Fair ω-Regular Games

Daniel Hausmann[1]*, Nir Piterman[1]*, Irmak Sağlam[2](✉)**, and Anne-Kathrin Schmuck[2]**

[1] University of Gothenburg, Gothenburg, Sweden
{hausmann,piterman}@chalmers.se
[2] Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany
{isaglam,akschmuck}@mpi-sws.org

Abstract. We consider two-player games over finite graphs in which both players are restricted by fairness constraints on their moves. Given a two player game graph $G = (V, E)$ and a set of fair moves $E_f \subseteq E$ a player is said to play *fair* in G if they choose an edge $e \in E_f$ infinitely often whenever the source node of e is visited infinitely often. Otherwise, they play *unfair*. We equip such games with two ω-regular winning conditions α and β deciding the winner of mutually fair and mutually unfair plays, respectively. Whenever one player plays fair and the other plays unfair, the fairly playing player wins the game. The resulting games are called *fair α/β games*.
We formalize fair α/β games and show that they are determined. For fair parity/parity games, i.e., fair α/β games where α and β are given each by a parity condition over G, we provide a polynomial reduction to (normal) parity games via a gadget construction inspired by the reduction of stochastic parity games to parity games. We further give a direct *symbolic fixpoint algorithm* to solve fair parity/parity games. On a conceptual level, we illustrate the translation between the gadget-based reduction and the direct symbolic algorithm which uncovers the underlying similarities of solution algorithms for fair and stochastic parity games, as well as for the recently considered class of fair games in which only one player is restricted by fair moves.

Keywords: games on graphs, fairness, two-player games, parity games

1 Introduction

Omega-regular games are a popular abstract modelling formalism for many core computational problems in the context of correct-by-construction synthesis of reactive software or hardware. This abstract view was initiated by the seminal work of Church [8] and its independent solutions by Büchi and Landweber and Rabin [18,5]. Since then these ideas have been refined and extended for solving the *reactive synthesis problems* [17,20,14].

* Supported by the ERC Consolidator grant D-SynMA (No. 772459).
** Supported by the DFG project SCHM 3541/1-1.

However, before using any such synthesis technique, the reactive software design problem at hand needs to be abstractly modelled as a two-player game. In order for the subsequently synthesized software to be 'correct-by-construction' this game graph needs to reflect all possible interactions between involved components in an abstract manner. Building such a game graph with the 'right' level of abstraction is a known severe challenge, in particular, if the synthesized software is interacting with existing components that already possess certain behavior. Here, part of the modelling challenge amounts to finding the 'right' power of both players in the resulting abstract game to ensure that winning strategies do not fail to exist due to an unnecessarily conservative overapproximation of modeling uncertainty (or the dual problem due to underapproximation).

In this context, *fairness* has been adopted as a notion to abstractly model known characteristics of the involved components in a very concise manner. *Fairness assumptions* have been used in model checking [1] and scheduler synthesis for the classical AMBA arbiter [16] or shared resource management [6]. Notably, fairness assumptions have also gained attention in cyber-physical system design [21,15,11] and robot motion planning [9,2]. In all these applications, fairness is used as an *assumption* that the synthesized (or verified) component can rely on. In particular, if these assumptions are modelled by *transition fairness* over a two-player game arena[3] $(V_\forall, V_\exists, E)$ – i.e., by a set of fair *environment* moves $E_f \subseteq E$ (i.e., with $V_\forall$ as their domain) that need to be taken infinitely often if the source node is seen infinitely often along a play – the resulting synthesis games can be solved efficiently [4,19].

While most existing work has only looked at fairness as an *assumption* to weaken the opponent in the synthesis game, all mentioned applications also naturally allow for scenarios where multiple components with intrinsically fair behavior are interacting with each other in a non-trivial manner. For example, the ability of a concurrent process to eventually free a shared resource might depend on how fair re-allocation is implemented in other threads. On an abstract level, the formal reasoning about such scenarios requires to understand how the interactive decision making of two dependent processes is influenced by intrinsic fairness constraints imposed on their decisions. Algorithmically, these synthesis questions require fairness restrictions on both players in a game, i.e., do not restrict the domain of fair moves E_f to one player only. We refer to such games simply as *fair games*.

Motivating Example. In order to better illustrate the challenges arising from solving such fair games, consider two robots in a shared workspace with narrow passages between adjacent regions that only one robot can pass at a time. One robot (say the green one) has an ω-regular objective α that specifies desired sequences of visited regions in the workspace. The other (red) robot tries to prevent the green robot from achieving this sequence. In order to rule out trivial spoiling strategies of the red robot, both robots need to implement a tie-breaking

[3] Whenever we interpret players in a one-sided manner as environment and system, we choose the environment player as the $\forall$-player, as we need to take all possible environment moves into account. Similarly, the system is the $\exists$-player in this scenario.

mechanism for obstacle avoidance, i.e., they must eventually move left or right if an obstacle blocks their way.

Now consider the scenario where both robots are facing each other at a gate, as depicted in Fig. 1. While both robots block the gate from one side, neither of them can move forward, but if the green robots moves left or the red robot moves right, the other robot can take the gate to reach region A. With the mentioned requirement for tie-breaking, none of the robots is allowed to block the gate forever and both eventually have to move to the side.

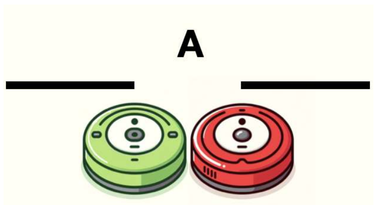

Fig. 1: Deadlock caused by fairness constraints of two robots facing a door.

Now let us assume that region A is important for both robots, hence, both robots have an incentive to enter region A first, to then move the game to an area preferable to them. However, the robot who breaks the tie first, (i.e., fulfills its fairness condition first) allows the *other* robot to enter region A first, which gives both robots the incentive to behave unfair. While it is very intuitive to make a player lose when she plays unfair and the other player plays fair, it is unclear who wins the game if both players play unfair.

To resolve this issue, we can make the objectives of the robots completely adversarial by assigning one of the players (say, green) the winner in a play where both players play unfair. In the above example, this would give the red robot the incentive to break the tie first. While this makes it harder for the red robot to spoil the objective of the green one, we might be interested in a more symmetric game which does not favor the green robot in all non-determined states of the graph. We therefore consider a second ω-regular objective β that determines the winner of (mutually) unfair plays. This results in fair games $G = (A, \alpha, \beta)$ which are determined (as shown in Sec. 3).

Contribution. Motivated by the above mentioned examples where interactive decision making of two dependent processes is influenced by intrinsic fairness constraints imposed on their decisions, this paper studies *fair games* $G = (A, \alpha, \beta)$ as their abstraction. In particular, we give solution algorithms for these games when both α and β are parity conditions induced by two different priority functions over the node set. We call such games *fair parity/parity games.*

Obviously, the previously discussed one-sided version of fair games, which we call $\forall$-*fair games* (as only the $\forall$-player (i.e., the environment) is restricted by strong transition fairness), is a special case of fair games. Both enumerative [19] and symbolic solution algorithms [4] have recently been proposed for $\forall$-

fair games, showing that strong transition fairness can be handled efficiently in both types of algorithms. This observation is closely related to a result for *stochastic games*, i.e., two-player games with an *additional* 'half' player that takes all its moves uniformly at random. For the purpose of qualitative analysis, such stochastic parity games have been shown in [7] to be reducible to (standard) parity games by the use of "gadgets" that turn stochastic nodes into a small sequence of $\forall$- and $\exists$-player nodes. While it is known that stochastic games can be reduced to $\forall$-*fair games* (and hence, fair games), it was not investigated how the different solution approaches compare. The main conceptual contribution of this paper is a unified understanding of all these solution approaches for the general class of fair games.

Concretely, our contribution is three-fold:
(1) We formalize fair games as a generalization of $\forall$-fair games and stochastic games such that they are determined.
(2) We show a reduction of fair parity/parity games to (standard) parity games, inspired by the gadget-based reduction of stochastic parity games to parity games in [7]. This reduction enables the use of parity game solvers over the reduced game (in particular enumerative ones such as Zielonka's algorithm [24]) and gives a gadget-based reduction of $\forall$-fair parity games to parity games as a corollary.
(3) We then show how our gadget construction can be used to define a *symbolic fixpoint algorithm* to solve fair parity/parity games directly (without the need for a reduction). We show the direct symbolic algorithm for $\forall$-fair parity games in [4] coinciding with our algorithm for this particular subclass of fair games.

With this, we believe that this paper uncovers the underlying similarities of solution algorithms for fair, $\forall$-fair and stochastic parity games. Further, we show how these conceptual similarities can be used to build both *enumerative* and direct *symbolic* algorithms. This is of interest as both are known to have complementary strengths, depending on how the synthesis instance is provided, and this connection was, to the best of our knowledge, not known before.

All omitted proofs are available in the extended version of this paper [10].

2 Preliminaries

We introduce infinite-duration ω-regular two-player games over finite graphs with additional *strong transition fairness conditions* on both players. For readability, we call the considered games (and their respective notions) simply *fair*.

