Admissiblity in Concurrent Games

Nicolas Basset, Gilles Geeraerts, Jean-François Raskin, Ocan Sankur

Admissibility in Concurrent Games[*,†]

Nicolas Basset[1], Gilles Geeraerts[2], Jean-François Raskin[3], and Ocan Sankur[4]

1 **Université libre de Bruxelles, Brussels, Belgium**
`nicolas.basset@ulb.ac.be`
2 **Université libre de Bruxelles, Brussels, Belgium**
`gilles.geeraerts@ulb.ac.be`
3 **Université libre de Bruxelles, Brussels, Belgium**
`jraskin@ulb.ac.be`
4 **CNRS, IRISA, Rennes, France**
`ocan.sankur@irisa.fr`

Abstract

In this paper, we study the notion of *admissibility* for randomised strategies in *concurrent games*. Intuitively, an admissible strategy is one where the player plays 'as well as possible', because there is no other strategy that *dominates* it, i.e., that wins (almost surely) against a superset of adversarial strategies. We prove that admissible strategies always exist in concurrent games, and we characterise them precisely. Then, when the objectives of the players are ω-regular, we show how to perform *assume-admissible synthesis*, i.e., how to compute admissible strategies that win (almost surely) under the hypothesis that the other players play admissible strategies only.

1998 ACM Subject Classification D.2.4 Software/Program Verification, F.3.1 Specifying and Verifying and Reasoning about Programs, I.2.2 Program Synthesis

Keywords and phrases Multi-player games, admissibility, concurrent games, randomized strategies

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.123

1 Introduction

In a concurrent n-player game played on a graph, all n players *independently* and *simultaneously* choose moves at each round of the game, and those n choices determine the next state of the game [14]. Concurrent games generalise turn-based games and it is well-known that, while deterministic strategies are sufficient in the turn-based case, randomised strategies are necessary for winning with probability one even for reachability objectives. Intuitively, randomisation is necessary because, in concurrent games, in each round, players choose their moves simultaneously. Randomisation makes it possible to choose a good move with some probability without the knowledge of the moves that the other players are simultaneously choosing. As a consequence, there are two classical semantics that are considered to analyse these games *qualitatively*: winning with certainty (sure semantics in the terminology of [14]), and winning with probability one (almost sure semantics in the terminology of [14]). We consider both semantics here.

[*] An extended version of this article is available in [4], `http://arxiv.org/abs/1702.06439`.
[†] This work was partially supported by the ERC Starting grant 279499 (inVEST), the ARC project « Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond » (Fédération Wallonie-Bruxelles), J.-F. Raskin is Professeur Francqui de Recherche.

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 123; pp. 123:1–123:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

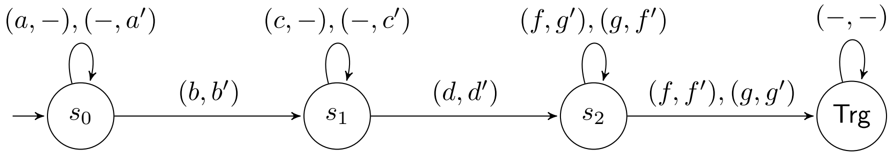
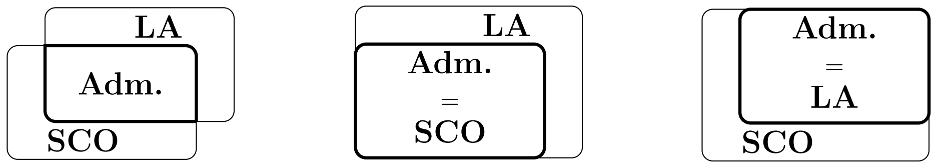


Figure 1 A concurrent game where Player 1 and 2 want to reach Trg and s_2 respectively.



(a) Concurrent games. **(b)** Turn-based games. **(c)** Safety games.

Figure 2 The relationships between the classes of Admissible, LA, and SCO strategies for three families of games. All the inclusions are strict.

Previous papers on concurrent games are mostly concerned with two-player zero-sum games, i.e. two players that have fully antagonistic objectives. In this paper, we consider the *more general setting* of n-player non zero-sum concurrent games in which each player has its own objective. The notion of winning strategy is not sufficient to study non zero-sum games and other solution concepts have been proposed. One such concept is the notion of *admissible strategy* [1].

For a player with objective Φ, a strategy σ is said to be *dominated* by a strategy σ' if σ' does as well as σ with respect to Φ against all the strategies of the other players and strictly better for some of them. A strategy σ is *admissible* for a player if it is not dominated by any other of his strategies. Clearly, playing a strategy which is not admissible is sub-optimal and a rational player should only play admissible strategies. While recent works have studied the notion of admissibility for n-player non zero-sum game graphs [5, 15, 10, 8, 7], they are all concerned with the special case of turn-based games and this work is the first to consider the more general concurrent games.

Throughout the paper, we consider the running example in Figure 1. This is a concurrent game played by two players. Player 1's objective is to reach Trg, while Player 2 wants to reach s_2. Edges are labelled by pairs of moves of both players which activate that transition (where $-$ means 'any move'). It is easy to see that no player can enforce its objective with or without randomisation, so, there is no winning strategy in this game for either player. This is because moving from s_0 to s_1 and from s_1 to s_2 requires the cooperation of both players. Moreover, the transitions from s_2 behave as in the classical 'matching pennies' game: player 1 must chose between f and g; player 2 between f' and g'; and the target is reached only when the choices 'match'. So, randomisation is needed to make sure Trg is reached with probability one, from s_2. In the paper, we will describe the dominated and admissible strategies of this game.

Technical contributions. First, we study the notion of admissible strategies for both the *sure* and *almost sure* semantics of concurrent games. We show in Theorem 8 that in both semantics admissible strategies always exist. The situation is thus similar to the turn-based case [5, 10]. Nevertheless, the techniques used in this simpler case do not generalise easily to the concurrent case and we need substantially more involved technical tools here. To obtain our universal existence result, we introduce two weaker solution concepts: *locally admissible moves* and *strongly cooperative optimal* strategies. While cooperative optimal strategies were already introduced in [7] and shown equivalent to admissible strategies in the turn based setting, they are strictly weaker than admissible strategies in the concurrent setting (both for the sure and the almost sure semantics), and they need to be combined with the notion of locally admissible moves to fully characterise admissible strategies. In the special case of safety objectives, we can show that admissible strategies are exactly those that always play locally admissible moves. This situation is depicted in Figure 2.

Second, we build on our characterisation of admissible strategies based on the notions of locally admissible moves and strongly cooperative optimal strategies to obtain algorithms to solve the *assume admissible synthesis* problem for concurrent games. In the assume admissible synthesis problem, we ask whether a given player has an *admissible* strategy that is *winning* against *all admissible* strategies of the other players. So this rule relaxes the classical synthesis rule by asking for a strategy that is winning against the admissible strategies of the other players only and not against all of them. This is reasonable as in a multi-player game, each player has his own objective which is generally not the complement of the objectives of the other players. The assume-admissible rule makes the hypothesis that players are rational, hence they play admissible strategies and it is sufficient to win against those strategies. Our algorithm is applicable to all ω-regular objectives and it is based on a reduction to a zero-sum two-player game in the sure semantics. While this reduction shares intuitions with the reduction that we proposed in [8] to solve the same problem in the turn-based case, our reduction here is based on games with imperfect information [18]. In contrast, in the turn-based case, games of perfect information are sufficient. The correctness and completeness of our reduction are proved in Theorem 11.

Related works. Concurrent *two player zero-sum* games are studied in [14] and [11]. We rely on the algorithms defined in [11] to compute states from which players have almost surely winning strategies. States where players have (deterministic) winning strategies can be computed by a reduction to more classical turn-based game graphs [2]. Nash equilibria have been studied in concurrent games [6], but without randomised strategies. None of those papers consider the notion of admissibility.

We use the notion of admissibility to obtain synthesis algorithms for systems composed of several sub-systems starting from non zero-sum specifications. Other approaches have been proposed based the notion of Nash equilibria (which suffer from the well-known limitation of non-credible threats): assume-guarantee synthesis [12] and rational synthesis [16, 17]. Those works assume the simpler setting of turn-based games and so they do not deal with randomised strategies.

