Beyond Winning Strategies: Admissible and Admissible Winning Strategies for Quantitative Reachability Games

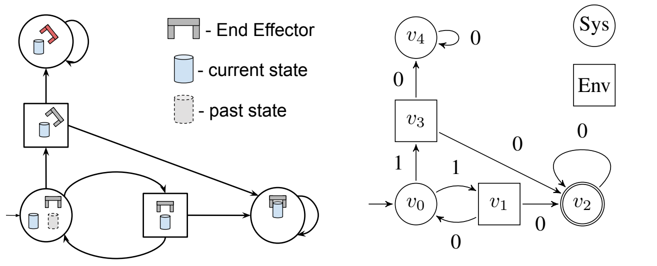
Karan Muvvala, **Qi Heng Ho** and **Morteza Lahijanian**
University of Colorado at Boulder, CO, USA
{karan.muvvala, qi.ho, morteza.lahijanian}@colorado.edu

Abstract

Classical reactive synthesis approaches aim to synthesize a reactive system that always satisfies a given specification. These approaches often reduce to playing a two-player zero-sum game where the goal is to synthesize a winning strategy. However, in many pragmatic domains, such as robotics, a winning strategy does not always exist, yet it is desirable for the system to make an effort to satisfy its requirements instead of “giving up.” To this end, this paper investigates the notion of *admissible* strategies, which formalize “doing-your-best”, in quantitative reachability games. We show that, unlike the qualitative case, memoryless strategies are not sufficient to capture *all* admissible strategies, making synthesis a challenging task. In addition, we prove that admissible strategies always exist but may produce undesirable optimistic behaviors. To mitigate this, we propose *admissible winning* strategies, which enforce the best possible outcome while being admissible. We show that both strategies always exist but are not memoryless. We provide necessary and sufficient conditions for the existence of both strategies and propose synthesis algorithms. Finally, we illustrate the strategies on gridworld and robot manipulator domains.

1 Introduction

Reactive Synthesis is the problem of automatically generating reactive systems from logical specifications, first proposed by [Church, 1963]. Its applications span a wide range of domains, including robotics [McMahon *et al.*, 2023; He *et al.*, 2017; Kress-Gazit *et al.*, 2018], program synthesis [Pnueli and Rosner, 1989], distributed systems [Filippidis and Murray, 2016], formal verification [Kupferman and Vardi, 2001], and security [Zhou and Foley, 2003]. Existing approaches to reactive synthesis usually boil down to computing a strategy over a game between a System (Sys) and an Environment (Env) player, with the goal of finding a *winning* strategy, which guarantees the Sys player achieves its objectives regardless of the Env player’s moves. In quantitative settings, the game incorporates a payoff requirement as part of the Sys player’s objectives. In many scenarios, however,



(a) Manipulation Domain (b) Two-player game abstraction

Figure 1: (a) Robotic manipulator in the presence of human. (b) Game abstraction, where weights represent robot energy.

a winning strategy may not exist, resulting in a failure in the synthesis algorithm. Prior works relax this requirement using the notions of *best effort* [Aminof *et al.*, 2021] and *admissibility* [Berwanger, 2007; Brandenburger *et al.*, 2008]. Best effort is specific to qualitative games, while admissibility is recently studied in quantitative games under strong assumptions on the Env player, namely, rationality and known objectives [Brenguier *et al.*, 2016]. This paper aims to study admissible strategies without these assumptions in quantitative reachability games with a particular focus on applications in robotics.

Consider the example in Fig. 1 with a robot (Sys player) and a human (Env player) operating in shared workspace. The robot is tasked with grasping the bin. Since the human can intervene by moving the bin before the robot completes its grasp, there is no winning strategy that enforces completion of the task under the *worst-case* Env strategy. However, in such cases it is still desirable for the robot to make an effort to satisfy its requirements instead of “giving up.”

To this end, this paper studies admissible strategies [Faella, 2009; Berwanger, 2007] in quantitative reachability games without assumptions on rationality and the objectives of the Env player. These games can model quantitative reactive synthesis for finite-behaviors expressed in, e.g., syntactically co-safe Linear Temporal Logic (LTL) [Kupferman and Vardi, 2001] and LTL over finite behaviors (LTL_f) [De Giacomo and Vardi, 2013]. We show that admissible strategies relax requirement of winning strategies, and always exist. We prove that, unlike the qualitative setting, quantitative admissible strategies are generally history-dependent even for finite payoff functions. Then, we show that such strategies can produce overly optimistic behaviors, which may be undesirable for robotics applications. To mitigate this, we propose *admissible winning* strategies, which have the desirable property of enforcing specification satisfaction when possible while being admissible. We prove that similar to admissible strategies, admissible winning strategies always exist and may require finite memory. Then, we provide necessary and sufficient conditions for both strategies and propose synthesis algorithms. Finally, we provide various robotic examples to show that admissible and admissible winning strategies provide desirable and flexible behaviors without a-priori knowledge of the objectives of other agents.

Our contributions are fourfold: (i) analysis of admissible strategies in quantitative reachability games without assumptions on rationality and objective of the Env player, including proofs of their existence, finite memory requirement, and necessary and sufficient conditions, (ii) introduction and analysis of novel notion of admissible winning strategies to mitigate over-optimism in admissible strategies, (iii) synthesis algorithms for both strategies, and (iv) illustrative examples on gridworld and manipulation domains, showing emergent behaviors under these strategies.

Related Work. Several works explore alternatives to winning strategies. Specifically, Faella investigates various concepts in qualitative games with reachability objectives within a zero-sum framework. They focus solely on the Sys player's objective without making any assumptions about the Env player. They use admissibility to define the notion of *best-effort* (BE). Aminof *et al.*; De Giacomo *et al.* further examine the complexity of synthesizing BE strategies, showing that it can be reduced to standard algorithms, with memoryless strategies being sufficient. In contrast, our work considers quantitative reachability games where the objective is to reach a goal state with minimal total cost. We show that memoryless strategies are insufficient in our context, and our synthesis approach does not reduce to standard algorithms.

The notion of admissiblity has also been explored in normal form games [Brandenburger *et al.*, 2008; Apt, 2011]. In qualitative games with logical specifications (extensive form), admissiblity has been investigated for n-player infinite games, where each player has their own objective and is assumed to play admissibly with respective to that objective – referred to as *assume admissible* (AA). Berwanger was the first to formalize this notion of AA for games played on graphs. Subsequently, Brenguier *et al.*; Brenguier *et al.* establish the complexity and give algorithms for ω-regular objectives. In our settings, we consider reachability games that terminate in finite time. Notably, we make no assumptions about the Env player, i.e., we neither know Env player's objective nor require them to play admissibly.