Infinite Sequences. We denote the set of infinite sequences over a set U by U^ω. We often view sequences $\tau = u_1 u_2 \ldots \in U^\omega$ as functions $\tau : \mathbb{N} \to U$, writing $\tau(i) = u_i$. Furthermore, we let $\mathsf{Inf}(\tau) := \{u \in U \mid \forall i.\, \exists j > i.\, \tau(j) = u\}$ denote the set of elements of U that occur infinitely often in τ. Given a function $f : U \to W$, we denote by $f(\tau) \in W^\omega$ the pointwise application of f to τ. Given a natural number n, we write $[n] := \{1, \ldots, n\}$.

Fair Game Arenas. A *fair game arena* $A = (V_\exists, V_\forall, E, E_f)$ consists of a set of *nodes* $V = V_\exists \cup V_\forall$ that is partitioned into the sets of *existential nodes* $V_\exists$ and *universal nodes* $V_\forall$, together with a set $E \subseteq V \times V$ of *moves* that is partitioned into the set $E_f \subseteq E$ of *fair moves* and the set $E \setminus E_f$ of *normal moves*. If $E_f = \emptyset$, then we sometimes omit this component for brevity. Given a node $v \in V$ and a binary relation $R \subseteq V \times V$, we write $R(v)$ to denote the set $\{w \in V \mid (v, w) \in R\}$. We assume that E is right-total, that is, $E(v) \neq \emptyset$ for all $v \in V$. We call a node v *fair*, if it is the source node of a fair edge, i.e., $E_f(v) \neq \emptyset$ and collect all fair nodes in the set $V^{\mathsf{fair}} = \{v \in V \mid E_f(v) \neq \emptyset\}$ and define $V^{\mathsf{n}} = V \setminus V^{\mathsf{fair}}$ to be the set of nodes that are not fair ('normal nodes'). We denote $V_\exists^{\mathsf{fair}} = V^{\mathsf{fair}} \cap V_\exists$ and $V_\forall^{\mathsf{fair}} = V^{\mathsf{fair}} \cap V_\forall$.

Plays. A *play* $\tau = v_0 v_1 \ldots$ on A is an infinite sequence of nodes s.t. $v_{i+1} \in E(v_i)$ for all $i \geq 0$. Given a play $\tau = v_0 v_1 \ldots$, we define the associated sequence of moves $\tau_m = (v_0, v_1)(v_1, v_2) \ldots$. Additionally, if i is a player in $\{\exists, \forall\}$, we denote the other player by $1 - i$. We let $\mathsf{plays}(A)$ denote the set of all plays on A.

For a player $i \in \{\exists, \forall\}$, a play τ is *i-fair* if for all nodes $v \in V_i \cap \mathsf{Inf}(\tau)$ holds that $E_f|_v \subseteq \mathsf{Inf}(\tau_m)$, where $E_f|_v = \{(v, v') \in E_f \mid v' \in V\}$ denotes the set of fair edges that start at $v \in V$. Given a play τ, we write $\mathsf{fair}_i(\tau)$ to indicate that τ is i-fair. We call a play *mutually fair* if it is both $\exists$- and $\forall$-fair and *mutually unfair* if it is neither $\exists$- nor $\forall$-fair.

Strategies. A strategy for player $i \in \{\exists, \forall\}$ (or, an *i-strategy*) is a function $p : V^* \cdot V_i \to V$ where for each $u \cdot v \in V^* \cdot V_i$ it holds that $p(u \cdot v) \in E(v)$. A strategy p is called *positional* if $p(u \cdot v) = p(w \cdot v)$ for all $u, w \in V^*$ and $v \in V_i$.

A strategy p for player i is said to *admit* a play $\tau = v_0 v_1 \ldots$ if for all $k \in \mathbb{N}$, $v_k \in V_i$ implies $p(v_0 \ldots v_k) = v_{k+1}$. Alternatively, τ is said to be *compliant* with p. We write Σ for the set of $\exists$-strategies and Π for the set of $\forall$-strategies. Starting from a node $v \in V$, any two strategies $s \in \Sigma$ and $t \in \Pi$ induce a unique play $\mathsf{play}_v(s, t)$ in the game arena. If we do not care about the initial node of the play, we simply write $\mathsf{play}(s, t)$.

A strategy for player $i \in \{\exists, \forall\}$ is an *i-fair strategy* if every play it admits is i-fair. We write Σ^{fair} (resp. Π^{fair}) for the set of $\exists$-fair (resp. $\forall$-fair) strategies.

Omega-regular Winning Conditions. We consider winning conditions given by an ω-regular [22,13] language $\varphi \subseteq V^\omega$ over the node set V. In particular, we write $\varphi = \bot$ and $\varphi = \top$ to denote the trivial winning conditions $\emptyset$ and V^ω, respectively. In particular, we focus our attention to *parity* winning conditions. For a priority function $\lambda : V \to [k]$ that maps nodes of a game arena to the natural numbers bounded by k for some $k \in \mathbb{N}$, the Parity(λ) condition is given via $\varphi = \{\tau \in V^\omega \mid \max(\mathsf{Inf}(\lambda(\tau))) \text{ is even}\}$.

Omega-regular Games. An ω-regular game is traditionally defined via a tuple $G = (A, \alpha)$ where A is a game arena *without fair edges*, i.e. $E_f = \emptyset$ and $\alpha \subseteq V^\omega$ an ω-regular winning condition. An $\exists$-strategy $s \in \Sigma$ is said to be *winning* (for $\exists$) from a node $v \in V$, if for all $t \in \Pi$, $\mathsf{play}_v(s, t) \in \alpha$. Dually, a $\forall$-strategy $t \in \Pi$ is said to be winning (for $\forall$) from a node $v \in V$, if for all $s \in \Sigma$, $\mathsf{play}_v(s, t) \notin \alpha$. In ω-regular games, every node $v \in V$ is won by one *and only one* of the players [12,13]. This property of a game is called determinacy, and ω-regular games are *determined*. We denote the nodes from which $\exists$ (resp. $\forall$) has a winning strategy in G by $\mathsf{Win}_\exists(G)$ (resp. $\mathsf{Win}_\forall(G)$). When G is clear from the context, we drop the parenthesis and write $\mathsf{Win}_\exists$ and $\mathsf{Win}_\forall$ instead. Determinacy then amounts to $\mathsf{Win}_\exists \cup \mathsf{Win}_\forall = V$ and $\mathsf{Win}_\exists \cap \mathsf{Win}_\forall = \emptyset$.

Node Conventions for Figures. Throughout this paper, in all figures, the rectangular nodes represent $\forall$-player nodes and the nodes with round corners represent $\exists$-player nodes.

3 Fair Games

As already outlined in the motivating example in Sec. 1, the interpretation of winning conditions over fair games influences the characteristics of resulting winning strategies. To formalize this intuition, we will first recall a particular subclass of fair games, namely those where only one player is restricted by an additional fairness condition, in Subsec. 3.1. We will then use these games to motivate winning semantics for the general class of fair games.

3.1 Determinacy of $\forall$-Fair Games

A $\forall$-fair game is a tuple $G = (A, \alpha)$ where A is a game arena with $V^{\mathsf{fair}} \subseteq V_\forall$ (*called a $\forall$-fair game arena*), and α is an ω-regular winning condition.

In $\forall$-fair games, fairness constraints typically model known behavior of existing components that the $\exists$-player (i.e., the to be synthesized system) can rely on. This is formalized by defining that the $\exists$-player wins a $\forall$-fair game with winning condition α from node v if

$$\exists s \in \Sigma.\, \forall t \in \Pi^{\mathsf{fair}}.\, \mathsf{play}_v(s,t) \in \alpha. \tag{1a}$$

That is, $\exists$-player (or shortly, $\exists$) wins if they have a strategy that can win against all $\forall$-fair $\forall$-strategies.

Our intuition tells us that this can be converted to reasoning about general strategies for $\forall$-player (or shortly, $\forall$) by allowing $\exists$ to win whenever $\forall$ plays unfairly. In order to see this, we can look at the complement of Eq. (1a), i.e., the description of when $\forall$ wins; namely, $\forall s \in \Sigma.\, \exists t \in \Pi^{\mathsf{fair}}.\, \mathsf{play}_v(s,t) \notin \alpha$. We can replace the quantification over fair strategies with a quantification over all strategies but require that, in addition to refuting α, the resulting play be fair: $\forall s \in \Sigma.\, \exists t \in \Pi.\, \mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \wedge \mathsf{play}_v(s,t) \notin \alpha$. Indeed, as we show in the extended version of this paper [10, App. A - Lem. 2], if strategy $t \in \Pi$ satisfies $\mathsf{fair}_\forall(\mathsf{play}_v(s,t))$ then we can find a fair strategy $t' \in \Pi^{\mathsf{fair}}$ with which $\mathsf{play}_v(s,t)$ is compliant. This $\forall$-fair strategy would also stop s from winning. Due to determinacy of ω-regular games, we know that the last condition is equivalent to $\exists t \in \Pi.\, \forall s \in \Sigma.\, \mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \wedge \mathsf{play}_v(s,t) \notin \alpha$. In particular, this implies that t is fair. We conclude that the complement of Eq. (1) is the following equation:

$$\exists t \in \Pi^{\mathsf{fair}}.\, \forall s \in \Sigma.\, \mathsf{play}_v(s,t) \notin \alpha. \tag{1b}$$

This statement is equivalent to the determinacy of $\forall$-fair games: either $\exists$-player has a winning strategy or $\forall$-player has a winning $\forall$-fair strategy, and the two cannot be true simultaneously.