Finally, in [13], Damm and Finkbeiner use the notion of *dominant strategy* to provide a compositional semi-algorithm for the (undecidable) distributed synthesis problem. However, the notion of dominant strategy is strictly stronger than the notion of admissible strategies, and dominant strategies are not guaranteed to exist, unlike admissible ones.

2 Preliminaries

Concurrent games played on graphs. Let $P = \{1, 2, \ldots n\}$ be a set of players. A *concurrent game* played on a finite graph by the players in P is a tuple $\mathcal{G} = (S, \Sigma, s_{\mathsf{init}}, (\Sigma_p)_{p\in P}, \delta)$ where,

(i) S is a finite set of *states*; and $s_{\mathsf{init}} \in S$ the *initial state*;

(ii) Σ is a finite set of *actions*;

(iii) For all $p \in P$, $\Sigma_p : S \to 2^\Sigma \setminus \{\emptyset\}$ is an *action assignment* that assigns, to all states $s \in S$, the set of actions available to player p from state s.

(iv) $\delta : S \times \Sigma \times \ldots \times \Sigma \to S$ is the *transition function*.

We write $\Sigma(s) = \Sigma_1(s) \times \ldots \times \Sigma_n(s)$ for all $s \in S$. It is often convenient to consider a player p separately and see the set of all other players $P \setminus \{p\}$ as a single player denoted $-p$. Hence, the set of actions of $-p$ in state s is: $\Sigma_{-p}(s) =_{\text{def}} \prod_{q \in P\setminus\{p\}} \Sigma_q(s)$. We assume that $\Sigma_i(s) \cap \Sigma_j(s) = \emptyset$ for all $s \in S$ and $i \neq j$. We denote by $\text{Succ}(s, a) = \{\delta(s, a, b) \mid b \in \Sigma_{-p}(s)\}$ the set of possible *successors* of the state $s \in S$ when player p performs action $a \in \Sigma_p(s)$. A particular case of concurrent games are the *turn-based games*. A game $\mathcal{G} = (S, \Sigma, s_{\mathsf{init}}, (\Sigma_p)_{p\in P}, \delta)$ is turn-based iff for all states $s \in S$, there is a unique player p s.t. the successors of s depend only on p's choice of action, i.e., $\text{Succ}(s, a)$ contains exactly one state for all $a \in \Sigma_p(s)$.

A *history* is a finite path $h = s_1 s_2 \ldots s_k \in S^*$ s.t.

(i) $k \in \mathbb{N}$;

(ii) $s_1 = s_{\mathsf{init}}$; and

(iii) for every $2 \leq i \leq k$, there exists $(a_1, \ldots, a_n) \in \Sigma^{|P|}$ with $s_i = \delta(s_{i-1}, a_1, \ldots, a_n)$.

The *length* $|h|$ of a history $h = s_1 s_2 \ldots s_k$ is its number of states k; for every $1 \leq i \leq k$, we denote by h_i the state s_i and by $h_{\leq i}$ the history $s_1 s_2 \ldots s_i$. We denote by $\text{last}(h)$ the last state of h, that is, $\text{last}(h) = h_{|h|}$. A *run* is defined similarly as a history except that its length is infinite. For a run $\rho = s_1 s_2 \ldots \in S^\omega$ and $i \in \mathbb{N}$, we also write $\rho_{\leq i} = s_1 s_2 \ldots s_i$ and $\rho_i = s_i$. Let $\texttt{Hist}(\mathcal{G})$ (resp. $\texttt{Runs}(\mathcal{G})$) denote the set of histories (resp. runs) of $\mathcal{G}$. The game is played from the initial state s_{init} for an infinite number of rounds, producing a run. At each round $i \geq 0$, with current state s_i, all players p select simultaneously an action $a_p^i \in \Sigma_p(s_i)$, and the state $\delta(s_i, a_1^i, \ldots, a_n^i)$ is appended to the current history. The selection of the action by a player is done according to strategies defined below.

Randomised moves and strategies. Given a finite set A, a probability distribution on A is a function $\alpha : A \to [0, 1]$ such that $\sum_{a\in A} \alpha(a) = 1$; and we let $\text{Supp}(\alpha) = \{a \mid \alpha(a) > 0\}$ be the support of α. We denote by $\alpha(B) = \sum_{a\in B} \alpha(a)$ the probability of a given set B according to α. The set of probability distributions on A is denoted by $\mathcal{D}(A)$. A *randomised move* of player p in state s is a probability distribution on $\Sigma_p(s)$, that is, an element of $\mathcal{D}(\Sigma_p(s))$. A randomised move that assigns probability 1 to an action and 0 to the others is called a *Dirac move*. We will henceforth denote randomised moves as sums of actions weighted by their respective probabilities. For instance $0.5f + 0.5g$ denotes the randomised move that assigns probability 0.5 to f and g (and 0 to all other actions). In particular, we denote by b a Dirac move that assigns probability 1 to action b.

Given a state s and a tuple $\boldsymbol{\beta} = (\beta_p)_{p\in P} \in \prod_{p\in P} \mathcal{D}(\Sigma_p(s))$ of randomised moves from s, one per player, we let $\delta_r(s, \boldsymbol{\beta}) \in \mathcal{D}(S)$ be the probability distribution on states s.t. for all $s' \in S$: $\delta_r(s, \boldsymbol{\beta})(s') = \sum_{\boldsymbol{a}\mid\delta(s,\boldsymbol{a})=s'} \boldsymbol{\beta}(\boldsymbol{a})$, where $\boldsymbol{\beta}(a_1, \ldots, a_n) = \prod_{i=1}^n \beta_i(a_i)$. Intuitively, $\delta_r(s, \boldsymbol{\beta})(s')$ is the probability to reach s' from s when the players play according to $\boldsymbol{\beta}$.

A *strategy* for player p is a function σ from histories to randomised moves (of player p) such that, for all $h \in \texttt{Hist}(\mathcal{G})$: $\sigma(h) \in \mathcal{D}(\Sigma_p(\text{last}(h)))$. A strategy is called Dirac at history h, if $\sigma(h)$ is a Dirac move; it is called Dirac if it is Dirac at all histories. We denote by $\Gamma_p(\mathcal{G})$ the set of player-p strategies in the game, and by $\Gamma_p^{det}(\mathcal{G})$ the set of player-p strategies that only use Dirac moves (those strategies are also called *deterministic*); we might omit $\mathcal{G}$ if it is clear from context. A *strategy profile* $\boldsymbol{\sigma}$ for a subset $A \subseteq P$ of players is a tuple $(\boldsymbol{\sigma}_p)_{p\in A}$ with $\boldsymbol{\sigma}_p \in \Gamma_p$ for all $p \in A$. When the set of players A is omitted, we assume $A = P$. Let $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_p)_{p\in P}$ be a strategy profile. Then, for all players p, we let $\boldsymbol{\sigma}_{-p}$ denote the restriction of $\boldsymbol{\sigma}$ to $P \setminus \{p\}$ (hence, $\boldsymbol{\sigma}_{-p}$ can be regarded as a strategy of player $-p$ that returns, for all histories h, a randomised move from $\prod_{p\in P\setminus\{p\}} \mathcal{D}(\Sigma_p(s)) \subseteq \mathcal{D}(\Sigma_{-p}(\text{last}(h)))$). We sometimes denote $\boldsymbol{\sigma}$ by the pair $(\boldsymbol{\sigma}_p, \boldsymbol{\sigma}_{-p})$. Given a history h, we let $(\boldsymbol{\sigma}_p)_{p\in A}(h) = (\boldsymbol{\sigma}_p(h))_{p\in A}$.

Let h be a history and let ρ be a history or a run. Then, we write $h \subseteq_{\mathsf{pref}} \rho$ iff h is a prefix of ρ, i.e., $\rho_{\leq|h|} = h$. Given two strategies $\sigma, \sigma' \in \Gamma_p$, and a history h, we let $\sigma\langle h \leftarrow \sigma'\rangle$ be the strategy that follows σ and *shifts* to σ' as soon as h has been played (i.e. $\sigma\langle h \leftarrow \sigma'\rangle$ is s.t. for all histories h': $\sigma\langle h \leftarrow \sigma'\rangle(h') = \sigma'(h')$ if $h \subseteq_{\mathsf{pref}} h'$; and $\sigma\langle h \leftarrow \sigma'\rangle(h') = \sigma(h')$ otherwise).