The work closest to ours is by [Brenguier *et al.*, 2016], who study admissibility in quantitative settings. They extend prior work [Brenguier *et al.*, 2014; Brenguier *et al.*, 2015] on infinite duration qualitative games to quantitative objectives. They give necessary and sufficient conditions for admissible strategies. Unlike their work, in our setting, we consider finite duration games and appropriately define our payoff over finite traces. We show our game is always determined and thus optimal worst-case and cooperative strategies always exist. While our analysis shares some conceptual similarities with theirs, addressing the finite play setting requires a distinct theoretical approach compared to infinite plays. Brenguier *et al.* give a sketch of their algorithm based on parity games. We, however, present a detailed yet simpler synthesis algorithm. We also analyze emergent behavior under admissible strategies in robotics settings. We observe that these strategies can be overly optimistic. To address this, we identify the underlying cause and propose the concept of admissible winning strategies to mitigate such optimism.

2 Problem Formulation

The overarching goal of this work is quantitative reactive synthesis for LTL$_f$ or cosafe LTL specifications where satisfaction cannot necessarily be guaranteed. This problem reduces to reachability analysis in quantitative games played between the Sys and Env players [Baier and Katoen, 2008]. For the sake of generality, we focus on these games.

2.1 2-Player Quantitative Games and Strategies

Definition 1 (2-player Quantitative Game)**.** *A two-player turn-based quantitative game is a tuple* $\mathcal{G} = (V, v_0, A_s, A_e, \delta, C, V_f)$*, where*

- $V = V_s \cup V_e$ *is a finite set of states, where* V_s *and* V_e *are disjoint and belong to the Sys and Env player,*
- $v_0 \in V$ *is the initial state,*
- A_s *and* A_e *are the finite sets of actions for the Sys and Env player, respectively,*
- $\delta : V \times (A_s \cup A_e) \to V$ *is the transition function such that, for* $i, j \in \{s, e\}$ *and* $i \neq j$*, given state* $v \in V_i$ *and action* $a \in A_i$*, the successor state is* $\delta(v, a) \in V_j$*,*
- $C : V \times (A_s \cup A_e) \to \mathbb{N}^0$ *is the cost (energy) function such that, for every* $(v, a) \in V_s \times A_s$*,* $C(v, a) > 0$*, otherwise* 0*, and*
- $V_f \subseteq V$ *is a set of goal (final) states.*

We assume that the game is non-blocking. There is at least one outgoing transition from every state, i.e., $\forall v \in V, \exists a \in A$ s.t. $\delta(v, a) \neq \emptyset$. Note that the Env action cost is zero since we are solely interested in the Sys player objectives (action costs) and make no assumption about the objective of the Env player. Finally, we assume that our transition function is deterministic and injective, i.e., $\delta(v, a) = \delta(v, a')$ iff $a = a'$.

The evolution of game $\mathcal{G}$ starts from v_0 and is played in turns between the Sys and Env player. At state $v \in V_i$, where $i \in \{s, e\}$, Player i picks an action $a \in A_i$ and incurs cost $C(v, a)$. Then, the game evolves to the next state according to the transition function $\delta(v, a) \in V_j$, where $j \neq i$. Then, Player j picks an action, and the process repeats. The game terminates if a goal state in V_f is reached. For the remainder of the paper, all definitions are provided with respect to the game $\mathcal{G}$. For brevity, we omit explicitly restating this context.

The players choose actions according to a strategy. Formally,

Definition 2 (Strategy)**.** *A strategy σ (τ) is a function that maps a finite sequence of states to a Sys (Env) action, such that $\sigma : V^* \cdot V_s \to A_s$ and $\tau : V^* \cdot V_e \to A_e$, where $\cdot$ is the concatenation operator. We denote Σ and T as the set of all strategies for the Sys and Env player, respectively. A strategy is called* memoryless *or* positional *if it only depends on the last state in the sequence.*

Given strategies σ and τ, a unique sequence of states, called *play* and denoted by $P^{v_0}(\sigma, \tau)$, is induced from v_0. Note that the play is unique for terminating plays only. A play can be finite $P^{v_0}(\sigma, \tau) := v_0 v_1 \dots v_n \in V^*$ or infinite $P^{v_0}(\sigma, \tau) := v_0 v_1 \dots \in V^\omega$. We denote by $\text{Plays}^v := \{P^v(\sigma, \tau) \mid \sigma \in \Sigma,\ \tau \in \mathrm{T}\}$ the set of plays starting from v under every Sys and Env strategy. $\text{Plays}^v(\sigma)$ is the set of plays induced by a fixed strategy σ and every Env strategy. We note that a finite play occurs iff a goal state is reached.

A finite prefix of a play is called the history h. We define $|h|$ the length of the history and h_j for $0 \le j \le |h| - 1$ as the $(j + 1)^{th}$ state in the sequence. The last vertex of a history h is defined as $\text{last}(h) := h_{|h|-1}$. We denote the set of plays with common prefix h as $\text{Plays}^h := \{P^h = h \cdot P \mid P \in \text{Plays}^v(\sigma, \tau), \sigma \in \Sigma, \tau \in \mathrm{T}, v = \delta(\text{last}(h), \sigma(h)) \text{ if } v \in V_s, \text{ else } v = \delta(\text{last}(h), \tau(h))\}$.

In a qualitative reachability game, the objective of the Sys player is to choose σ such that every play in $\text{Plays}^{v_0}(\sigma)$ reaches a state in V_f. In a quantitative reachability game, the Sys player has an additional objective of minimizing the total cost of its actions along the play, called the payoff.

Definition 3 (Total Payoff)**.** *Given strategies $\sigma \in \Sigma$ and $\tau \in \mathrm{T}$,* total payoff *is defined as the sum of all the action costs given by C along the induced play $P^{v_0}(\sigma, \tau) = v_0 v_1 \dots v_n$ where $n \in \mathbb{N} \cup \{\infty\}$, i.e.,*

$$\text{Val}(P^{v_0}(\sigma, \tau)) := \sum_{i=0}^{n-1} C(v_i, a_i), \tag{1}$$

where $a_i = \sigma(v_0 \dots v_i)$ if $v_i \in V_s$, else $a_i = \tau(v_0 \dots v_i)$.

Note that, for a play with infinite length $|P^{v_0}(\sigma, \tau)| = \infty$, $\text{Val}(P^{v_0}(\sigma, \tau)) = \infty$. We now define two notions of payoff for a game $\mathcal{G}$ that formalizes best-case and worst-case scenarios for the Sys player with respect to Val.