3.2 From ∀-Fair Games to Defining Determined Fair Games

Given a fair game arena A and an ω-regular objective α, a natural attempt to define winning regions in fair games would be to generalize Eq. (1) to

$$v \in \mathsf{Win}_\exists \text{ if } \exists s \in \Sigma^{\mathsf{fair}}.\, \forall t \in \Pi^{\mathsf{fair}}.\, \mathsf{play}_v(s,t) \in \alpha, \text{ and} \tag{2a}$$

$$v \in \mathsf{Win}_\forall \text{ if } \exists t \in \Pi^{\mathsf{fair}}.\, \forall s \in \Sigma^{\mathsf{fair}}.\, \mathsf{play}_v(s,t) \notin \alpha. \tag{2b}$$

However, in this case, $\mathsf{Win}_\exists \cup \mathsf{Win}_\forall \neq V$. Indeed, equations (2a) and (2b) are not complements of each other, that is,

$$\exists s \in \Sigma^{\mathsf{fair}}.\, \forall t \in \Pi^{\mathsf{fair}}.\, \mathsf{play}(s,t) \in \alpha \qquad \not\Leftrightarrow \qquad \forall t \in \Pi^{\mathsf{fair}}.\, \exists s \in \Sigma^{\mathsf{fair}}.\, \mathsf{play}(s,t) \in \alpha.$$

This observation makes a fair game in which winning regions are defined via Eq. (2) undetermined. The undetermined nodes $O \subseteq V$ – nodes from which none of the players has a fair winning strategy – form a separate partition of nodes, i.e., $V = \mathsf{Win}_\exists \uplus \mathsf{Win}_\forall \uplus O$. To see this, consider the following example.

Example 1. Consider the fair game arena depicted in Fig. 2 where fair edges are shown by dashed lines, $\alpha = \text{Parity}(\lambda)$ and each node is labeled by its priority assigned by λ. We observe that the existential player cannot enforce reaching the even node with a ∃-fair strategy from the two middle nodes. Every ∃-fair ∃-strategy s has a counter ∀-fair ∀-strategy: choose the fair edge outgoing from the square node *after* s chooses the fair edge outgoing from the node with round corners. On the other hand, the universal player cannot prevent the play from reaching the even node with a ∀-fair strategy from these nodes for exactly the same reason. Hence, the middle two nodes are neither in $\mathsf{Win}_\exists$ nor in $\mathsf{Win}_\forall$. That is, these two nodes are undetermined; therefore they form O.

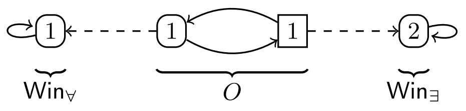

Fig. 2: A simple fair game arena discussed in Ex. 1.

In order to better understand the distinction between Equations 2a and 2b, we rely again on translation to ω-regular games. Consider the following reformulation of Eq. (2a):

$$\exists s \in \Sigma. \forall t \in \Pi. \mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \wedge (\mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \Rightarrow \mathsf{play}_v(s,t) \in \alpha). \tag{3a}$$

Similarly, the following is a reformulation of Eq. (2b):

$$\exists t \in \Pi. \forall s \in \Sigma. \mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \wedge (\mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \Rightarrow \mathsf{play}_v(s,t) \notin \alpha). \tag{3b*}$$

From determinacy of ω-regular games, the negation of the latter is:

$$\exists s \in \Sigma. \forall t \in \Pi. \mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \Rightarrow (\mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \wedge \mathsf{play}_v(s,t) \in \alpha). \tag{3b}$$

We formally prove the equivalences of Eqs. (2a) and (3a) and Eqs. (2b) and (3b) in [10]. It is not hard to see that the difference between Eq. (3a) and Eq. (3b) is in the way fairness is handled. Namely, in Eq. (3a) $\exists$ loses whenever she plays unfairly regardless of how $\forall$ plays. Dually, in Eq. (3b) $\exists$ wins immediately when $\forall$ plays unfairly regardless of how $\exists$ plays. It follows that determinacy can be regained by deciding the winner of the four different combinations of fairness with an ω-regular winning condition each, as summarized in the following table.

	$\mathsf{fair}_\exists(\tau)$	$\neg\mathsf{fair}_\exists(\tau)$
$\mathsf{fair}_\forall(\tau)$	$\tau \in \alpha$	$\tau \in \gamma$
$\neg\mathsf{fair}_\forall(\tau)$	$\tau \in \delta$	$\tau \in \beta$

With this generalization, we obtain (3a) if $\beta = \gamma = \bot$ and $\delta = \top$, and (3b) if $\gamma = \bot$ and $\beta = \delta = \top$.

We note that the discussion of determinacy has crucial importance to the analysis of games and the decision of how to model particular scenarios. For example, if fairness of $\forall$-player arises from physical constraints (as, e.g., in [4]) then it might make sense to consider Eq. (2b), which corresponds to $\beta = \top$. Dually, if fairness of $\exists$-player must be adhered to, then it makes sense to consider Eq. (2a), which corresponds to $\beta = \bot$. Our formulation allows to further fine tune what happens when both act unfairly by adjusting β.

Given the intuition that fairness constraints are actually additional obligations for both players, the choice of $\gamma = \bot$ and $\delta = \top$ assumed in Equations (2)-(3) is very natural. However, allowing mutually unfair plays to be decided by a different ω-regular winning condition β, allows games with more symmetric winning semantics e.g., by setting $\beta = \alpha$. We therefore restrict our attention in this paper to fair games with two winning conditions α and β while if i-player plays fairly but $(1-i)$-player plays unfairly, i-player wins, i.e., $\gamma := \bot$ and $\delta := \top$. This is formalized next.

Definition 1 (Fair Games). *A* fair game $G = (A, \alpha, \beta)$ *consists of a fair game arena A together with two (ω-regular) winning conditions $\alpha, \beta \subseteq \mathsf{plays}(A)$ where α and β determine the winner of mutually fair and mutually unfair plays, respectively. In fair games, a play that is i-fair and $(1-i)$-unfair is won by player i. Formally, in the fair game $G = (A, \alpha, \beta)$, $v \in \mathsf{Win}_\exists$ if and only if,*

$$\exists s \in \Sigma. \forall t \in \Pi.\, \mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \wedge (\mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \Rightarrow \mathsf{play}_v(s,t) \in \alpha) \\ \vee(\neg\mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \wedge \neg\mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \wedge \mathsf{play}_v(s,t) \in \beta) \quad (4)$$

The determinacy of fair games follows trivially from the formulation. It follows that the complement of Eq. (4) is the $\forall$ winning region, defined symmetrically by $v \in \mathsf{Win}_\forall$ if and only if

$$\exists t \in \Pi. \forall s \in \Sigma.\, \mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \wedge (\mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \Rightarrow \mathsf{play}_v(s,t) \notin \alpha) \\ \vee(\neg\mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \wedge \neg\mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \wedge \mathsf{play}_v(s,t) \notin \beta)$$

Notation. We call a fair game $G = (A, \alpha, \beta)$ a fair α/β game. Further, if α or β are given by mentioned winning conditions(e.g. $\alpha = \text{Parity}(\lambda)$, $\beta = \bot$), with slight abuse of notation, we refer to the game with the name of the objectives (e.g. fair parity/$\perp$ game).

Remark 1. Stochastic games allow for an additional set V_s of stochastic game nodes that belong to neither $\exists$ nor $\forall$, and for which the stochasticity is resolved uniformly at random. It is known that for purposes of qualitive analysis (i.e., the computation of almost-sure winning strategies), stochastic games can be seen as the special case of $\forall$-fair games in which $E(v) \subseteq E_f$ holds for all stochastic nodes $v \in V_s$, and $E_f \cap E(v) = \emptyset$ for all non-stochastic nodes $v \in V_\exists \cup V_\forall$, that is, all stochastic edges are fair edges, but no non-stochastic edges are fair edges. This encoding treats stochastic branching as adversarial for the system ($\exists$-player).

3.3 Mutually Fair Strategies in Fair Parity Games

In Subsec. 3.2 and in particular in Ex. 1 we have discussed the mutually unfair plays and strategies that take such plays into account in fair α/β games. In this section, we start restricting our attention to fair parity/β games (as this will be our focus for the rest of the paper) and discuss the particularities of mutually fair strategies in such games. We will do this with the help of the games $G_1 - G_4$ depicted in Fig. 3. No mutually unfair plays exist in any of these games. This is because on all given arenas the unfair behaviour of one player makes the play trivially fair for the other. Therefore, the winning regions are independent of β.