Probability measure and outcome of a profile. Given a history h, we let $\text{Cyl}(h) = \{\rho \mid h \subseteq_{\mathsf{pref}} \rho\}$ be the *cylinder* of h. To each strategy profile $\boldsymbol{\sigma}$, we associate a probability measure $\mathbb{P}_{\boldsymbol{\sigma}}$ on certain sets of runs. First, for a history h, we define $\mathbb{P}_{\boldsymbol{\sigma}}(\text{Cyl}(h))$ inductively on the length of h: $\mathbb{P}_{\boldsymbol{\sigma}}(\text{Cyl}(s_{\mathsf{init}})) = 1$, and $\mathbb{P}_{\boldsymbol{\sigma}}(\text{Cyl}(h's')) = \mathbb{P}_{\boldsymbol{\sigma}}(\text{Cyl}(h')) \cdot \delta_r(\text{last}(h'), \boldsymbol{\sigma}(h'))(s')$when $|h| > 1$ and $h = h's'$. Based on this definition, we can extend the definition of $\mathbb{P}_{\boldsymbol{\sigma}}$ to any Borel set of runs on cylinders. In particular, the function $\mathbb{P}_{\boldsymbol{\sigma}}$ is well-defined for all ω-regular sets of runs, that we will consider in this paper [19]. We extend the $\mathtt{Hist}$ notation and let $\mathtt{Hist}(\boldsymbol{\sigma})$ be the set of histories h such that $\mathbb{P}_{\boldsymbol{\sigma}}(\text{Cyl}(h)) > 0$. Given a profile $\boldsymbol{\sigma}$ we denote by $\mathtt{Outcome}(\boldsymbol{\sigma})$ the set of runs ρ s.t. all prefixes h of ρ belong to $\mathtt{Hist}(\boldsymbol{\sigma})$. In particular, $\mathbb{P}_{\boldsymbol{\sigma}}(\mathtt{Outcome}(\boldsymbol{\sigma})) = 1$. Note that when $\boldsymbol{\sigma}$ is composed of Dirac strategies then $\mathtt{Outcome}(\boldsymbol{\sigma})$ is a singleton. The outcome (set of histories) of a strategy $\sigma \in \Gamma_p$, denoted by $\mathtt{Outcome}(\sigma)$ ($\mathtt{Hist}(\sigma)$), is the union of outcomes (set of histories, respectively) of profile $\boldsymbol{\sigma}$ s.t. $\boldsymbol{\sigma}_p = \sigma$.

Winning conditions. To determine the gain of all players in the game $\mathcal{G}$, we define *winning conditions* that can be interpreted with two kinds of semantics denoted by the symbols $\mathtt{S}$ for the *sure semantics* or and $\mathtt{A}$ for the *almost sure semantics*. A winning condition Φ is a subset of $\mathtt{Runs}(\mathcal{G})$ called *winning runs*. From now on, we assume that concurrent games are equipped with a function Φ, called the winning condition, and mapping all players $p \in P$ to a winning condition $\Phi(p)$. A profile $\boldsymbol{\sigma}$ is $\mathtt{A}$-winning for $\Phi(p)$ if $\mathbb{P}_{\boldsymbol{\sigma}}(\Phi) = 1$ which we write $\mathcal{G}, \boldsymbol{\sigma} \models^{\mathtt{A}} \Phi(p)$. A profile $\boldsymbol{\sigma}$ is $\mathtt{S}$-winning for $\Phi(p)$ if $\mathtt{Outcome}(\mathcal{G}, \boldsymbol{\sigma}) \subseteq \Phi(p)$ which we write $\mathcal{G}, \boldsymbol{\sigma} \models^{\mathtt{S}} \Phi(p)$. Note that when $\boldsymbol{\sigma}$ is Dirac, the two semantics coincide: $\mathcal{G}, \boldsymbol{\sigma} \models^{\mathtt{S}} \Phi(p)$ iff $\mathcal{G}, \boldsymbol{\sigma} \models^{\mathtt{A}} \Phi(p)$. The profile $\boldsymbol{\sigma}$ is $\mathtt{A}$-winning from h if $h \in \mathtt{Hist}(\mathcal{G}, \boldsymbol{\sigma})$ and $\mathbb{P}_{\boldsymbol{\sigma}}(\Phi(p) \mid \text{Cyl}(h)) = \mathbb{P}_{\boldsymbol{\sigma}}(\Phi(p) \cap \text{Cyl}(h))/\mathbb{P}_{\boldsymbol{\sigma}}(\text{Cyl}(h)) = 1$ which we denote $\mathcal{G}, \boldsymbol{\sigma} \models^{\mathtt{A}}_h \Phi(p)$. The profile $\boldsymbol{\sigma}$ is $\mathtt{S}$-winning from h if $\{\rho \in \mathtt{Outcome}(\mathcal{G}, \boldsymbol{\sigma}) \mid h \subseteq_{\mathsf{pref}} \rho\} \subseteq \Phi(p)$, which we denote $\mathcal{G}, \boldsymbol{\sigma} \models^{\mathtt{S}}_h \Phi(p)$. We often omit $\mathcal{G}$ in notations when clear from the context. Most of our definitions and results hold for both semantics and we often state them using the symbol $\star \in \{\mathtt{S}, \mathtt{A}\}$ as in the following definition. Given a semantics $\star \in \{\mathtt{S}, \mathtt{A}\}$, a strategy σ for player p (from a history h) is called $\star$-winning for player p if for every $\tau \in \Gamma_{-p}$, the profile (σ, τ) is $\star$-winning for player p (from h). Note that a strategy σ for player p is $\mathtt{S}$-winning iff $\mathtt{Outcome}(\sigma) \subseteq \Phi(p)$. We often describe winning conditions using standard linear temporal operators $\Box$ and $\Diamond$; *e.g.* $\Box\Diamond S$ means the set of runs that visit infinitely often S. See [3] for a formal definition.

A winning condition $\Phi(p)$ is *prefix-independent* if for all $s_1 s_2 \ldots \in \Phi(p)$, and all $i \geq 1$: $s_i s_{i+1} \ldots \in \Phi(p)$. When $\Phi(p)$ contains all runs that do not visit some designated set $\mathsf{Bad}_p \subseteq S$ of states, we say that $\Phi(p)$ is a *safety condition.* A safety game is a game whose winning condition Φ is such that $\Phi(p)$ is a safety condition for all players p. Without loss of generality, we assume that safety games are so-called *simple* safety games: a safety game $(S, \Sigma, s_{\mathsf{init}}, (\Sigma_p)_{p\in P}, \delta)$ is *simple* iff for all players p, for all $s \in S$: $s \in \mathsf{Bad}_p$ implies that no $s' \notin \mathsf{Bad}_p$ is reachable from s. That is, once the safety condition is violated, then it remains violated forever at all future histories.

▶ **Example 1.** Let us consider three player-1 strategies in Figure 1.
(i) σ_1 is any strategy that plays a in s_0;
(ii) σ_2 is any strategy that plays b in s_0, d in s_1 and f in s_2; and
(iii) σ_3 is any strategy that plays b in s_0, d in s_1, and $0.5f + 0.5g$ in s_2.
Clearly, σ_1 never allows one to reach $\mathtt{Trg}$ while some runs respecting σ_2 and σ_3 do (remember that there is no $\star$-winning strategy in this game). We will see later that the best choice of player 1 (among σ_2, σ_3) depends on the semantics we consider. In the almost-sure semantics, σ_3 is 'better' for player 1, because σ_3 is an $\mathtt{A}$-winning strategy from all histories ending in s_2, while σ_2 is not. On the other hand, in the sure semantics, playing σ_2 is 'better' for player 1 than σ_3. Indeed, for all player-2 strategies τ, either $\mathtt{Outcome}(\sigma_3, \tau)$ contains only runs that do not reach s_2 (hence, do not reach $\mathtt{Trg}$ either), or $\mathtt{Outcome}(\sigma_3, \tau)$ contains at least a run that reaches s_2, but, in this case, it also contains a run of the form hs_2^ω that does not reach $\mathtt{Trg}$ (because, intuitively, player 1 plays *both* f and g from s_2). So, σ_3 is not $\mathtt{S}$-winning against any τ, while σ_2 wins at least against a player 2 strategy that plays b' in s_0, d' in s_1 and f' in s_2. We formalise these intuitions in the next section.