Definition 4 (Cooperative & Adversarial Values)**.** *Given h, Sys (σ) and Env (τ) strategies compatible with h, let $P^h(\sigma, \tau)$ denote a play that extends history h. The* cooperative value *$\text{cVal}(h, \sigma)$ is the payoff of the play such that the Env player plays minimally, i.e.,*

$$\text{cVal}(h, \sigma) = \inf_{\tau \in \mathrm{T}} \text{Val}(P^h(\sigma, \tau)). \tag{2}$$

Similarly, the adversarial value *$\text{aVal}(h, \sigma)$ is the payoff where the Env player plays maximally, i.e.,*

$$\text{aVal}(h, \sigma) = \sup_{\tau \in \mathrm{T}} \text{Val}(P^h(\sigma, \tau)). \tag{3}$$

We denote by $\text{cVal}(h)$ and $\text{aVal}(h)$ the optimal cooperative and adversarial values for history h, respectively, i.e.,

$$\text{cVal}(h) = \inf_{\sigma \in \Sigma} \text{cVal}(h, \sigma) \ \textit{ and } \ \text{aVal}(h) = \inf_{\sigma \in \Sigma} \text{aVal}(h, \sigma).$$

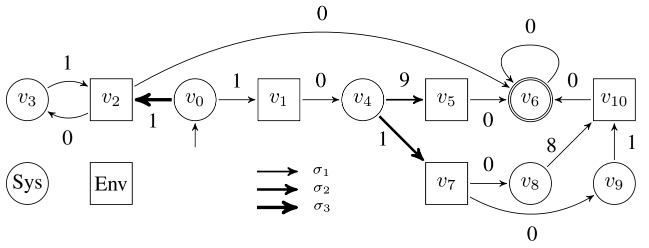


Figure 2: Illustrative game. v_0 is initial and v_6 is goal state.

For a given h, we say Sys strategy $\sigma_{win} \in \Sigma$ is *winning* if σ_{win} is compatible with h and $\text{aVal}(h, \sigma_{win}) < \infty$. If σ_{win} exists, the Sys player can force a visit to V_f for all Env strategies, guaranteeing its reachability objective. In classical quantitative reactive synthesis, the interest is in a σ_{win} that achieves the optimal $\text{aVal}(v_0)$.

2.2 Beyond Winning Strategies

In many applications, when a winning strategy does not exist, it is desirable for the Sys player to adopt a strategy that ensures the best possible outcome. For game $\mathcal{G}$ in Fig. 1b, σ_{win} does not exist but it is better for the Sys player to keep trying, i.e., move to v_1 or v_3 rather than giving up. To capture this intuition, we turn to the classical notion of dominance [Leyton-Brown and Shoham, 2008].

Definition 5 (Dominance [Brenguier *et al.*, 2016; Berwanger, 2007])**.** *Given two Sys strategies $\sigma, \sigma' \in \Sigma$ and initial state $v \in V$, we say*

- σ very weakly dominates σ'*, denoted by $\sigma \succeq \sigma'$, if σ does at least as well as σ':*

$$\text{Val}(P^v(\sigma, \tau)) \le \text{Val}(P^v(\sigma', \tau)) \quad \forall \tau \in \mathrm{T}. \tag{4}$$

- σ weakly dominates σ'*, denoted by $\sigma \succ \sigma'$, if $\sigma \succeq \sigma'$ and σ sometimes does better than σ' :*

$$\text{Val}(P^v(\sigma, \tau)) < \text{Val}(P^v(\sigma', \tau)) \quad \exists \tau \in \mathrm{T}. \tag{5}$$

Dominance induces a partial order on Sys strategies, whose maximal elements are called admissible strategies. Strategies that are not dominated are always admissible.

Definition 6 (Admissible Strategy)**.** *A strategy σ is called* admissible *if it is not* weakly dominated *by any other Sys player strategy, i.e., $\nexists \sigma' \in \Sigma$ s.t. $\sigma' \succ \sigma$.*

Example 1. *In the game in Fig. 2, let $\sigma_1 : (v_0 \to v_1), (v_4 \to v_5)$, $\sigma_2 : (v_0 \to v_1), (v_4 \to v_7)$, and $\sigma_3 : (v_0 \to v_2), (v_3 \to v_2)$. As $\sigma_2 \succ \sigma_1$, σ_1 is not admissible. But, σ_3 is not weakly dominated by σ_2 as there exists a play under which σ_3 does strictly better than σ_2 for every play in* $\text{Plays}^{v_0}(\sigma_2)$*. Hence, σ_3 and σ_2 are both admissible. This example demonstrates that (i) admissible strategies can be overly optimistic, and (ii) they differ from winning strategies.*

Brandenburger *et al.* rationalize a player playing admissibly to be doing their *best*. Thus, the first problem we consider is the synthesis of admissible strategies.

Problem 1. *Given a 2-player quantitative reachability game $\mathcal{G}$ and energy budget $\mathcal{B} \in \mathbb{N}^+$, synthesize the set of all admissible strategies Σ_{adm} such that, for every $\sigma \in \Sigma_{adm}$, there exists $\tau \in \mathrm{T}$ under which $\mathrm{Val}(P^{v_0}(\sigma, \tau)) \leq \mathcal{B}$.*

Intuitively, $\sigma \in \Sigma_{adm}$ allows the Sys player to do its best without any assumptions about the Env player. In qualitative settings, winning strategies are always admissible [Brenguier *et al.*, 2015, Lemma 8]. In quantitative settings, however, winning strategies are not necessarily admissible (e.g., σ_1 in Fig. 2). Thus, to compute enforceable admissible strategies, we introduce the notion of an admissible winning strategy.

Definition 7 (Admissible Winning Strategy). *Strategy σ is called* admissible winning *for Sys player iff it is admissible and $\forall h \in \mathrm{Plays}^{v_0}(\sigma)$ if $\mathrm{aVal}(h) < \infty$ then $\mathrm{aVal}(h, \sigma) < \infty$.*

For the game in Fig. 2, σ_3 is admissible but not enforcing as it cannot ensure always reaching the goal state. σ_1 is winning but not admissible. σ_2 is admissible winning, which is more desirable than the other two. In this work, we also consider the synthesis problem of such strategies.

Problem 2. *Given a 2-player quantitative reachability game $\mathcal{G}$ and energy budget $\mathcal{B} \in \mathbb{N}^+$, synthesize the set of all admissible winning strategies Σ_{adm}^{win} such that, for every $\sigma \in \Sigma_{adm}^{win}$ and $\forall h \in \mathrm{Plays}^{v_0}(\sigma)$, if $\mathrm{aVal}(h) < \infty$ then $\mathrm{aVal}(h, \sigma) \leq \mathcal{B}$.*

In Section 3, we show how to solve Problem 1 by providing necessary and sufficient conditions for a strategy to be admissible. In Section 4, we identify the class of admissible strategies that are admissible winning and give an algorithm to solve Problem 2. Due to space constraints, the proofs of all theoretical claims are provided in Appendix.