In game G_1, both nodes are won by $\exists$. $\forall$-player loses node 3 since taking the self loop on 3 makes the play visit 3 infinitely often, however, it forces $\forall$ to play fairly, implying that they must take the edge to 4 infinitely often. Therefore, any $\forall$-fair play is won by $\exists$ since the priority 4 is seen infinitely often. Also note that if $\forall$-player decides not to play fairly, they immediately lose since all plays are trivially $\exists$-fair. The trivial winning $\exists$-strategy is depicted by red edges.

To get to game G_2, we append node 1 to the left of G_1. Here, all the nodes are won by $\forall$. This is because $\forall$-player wins node 3 by eventually taking the outgoing edge to 1 and then staying in 1 forever with the self-loop. By doing so $\forall$ evades his obligation to take the fair edges by forcing each play to see node 3 a finite number of times. One winning $\forall$-strategy is depicted by blue edges.

To get to game G_3, we append node 5 to the right of game G_1. Again, all the nodes are won by $\forall$ even though this time he cannot evade taking his fair edges. In this game $\forall$ wins due to the obligation of $\exists$ to play fairly. In a play starting from 3, $\forall$ must eventually take the outgoing edge to 4. From there on, the play will visit node 4 infinitely often, forcing $\exists$ to take his outgoing edge to 5 infinitely often. As a consequence, in every mutually fair play 5 is seen infinitely often. Therefore, the game is won by $\forall$. A winning $\forall$-strategy is depicted by blue edges on the figure, with the interpretation that blue edges from node 3 are taken alternatingly (in every sequence).

Finally, to get to game G_4, we append two nodes to game G_3. This time, all the nodes are won by $\exists$. $\exists$-player still needs to take their fair outgoing edges to 5 (and this time, also to the new node 1) infinitely often. But this time she can also take the outgoing edge to 6 infinitely often and thereby win the game. A winning

∃-strategy is depicted by red edges on the figure, again with the interpretation that red edges from node 4 are taken alternatingly (in every sequence).

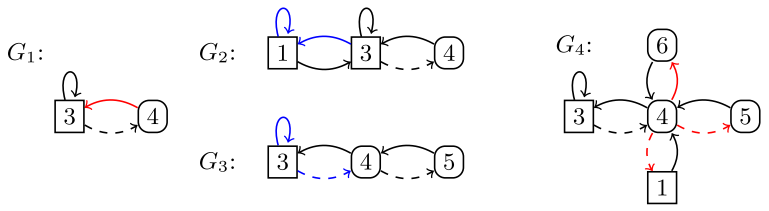

Fig. 3: Four fair parity/β games: dashed lines represent fair edges. Games G_1 and G_4 are won by ∃-player and G_2 and G_3 are won by ∀-player. In each case, a respective winning strategy is shown by colored edges. A set of colored edges represents a strategy that takes only the colored edges in the game, and whenever a source node is visited all its colored outgoing edges are taken alternatingly.

4 Reduction to Parity Games

In this section, we show how fair parity games can be reduced to parity games without fairness constraints. We show that there is a linear reduction to parity games in the case that α is a parity objective and $\beta = \top$ or $\beta = \bot$; for the case that β is a non-trivial parity objective, we show that there still is a quadratic reduction. Our reductions work by replacing each fair node in the fair game with a 3-step parity gadget. This construction is inspired by the work of Chatterjee et al. [7] where the qualitative analysis of stochastic parity games is reduced to solving parity games.

We give the formal reduction for fair parity/$\bot$ games in Subsec. 4.1 and extend it to fair parity/parity games in Subsec. 4.2. The extended version contains a discussion of the reduction for a restricted case of fair parity/$\bot$ games (fair Büchi/$\bot$ games), which can serve as a hand-holding introduction to the section.

4.1 Reduction of Fair Parity/$\bot$ Games

Let $G = (A, \text{Parity}(\lambda), \bot)$ where $A = (V_\exists, V_\forall, E, E_f)$ is a fair game arena, $V = V_\exists \cup V_\forall$ and $\lambda : V \to [2k]$ is the priority function.

The reduction to parity games replaces fair nodes $v \in V^{\mathsf{fair}}$ in G with the gadgets given in Fig. 4. Nodes $v \in V_\exists^{\mathsf{fair}}$ in G are replaced with one of the gadgets on the top (i.e. the incoming edges to v are redirected to v in the root, and the outgoing edges on the third level lead to $E(v)$ and $E_f(v)$, which are the outgoing edges and outgoing fair edges of v in G, resp.) and nodes $v \in V_\forall^{\mathsf{fair}}$ in G are replaced with one of the gadgets at the bottom. The gadgets on the left are called *existential* gadgets and the ones on the right are called *universal* gadgets, referring to the player picking the first move. Nodes in V^{n} are not altered.

Even though the proof works for all combinations of the gadgets (i.e. one can replace each $v \in V_\exists^{\mathsf{fair}}$ ($v \in V_\forall^{\mathsf{fair}}$) with any of the gadgets on the top (bottom)), due to space constraints we give the intuition only for the existential gadgets.

Imagine all $v \in V^{\mathsf{fair}}$ are replaced with their existential gadgets. Within a subgame that starts at a fair node $v \in V^{\mathsf{fair}}$, the two players intuitively interact as follows. The $\exists$-player gets to pick a number i, indicating the priorities $(2i-1$ or $2i)$ they intend to visit infinitely often in any play that visits v infinitely often. In turn, $\forall$-player gets to either pick an outgoing edge at v (for this, he pays the price of seeing the even priority $2i$), or allow $\exists$ to pick an outgoing edge (in which case he is rewarded with a visit to the odd priority $2i-1$). Depending on the owner of v, the edge picked by $\forall$ (if $v \in V_{\exists}^{\mathsf{fair}}$), or the edge picked by $\exists$ (if $v \in V_{\forall}^{\mathsf{fair}}$) is required to be contained in E_f. Thus $\forall$ can insist on exploring fair edges at $V_{\exists}^{\mathsf{fair}}$ nodes, but has to pay a price for it; dually, $\forall$ eventually has to allow $\exists$ to explore the fair edges at $V_{\forall}^{\mathsf{fair}}$ nodes to win.

In the full reduced game defined formally in the proof of Thm. 1 below, the owner of a fair node v can *fairly* win from v by either avoiding v from some point on forever, or eventually allowing the opponent player to explore all fair edges leading out of that node. The owner wins by playing *unfairly* if and only if the opponent also plays unfairly and the owner is the $\forall$-player.

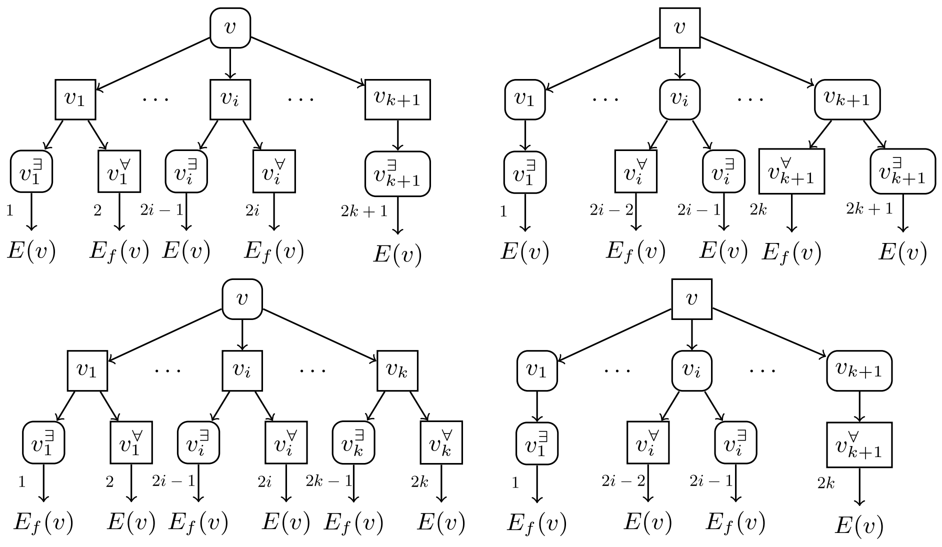

Fig. 4: Existential (left) and universal (right) gadgets for $v \in V_{\exists}^{\mathsf{fair}}$ (top) and $v \in V_{\forall}^{\mathsf{fair}}$ (bottom) in fair parity/$\perp$ games. For $i \in [1, k+1]$, priorities of nodes $v_i^{\exists}$ and $v_i^{\forall}$ are given below them, priorities of nodes v_i are ignored, and the priority of v is unaltered.