3 Admissibility

In this section, we define the central notion of the paper: admissibility [5, 9]. Intuitively, a strategy is admissible when it plays 'as well as possible'. Hence the definition of admissible strategies is based on a notion of domination between strategies: a strategy σ' dominates another strategy σ when σ' wins every time σ does. Obviously, players have no interest in playing dominated strategies, hence admissible strategies are those that are not dominated. Apart from these (classical) definitions, we characterise admissible strategies as those that satisfy two weaker notions: they must be both *strongly cooperative optimal* and play only *locally-admissible moves*. Finally, we discuss important characteristics of admissible strategies that will enable us to perform assume-admissible synthesis (see Section 4).

In this section, we fix a game $\mathcal{G}$, a player p, and, following our previous conventions, we denote by Γ_{-p} the set $\{\boldsymbol{\sigma}_{-p} \mid \boldsymbol{\sigma} \in \Gamma\}$.

Admissible strategies. We first recall the classical notion of *admissible strategy* [5, 1]. Given two strategies $\sigma, \sigma' \in \Gamma_p$, we say that σ is $\star$-*weakly dominated* by σ', denoted $\sigma \preccurlyeq^\star \sigma'$, if for all $\tau \in \Gamma_{-p}$: $(\sigma, \tau) \models^\star \Phi(p)$ *implies* $(\sigma', \tau) \models^\star \Phi(p)$. This indeed captures the idea than σ' is *not worse* that σ, because it wins (for p) every time σ does. Note that $\preccurlyeq^\star$ is not anti-symmetric, hence we write $\sigma \approx^\star \sigma'$ when σ and σ' are equivalent, i.e. $\sigma \preccurlyeq^\star \sigma'$ and $\sigma' \preccurlyeq^\star \sigma$. In other words $\sigma \approx^\star \sigma'$ iff for every $\tau \in \Gamma_{-p}$, $(\sigma, \tau) \models^\star \Phi(p) \Leftrightarrow (\sigma', \tau) \models^\star \Phi(p)$. When $\sigma \preccurlyeq^\star \sigma'$ but $\sigma' \not\preccurlyeq^\star \sigma$ we say that σ is $\star$-*dominated* by σ', and we write $\sigma \prec^\star \sigma'$. Observe that $\sigma \prec^\star \sigma'$ holds if and only if $\sigma \preccurlyeq^\star \sigma'$ and there exists at least one $\tau \in \Gamma_{-p}$, such that $(\sigma, \tau) \not\models^\star \Phi(p)$ and $(\sigma', \tau) \models^\star \Phi(p)$. That is, σ' is now *strictly better* than σ. Then, *a strategy σ is $\star$-**admissible** iff there is no strategy σ' s.t. $\sigma \prec^\star \sigma'$*, i.e., σ is $\star$-admissible iff it is not $\star$-dominated.

▶ **Example 2.** Let us continue our running example, by formalising the intuitions we have sketched in Example 1. Since σ_1 does not allow to reach the target, while some runs respecting σ_2 and σ_3 do, we have: $\sigma_1 \prec^\star \sigma_2$ and $\sigma_1 \prec^\star \sigma_3$. Moreover, $\sigma_2 \prec^{\mathtt{A}} \sigma_3$ because σ_3 is $\mathtt{A}$-winning from any history that ends in s_2 while σ_2 is not because it does not $\mathtt{A}$-win against a player 2 strategy that would always play g' in s_2 (and both strategies behave the same way in s_0 and s_1). On the other hand, $\sigma_3 \prec^{\mathtt{S}} \sigma_2$ since we saw in Example 1 that every profile containing σ_3 is not $\mathtt{S}$-winning while some profiles containing σ_2 are. We will see later that σ_3 is $\mathtt{A}$-admissible and σ_2 is $\mathtt{S}$-admissible.

Values of histories. Before we discuss strongly cooperative optimal and locally admissible strategies, we associate *values* to histories. Let h be a history, and σ be a strategy of player p. Then, the *value of h w.r.t. σ for semantics* $\star \in \{\mathtt{S}, \mathtt{A}\}$ is defined as follows. $\chi^\star_\sigma(h) = 1$ if σ is $\star$-winning from h; $\chi^\star_\sigma(h) = 0$ if there are $\tau \in \Gamma_{-p}$ and $\tau' \in \Gamma_{-p}$ s.t. $(\sigma, \tau) \models^\star_h \Phi(p)$, and $(\sigma, \tau') \not\models^\star_h \Phi(p)$; and -1 otherwise.

Value $\chi^\star_\sigma(h) = 1$ corresponds to the case where σ is $\star$-winning for player p from h (thus, against all possible strategies in Γ_{-p}). When $\chi^\star_\sigma(h) = 0$, σ is *not* $\star$-winning from h (because of τ' in the definition), but the other players can still help p to reach his objective (by playing some τ s.t. $(\sigma, \tau) \models^\star_h \Phi(p)$, which exists by definition). Last, $\chi^\star_\sigma(h) = -1$ when there is no hope for p to $\star$-win, even with the collaboration of the other players. In this case, there is no τ s.t. $(\sigma, \tau) \models^\star_h \Phi(p)$. Hence, having $\chi^\star_\sigma(h) = -1$ is stronger than saying that σ is not winning—when σ is not winning, we could have $\chi^\star_\sigma(h) = 0$ as well.

We define the value of a history h for player p as the best value he can achieve with his different strategies: $\chi^\star_p(h) = \max_{\sigma \in \Gamma_p} \chi^\star_\sigma(h)$. Last, for $v \in \{-1, 0, 1\}$, let $\mathtt{Val}^\star_{p,v}$ be the set of histories h s.t. $\chi^\star_p(h) = v$.

Strongly cooperative optimal strategies. We are now ready to define *strongly cooperative optimal* (SCO) strategies. Recall that, in the classical setting of turn-based games, admissible strategies are exactly the SCO strategies [9]. We will see that this condition is still necessary but not sufficient in the concurrent setting.

A strategy σ of Player p is $\star$-SCO at h iff $\chi^\star_\sigma(h) = \chi^\star_p(h)$; and σ is $\star$-SCO iff it is $\star$-SCO at all $h \in \mathtt{Hist}(\sigma)$. Intuitively, when σ is a $\star$-SCO strategy of Player p, the following should hold:

(i) if p has a $\star$-winning strategy from h (i.e. $\chi^\star_p(h) = 1$), then, σ should be $\star$-winning (i.e. $\chi^\star_\sigma(h) = 1$); and

(ii) otherwise if p has no $\star$-winning strategy from h but still has the opportunity to $\star$-win with the help of other players (hence $\chi^\star_p(h) = 0$), then, σ should enable the other players to help p fulfil his objective (i.e. $\chi^\star_\sigma(h) = 0$).

Observe that when $\chi^\star_p(h) = -1$, no continuation of h is $\star$-winning for p, so $\chi^\star_\sigma(h) = -1$ for all strategies σ.

▶ **Example 3.** Consider again the example in Figure 1. For the almost-sure semantics, we have $\mathtt{Val}^{\mathtt{A}}_{p,1} = \{h \mid \mathrm{last}(h) \in \{s_2, \mathsf{Trg}\}\}$, and $\mathtt{Val}^{\mathtt{A}}_{p,0} = \{h \mid \mathrm{last}(h) \in \{s_0, s_1\}\}$. For the sure semantics, we have: $\mathtt{Val}^{\mathtt{S}}_{1,1} = \{h \mid \mathrm{last}(h) = \mathsf{Trg}\}$, and $\mathtt{Val}^{\mathtt{S}}_{1,0} = \{h \mid \mathrm{last}(h) \neq \mathsf{Trg}\}$. Consider again the three strategies σ_1, σ_2 and σ_3 from Example 1. We see that σ_2 is $\mathtt{S}$-SCO but it is not $\mathtt{A}$-SCO because, for all profiles h ending in s_2: $\chi^{\mathtt{A}}_{\sigma_2}(h) = 0$ while $h \in \mathtt{Val}^{\mathtt{A}}_{1,1}$. On the other hand, σ_3 is $\mathtt{A}$-SCO; but it is not $\mathtt{S}$-SCO. Indeed, one can check that, for all strategies $\tau \in \Gamma_2$: if $\mathtt{Outcome}(\sigma_3, \tau)$ contains a run reaching Trg, then it also contains a run that cycles in s_2. So, for all such strategies τ, $\mathtt{Outcome}(\sigma_3, \tau) \not\models^{\mathtt{S}} \Phi(1)$, hence $\chi^{\mathtt{S}}_{\sigma_3}(h) = -1$ for all histories that end in s_2; while $\chi^{\mathtt{S}}_p(h) = 0$ since $\chi^{\mathtt{S}}_{\sigma'}(h) = 0$ for all Dirac strategies σ'.