3 Admissible Strategies

To have a sound and complete algorithm for the synthesis of Σ_{adm}, we need to first understand the characteristics and properties of admissible strategies. Prior works in qualitative reachability games show that synthesis can be reduced to strategies with *value-preserving* property (defined below). We show that in our quantitative setting, this property does *not* hold. Thus, we investigate the appropriate conditions that characterize admissible strategies. We identify two classes of strategies that are not only sufficient but also necessary for a strategy to be admissible. Finally, we show how memoryless strategies are not sufficient for admissibility and provide a synthesis algorithm.

3.1 Admissible Strategies are not Value-Preserving

The reachability objective in $\mathcal{G}$ naturally partitions the set of states V into three subsets: the set of states from which the Sys player (i) can force a visit to V_f under every Env strategy, (ii) cannot reach V_f under any Env strategy, and (iii) may reach V_f only under some Env strategies. We can formalize these sets using $\mathrm{cVal}(v)$ and $\mathrm{aVal}(v)$:

winning region: $V_{win} = \{v \in V \mid \mathrm{aVal}(v) < \infty\}$,
losing region: $V_{los} = \{v \in V \mid \mathrm{aVal}(v) = \mathrm{cVal}(v) = \infty\}$,
pending region: $V_{pen} = \{v \in V \mid \mathrm{aVal}(v) = \infty, \mathrm{cVal}(v) < \infty\}$.

	SC	**WCo-Op**	**mSC**	Adm.	Adm. winning
Value-preserving	✗	✓	✓	✗	✓
Winning	✗	✓	✓	✗	✓
Memoryless	✗	✓	✗	✗	✗
Algorithm	Sec. 3	Sec. 3	Sec. 4	Sec. 3	Sec. 4

Table 1: Properties of new strategies defined for admissibility.

Note V_{win}, V_{los}, and V_{pen} define a partition for V, i.e., their union is V and their pair-wise intersection is the empty set. Based on these sets, we characterize value-preserving strategies according to the notion of value for each state. Let $\mathrm{sVal} : V \to \{-1, 0, 1\}$ be a state-value function such that $\mathrm{sVal}(v) = 1$ if $v \in V_{win}$, 0 if $v \in V_{pen}$, and -1 if $v \in V_{los}$.

Definition 8 (Value-Preserving). *We say history h is value-preserving if $\mathrm{sVal}(h_j) \leq \mathrm{sVal}(h_{j+1})$ for all $0 \leq j < |h| - 1$. Strategy σ is value preserving if every $h \in \mathrm{Plays}^{v_0}(\sigma)$ is value preserving.*

Let us now look at two classical notions for strategies defined for quantitative games and discuss their value-preserving property. We say σ is a *worst-case optimal* strategy (**WCO**) if $\mathrm{aVal}(h, \sigma) = \mathrm{aVal}(h)$. If at the current state v, $\mathrm{sVal}(v) = 1$, then an optimal winning strategy σ_{win} exists such that all plays in $\mathrm{Plays}^{v}(\sigma_{win})$ are value-preserving. Since σ_{win} is **WCO**, every **WCO** strategy in V_{win} is also value preserving. If $\mathrm{sVal}(v) \neq -1$, a *cooperatively-optimal* (**Co-Op**) strategy σ exists such that $\mathrm{cVal}(h, \sigma) = \mathrm{cVal}(h)$. Unlike **WCO**, **Co-Op** strategies are not value-preserving. In Fig. 2, σ_1 and σ_2 are **WCO** as they ensure the lowest payoff of 10 if Env is adversarial while σ_3 is **Co-Op** as the corresponding payoff of 1 is the lowest for a cooperative Env. Further, notice that σ_3, while admissible, is not value preserving. The following lemma formalizes this observation.

Lemma 1. *Admissible strategies are* not *always value preserving.*

Unlike the qualitative setting [Faella, 2009; Aminof *et al.*, 2020], Lemma 1 shows that we cannot characterize admissible strategies solely on the basis of **WCO** strategies as they are not value-preserving in our quantitative setting. Below, we derive two new categories of strategies from **WCO** and **Co-Op** that are always admissible. We then discuss their properties and show that they are also necessary conditions for admissibility. Table 1 summarizes properties of all new strategies we define hereafter.

3.2 Characterization of Admissible Strategies

Note that, for every history h, the strategies that are cooperative optimal have the least payoff. Thus, every σ that is **Co-Op** is admissible as there does not exist σ' that weakly dominates it. We now define *strongly cooperative* condition (**SC**) which generalizes **Co-Op**. Intuitively, strategies that are **SC** have a lower payoff than the worst-case optimal payoff at h. In case, a lower payoff cannot be obtained, **SC** are worst-case optimal.

Definition 9 (**SC**). *Strategy σ is* Strongly Cooperative (**SC**) *if for every $h \in \mathrm{Plays}^{v_0}(\sigma)$ one of the following two conditions holds: (i) if $\mathrm{cVal}(h) < \mathrm{aVal}(h)$ then $\mathrm{cVal}(h, \sigma) <$*

$\mathrm{aVal}(h)$*, or (ii) if* $\mathrm{cVal}(h) = \mathrm{aVal}(h)$ *then* $\mathrm{aVal}(h,\sigma) = \mathrm{cVal}(h,\sigma) = \mathrm{aVal}(h)$.

In the game in Fig. 2, both σ_2 and σ_3 are $\mathbf{SC}$ strategies.

Let σ' be a strategy that is not $\mathbf{SC}$. If $\mathrm{cVal}(h,\sigma') > \mathrm{aVal}(h)$ then σ' always has a payoff worse than a $\mathbf{WCO}$ strategy. If $\mathrm{cVal}(h,\sigma') = \mathrm{aVal}(h)$ then σ' does as well as a $\mathbf{WCO}$ strategy but never better. Thus, σ' does not weakly dominate a $\mathbf{SC}$ strategy, resulting in the following lemma.

Lemma 2. *All* $\mathbf{SC}$ *strategies are admissible.*

From Lemma 2, it suffices for us to show that $\mathbf{SC}$ strategies always exist to prove that admissible strategies always exist. Unfortunately, $\mathbf{SC}$ strategies are history-dependent, i.e., we need to reason over every state along a history to check for admissibility. This is formalized as follows.

Theorem 1. *Memoryless strategies are* not *sufficient for* $\mathbf{SC}$ *strategies.*

We now look at another interesting class of strategy that is always admissible. For every history h, $\mathbf{WCO}$ strategies always guarantee the worst-case payoff. A notable subset of $\mathbf{WCO}$ strategies are those that are also $\mathbf{Co\text{-}Op}$. We call such strategies *Worst-case Cooperative Optimal* ($\mathbf{WCo\text{-}Op}$).