Theorem 1. *Let $G = (A, Parity(\lambda), \perp)$ where $A = (V_{\exists}, V_{\forall}, E, E_f)$ is a fair game arena, $V = V_{\exists} \uplus V_{\forall}$ and $\lambda : V \to [2k]$ is the priority function. Then there exists a parity game G' on the node set V' with $V \subseteq V'$ and $|V'| \leq n(3k+3)$ over $2k+1$ priorities such that for $i \in \{\exists, \forall\}$, $\mathsf{Win}_i(G) = \mathsf{Win}_i(G') \cap V$.*

Proof (Sketch). Let $G' = (V'_{\exists}, V'_{\forall}, E', \Omega : V' \to [2k+1])$ be the parity game obtained by replacing the fair nodes in G with an arbitrary combination of their corresponding existential and universal gadgets in Fig. 4. Let $V' = V'_{\exists} \cup$

$V'_\forall = V \cup V^{\mathsf{gad}}$ where V represent the nodes coming from G and V^{gad} represent the nodes coming from the gadgets. Note that the maximum priority in G' is $\max_{\mathsf{odd}} = 2k+1$ which comes only from the gadget nodes V^{gad}. The maximum even priority in G' is $\max_{\mathsf{even}} = 2k$ which can come from both V^{gad} and V. It is easy to see that $|V'| \leq n(3k+3)$ and G' uses priorities $[2k+1]$. To prove the correctness, we recall that the winning regions for fair parity/$\bot$ games are given via Eq. (3a), i.e. $v \in \mathsf{Win}_\exists(G)$ if and only if

$$\exists s \in \Sigma . \forall t \in \Pi . \mathsf{fair}_\exists(\mathsf{play}_v(s,t)) \wedge (\mathsf{fair}_\forall(\mathsf{play}_v(s,t)) \Rightarrow \mathsf{play}_v(s,t) \in \alpha). \quad \text{(3a)}$$

($\Rightarrow$) We will first show, $v \in \mathsf{Win}_\exists(G') \cap V \Rightarrow v \in \mathsf{Win}_\exists(G)$. To do so, we will take a (positional) winning $\exists$-strategy s' in G' and construct an $\exists$-strategy s in G such that s is $\exists$-winning in G i.e., s realizes Eq. (3a). That is, for a play ρ in G that starts from v and compliant with s Eq. (3a-s) holds.

$$\mathsf{fair}_\exists(\rho) \wedge (\mathsf{fair}_\forall(\rho) \Rightarrow \rho \in \alpha) \quad \text{(3a-s)}$$

For this we will show the two parts of the conjunction separately. We will show (i) $\mathsf{fair}_\exists(\rho)$, i.e. $s \in \Sigma^{\mathsf{fair}}$, (ii) $\mathsf{fair}_\forall(\rho) \Rightarrow \rho \in \alpha$, i.e. every $\forall$-fair play compliant with s is $\exists$-winning w.r.t. the parity condition.

Construction of the s'-subgame $G'_{s'}$: Let s' be a positional $\exists$-strategy winning every play from v in G'. We will denote the subgame of G' where $\exists$ nodes have only the outgoing edges $u \to s'(u)$ by $G'_{s'}$, and call it *the s'-subgame.* Recall that all plays that start from v in $G'_{s'}$ are $\exists$-winning.

Notation of n_u and $succ(u)$: For the existential gadgets for both $V_\exists^{\mathsf{fair}}$ and $V_\forall^{\mathsf{fair}}$, we call the index of the unique successor of u in $G'_{s'}$, n_u. That is, $s'(u) = u_{n_u}$. For the same gadgets, we will denote $s'(u^\exists_{n_u})$ with $succ(u)$. For the universal gadgets for both $V_\exists^{\mathsf{fair}}$ and $V_\forall^{\mathsf{fair}}$, we will let n_u denote the index of the rightmost child of u that is sent to its right child by s'. That is, n_u is the largest index i such that $s'(u_i) = u_i^\exists$. For the same gadgets, we will denote $s'(u^\exists_{n_u})$ with $succ(u)$.

Construction of s: We define $s : V^* \cdot V_\exists \to V$ as follows. For $u \in V_\exists^{\mathsf{fair}}$: 1. If $n_u = k+1$, we set $s(u) = succ(u)$. 2. Otherwise, $s(u)$ cycles through the set $\{succ(u), E_f(u)\}$ starting from $succ(u)$. For $u \in V_\exists \setminus V_\exists^{\mathsf{fair}}$, we set $s(u) = s'(u)$.

Constraining $G'_{s'}$ with n_u: Here we will constrain $G'_{s'}$ to its subgame by limiting the choices of $\forall$-player from a u replaced by the universal gadget. For every universal gadget encountered in $G'_{s'}$, we limit the choices of $u \in V_\forall^{\mathsf{fair}}$ to only $u \to u_{n_u}$ and $u \to u_{n_u+1}$ (if it exists). So, we remove all the other branches of u out of $G'_{s'}$. We call the remaining game $LG'_{s'}$, standing for *limited* $G'_{s'}$. Note that as $LG'_{s'}$ is a subgame of $G'_{s'}$, it is still $\exists$-winning.

$\exists$-extension: Let ρ be some play in G compliant with s. We define a play ρ' that is called *the $\exists$-extension of* $\rho = u^1 u^2 \dots$ as follows: ρ' is the play on $LG'_{s'}$ that follows ρ while 'prioritising existential nodes on $succ(u)$'. What is meant by this is, for a $u^i \in V^{\mathsf{fair}}$, if $u^{i+1} = succ(u^i)$, then ρ' takes the unique branch in $LG'_{s'}$ that leads to u^{i+1} while passing through an existential node $(u^i)^\exists_j$. That is, regardless of which gadget u^i is replaced by, ρ' takes the branch

$$u^i \to u^i_{n_{u^i}} \to (u^i)^\exists_{n_{u^i}} \to succ(u^i) = u^{i+1} \quad \text{(branch 1)}$$

On the other hand if $u^{i+1} \neq succ(u^i)$, then ρ' takes the only other branch in $LG'_{s'}$, that is (branch 2) is taken as

1. If $u^i \in V^{\mathsf{fair}}$ is replaced by an $\exists$-gadget, then $u^i \to u^i_{n_{u^i}} \to (u^i)^{\forall}_{n_{u^i}} \to u^{i+1}$,
2. If $u^i \in V^{\mathsf{fair}}$ is replaced by a $\forall$-gadget, then $u^i \to u^i_{n_{u^i}+1} \to (u^i)^{\forall}_{n_{u^i}+1} \to u^{i+1}$,

Note that these branches do not leave out any possible transition in ρ. That's because 1. all the successors of a $V^{\mathsf{fair}}_{\forall}$ node are covered by one of the branches since (branch 2) leads the universal node $(u^i)^{\forall}_{n_{u^i}}$ or $(u^i)^{\forall}_{n_{u^i}+1}$, which can pick any successor of u^i. 2. all the successors of a $V^{\mathsf{fair}}_{\exists}$ node are covered by one of the branches, since by construction of s, all the successors of u^i in ρ are in the set $\{succ(u^i)\} \cup E_f(u^i)$, where (branch 1) covers the $succ(u^i)$ successors, and (branch 2) covers the $E_f(u^i)$ successors since in this case the universal node $(u^i)^{\forall}_{n_{u^i}}$ or $(u^i)^{\forall}_{n_{u^i}+1}$ can pick any fair successor of u^i.

For $u^i \neq V^{\mathsf{fair}}$, ρ' just takes $u^i \to u^{i+1}$.

So ρ' is well defined, and is a play in $LG'_{s'}$ that starts from v. Thus, ρ' is $\exists$-winning. Observe that if we remove the gadget nodes from ρ', we get ρ. That is, the restriction of ρ' to V, $\rho' \mid_V = \rho$.

(i) $\mathsf{fair}_{\exists}(\rho)$: Observe that for any ρ in G compliant with s, by construction of s, the only nodes $u \in V^{\mathsf{fair}}_{\exists}$ that ρ may not be fair on, are those for which $n_u = k+1$. So we only need to show that such nodes are seen only finitely often in ρ. Since $\rho|_V = \rho'$, that is equivalent to showing such a u cannot be seen infinitely often in its $\exists$-extension, ρ'. If it is seen infinitely often in ρ', then regardless of the gadget u is replaced with, the branch $u \to u_{k+1} \to u^{\exists}_{k+1}$ is evoked infinitely often, signalling the largest priority $2k+1$. Therefore, ρ' is won by $\forall$-player, giving a contradiction. Therefore, we conclude ρ is $\exists$-fair.

(ii) $\mathsf{fair}_{\forall}(\rho) \Rightarrow \rho \in \alpha$: Let ρ be $\forall$-fair. Look at the $\exists$-extension ρ' of ρ. Let m be the largest (even) priority in $\mathsf{Inf}(\rho')$. Due to $\rho' \mid_V = \rho$, all we need to show is the existence of a $u \in \mathsf{Inf}(\rho' \mid_V)$ that has priority m. Then it automatically implies that the maximum priority in $\mathsf{Inf}(\rho)$ is m, and thus ρ is $\exists$-winning.

We will proceed with proof by contradiction and assume that the priority m appears only in $V^{\mathsf{gad}} \cap \mathsf{Inf}(\rho')$. Now let F be the subgame of $LG'_{s'}$ that consists of nodes and edges taken infinitely often in ρ'. Then, priority m appears in $V^{\mathsf{gad}} \cap F$. These gadget nodes must exist in F due to nodes

– $u \in V^{\mathsf{fair}}$ replaced by existential gadgets, and with $n_u = m\backslash 2$ (which corresponds to (branch 2)-1), or
– $u \in V^{\mathsf{fair}}$ replaced by universal gadgets, and with $n_u = m\backslash 2 - 1$ (which corresponds to (branch 2)-2)

Note that for all such nodes u, (branch 1) of u is also in F. This is because $u \to succ(u)$ is taken infinitely often in ρ. For $u \in V^{\mathsf{fair}}_{\exists}$, this is due to the construction of s, for $u \in V^{\mathsf{fair}}_{\forall}$, this is due to ρ being $\forall$-fair (remember, in this case $succ(u) \in E_f(u)$).