Next, let us build a strategy σ'_3 that is $\mathtt{A}$-dominated by σ_3 (hence, not $\mathtt{A}$-admissible), but $\mathtt{A}$-SCO. We let σ'_3 play as σ_3 except that σ'_3 plays c the first time s_1 is visited (hence ensuring that the self-loop on s_1 will be taken after the first visit to s_1). Now, σ'_3 is $\mathtt{A}$-dominated by σ_3, because

(i) σ_3 $\mathtt{A}$-wins every time σ'_3 does; but

(ii) σ'_3 does not $\mathtt{A}$-win against the player 2 strategy τ that plays d' only when s_1 is visited for the first time, while σ_3 $\mathtt{A}$-wins against τ.

However, σ'_3 is SCO because playing c keeps the value of the history equal to $0 = \chi^{\mathtt{A}}_1(h)$ (intuitively, playing c once does not prevent the other players from helping in the future).

As similar example can be built in the $\mathtt{S}$ semantics. Thus, **there are $\star$-SCO strategies which are not admissible**, so, being $\star$-SCO is not a sufficient criterion for admissibility.

Locally admissible moves and strategies. Let us now discuss another criterion for admissibility, which is more *local* in the sense that it is based on a domination between *moves* available to each player after a given history. Let h be a history, and let α and α' be two randomised moves in $\mathcal{D}(\Sigma_p)$. We say that α is $\star$-*weakly dominated* at h by α' (denoted $\alpha \leqslant_h^\star \alpha'$) iff for all $\sigma \in \Gamma_p$ such that $h \in \mathtt{Hist}(\sigma)$ and $\sigma(h) = \alpha$, there exists $\sigma' \in \Gamma_p$ s.t. $h \in \mathtt{Hist}(\sigma')$, $\sigma'(h) = \alpha'$ and $\sigma \preccurlyeq^\star \sigma'$. Observe that the relation $\leqslant_h^\star$ is not anti-symmetric. We let $\simeq_h^\star$ be the equivalence relation s.t. $\alpha \simeq_h^\star \beta$ iff $\alpha \leqslant_h^\star \beta$ and $\beta \leqslant_h^\star \alpha$. When $\alpha \leqslant_h^\star \alpha'$ but $\alpha' \not\leqslant_h^\star \alpha$ we say that α is $\star$-*dominated* at h by α' and denote this by $\alpha <_h^\star \alpha'$. When a randomised move α is not $\star$-dominated at h, we say that α is $\star$-*locally-admissible* ($\star$-LA) at h. This allows us to define a more local notion of dominated strategy: *a strategy σ of player p is* ***$\star$-locally-admissible*** *($\star$-LA) if $\sigma(h)$ is a $\star$-LA move at h, for all histories h.*

▶ **Example 4.** Consider the Dirac move f and the non-Dirac move $0.5f + 0.5g$ played from s_2 in the example in Figure 1. One can check that $0.5f + 0.5g <_{s_2}^{\mathtt{S}} f$. Indeed, consider a strategy σ s.t. $\sigma(h) = 0.5f + 0.5g$ for some h with $\mathrm{last}(h) = s_2$. Then, playing $\sigma(h)$ from h will never allow Player 1 to reach Trg *surely* at the next step, whatever Player 2 plays; while playing, for instance, f (Dirac move) ensures player 1 to reach Trg surely at the next step, against a Player-2 strategy that plays f'. Thus, σ_2 is $\mathtt{S}$-LA but σ_3 is not.

On the other hand, after every randomised move played in state s_2, the updated state is s_2 or s_3 from which $\mathtt{A}$-winning strategies exist, thus $f \simeq_h^{\mathtt{A}} g \simeq_h^{\mathtt{A}} \lambda f + (1-\lambda)g$ for all $\lambda \in [0,1]$ and all histories h s.t. $\mathrm{last}(h) = s_2$ (so, in particular, $\lambda f + (1-\lambda)g \leqslant_h^{\mathtt{A}} f$ and $\lambda f + (1-\lambda)g \leqslant_h^{\mathtt{A}} g$). It follows that both σ_2 and σ_3 are $\mathtt{A}$-LA. However, in the long run, player 1 needs to play $\lambda f + (1-\lambda)g$, with $\lambda \in (0,1)$, infinitely often in order to $\mathtt{A}$-win. In fact, σ_3 is $\mathtt{A}$-winning from s_2 while σ_2 is not. Thus, **there are $\star$-LA strategies which are not admissible**, so being $\star$-LA is not a sufficient criterion for $\star$-admissibility.

We close this section by several lemmata that allow us to better characterise the notion of LA strategies. First, we observe that, while randomisation might be necessary for winning in certain concurrent games (for example, in Figure 1, no Dirac move allows player 1 to reach Trg *surely* from s_2, while playing repeatedly f and g with equal probability ensures to reach Trg with probability 1) randomisation is useless when a player wants to play only locally admissible moves. This is shown by the next Lemma (point (1)), saying that, *if a randomised move α plays some action a with some positive probability, then α is weakly dominated by the Dirac move a.* However, this does not immediately allow us to characterise admissible moves: some Dirac moves could be dominated (hence non-admissible), and some non-Dirac moves could be admissible too. Points (2) and (3) elucidate this: *among Dirac moves, the non-dominated ones are admissible*, and *a non-Dirac move is admissible iff all the Dirac moves that occur in its support are admissible and equivalent to each other.*

▶ **Lemma 5.** *For all histories h and all randomised moves α:*
(i) *For all $a \in Supp(\alpha)$: $\alpha \leqslant_h^\star a$;*
(ii) *Dirac moves that are not $\star$-dominated at h by another Dirac move are $\star$-LA;*
(iii) *A move α is $\star$-LA at h iff, for all $a \in Supp(\alpha)$:*
 1. *a is $\star$-LA at h; and*
 2. *$a \simeq_h^\star b$ for all $b \in Supp(\alpha)$.*

▶ **Example 6.** As we have seen in Example 4, $0.5f + 0.5g <_{s_2}^{\mathtt{S}} f$. Note that a strategy σ' s.t. $\sigma'(h) = 0.5f + 0.5g$ for all h with $\mathrm{last}(h) = s_2$ has value $\chi_{\sigma'}^{\mathtt{S}}(h) = -1$, while $\chi_1^{\mathtt{S}}(h) = 0$.

This example seems to suggest that the local dominance of two moves coincide with the natural order on the values of histories that are obtained when playing those moves (in other words $x <_h^\star y$ would hold iff the value of the history obtained by playing x is smaller than or equal to the value obtained by playing y). This is not true for histories of value 0: we have seen that a and b are $\leqslant_h^\star$-incomparable, yet playing a or b from s_0 yields a history with value 0 in all cases (even when s_1 is reached). The next Lemma gives a precise characterisation of the dominance relation between Dirac moves in terms of values:

▶ **Lemma 7.** *For all players p, histories h with $\text{last}(h) = s$ and Dirac moves $a, b \in \Sigma_p(s)$: $a \leqslant_h^\star b$ if, and only if the following conditions hold for every $c \in \Sigma_{-p}(s)$ where we write $s_{(a,c)} = \delta(s, (a, c))$ and $s_{(b,c)} = \delta(s, (b, c))$:*

(i) $\chi_p^\star(hs_{(a,c)}) \leq \chi_p^\star(hs_{(b,c)})$;

(ii) *if* $\chi_p^\star(hs_{(a,c)}) = \chi_p^\star(hs_{(b,c)}) = 0$ *then* $s_{(a,c)} = s_{(b,c)}$.

Characterisation and existence of admissible strategies. Equipped with our previous results, we can now establish the main results of this section. First, we show that *$\star$-admissible strategies are exactly those that are both $\star$-LA and $\star$-SCO* (Theorem 8(i)). Then, we show that *admissible strategies always exist* in concurrent games (Theorem 8(ii)).

