \mathbb{P}(M_k = m' \mid W_k = w')$. Up to this point, we have established that

$$\mu_w(m) = \sum_{m' \in M} \alpha_{\mathsf{up}}(m', s_k, \bar{a}_k)(m) \cdot \mathbb{P}(M_k = m' \mid W_{k+1} = w). \tag{A.2}$$

Using Bayes' theorem, we can show a relation between the probability $\mathbb{P}(M_k = m' \mid W_{k+1} = w)$ and $\mu_{w'}(m')$. Let us write h' in the following for the history $w's_k$ given by w without its last action. We note that $\{W_{k+1} = w\}$ and $\{H_k = h'\} \cap \{\bar{A}_k = \bar{a}_k\}$ both denote the same set. We have the following chain of equations:

$$\begin{aligned}&\mathbb{P}(M_k = m' \mid W_{k+1} = w)\\ &\quad= \mathbb{P}(M_k = m' \wedge H_k = h' \mid W_{k+1} = w)\\ &\quad= \frac{\mathbb{P}(W_{k+1} = w \mid M_k = m' \wedge H_k = h') \cdot \mathbb{P}(M_k = m' \wedge H_k = h')}{\mathbb{P}(W_{k+1} = w)}\\ &\quad= \frac{\mathbb{P}(\bar{A}_k = \bar{a}_k \mid M_k = m' \wedge H_k = h') \cdot \mathbb{P}(M_k = m' \mid H_k = h')}{\mathbb{P}(\bar{A}_k = \bar{a}_k \mid H_k = h')}.\end{aligned}$$

The first equality is a consequence of $W_{k+1} = w$ implying $H_k = h'$. Bayes' theorem is used between lines two and three. To go from the third to the fourth line, both the numerator and denominator of the fraction have been multiplied by $\mathbb{P}(H_k = h')$ and the definition of conditional probabilities has been used to rewrite the denominator and the rightmost factor of the numerator.

We now analyse the three terms of the fraction above. The probability $\mathbb{P}(M_k = m' \mid H_k = h')$ is equal to the probability $\mathbb{P}(M_k = m' \mid W_k = w')$. This is exactly $\mu_{w'}(m')$.

Next, we obtain from the independence of the action choices of both players and how the action probabilities are computed that

$$\mathbb{P}(\bar{A}_k = \bar{a}_k \mid M_k = m' \wedge H_k = h') = \alpha_{\mathsf{nxt}}(m', s_k)(a^{(1)}) \cdot \sigma_2(h')(a^{(2)}).$$

We use the previous equation to analyse the final term of the quotient. We have

$$\begin{aligned}&\mathbb{P}(\bar{A}_k = \bar{a}_k \mid H_k = h')\\ &\quad= \sum_{\substack{m'' \in M\\ \mathbb{P}(M_k = m'' \mid H_k = h') > 0}} \mathbb{P}(\bar{A}_k = \bar{a}_k \mid M_k = m'' \wedge H_k = h') \cdot \mathbb{P}(M_k = m'' \mid H_k = h')\\ &\quad= \sigma_2(h')(a_k^{(2)}) \cdot \sum_{m'' \in M} \alpha_{\mathsf{nxt}}(m'', s_k)(a_k^{(1)}) \cdot \mu_{w'}(m'').\end{aligned}$$

By injecting the above in Equation (A.2), we directly obtain Equation (A.1) (note that any term appearing in a denominator is non-zero by the assumption $\mathbb{P}(W_{k+1} = w) > 0$). This concludes the explanation on how to derive the formula to update the distribution over memory states following a state transition and the choice of a pair of actions.