Next, we will remove from F all priority m gadget nodes (and everything reachable only from those nodes). That is, we will prune out (branch 2) of all the nodes that bring in m priority gadget nodes to F. Due to the remaining (branch 1)s, this pruning does not cause any dead-ends. Let's call this pruned subgame of F, H. Observe once more that all plays in H are $\exists$-winning. However, the maximum priority in H is $m-1$. This is due to the remaining (branch 1)s of the pruned nodes having this priority. This implies that all infinite plays starting in H get trapped in a subgame H' of H that doesn't have nodes with priority $m-1$. Since non of the nodes in $V^{\mathsf{fair}} \cap H'$ cause a gadget node with priority m, non of their branches get pruned. That is, all nodes in H' have the same outgoing edges in H' and in F. Then any play that start in H' in F, does not leave H', making H' exactly the set of nodes seen infinitely often in ρ', i.e. $H' = F$. This contradicts our initial assumption that maximum priority seen infinitely often in ρ' being m; therefore proving ρ is $\exists$-winning.

The proof of direction ($\Leftarrow$) is similar to the proof of ($\Rightarrow$), and can be found in detail in the extended version [10].

Remark 2 (Reduction of parity/$\top$ games). In the gadgets from Fig. 4, in order to play unfairly from a $v \in V_{\exists}^{\mathsf{fair}}$, $\exists$-player has to take its rightmost branch and signal priority $\max_{\mathsf{odd}}$, whereas to play unfairly from $v \in V_{\forall}^{\mathsf{fair}}$, $\forall$-player has to take the rightmost branch and signal $\max_{\mathsf{even}}$. Since $\max_{\mathsf{odd}} > \max_{\mathsf{even}}$, this dynamic ensures mutually unfair plays are $\forall$-winning. The gadgets for a fair parity/$\top$ game with $\lambda : V \to [2k]$ can be constructed as follows with the addition of priority $2k+2$: Take the gadgets from Fig. 4. In the existential gadget for $V_{\exists}^{\mathsf{fair}}$ add another branch $\to v_{k+1}^{\forall} \to E_f(v)$ to v_{k+1} and in the universal gadget for $V_{\exists}^{\mathsf{fair}}$ add a rightmost branch $\to v_{k+2} \to v_{k+2}^{\forall} \to E_f(v)$. In the existential gadget for $V_{\forall}^{\mathsf{fair}}$ add a rightmost branch $\to v_{k+1} \to v_{k+1}^{\exists} \to E_f(v)$ and in the universal gadget for $V_{\forall}^{\mathsf{fair}}$ add another branch $\to v_{k+1}^{\exists} \to E_f(v)$ to v_{k+1}.

All the newly added gadget nodes have priority $2k+2$ and therefore $\max_{\mathsf{even}} = 2k+2 > \max_{\mathsf{odd}} = 2k+1$, which ensures that mutually unfair plays are $\exists$-winning. The correctness of the construction follows as a corollary of the reduction of fair parity/parity games given in the next section.

4.2 Reduction of Fair Parity/Parity Games

In this section, we present a quadratic reduction from fair parity/parity to parity games. So let $G = (A, \mathrm{Parity}(\lambda), \mathrm{Parity}(\Gamma))$ where $A = (V_{\exists}, V_{\forall}, E, E_f)$ is a fair game arena with $V = V_{\exists} \uplus V_{\forall}$ and priority functions $\lambda : V \to [2k]$, $\Gamma : V \to [d]$.

The reduction is based on ideas from the previous section, in particular adapting the basic structure of the introduced gadgets. However, in order to correctly treat mutually unfair plays according to the additional parity objective Γ, we annotate game nodes $v \in V$ with two memory values $p \in [d]$ and $b \in \{\exists, \forall\}$. The former is used to store the maximal priority according to Γ that the play has *recently* seen; this value is signalled (and reset after signalling) from time to time in the reduced game. The value b is used to decide (at certain nodes) whether the memory value is signalled, or not.

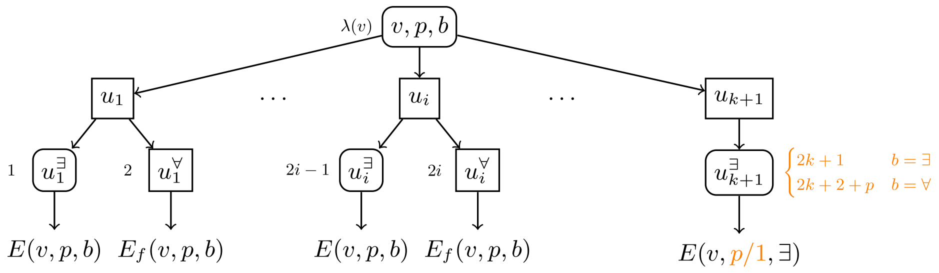

Fig. 5: Gadget for $v \in V_\exists^{\mathsf{fair}}$ in fair parity/parity games; u abbreviates (v, p, b).

It indicates the player that has last taken the rightmost branch in the gadget for one of its fair nodes. If this bit keeps flipping between $\exists$ and $\forall$ forever, then both players intuitively insist on keeping control in one of their respective fair nodes, enabling a mutually unfair play; in the reduced game, the memory content p is signalled (and then reset to 1) whenever the value flips from $\forall$ to $\exists$.

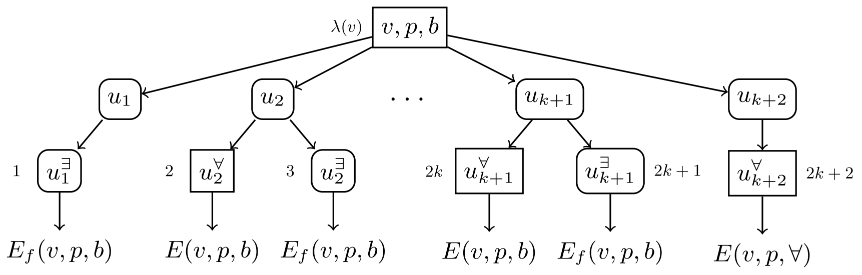

Fig. 6: Gadget for $v \in V_\forall^{\mathsf{fair}}$ in fair parity/parity games; u abbreviates (v, p, b).

Formally, the reduction is as follows. The game is based on the set $V \times [d] \times [1]$ of base nodes, where we use $[1]$ to denote $\{\exists, \forall\}$; intuitively, a node (v, p, b) from this set corresponds to $v \in V$, annotated with memory values p and b as described above. In order to succinctly refer to the combination of taking a move in G and updating the memory components, we overload notation and put

$$E(v,p,b) = \{(w,p',b) \in V \times [d] \times [1] \mid w \in E(v) \text{ and } p' = \max(p, \Gamma(v))\}$$
$$E_f(v,p,b) = \{(w,p',b) \in V \times [d] \times [1] \mid w \in E_f(v) \text{ and } p' = \max(p, \Gamma(v))\}$$

for $(v, p, b) \in V \times [d] \times [1]$. Thus a triple (w, p', b) is contained in $E(v, p, b)$ if there is a move $(v, w) \in E$ and p' is the maximum of the previous memory value p and the current priority $\Gamma(v)$ at v; in $E_f(v, p, b)$, we require $(v, w) \in E_f$ instead.

In both functions, the argument b is used to explicitly set this component of the memory to either $\exists$ or $\forall$. The reduced game consists of subgames that start at annotated nodes $u = (v, p, b) \in V \times [d] \times [1]$. In case that $v \in V^{\mathsf{n}}$, the game just proceeds according to $E(v, p, b)$, with ownership of (v, p, b) determined by whether $v \in V_\exists$ or $v \in V_\forall$; this corresponds to taking a move at a normal node in G, but updating the memory component p, and keeping the component b without modifying it. For fair nodes $v \in V^{\mathsf{fair}}$, the subgame consists of three levels, and after these three steps leads back to a node from $V \times [d] \times [1]$. Fig. 5 and 6 show the subgames that start at $(v, b, p) \in V \times [d] \times [1]$ such that $v \in V_\exists^{\mathsf{fair}}$ and $v \in V_\forall^{\mathsf{fair}}$, respectively, adapting the existential gadget for $v \in V_\exists^{\mathsf{fair}}$ and the universal one for $v \in V_\forall^{\mathsf{fair}}$.

The rightmost branches in these gadgets overwrite the last component b with $\exists$ and $\forall$, respectively. The colored values in the right-most branch in the Fig. 5 gadget depend on the value of b. If $b = \forall$ (corresponding to $\forall$-player being the one that last has taken the right-most branch), then the priority $2k + 2 + p$ is signalled and the memory value p is reset to 1; if $b = \exists$ (corresponding to $\exists$-player having taken the right-most branch last), then the priority $2k + 1$ is signalled and the memory value p does not change.