▶ **Theorem 8** (Characterisation and existence of admissible strategies)**.** *The following holds for all strategies σ in a concurrent game with semantics $\star \in \{\mathtt{S}, \mathtt{A}\}$:*

(i) *σ is $\star$-admissible iff σ is $\star$-LA and $\star$-SCO; in the special case of simple safety objectives, if σ is $\star$-LA then σ is $\star$-admissible.*

(ii) *there is a $\star$-admissible strategy σ' such that $\sigma \preccurlyeq^\star \sigma'$.*

In particular, point (2) implies that admissible strategies always exist in concurrent games.

▶ **Example 9.** We consider again the example in Figure 1, and consider strategies σ_2 and σ_3 as defined in Example 1. Remember that these two strategies do their best to reach s_2, and that, from s_2, σ_2 plays deterministically f, while σ_3 plays f and g with equal probabilities. From Example 3, we know that σ_2 is $\mathtt{S}$-SCO but not $\mathtt{A}$-SCO; while σ_3 is $\mathtt{A}$-SCO but not $\mathtt{S}$-SCO. Indeed, we have already argued in Example 2 that σ_2 is not $\mathtt{A}$-admissible, and that σ_3 is not $\mathtt{S}$-admissible. However, from Example 4, we know that σ_2 is $\mathtt{S}$-LA and that σ_3 is $\mathtt{A}$-LA. So, by Theorem 8, σ_2 is $\mathtt{S}$-admissible and σ_3 is $\mathtt{A}$-admissible as expected.

Finally, we close the section by a finer characterisation of $\star$-admissible strategies. We show that:

(i) in the sure semantics, there is always an $\mathtt{S}$-admissible strategy that plays Dirac moves only; and

(ii) in the almost-sure semantics, there is always an $\mathtt{A}$-admissible strategy that plays Dirac moves only in histories of values 0 or -1.

The difference between the two semantics should not be surprising, as we know already that randomisation is sometimes needed to win (i.e., from histories of value 1) in the almost sure semantics:

▶ **Proposition 10.** *For all player-p strategies σ in a concurrent game:*

(i) *If σ is $\mathtt{S}$-admissible then there exists a Dirac strategy σ' such that $\sigma \simeq^{\mathtt{S}} \sigma'$.*

(ii) *If σ is $\mathtt{A}$-admissible then there exists a strategy σ' that plays only Dirac moves in histories of value ≤ 0 such that $\sigma \simeq^{\mathtt{A}} \sigma'$.*

4 Assume admissible synthesis

In this section we discuss an *assume-admissible synthesis* framework for concurrent games. With classical synthesis, one tries to compute *winning* strategies for all players, i.e., strategies that *always win* against *all possible strategies* of the other players. Unfortunately, it might be the case that such *unconditionally* winning strategies do not exist, as in our example. As explained in the introduction, the assume-admissible synthesis rule relaxes the classical synthesis rule: instead of searching for strategies that win unconditionally, the new rule requires winning against the *admissible strategies* of the other players. So, a strategy may satisfy the new rule while not winning unconditionally. We claim that winning against admissible strategies is well enough assuming that the players are *rational*; if we assume that players only play strategies that are good for achieving their objectives, *i.e.* admissible ones.

The general idea of the assume-admissible synthesis algorithm is to reduce the problem (in a concurrent n-player game) to the synthesis of a winning strategy *in a 2-player zero-sum concurrent game of imperfect information, in the* $\mathtt{S}$*-semantics* (even when the original assume-admissible problem is in the $\mathtt{A}$-semantics), where the objective of player 1 is given by an LTL formula. Such games are solvable using techniques presented in [11].

More precisely, from a concurrent game $\mathcal{G}$ in the semantics $\star \in \{\mathtt{S}, \mathtt{A}\}$ and player p, we build a game $\mathcal{G}_p^\star$ with the above characteristics, which is used to decide the assume-admissible synthesis rule. If such a solution exists, our algorithm constructs a witness strategy. For example, the game $\mathcal{G}_1^\star$ corresponding to the game in Figure 1 is given in Figure 3. The main ingredients for this construction are the following.

(i) In $\mathcal{G}_p^\star$, the protagonist is player p, and the second player is $-p$.

(ii) Although randomisation is needed to win in such games in general, we interpret $\mathcal{G}_p^\star$ in the $\mathtt{S}$-semantics only. In fact, we have seen that for the protagonist, Dirac moves suffice in states of value 0; so the only states where he might need randomisation are those of value 1 (randomisation does not matter if the value is -1 since the objective is lost anyway). Hence we define winning condition to be $\Phi(p) \vee \Diamond \mathtt{Val}^\star_{p,1}$ enabling us to consider only histories of values 0 in $\mathcal{G}_p^\star$; and thus hiding the parts of the game where randomisation might be needed. We also prove that we can restrict to Dirac strategies for $-p$ when it comes to admissible strategies.

(iii) In order to restrict the strategies to admissible ones, we only allow $\star$-LA moves in $\mathcal{G}_p^\star$. These moves can be computed by solving classical 2-player games ([2]) using Lemma 7. For example, in Figure 3, moves c and c' are removed since they are not $\mathtt{A}$-LA.

(iv) Last, since $\star$-admissible strategies are those that are both $\star$-LA and $\star$-SCO (see Theorem 8), we also need to ensure that the players play $\star$-SCO. This is more involved than $\star$-LA, as the $\star$-SCO criterion is not *local*, and requires information about the *sequence of actual moves* that have been played, which cannot be deduced, in a concurrent game, from the sequence of visited states. So, we store, in the states of $\mathcal{G}_p^\star$, the moves that have been played by all the players to reach the state. For example, in Figure 3, the state labelled by $s_1, (b, b')$ means that $\mathcal{G}$ has reached s_1, and that the last actions played by the players were b and b' respectively. However, players' strategies must not depend on this extra information since they do not have access to this information in $\mathcal{G}$ either. We thus interpret $\mathcal{G}_p^\star$ as a game of *imperfect information* where all the states labelled by the same state of $\mathcal{G}$ are in the same observation class. We can then encode that the players must play $\star$-SCO strategies in the new objective of the games, which will be given as an LTL formula, as we describe below.

To ensure we can effectively solve subproblems mentioned above, we consider ω-regular objectives. We also restrict ourselves to prefix-independent winning conditions to simplify the presentation. In the case of ω-regular objectives, prefix-independence is not a restrictive hypothesis (we can always compute the product of the game graph with a deterministic parity automaton that accepts the ω-regular objective and consider a parity winning condition). The values of the histories depend thus only on their last states, i.e. for all pairs of histories h_1 and h_2: $\text{last}(h_1) = \text{last}(h_2)$ implies that $\chi_p^\star(h_1) = \chi_p^\star(h_2)$. We denote by $\chi_p^\star(s)$ the value $\chi_p^\star(h)$ of all histories h s.t. $\text{last}(h) = s$. Last, we assume that a player cannot play the same action from two different states, i.e. $\forall s_1 \neq s_2$, $\Sigma_{s_1}(p) \cap \Sigma_{s_2}(p) = \emptyset$. Thus, we say that *move a is $\star$-LA* when a is $\star$-LA from all histories ending in the unique state where a is available.

The game $\mathcal{G}_p^\star$. Let us now describe precisely the construction of $\mathcal{G}_p^\star$. Given an n-player concurrent game $\mathcal{G} = (S, \Sigma, s_{\mathsf{init}}, (\Sigma_p)_{p\in P}, \delta)$ with winning condition Φ considered under semantics $\star \in \{\mathtt{S}, \mathtt{A}\}$, and given a player p, we define the two-player zero-sum concurrent game $\mathcal{G}_p^\star = (\overline{S}, \overline{\Sigma}, \overline{s_{\mathsf{init}}}, (\overline{\Sigma}_p, \overline{\Sigma}_{-p}), \overline{\delta})$ where:

(i) $\overline{S} = S \times \overline{\Sigma}^n \cup \{\overline{s_{\mathsf{init}}}\}$;
(ii) $\overline{\Sigma}$ is the set of Dirac $\star$-LA moves in Σ;
(iii) $\overline{s_{\mathsf{init}}} = s_{\mathsf{init}}$ is the initial state;
(iv) $\overline{\Sigma}_p$ is such that $\overline{\Sigma}_p(s)$ is the set of Dirac $\star$-LA moves of p in s, for all $s \in S$;
(v) $\overline{\Sigma}_{-p}$ is s.t. for all $s \in S$: $\overline{\Sigma}_{-p}(s)$ is the set of moves $\boldsymbol{a}$ of $-p$ in s s.t. for all $q \neq p$, a_q is a Dirac $\star$-LA move;
(vi) $\overline{\delta}$ updates the state according to δ, remembering the last actions played: $\overline{\delta}(\overline{s_{\mathsf{init}}}, \boldsymbol{b}) = (\delta(s_{\mathsf{init}}, \boldsymbol{b}), \boldsymbol{b})$ and $\overline{\delta}((s, \boldsymbol{a}), \boldsymbol{b}) = (\delta(s, \boldsymbol{b}), \boldsymbol{b})$ for all $s \in S$.