Definition 10 ($\mathbf{WCo\text{-}Op}$)**.** *Strategy* σ *is* Worst-case Cooperative Optimal ($\mathbf{WCo\text{-}Op}$) *if, for all* $h \in \mathrm{Plays}^{v_0}(\sigma)$,

$$\mathrm{aVal}(h,\sigma) = \mathrm{aVal}(h) \;\; \textit{and} \;\; \mathrm{cVal}(h,\sigma) = \mathrm{acVal}(h),$$

where $\mathrm{acVal}(h) := \min\{\mathrm{cVal}(h,\sigma) \mid \sigma \in \Sigma, \mathrm{aVal}(h,\sigma) \leq \mathrm{aVal}(h)\}$ *is the optimal adversarial-cooperative value of h.*

In this definition, acVal is a new notion that characterizes the minimum payoff that Sys player can obtain from the set of worst-case optimal strategies. In the game in Fig. 2, only σ_2 is $\mathbf{WCo\text{-}Op}$ strategy. That is because action $v_4 \to v_7$ belongs to an admissible strategy as it has the minimum cooperative value while ensuring the worst-case optimal payoff. Here $\mathrm{aVal}(v_4,\sigma_2) = \mathrm{aVal}(v_4) = 9$; $\mathrm{cVal}(v_4,\sigma_2) = 2$.

Let σ' be a strategy that is not $\mathbf{WCo\text{-}Op}$ and $\mathbf{WCO}$. If σ is $\mathbf{WCO}$, then the worst-case payoff of σ' is greater than σ's worst-case payoff. As there exists a play under σ' that does strictly worse than all plays under σ, it cannot dominate σ. Thus, σ is admissible.

Lemma 3. *All* $\mathbf{WCo\text{-}Op}$ *strategies are admissible.*

From Lemma 3, it suffices for us to show that $\mathbf{WCo\text{-}Op}$ strategies always exist for admissible strategies to always exist. Interestingly, unlike prior work [Brenguier *et al.*, 2016], in our case, $\mathbf{WCo\text{-}Op}$ strategies always exist and at least one is memoryless. The latter is desirable because memoryless strategies can be computed efficiently using fixed-point-based algorithms [Baier and Katoen, 2008].

Theorem 2. $\mathbf{WCo\text{-}Op}$ *strategies always exist and at least one is memoryless.*

The proof relies on the fact that memoryless strategies are sufficient for aVal and cVal, and proceeds by constructing a subgame $\bar{\mathcal{G}}$ for a given history h and showing that the cooperative value of the initial state in $\bar{\mathcal{G}}$ equals the $\mathrm{acVal}(h)$ of $\mathcal{G}$. A consequence of Theorem 2 is that a subset of admissible strategies, precisely $\mathbf{WCo\text{-}Op}$ strategies, are history-independent even for a payoff that is history-dependent. Thus, for σ to be admissible, it is sufficient to be $\mathbf{SC}$ or $\mathbf{WCo\text{-}Op}$, i.e.,

$$\big(\mathrm{cVal}(h,\sigma) < \mathrm{aVal}(h)\big) \vee \tag{6a}$$

$$\big(\mathrm{aVal}(h) = \mathrm{aVal}(h,\sigma) \wedge \mathrm{cVal}(h,\sigma) = \mathrm{acVal}(h)\big) \tag{6b}$$

As shown below (Theorem 3), $\mathbf{SC}$ and $\mathbf{WCo\text{-}Op}$ are also necessary conditions for admissibility. This becomes useful for synthesizing the set of all admissible strategies.

3.3 Existence of Admissible Strategies

By simplifying Eq. (6), we get the following theorem.

Theorem 3. *A strategy* σ *is admissible if, and only if,* $\forall h \in \mathrm{Plays}^{v_0}(\sigma)$ *with* $\mathrm{last}(h) \in V_s$*, the following holds*

$$\big(\mathrm{cVal}(h,\sigma) < \mathrm{aVal}(h)\big) \;\; \vee \tag{7a}$$

$$\big(\mathrm{aVal}(h) = \mathrm{aVal}(h,\sigma) = \mathrm{cVal}(h,\sigma) = \mathrm{acVal}(h)\big). \tag{7b}$$

The proof uses Lemma 2 and 3 for the sufficient conditions, and for the necessary conditions, it shows that $\neg$Eq.(7) $\implies \sigma \notin \Sigma_{adm}$. We now establish the existence of admissible strategies.

Theorem 4. *There always exists an admissible strategy* σ_{adm} *in a 2-player, turn-based, total-payoff, reachability game* $\mathcal{G}$*, and* σ_{adm} *solves Problem 1 if* $\mathrm{cVal}(v_0,\sigma_{adm}) \leq \mathcal{B}$.

The proof follows from Lemma 3 and Theorem 2. Observe that Theorem 3 characterizes the set of *all* admissible strategies and Theorem 4 establishes their existence in full generality, i.e., independent of $\mathcal{B}$, for 2-player turn-based games with reachability objectives. For computability considerations, we bound the payoffs associated with plays to a given budget $\mathcal{B}$ so that the set of all admissible strategies is finite. This is a reasonable assumption that is often used in, e.g., energy games with fixed initial credit and robotics application with finite resources [Chakrabarti *et al.*, 2003; Bouyer *et al.*, 2008; Muvvala and Lahijanian, 2023; Filiot *et al.*, 2010].

3.4 Admissible Strategy Synthesis

Given $\mathcal{G}$ and budget $\mathcal{B}$, we first construct a game tree arena that captures all plays with payoff less than or equal to $\mathcal{B}$. Next, we show that the payoff function on this tree is history-independent, which allows us to modify Theorem 3 and compute a finite set of aVals. We conclude the following from Lemma 2 and Theorem 1.

Corollary 1. *Memoryless strategies are* not *sufficient for admissible strategies.*

A consequence of Corollary 1 is that we cannot use a backward induction-based algorithm to compute these strategies. Instead, we use a forward search algorithm that starts from the initial state and recursively checks whether the admissibility constraints are satisfied.