Theorem 2. *Let $G = (A, Parity(\lambda), Parity(\Gamma))$ where $A = (V_\exists, V_\forall, E, E_f)$ is a fair game arena, $V = V_\exists \uplus V_\forall$ and $\lambda : V \to [2k]$ and $\Gamma : V \to [d]$ are priority functions. Then there exists a parity game G' with $6nd(k+2)$ nodes and $2k+2+d$ priorities with set $V \times [d] \times [1]$ of base nodes such that for all $v \in V$, $\exists$-player wins v in G if and only if $\exists$-player wins $(v, 1, \exists)$ in G'.*

Proof (Sketch). We construct the parity game G' following the above description, using the gadgets from Fig. 5 and 6 to treat fair nodes. The detailed construction and the correctness proof can be found in the extended version [10].

We obtain the following bound on strategy sizes for fair parity/parity games.

Lemma 1. *Let G be a fair parity/parity game on n nodes. Then for both players the memory requirement of winning strategies in G is at most $n^2 \cdot n^n$. Furthermore, for each player a family of fair parity/$\perp$ games $(G_n)_{n \in \mathbb{N}}$ exists such that for all n, every winning strategy for the respective player requires memory at least 2^n.*

Proof (Sketch). For the upper bound, we note that in a winning i-strategy for a fair parity/parity game, as constructed in the proof of Thm. 2, the nodes in $V_i \setminus V^{\mathsf{fair}}$ have strategies with quadratic memory, but the nodes in V_i^{fair} may have to traverse all their fair successors, and possibly one more successor. In the worst case, this requires an additional local memory of $|E_f(v)| + 1 \leq n$ for each $v \in V_i^{\mathsf{fair}}$, and causes an exponential blowup in the overall memory required.

For the lower bound, we consider the case for $\exists$-player; the result for $\forall$-player is obtained by switching the player's roles. Define the family $(G_n)_{n \in \mathbb{N}}$ of games by letting G_n (for $n \in \mathbb{N}$) have exactly $n+1$ nodes, one node x owned by $\forall$-player and n nodes y_i owned by $\exists$-player; let there be an edge from x to any node y_i and two *fair* edges from any node y_i back to x. Let all nodes have priority 0. Then any winning $\exists$-strategy in G_n necessarily is $\exists$-fair. There is a fair $\exists$-strategy s that uses one bit as local memory for each node $y_i \in V_\exists^{\mathsf{fair}}$, and therefore uses memory of overall size 2^n. The claim follows since there is no $\exists$-fair strategy that uses less memory than s, which is shown by induction on n. $\square$

5 Fixpoint Characterization of Winning Regions

In this section, we will characterize the winning regions in fair games with parity conditions by means of fixpoint expressions. Thereby we provide an alternative, symbolic route to solve such games, rather than by reducing to parity games. We start by briefly recalling details on Boolean fixpoint expressions.

Fixpoint expressions and fixpoint games. Let U be a finite set, let o be a fixed number and let $f : \mathcal{P}(U)^o \to \mathcal{P}(U)$ be a monotone function, that is, assume that whenever we have sets $X_j, Y_j \subseteq U$ such that $X_j \subseteq Y_j$ for all $1 \leq j \leq o$, then $f(X_1, \dots, X_o) \subseteq f(Y_1, \dots, Y_o)$. Then f and o induce the *fixpoint expression*

$$e = \eta_o X_o.\, \eta_{o-1} X_{o-1}. \dots .\nu X_2.\, \mu X_1.\, f(X_1, \dots, X_o) \tag{5}$$

where $\eta_i = \nu$ if i is even and $\eta_i = \mu$ if i is odd. We define the semantics of fixpoint expressions using parity games. Given a fixpoint expression e, the associated *fixpoint game* $G_e = (W_\exists, W_\forall, E, \text{Parity}(\kappa))$ for the priority function $\kappa : W_\exists \cup W_\forall \to [o]$ is the following parity game. We put $W_\exists = U \times \{1, \dots, o\}$, $W_\forall = \mathcal{P}(U)^o$. Moves and priorities are defined by

$$E(v,i) = \{\overline{Z} \in W_\forall \mid v \in f(\overline{Z})\} \qquad \kappa(v,i) = i$$
$$E(\overline{Z}) = \{(v,i) \mid v \in Z_i\} \qquad \kappa(\overline{Z}) = 0$$

for $(v,i) \in W_\exists$ and $\overline{Z} = (Z_1, \dots, Z_o) \in W_\forall$. Then we say that $v \in U$ is *contained* in e (denoted $v \in e$) if and only if $\exists$-player wins the node $(v,1)$ in G_e.

Remark 3. The above game semantics for fixpoint expressions has been shown to be equivalent to the more traditional Knaster-Tarski semantics [3]; the cited work takes place in a more general setting and therefore uses slightly more verbose parity games.

Next we present a fixpoint characterization of the winning regions in fair games of the form $G = (A, \text{Parity}(\lambda), \bot)$ where $A = (V_\exists, V_\forall, E, E_f)$ is a fair game arena, $V = V_\exists \uplus V_\forall$ and $\lambda : V \to [2k]$ a priority function. To be able to write fixpoint expressions over such games we define monotone operators on subsets of V by putting

$$\Diamond X = \{v \in V \mid E(v) \cap X \neq \emptyset\} \qquad \Box X = \{v \in V \mid E(v) \subseteq X\}$$
$$\Diamond_f X = \{v \in V \mid E_f(v) \cap X \neq \emptyset\} \qquad \Box_f X = \{v \in V \mid E_f(v) \subseteq X\}$$

for $X \subseteq V$ and also put $\mathsf{Cpre}(X) = (V_\exists \cap \Diamond X) \cup (V_\forall \cap \Box X)$. Then $\mathsf{Cpre}(X)$ is the set of nodes from which $\exists$-player can force the game to reach a node from X in one step. Also, we define $C_i = \{v \in V \mid \lambda(v) = i\}$ for $1 \leq i \leq 2k$.

Using this notation, we define a function $\mathsf{parity} : \mathcal{P}(V)^{2k} \to \mathcal{P}(V)$ by putting

$$\mathsf{parity}(X_1, \dots, X_{2k}) := (C_1 \cap \mathsf{Cpre}(X_1)) \cup \dots \cup (C_k \cap \mathsf{Cpre}(X_{2k}))$$

for $(X_1, \dots, X_{2k}) \subseteq \mathcal{P}(V)^{2k}$. This function is monotone and it is well-known (see e.g [23]) that the fixpoint induced by parity characterizes the winning region in parity games with priorities 1 through $2k$. This formula will still apply to 'normal' nodes V^{n} in the fixpoint characterization of fair parity games.

We follow the gadget constructions from Fig. 4 (using their *existential* versions) to define the following additional functions. For $1 \leq i < k$, put

$$\mathsf{Apre}_{\exists}(X_i, X_{i+1}) = \Diamond X_i \cap \Box_f X_{i+1} \qquad \mathsf{Apre}_{\forall}(X_i, X_{i+1}) = \Diamond_f X_i \cap \Box X_{i+1},$$

encoding nodes $(v^{\forall}, 2i)$ for $v \in V_{\exists}^{\mathsf{fair}}$ and $v \in V_{\forall}^{\mathsf{fair}}$, respectively (here, Apre stands for *alternative* predecessor function, as it encodes the additional $\forall$-choice of whether a fair edge is to be taken). Then, we let $I_p = \{i \mid i \text{ odd}, p \leq i < 2k\}$ denote the set of odd priorities that lie between p and $2k$, and put

$$\phi^{\mathsf{fair}}_{\exists,p} = \begin{cases} \bigcup_{i \in I_p} \mathsf{Apre}_{\exists}(X_i, X_{i+1}) \cup \Diamond X_{2k+1} & p \text{ is odd} \\ \bigcup_{i \in I_p} \mathsf{Apre}_{\exists}(X_i, X_{i+1}) \cup \Diamond X_{2k+1} \cup \Box_f X_p & p \text{ is even,} \end{cases}$$

$$\phi^{\mathsf{fair}}_{\forall,p} = \begin{cases} \bigcup_{i \in I_p} \mathsf{Apre}_{\forall}(X_i, X_{i+1}) & p \text{ is odd} \\ \bigcup_{i \in I_p} \mathsf{Apre}_{\forall}(X_i, X_{i+1}) \cup \Box X_p & p \text{ is even} \end{cases}$$

Using this notation, the winning region for the existential player in fair parity/$\bot$ games with priorities 1 through $2k$ can be characterized by the fixpoint expression induced by $2k+1$ and the function χ that is defined to map $(X_1, \dots, X_{2k+1}) \in \mathcal{P}(V)^{2k+1} \to \mathcal{P}(V)$ to the set

$$\begin{aligned} \chi(X_1, \dots, X_{2k+1}) = & (V^{\mathsf{n}} \cap \mathsf{parity}) \cup \\ & (V_{\exists}^{\mathsf{fair}} \cap \bigcup_{i \in [2k+1]} C_i \cap \phi^{\mathsf{fair}}_{\exists,i}) \cup \\ & (V_{\forall}^{\mathsf{fair}} \cap \bigcup_{i \in [2k+1]} C_i \cap \phi^{\mathsf{fair}}_{\forall,i}) \end{aligned}$$

The function χ therefore treats normal nodes from V^{n} in the same way as nodes in standard parity games are treated, but for fair nodes with priority i, the functions $\phi^{\mathsf{fair}}_{\exists,i}$ and $\phi^{\mathsf{fair}}_{\forall,i}$ are used to encode the respective gadget construction. The full fixpoint expression then is

$$e = \mu X_{2k+1}.\, \nu X_{2k}.\, \mu X_{2k-1} \, \dots \, \nu X_2.\, \mu X_1.\, \chi(X_1, \dots, X_{2k+1}) \tag{6}$$

The first result of this section is that the fixpoint expression (6) characterizes the winning region of $\exists$-player in fair parity/$\bot$ games.