Note that the game $\mathcal{G}_p^\star$ depends on whether $\star = \mathtt{A}$ or $\star = \mathtt{S}$ because the two semantics yield different sets of LA-moves. However, we interpret $\mathcal{G}_p^\star$ in the sure semantics, so both players can play Dirac strategies only in $\mathcal{G}_p^\star$.

Let us now explain how we obtain an imperfect information game by defining an *observation function* $\mathfrak{o}$. Note that histories in $\mathcal{G}_p^\star$ are of the form: $\overline{h} = \overline{s_{\mathsf{init}}}(s_1, \boldsymbol{a}_1)(s_2, \boldsymbol{a}_2)\cdots(s_n, \boldsymbol{a}_n)$. Then, let $\mathfrak{o} : \overline{S} \to S$ be the mapping that, intuitively, projects moves away from states. For example, in Figure 3, states with observation s_0 are in the dashed rectangle. That is: $\mathfrak{o}(s, \boldsymbol{a}) = s$ for all states s, and $\mathfrak{o}(\overline{s_{\mathsf{init}}}) = s_{\mathsf{init}}$. We extend $\mathfrak{o}$ to histories recursively: $\mathfrak{o}(\overline{s_{\mathsf{init}}}) = s_{\mathsf{init}}$ and $\mathfrak{o}(h(s_n, \boldsymbol{a}_n)) = \mathfrak{o}(h)s_n$. To make $\mathcal{G}_p^\star$ a game of imperfect information, we request that, in $\mathcal{G}_p^\star$, players play only strategies σ s.t. $\sigma(h_1) = \sigma(h_2)$ whenever $\mathfrak{o}(h_1) = \mathfrak{o}(h_2)$.

We relate the strategies in the original game $\mathcal{G}$ with the strategies in $\mathcal{G}_p^\star$, which we need to extract admissible strategies in $\mathcal{G}$ from the winning strategies in $\mathcal{G}_p^\star$ and thus perform assume-admissible synthesis. First, given a player-p strategy σ in $\mathcal{G}$ (i.e., $\sigma \in \Gamma_p(\mathcal{G})$), we say that a strategy $\overline{\sigma} \in \Gamma_p^{\mathrm{det}}(\mathcal{G}_p^\star)$ is a *realisation* of σ iff:

(i) $\overline{\sigma}$ is Dirac; and
(ii) $\overline{\sigma}(h) \in \mathrm{Supp}(\sigma(h))$ for all h.

Note that every $\star$-LA strategy $\sigma \in \Gamma_i(\mathcal{G})$ admits realisations σ in $\Gamma_i(\mathcal{G}_p^\star)$. Second, given a player-p Dirac strategy σ in $\mathcal{G}_p^\star$ (i.e., $\sigma \in \Gamma_p^{\mathrm{det}}(\mathcal{G}_p^\star)$) we say that $\hat{\sigma} \in \Gamma_p(\mathcal{G})$ is an *extension* of σ iff, for all $h \in \mathtt{Hist}(\mathcal{G}_p^\star, \sigma)$: $\hat{\sigma}(\mathfrak{o}(h)) = \sigma(h)$.

The assume-admissible synthesis technique. As explained above, the assume-admissible rule boils down to computing a winning strategy $\overline{\sigma}$ for player-p in $\mathcal{G}_p^\star$ w.r.t. the winning condition $\Phi_{\mathcal{G}_p^\star}$, and extracting, from $\overline{\sigma}$, the required admissible strategy in $\mathcal{G}$.

We will now formally define $\Phi_{\mathcal{G}_p^\star}$. Let p be a player (in $\mathcal{G}$); and let us denote by $\mathtt{st}(a)$ the (unique) state from which a is available, for all actions a. We define $\mathtt{AfterHelpMove}_p^\star$ as

$$\mathtt{AfterHelpMove}_p^\star = \{(s,\boldsymbol{a}) \in \overline{S} \mid \exists s' \in \mathrm{Succ}(\mathtt{st}(a_p), a_p) : \chi_p^\star(s') \geq 0 \wedge s' \neq s \wedge \chi_p^\star(s) = 0\}.$$

That is, when $(s,\boldsymbol{a}) \in \mathtt{AfterHelpMove}_p^\star$, in $\mathcal{G}$, player p has played a_p from $\mathtt{st}(a_p)$ and, due to player $-p$'s choice, $\mathcal{G}$ has reached s. However, with another choice of player $-p$, the game could have moved to a different state s' from which $-p$ can help p to win as $\chi_p^\star(s') \geq 0$. Intuitively, in runs that visit states of value 0 infinitely often, states from $\mathtt{AfterHelpMove}_p^\star$ should be visited infinitely often for player p to play SCO, i.e. such runs might not be winning, but this cannot be blamed on player p who has sought repeatedly the collaboration of the other players to enforce his objective. Observe further that the definition of this predicate requires the labelling of the states (by actions) we have introduced in $\mathcal{G}_p^\star$. For example, in Figure 3, $\mathtt{AfterHelpMove}_2^{\mathtt{A}} = \{(s_0,(a,b')),(s_1,(b,b'))\}$. We let $\Phi_0^\star(p) = \Diamond\neg\mathtt{Val}_{p,0}^\star \vee \Phi(p) \vee \Box\Diamond\mathtt{AfterHelpMove}_p^\star$ and $\Phi_1^\star(p) = (\Diamond\mathtt{Val}_{p,1}^\star) \to \Phi(p)$. Let us define $\Phi_{\mathcal{G}_p^\star} = \left(\bigwedge_{q\neq p}\Phi_0^\star(q)\wedge\Phi_1^\star(q)\right) \to \left(\Phi(p) \vee \Diamond\mathtt{Val}_{p,1}^\star\right)$.

▶ **Theorem 11** (Assume-admissible synthesis). *Player p has a $\star$-admissible strategy σ that is $\star$-winning against all player $-p$ $\star$-admissible strategies in $\mathcal{G}$ iff Player p has an $\mathtt{S}$-winning strategy in $\mathcal{G}_p^\star$ for the objective $\Phi_{\mathcal{G}_p^\star}$. Such a $\star$-admissible strategy σ can be effectively computed (from any player p $\mathtt{S}$-winning strategy in $\mathcal{G}_p^\star$).*

Let us explain how we build a strategy in $\mathcal{G}$ with the desired properties, from any player p strategy enforcing $\Phi_{\mathcal{G}_p^\star}$ in $\mathcal{G}_p^\star$. Remember that $\mathcal{G}_p^\star$ ensures that the players play $\star$-LA moves only. We will use $\Phi_{\mathcal{G}_p^\star}$ to make sure that, when SCO strategies are played by $-p$ (relying on the extra information we have encoded in the states), then p reaches a state of value 1. First, consider $\Phi_0^\star(q)$ for $q \neq p$. Runs that satisfy this formula are either those that visit states of value 0 only finitely often ($\Diamond\neg\mathtt{Val}_{q,0}^\star$); or those that stay in states of value 0, in which case they must be either winning ($\Phi(q)$) or visit infinitely often states where Player q could have been helped by the other players ($\Box\Diamond\mathtt{AfterHelpMove}_q^\star$). This is a necessary condition on runs visiting only value 0 states for the strategy to be SCO. Next, observe that $\Phi_1^\star(q)$ states that *if* a history of value 1 is entered *then* Player q must win. This allows us to understand the left part of the implication in $\Phi_{\mathcal{G}_p^\star}$: the implication can be read as 'if all other players play a $\star$-admissible strategy, then either p should win ($\Phi(p)$) or a state of value 1 for player p should eventually be visited ($\Diamond\mathtt{Val}_{p,1}^\star$)'. Then a strategy $\hat{\sigma}$ (in $\mathcal{G}$) that wins against admissible strategies can be extracted from a winning strategy $\overline{\sigma}$ (in $\mathcal{G}_p^\star$) in a straightforward way, *except* when $\overline{\sigma}$ enforces to reach a state of value 1 ($\Diamond\mathtt{Val}_{p,1}^\star$ in $\Phi_{\mathcal{G}_p^\star}$). In this case, σ cannot follow $\overline{\sigma}$, but must rather switch to a *winning* strategy, which:

(i) is guaranteed to exist since the state that has been reached has value 1; and

(ii) can be computed using classical techniques [11].