Game Tree Arena. The algorithm is outlined in Alg. 1. Given game $\mathcal{G}$ and budget $\mathcal{B}$, we construct a tree of plays $\mathcal{G}'$ by unrolling $\mathcal{G}$ until the payoff associated with a play exceeds $\mathcal{B}$ or a goal state is reached. Every play in $\mathcal{G}$ corresponds to a branch in $\mathcal{G}'$. Every play that reaches a goal state with a

Algorithm 1: Admissible Strategy Synthesis

```
Input  : Game $\mathcal{G}$, Budget $\mathcal{B}$
Output: Strategy $\Sigma_{adm}$
1  $\mathcal{G}' \leftarrow$ Unroll $\mathcal{G}$ up until payoff $\mathcal{B}$
2  aVal; cVal $\leftarrow$ ValueIteration ($\mathcal{G}'$)
3  forall $v$ in $\mathcal{G}'$ do  acVal($v$) $\leftarrow$ as per Def. 10;
4  if $\mathcal{B} <$ cVal($v_0$) then return $\mathcal{G}$;
5  $h$.push$\big((v_0, \{\delta(v_0, a_s)\})\big)$ # let $h$ be a stack
6  while $h \neq \emptyset$ do
7      $v, \{v'\} \leftarrow h[-1]$
8      try $v' \leftarrow$ next(iter($\{v'\}$)):
9          if $v \in V_s$ and ((8a) $\vee$ (8b)) holds then
10             $h$.push$\big((v', \{\delta(v', a_e)\})\big)$
11             $\Sigma_{adm} : h \rightarrow v'$ # Add only states in $h$
12         if $v \in V_e$ then $h$.push$\big((v', \{\delta(v', a_s)\})\big)$;
13     catch StopIteration: $h$.pop();
14 return $\Sigma_{adm}$
```

payoff $b \leq \mathcal{B}$ in $\mathcal{G}$ is a play that ends in a leaf node in $\mathcal{G}'$, which is marked as a goal state for Sys player. The leaf nodes in $\mathcal{G}'$ that correspond to the plays with payoff $b > \mathcal{B}$ in $\mathcal{G}$ are assigned a payoff of $+\infty$. Further, in $\mathcal{G}'$, the weights along all the edges are zero, and the payoffs are strictly positive only when a play reaches a leaf node, otherwise it is zero. By construction, the payoff function Val is history-independent in $\mathcal{G}'$. Then, Theorem 3 can be restated as the following lemma.

Lemma 4. *Given $\mathcal{G}'$, strategy σ is admissible if and only if $\forall h \in \text{Plays}^{v_0}(\sigma)$ with $\text{last}(h) \in V_s$ and $v' = \delta(\text{last}(h), \sigma(h))$, the following holds,*

$$\big(\text{cVal}(v') < \min\{\text{aValues}\}\big) \vee \tag{8a}$$

$$\big(\text{aVal}(v^\dagger) = \text{aVal}(v') = \text{cVal}(v') = \text{acVal}(v^\dagger)\big) \tag{8b}$$

where $\text{aValues} := \{\text{aVal}(v) \mid v \in h\}$ *is the set of adversarial values along history h and $v^\dagger = \text{last}(h)$.*

After constructing $\mathcal{G}'$, Alg. 1 computes aVal and cVal values for each state in $\mathcal{G}'$ using the Value Iteration algorithm [Brihaye *et al.*, 2017]. If $\mathcal{B} < \text{cVal}(v_0)$, then there does not exist a play that reaches a goal state in $\mathcal{G}'$. Thus, all strategies in $\mathcal{G}$ are admissible. To compute admissible strategies on $\mathcal{G}'$, we use a DFS algorithm to traverse every play and check if the admissibility criteria from Lemma 4 is satisfied. If yes, we add the history and the successor state to Σ_{adm}. We repeat this until every state in $\mathcal{G}'$ is explored. This algorithm is sound and complete with polynomial time complexity.

Theorem 5 (Sound and Complete)**.** *Given $\mathcal{G}$ and a budget $\mathcal{B}$, Alg. 1 returns the set of all admissible strategies Σ_{adm}. The algorithm runs in polynomial time when $\mathcal{B}$ is fixed and in pseudo-polynomial time when $\mathcal{B}$ is arbitrary.*

4 Admissible Winning Strategies

Although value preservation is a desirable attribute, Lemma 1 shows that admissible strategies do not ensure this. In contrast, an admissible winning strategy is value-preserving and enforces reaching a goal state from the winning region. For instance, for the game in Fig. 2, an admissible winning strategy commits to v_1 from v_0, which is value-preserving. This is desirable as it ensures reaching v_6, while the other admissible strategies do not.

Here, we identify the subset of admissible strategies that are not value-preserving (aka, optimistic strategy), and prove that if they exist, they must be $\mathbf{SC}$. Next, we propose $\mathbf{mSC}$ that are admissible winning and show that $\mathbf{mSC}$ and $\mathbf{WCo\text{-}Op}$ are admissible winning. Finally, we show that they always exist and give our synthesis algorithm.

4.1 Optimistic Strategy

In Sec. 3.1, we show that $\mathbf{SC}$ strategies are willing to risk a higher payoff $(\text{aVal}(\delta(\text{last}(h), \sigma(h))) > \text{cVal}(h, \sigma))$ in the hopes that the Env will cooperate, i.e., they are optimistic.

Definition 11 (Optimistic Strategy)**.** *Strategy σ is an optimistic strategy if, and only if, σ is admissible but not value preserving.*

We now show that if optimistic strategies exist, then they must be $\mathbf{SC}$ strategies. This implies, $\mathbf{WCo\text{-}Op}$ strategies are never optimistic strategies.

Lemma 5. *If an optimistic strategy exists, it must be* $\mathbf{SC}$ *strategy.*

Thus, to enforce value-preserving, we modify Def. 9.

Definition 12 ($\mathbf{mSC}$)**.** *For all $h \in \text{Plays}^{v_0}(\sigma)$, strategy σ is* $\mathbf{mSC}$*, if σ is* $\mathbf{SC}$ *and value-preserving, i.e., if* $\text{sVal}(\text{last}(h)) = 1$ *then* $\text{sVal}(\delta(\text{last}(h), \sigma(h))) = 1$.

We say that every strategy that is either $\mathbf{WCo\text{-}Op}$ or $\mathbf{mSC}$ is admissible winning as they are value preserving and admissible. In Fig. 2, both σ_2 and σ_3 are admissible, but σ_3 is not $\mathbf{mSC}$ as it is not value preserving. Thus, only σ_2 is admissible winning strategy.

Theorem 6. *A strategy σ is* admissible winning *if, and only if, $\forall h \in \text{Plays}^{v_0}(\sigma)$ with $\text{last}(h) \in V_s$, the following holds*

$$\Big(\big(\text{cVal}(h, \sigma) < \text{aVal}(h)\big) \wedge \tag{9a}$$

$$\big(\text{sVal}(\text{last}(h)) = 1 \implies \text{sVal}(\delta(\text{last}(h), \sigma(h))) = 1\big)\Big) \vee$$

$$\big(\text{aVal}(h) = \text{aVal}(h, \sigma) = \text{cVal}(h, \sigma) = \text{acVal}(h)\big) \tag{9b}$$

The additional term in Eq. (9a) in comparison to Eq. (6a) constrains a strategy σ that satisfies $\mathbf{SC}$ condition in the winning region to action(s) such that $\delta(\text{last}(h), \sigma(h)) \in V_{win}$. For all h with $\text{last}(h) \notin V_{win}$, the condition is the same as the $\mathbf{SC}$ condition in Eq. (6a).