Theorem 3. *Let $G = (A, Parity(\lambda), \bot)$ where $A = (V_{\exists}, V_{\forall}, E, E_f)$ is a fair game arena, $V = V_{\exists} \uplus V_{\forall}$ and $\lambda : V \to [2k]$ is the priority function. Then the fixpoint expression given in* (6) *characterizes* $\mathsf{Win}_{\exists}(G)$.

Proof (Sketch). The proof is by mutual transformation of winning strategies in G and in the semantic game G_e for (6). The full proof can be found in [10].

We note that for $\forall$-fair parity games ($V_\exists^{\mathsf{fair}} = \emptyset$), Eq. (6) instantiates to the fixpoint characterization given in [4]; it follows that the parity game reductions from Sec. 4 apply to the one-sided fair parity games considered in [4] as well.

For fair parity/parity games, we obtain a similar fixpoint characterization, encoding the reduction to parity games presented in Subsec. 4.2 along the lines of Figures 5 and 6. Here, all involved functions work over (subsets of) the set $V \times [d] \times [1]$ of base nodes, consisting of game nodes that are annotated with memory values. The definition of the fixpoint expression for fair parity/parity games is straight-forward but somewhat technical since the updating and resetting mechanisms for the memory values have to be accommodated. For brevity, we refrain from elaborating the required notation and the full fixpoint expression here, and state just the main result that yields a symbolic fixpoint algorithm for fair parity/parity games; full details can be found in the extended version [10].

Theorem 4. *Let $G = (A, \mathit{Parity}(\lambda), \mathit{Parity}(\Gamma))$ where $A = (V_\exists, V_\forall, E, E_f)$ is a fair game arena, $V = V_\exists \uplus V_\forall$ and $\lambda : V \to [2k]$, $\Gamma : V \to [d]$ are priority functions. Then there is a fixpoint expression over $V \times [d] \times [1]$ with alternation depth $2(k+1)+d$ that characterizes $\mathsf{Win}_\exists(G)$.*

Proof (Sketch). Again the proof is by mutual transformation of winning strategies in G and in the semantic game G_e for the fixpoint expression. The full proof can be found in the extended version [10].

6 Conclusion

We introduce two-player games with local transition-fairness constraints for both players, allowing two objectives α and β to decide the winner of plays in which both players play fair and both players play unfair, respectively. We show the determinacy of this class of games in the case that α and β are ω-regular objectives. In the special case that both α and β are parity conditions, there is a reduction to standard parity games with blow-up quadratic in the number of priorities used by α and β; if $\beta = \top$ or $\beta = \bot$, the reduction becomes even linear. We present both enumerative and symbolic methods to realize this reduction; in the process, we also obtain an exponential tight bound on the memory required by winning strategies in fair parity/parity games. We expect that the central idea behind the reduction generalizes from parity objectives to more general settings such as fair games in which α and β are Rabin, Streett, or even Emerson-Lei conditions, but leave this issue for future work.

Acknowledgments

We are grateful to an anonymous referee for contributing the proof of the lower bound in Lemma 1.

References

1. Aminof, B., Ball, T., Kupferman, O.: Reasoning about systems with transition fairness. In: Baader, F., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning, 11th International Conference, LPAR 2004, Montevideo, Uruguay, March 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3452, pp. 194–208. Springer (2004). https://doi.org/10.1007/978-3-540-32275-7_14
2. Aminof, B., Giacomo, G.D., Rubin, S.: Stochastic fairness and language-theoretic fairness in planning in nondeterministic domains. In: Beck, J.C., Buffet, O., Hoffmann, J., Karpas, E., Sohrabi, S. (eds.) Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020. pp. 20–28. AAAI Press (2020), https://ojs.aaai.org/index.php/ICAPS/article/view/6641
3. Baldan, P., König, B., Mika-Michalski, C., Padoan, T.: Fixpoint games on continuous lattices. Proc. ACM Program. Lang. **3**(POPL), 26:1–26:29 (2019). https://doi.org/10.1145/3290339
4. Banerjee, T., Majumdar, R., Mallik, K., Schmuck, A., Soudjani, S.: Fast symbolic algorithms for omega-regular games under strong transition fairness. TheoretiCS **2** (2023). https://doi.org/10.46298/theoretics.23.4
5. Büchi, J., Landweber, L.: Solving sequential conditions by finite-state strategies. Trans. Amer. Math. Soc. **138**, 295–311 (1969)
6. Chatterjee, K., de Alfaro, L., Faella, M., Majumdar, R., Raman, V.: Code aware resource management. Formal Methods Syst. Des. **42**(2), 146–174 (2013). https://doi.org/10.1007/s10703-012-0170-4
7. Chatterjee, K., Jurdzinski, M., Henzinger, T.: Simple stochastic parity games. In: In Proceedings of the International Conference for Computer Science Logic (CSL). pp. 100–113 (2003), http://chess.eecs.berkeley.edu/pubs/729.html
8. Church, A.: Application of recursive arithmetic to the problem of circuit synthesis. Journal of Symbolic Logic **28**(4) (1963)
9. D'Ippolito, N., Rodríguez, N., Sardiña, S.: Fully observable non-deterministic planning as assumption-based reactive synthesis. J. Artif. Intell. Res. **61**, 593–621 (2018). https://doi.org/10.1613/jair.5562
10. Hausmann, D., Piterman, N., Sağlam, I., Schmuck, A.K.: Fair ω-regular games (2024), extended version available at https://arxiv.org/abs/310.13612
11. Majumdar, R., Mallik, K., Schmuck, A.K., Soudjani, S.: Symbolic control for stochastic systems via finite parity games. Nonlinear Analysis: Hybrid Systems **51**, 101430 (2024). https://doi.org/10.1016/j.nahs.2023.101430
12. Martin, D.: Borel determinacy. Annals of Mathematics **65**, 363–371 (1975)
13. Mazala, R.: Infinite games. In: Grädel, E., Thomas, W., Wilke, T. (eds.) Automata, Logics, and Infinite Games: A Guide to Current Research. Lecture Notes in Computer Science, vol. 2500, pp. 23–42. Springer (2001)
14. Meyer, P.J., Sickert, S., Luttenberger, M.: Strix: Explicit reactive synthesis strikes back! In: 30th International Conference on Computer Aided Verification. Lecture Notes in Computer Science, vol. 10981, pp. 578–586. Springer (2018)
15. Nilsson, P., Ozay, N., Liu, J.: Augmented finite transition systems as abstractions for control synthesis. Discret. Event Dyn. Syst. **27**(2), 301–340 (2017). https://doi.org/10.1007/s10626-017-0243-z

16. Piterman, N., Pnueli, A., Sa'ar, Y.: Synthesis of reactive(1) designs. In: Proceedings of the 7th International Conference on Verification, Model Checking, and Abstract Interpretation. p. 364–380. VMCAI'06, Springer-Verlag, Berlin, Heidelberg (2006)
17. Pnueli, A., Rosner, R.: A framework for the synthesis of reactive modules. In: Proc. Intl. Conf. on Concurrency: Concurrency 88. Lecture Notes in Computer Science, vol. 335, pp. 4–17. Springer-Verlag (1988)
18. Rabin, M.: Decidability of second order theories and automata on infinite trees. Trans. Amer. Math. Soc. **141**, 1–35 (1969)
19. Sağlam, I., Schmuck, A.K.: Solving odd-fair parity games. In: 42th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (2023), (to appear)
20. Schewe, S., Finkbeiner, B.: Bounded synthesis. In: 4th Int. Symp. on Automated Technology for Verification and Analysis. Lecture Notes in Computer Science, vol. 4218, pp. 245–259. Springer (2006)
21. Thistle, J.G., Malhamé, R.: Control of ω-automata under state fairness assumptions. Systems & control letters **33**(4), 265–274 (1998)
22. Thomas, W.: Languages, automata, and logic. Handbook of Formal Language Theory **III**, 389–455 (1997)
23. Walukiewicz, I.: Monadic second-order logic on tree-like structures. Theor. Comput. Sci. **275**(1-2), 311–346 (2002). https://doi.org/10.1016/S0304-3975(01)00185-2
24. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theor. Comput. Sci. **200**(1-2), 135–183 (1998). https://doi.org/10.1016/S0304-3975(98)00009-7