The strategy $\hat{\sigma}$ is not necessarily admissible but by Theorem 8 (1), there is an admissible strategy σ with $\hat{\sigma} \preccurlyeq^\star \sigma$. By weak domination, σ wins against more profiles than $\hat{\sigma}$, in particular, against the profiles of admissible strategies of the other players.

▶ **Example 12.** In our running example, observe that $\neg\mathtt{Val}_{2,0}^{\mathtt{A}} = \mathtt{Val}_{2,1}^{\mathtt{A}} = \{Win\}$ since there is no state of value -1 in $\mathcal{G}$. Hence, $\Phi(2) = \Diamond Win = \Diamond\mathtt{Val}_{2,1}^{\mathtt{A}} = \Diamond\neg\mathtt{Val}_{2,0}^{\mathtt{A}}$. Finally, $\mathtt{AfterHelpMove}_2^{\mathtt{A}} = \{(s_0,(a,b')),(s_1,(b,b'))\}$, so, after simplification: $\Phi_{\mathcal{G}_1^{\mathtt{A}}} = [\Diamond Win \vee \Box\Diamond((s_0,(a,b')) \vee (s_1,(b,b')))] \to \Diamond Win$. Thus, to win in $\mathcal{G}_1^{\mathtt{A}}$ (under the *sure* semantics), player 1 must ensure to reach Win as long as player 2 visits the set of *bold* states in Figure 3 infinitely often. A winning strategy $\overline{\sigma}$ in $\mathcal{G}_1^{\mathtt{A}}$ consists in (eventually) always playing b from

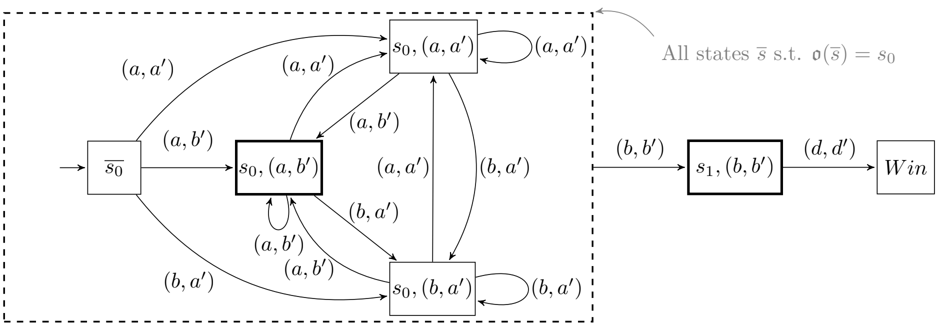


Figure 3 The game $\mathcal{G}_1^{\mathtt{A}}$ obtained from the game in Figure 1. Bold states $\big(s_0, (a, b')\big)$ and $\big(s_1, (b, b')\big)$ are the states of $\mathtt{AfterHelpMove}_2^{\mathtt{A}}$. There is a (b, b')-labelled transition from all states in the dashed rectangle to $\big(s_1, (b, b')\big)$.

all states in the dashed rectangle; and d from $\big(s_1, (b, b')\big)$. Observe that this strategy is compatible with $\mathfrak{o}$. From $\overline{\sigma}$, we can extract an admissible player 1 strategy in $\mathcal{G}$: always play b in s_0; always play d in s_1; and play a winning strategy from s_2 (which is of value 1), for instance: always play $0.5f + 0.5g$ from s_2 like σ_3 does.

We conclude by two remarks on simple safety games and on the choice of our semantics. First, note that assume-admissible synthesis is simpler in simple safety games, since the admissible strategies are exactly the $\star$-LA strategies in this case (see Theorem 8). So, one can build $\mathcal{G}_p$ from $\mathcal{G}$ by pruning the actions which are not $\star$-LA (the labelling by actions is not necessary anymore), and look for a player p winning strategy. Second, in the semantics of concurrent games considered in this paper, players see, at each step, the transition taken but not the actual moves of the other player even once they are played. An alternative semantics could be that the players discover simultaneously the moves of other players after each step, as in the Rock-Paper-Scissors game. The former semantics is more general than the latter since moves played at the preceding round can always be encoded in the current state (as we did in the construction of $\mathcal{G}_p^\star$). Our results remain meaningful in this simpler case (in particular the characterisation of admissible strategies), but assume-admissible synthesis can be performed by reducing to games with *perfect* information.

References

1 Brandenburger Adam, Friedenberg Amanda, H Jerome, et al. Admissibility in games. *Econometrica*, 2008.

2 Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002. doi:10.1145/585265.585270.

3 Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

4 Nicolas Basset, Gilles Geeraerts, Jean-François Raskin, and Ocan Sankur. Admissibility in concurrent games. *CoRR*, abs/1702.06439, 2017. URL: http://arxiv.org/abs/1702.06439.

5 Dietmar Berwanger. Admissibility in infinite games. In *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, number 4393 in Lecture Notes in Computer Science, pages 188–199. Springer, 2007. doi:10.1007/978-3-540-70918-3_17.

6 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Nash equilibria in concurrent games with Büchi objectives. In *Proceedings of the 31st Conference on Found-*

ations of Software Technology and Theoretical Computer Science (FSTTCS'11), volume 13 of *Leibniz International Proceedings in Informatics*, pages 375–386, Mumbai, India, dec 2011. Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.FSTTCS.2011.375`.

7 Romain Brenguier, Guillermo A. Pérez, Jean-François Raskin, and Ocan Sankur. Admissibility in Quantitative Graph Games. In Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen, editors, *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016)*, volume 65 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.FSTTCS.2016.42`.

8 Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. In Luca Aceto and David de Frutos-Escrig, editors, *CONCUR*, volume 42 of *LIPIcs*, pages 100–113. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. `doi:10.4230/LIPIcs.CONCUR.2015.100`.

9 Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. *Acta Inf.*, 54(1):41–83, 2017. `doi:10.1007/s00236-016-0273-2`.

10 Romain Brenguier, Jean-François Raskin, and Mathieu Sassolas. The complexity of admissibility in omega-regular games. In *CSL-LICS '14, 2014*. ACM, 2014. `doi:10.1145/2603088.2603143`.

11 Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. Qualitative concurrent parity games. *ACM Trans. Comput. Log.*, 12(4):28:1–28:51, 2011. `doi:10.1145/1970398.1970404`.

12 Krishnendu Chatterjee and Thomas A. Henzinger. Assume-guarantee synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 - April 1, 2007, Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, pages 261–275. Springer, 2007.

13 Werner Damm and Bernd Finkbeiner. Automatic compositional synthesis of distributed systems. In *FM 2014: Formal Methods - 19th International Symposium, Singapore, May 12-16, 2014. Proceedings*, volume 8442 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2014.

14 Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. Concurrent reachability games. *Theor. Comput. Sci.*, 386(3):188–217, 2007. `doi:10.1016/j.tcs.2007.07.008`.

15 Marco Faella. Admissible strategies in infinite games over graphs. In *MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2009.

16 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010.

17 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. In *Multi-Agent Systems - 12th European Conference, EUMAS 2014, Prague, Czech Republic, December 18-19, 2014, Revised Selected Papers*, pages 219–235. Springer, 2014.

18 Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3(3), 2007. `doi:10.2168/LMCS-3(3:4)2007`.

19 Moshe Y Vardi. Automatic verification of probabilistic concurrent finite state programs. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 327–338. IEEE, 1985.