Lemma 6. *There always exists an admissible winning σ^{win}_{adm} strategy in a 2-player, turn-based, total-payoff, reachability games $\mathcal{G}$, and σ^{win}_{adm} solves Problem 2 if* $\text{aVal}(v_0, \sigma^{win}_{adm}) \leq \mathcal{B}$.

4.2 Admissible Winning Strategy Synthesis

Here, we give an algorithm to solve Problem 2. Similar to admissible strategies, admissible winning strategies are history-dependent as shown by the next theorem.

Theorem 7. *Memoryless strategies are* not *sufficient for admissible winning strategies.*

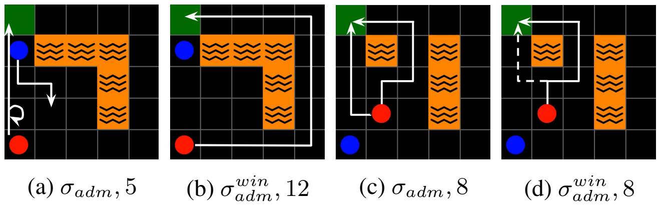

(a) $\sigma_{adm}, 5$ (b) $\sigma_{adm}^{win}, 12$ (c) $\sigma_{adm}, 8$ (d) $\sigma_{adm}^{win}, 8$

Figure 3: Gridworld Example. Sub-captions show strategy and $\mathcal{B}$.

The algorithm is similar to Alg. 1 except for the admissibility checking criteria. The tree construction, syntax, and semantics are as outlined in Sec. 3.4. We modify admissibility checking criteria in Line 9 of Alg. 1 to $\text{cVal}(v') < \min\{\text{aValues}\}$ and $\neg(v \in V_{win}) \vee (v' \in V_{win})$ This modification expresses the value-preserving term from Eq. (9a).

Theorem 8 (Sound and Complete)**.** *Given $\mathcal{G}$ and budget $\mathcal{B}$, the algorithm described returns the set of all admissible winning strategies Σ_{adm}^{win} and has same time complexity as Alg. 1.*

5 Illustrative Examples

We now discuss the emergent behavior under admissible and admissible winning strategies in two settings: (i) a gridworld domain, where two agents take actions in turns, and (ii) a manipulator domain, where a robotic arm and a human operate in a shared workspace. For both domains, the task φ is specified using LTL_f formulas. We first construct a game abstraction $\mathcal{G}$. Next, we construct a Deterministic Finite Automaton for the task [Fuggitti, 2019] and take the product to construct the product game, where the objective for the Sys player is to reach a set of goal (accepting) states. Additional experiments and empirical validations of PTIME complexity of Alg. 1 for fixed $\mathcal{B}$ are provided in the extended version [Muvvala *et al.*, 2025]. Code is available on Github [Muvvala, 2025].

Gridworld. Fig. 3 illustrates a gridworld domain with Sys (red), Env (blue), goal (green), and lava (orange) states. The objective for Sys is to reach the goal state while avoiding the Env player. The players should not enter lava and Env player cannot traverse through goal state. Action cost is 1 for all Sys player actions. The game is played in turns starting with Sys where both players take a step in each cardinal direction.

Figs. 3a and 3c illustrate the initial position of both players for two different scenarios. Figs. 3a-3b and 3c-3d illustrate σ_{adm} and σ_{adm}^{win} for different budgets. In Figs. 3a-3b, a winning strategy exists if $\mathcal{B} \geq 12$. Hence, for $\mathcal{B} = 5$ (Fig. 3a), only σ_{adm} exists, which relies on Env's cooperation to reach goal state. For $B = 12$ (Fig. 3b), σ_{adm}^{win} exists and commits to go around and reach the goal state. Note that σ_{adm} (not shown) also exists but does not commit to go around.

In Fig. 3c, we start in the winning region. Despite $v_0 \in V_{win}$, σ_{adm} chooses to go west (in V_{pen}) or north (in V_{win}). In contrast, σ_{adm}^{win} in Fig. 3d stays in the winning region (goes north) as it is the sole winning strategy that is also admissible, with a potentially shorter path (dashed) if Env cooperates.

Comparison against Best-Effort (BE): Here we illustrate differences between BE and admissible strategies. Besides qualitative vs quantitative, in the winning region, a major difference is that every σ_{win} is also BE, but not every σ_{win} is admissible winning. In Fig. 3d, if Env player stays in the left corner (cooperative), σ_{adm}^{win} *commits* to dashed-line strategy as the solid-line strategy is not $\mathbf{Co\text{-}Op}$, whereas BE does not distinguish between the two strategies as both are winning.

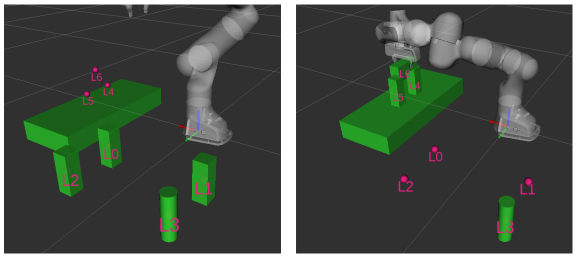


Figure 4: Left: Initial setup. Right: σ_{adm} for $\mathcal{B} = 13$. Task: $\varphi = (p_{04} \wedge p_{15} \wedge \bigcirc(p_{26} \vee p_{36}))$ where p_{ij} is block i at loc j. Blocks 0 to 2 are boxes and block 3 is a cylinder. Supplementary Video: https://youtu.be/t0TMOC_PrNk

Manipulator Domain. This domain considers a robotic arm (Sys) operating in presence of a human (Env), as described in [Muvvala *et al.*, 2022]. The task for the robot is to build an arch with two boxes as support and one block (cylinder or box) on top as shown in Fig. 4. The human can choose to either move a block back to the initial position or not intervene. The robot's actions are `transit`, `grasp`, `transfer`, and `release` with unit cost for each action, except for transferring the cylinder, which costs two units.

A winning strategy from the initial state does not exist as the human can always undo robot actions. However, an admissible strategy σ_{adm} that solves Problem 1 exists. Here, σ_{adm} continually attempts to build the arch with either a box or the cylinder on top. Interestingly, a subset of σ_{adm}, $\mathbf{WCo\text{-}Op}$ strategies, commit to building an arch with the box on top as it ensures the least payoff among all $\mathbf{WCO}$ strategies. Note that synthesizing $\mathbf{WCo\text{-}Op}$ is easy as it can be constructed from $\mathbf{WCO}$ and $\mathbf{Co\text{-}Op}$ (Line 2 of Alg. 1). Thus, σ_{adm} allows the robot to operate beyond the winning region while more desirable (less costly) behaviors can also be extracted via $\mathbf{WCo\text{-}Op}$ strategies.

Comparison against Best-Effort (BE): BE strategies build arch with either a box or cylinder on top, whereas $\mathbf{WCo\text{-}Op}$ strategies use a box on top, which is more desirable.

6 Conclusion

This paper relaxes the requirement of winning strategies in quantitative, reachability games using the notion of admissibility. While we show that admissible strategies are desirable in such settings, they are hard to synthesize due to their history-dependence. We show that such strategies can produce overly optimistic behaviors, and propose admissible winning strategies to mitigate them. We specifically show that admissible winning strategies are appropriate for robotics applications and their synthesis does not require more effort than synthesizing admissible strategies. Future work should explore classes of admissible winning strategies that are more risk-averse, e.g., regret-minimizing admissible strategies.

Acknowledgments

This work was supported in part by the University of Colorado Boulder and by Strategic University Research Partnership (SURP) grants from the NASA Jet Propulsion Laboratory (JPL) (RSA 1688009 and 1704147). The authors thank the reviewers for their constructive feedback.

References

[Aminof *et al.*, 2020] Benjamin Aminof, Giuseppe De Giacomo, Alessio Lomuscio, Aniello Murano, Sasha Rubin, et al. Synthesizing strategies under expected and exceptional environment behaviors. In *IJCAI*, pages 1674–1680, 2020.

[Aminof *et al.*, 2021] Benjamin Aminof, Giuseppe De Giacomo, Rubin Sasha, et al. Best-effort synthesis: Doing your best is not harder than giving up. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, 2021.

[Apt, 2011] Krzysztof R. Apt. *A Primer on Strategic Games*, page 1–37. Cambridge University Press, 2011.

[Baier and Katoen, 2008] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[Berwanger, 2007] Dietmar Berwanger. Admissibility in infinite games. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 188–199. Springer, 2007.

[Bouyer *et al.*, 2008] Patricia Bouyer, Uli Fahrenberg, Kim G Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. In *Formal Modeling and Analysis of Timed Systems: 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings 6*, pages 33–47. Springer, 2008.

[Brandenburger *et al.*, 2008] Adam Brandenburger, Amanda Friedenberg, and H Jerome Keisler. Admissibility in games 1. *Econometrica*, 76(2):307–352, 2008.

[Brenguier *et al.*, 2014] Romain Brenguier, Jean-François Raskin, and Mathieu Sassolas. The complexity of admissibility in omega-regular games. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2014.

[Brenguier *et al.*, 2015] Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-Admissible Synthesis. In *26th International Conference on Concurrency Theory (CONCUR 2015)*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 100–113, 2015.

[Brenguier *et al.*, 2016] Romain Brenguier, Guillermo A. Pérez, Jean-Francois Raskin, and Ocan Sankur. Admissibility in Quantitative Graph Games. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016)*, volume 65, pages 42:1–42:14, 2016.

[Brihaye *et al.*, 2017] Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege. Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games. *Acta Informatica*, 54:85–125, 2017.

[Chakrabarti *et al.*, 2003] Arindam Chakrabarti, Luca De Alfaro, Thomas A Henzinger, and Mariëlle Stoelinga. Resource interfaces. In *International Workshop on Embedded Software*, pages 117–133. Springer, 2003.

[Church, 1963] Alonzo Church. Application of recursive arithmetic to the problem of circuit synthesis. *Journal of Symbolic Logic*, 28(4):289–290, 1963.

[De Giacomo and Vardi, 2013] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Int. Joint Conf. on Artificial Intelligence*, IJCAI '13, page 854–860. AAAI Press, 2013.

[De Giacomo *et al.*, 2023] Giuseppe De Giacomo, Gianmarco Parretti, and Shufang Zhu. Symbolic LTL_f best-effort synthesis. In *European Conference on Multi-Agent Systems*, pages 228–243. Springer, 2023.

[Faella, 2009] Marco Faella. Admissible strategies in infinite games over graphs. In *International Symposium on Mathematical Foundations of Computer Science*, pages 307–318. Springer, 2009.

[Filiot *et al.*, 2010] Emmanuel Filiot, Tristan Le Gall, and Jean-François Raskin. Iterated regret minimization in game graphs. In *Mathematical Foundations of Computer Science 2010: 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23-27, 2010. Proceedings 35*, pages 342–354. Springer, 2010.

[Filippidis and Murray, 2016] Ioannis Filippidis and Richard M. Murray. Symbolic construction of gr(1) contracts for systems with full information. In *2016 American Control Conference (ACC)*, pages 782–789, 2016.

[Fuggitti, 2019] Francesco Fuggitti. Ltlf2dfa tool. http://ltlf2dfa.diag.uniroma1.it, 2019.

[He *et al.*, 2017] Keliang He, Morteza Lahijanian, Lydia E Kavraki, and Moshe Y Vardi. Reactive synthesis for finite tasks under resource constraints. In *Int. Conf. on Intel. Robots and Sys.*, pages 5326–5332, 2017.

[Kress-Gazit *et al.*, 2018] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):211–236, 2018.

[Kupferman and Vardi, 2001] Orna Kupferman and Moshe Y Vardi. Model checking of safety properties. *Formal Methods in System Design*, 19(3):291–314, 2001.

[Leyton-Brown and Shoham, 2008] Kevin Leyton-Brown and Yoav Shoham. *Further Solution Concepts for Normal-Form Games*, pages 15–30. Springer International Publishing, Cham, 2008.

[McMahon *et al.*, 2023] Jay McMahon, Nisar Ahmed, Morteza Lahijanian, Peter Amorese, Taralicin Deka, Karan Muvvala, Kian Shakerin, Trevor Slack, and Shohei Wakayama. Reason-recourse software for science operations of autonomous robotic landers. In *2023 IEEE Aerospace Conference*, pages 1–11, 2023.

[Muvvala and Lahijanian, 2023] Karan Muvvala and Morteza Lahijanian. Efficient symbolic approaches for quantitative reactive synthesis with finite tasks. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8666–8672, 2023.

[Muvvala *et al.*, 2022] Karan Muvvala, Peter Amorese, and Morteza Lahijanian. Let's collaborate: Regret-based reactive synthesis for robotic manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4340–4346, 2022.

[Muvvala *et al.*, 2025] Karan Muvvala, Qi Heng Ho, and Morteza Lahijanian. Beyond winning strategies: Admissible and admissible winning strategies for quantitative reachability games. *arXiv preprint arXiv:2408.13369*, 2025.

[Muvvala, 2025] Karan Muvvala. Admissible strategy synthesis toolbox. https://github.com/aria-systems-group/PDDLtoSim, 2025.

[Pnueli and Rosner, 1989] Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 179–190, 1989.

[Zhou and Foley, 2003] Hongbin Zhou and Simon N. Foley. Fast automatic synthesis of security protocols using backward search. In *Proceedings of the 2003 ACM Workshop on Formal Methods in Security Engineering*, FMSE '03, page 1–10, New York, NY, USA, 2003. Association for Computing Machinery.