Fast Symbolic Algorithms for Omega-Regular Games under Strong Transition Fairness

Technical Report MPI-SWS-2020-007r
July 2021

TAMAJIT BANERJEE, IIT Delhi, India
RUPAK MAJUMDAR, MPI-SWS, Germany
KAUSHIK MALLIK, MPI-SWS, Germany
ANNE-KATHRIN SCHMUCK, MPI-SWS, Germany
SADEGH SOUDJANI, Newcastle University, UK

We consider fixpoint algorithms for two-player games on graphs with ω-regular winning conditions, where the environment is constrained by a *strong transition fairness* assumption. Strong transition fairness is a widely occurring special case of strong fairness, which requires that any execution is strongly fair with respect to a specified set of live edges: whenever the source vertex of a live edge is visited infinitely often along a play, the edge itself is traversed infinitely often along the play as well.

We show that, surprisingly, strong transition fairness retains the algorithmic characteristics of the fixpoint algorithms for ω-regular games—the new algorithms can be obtained simply by replacing certain occurrences of the controllable predecessor by a new almost sure predecessor operator. For Rabin games with k pairs, the complexity of the new algorithm is $O(n^{k+2}k!)$ symbolic steps, which is independent of the number of live edges in the strong transition fairness assumption. Further, we show that GR(1) specifications with strong transition fairness assumptions can be solved with a 3-nested fixpoint algorithm, same as the usual algorithm. In contrast, strong fairness necessarily requires increasing the alternation depth depending on the number of fairness assumptions.

We get symbolic algorithms for (generalized) Rabin, parity and GR(1) objectives under strong transition fairness assumptions as well as a direct symbolic algorithm for qualitative winning in stochastic ω-regular games that runs in $O(n^{k+2}k!)$ symbolic steps, improving the state of the art. Previous approaches for handling fairness assumptions would either increase the alternation depth of the fixpoint algorithm or require an up-front automata-theoretic construction that would increase the state space, or both.

Finally, we have implemented a BDD-based synthesis engine based on our algorithm. We show on a set of synthetic and real benchmarks that our algorithm is scalable, parallelizable, and outperforms previous algorithms by orders of magnitude.

All proofs can be found in the appendix.

1 INTRODUCTION

Symbolic algorithms for two-player graph games are at the heart of many problems in the automatic synthesis of correct-by-construction hardware, software, and cyber-physical systems from logical specifications. The problem has a rich pedigree, going back to Church [Church 1963] and a sequence of seminal results [Buchi and Landweber 1969; Emerson and Jutla 1988, 1991; Gurevich and Harrington 1982; Kupferman and Vardi 2005; Pnueli and Rosner 1989; Rabin 1969; Zielonka 1998]. A chain of reductions can be used to reduce the synthesis problem for ω-regular specifications to finding winning strategies in two-player games on graphs, for which (symbolic) algorithms are known (see, e.g., [Emerson and Jutla 1991; Piterman and Pnueli 2006; Pnueli and Rosner 1988; Zielonka 1998]). These reductions and algorithms form the basis for algorithmic reactive synthesis.

This report refines and supersedes the original version published in February 2021 as Technical Report MPI-SWS-2020-007.
Authors' addresses: Tamajit Banerjee, Department of Computer Science and Engineering, IIT Delhi, India, cs1190408@iitd.ac.in; Rupak Majumdar, MPI-SWS, Germany, rupak@mpi-sws.org; Kaushik Mallik, MPI-SWS, Germany, kmallik@mpi-sws.org; Anne-Kathrin Schmuck, MPI-SWS, Germany, akschmuck@mpi-sws.org; Sadegh Soudjani, Newcastle University, UK, sadegh.soudjani@ncl.ac.uk.

In practice, it is often the case that no solution exists to a given synthesis problem, but for "uninteresting" reasons. For example, consider synthesizing a mutual exclusion protocol from a specification that requires (1) that at most one of two processes can be in the critical section at any time and (2) that a process wishing to enter the critical section is eventually allowed to do so. As stated, there may not be a feasible solution to the problem because a process within the critical section may decide to stay there forever. Similarly, in a synthesis problem involving concurrent threads, no solution may exist simply because the scheduler may decide never to pick a particular thread. Fairness assumptions rule out such uninteresting conditions by constraining the possible behaviors of the environment. The winning condition under fairness is of the form

$$\text{Fairness Assumption} \implies \omega\text{−regular Specification.} \tag{1}$$

For example, a fairness constraint can state that whenever a process is in its critical section, it must eventually leave it or that, if a thread is enabled infinitely often, then it is picked by the scheduler infinitely often. These two examples, and in fact many other practical instances of fairness, actually fall into a particular subclass of fairness assumptions, called *strong transition fairness* [Baier and Katoen 2008; Francez 1986; Queille and Sifakis 1983]. A strong transition fairness assumption can be modeled by a set of *live* environment transitions in the underlying two-player game graph. Whenever the source vertex of a live transition is visited infinitely often, the transition will be taken infinitely often by the environment. Unfortunately, despite the widespread prevalence of strong transition fairness, current symbolic algorithms for solving games do not take advantage of their special structure in the winning condition (1) and no algorithm better than those for general (Streett) liveness conditions is known.

In this paper, we show a surprising syntactic transformation that modifies well-known symbolic fixpoint algorithms for solving two-player games on graphs *without* fairness assumptions, such that the modified fixed point solves the game for the winning condition (1) whenever the given fairness assumption can be specified as *strong transition fairness*. To appreciate the simplicity of our modification, let us consider the well-known fixpoint algorithm for parity games [Emerson and Jutla 1991] given by the μ-calculus formula

$$\nu Y_{2k}.\ \mu X_{2k-1}.\ \ldots \nu Y_2.\ \mu X_1 (C_1 \cap \mathrm{Cpre}(X_1)) \cup (C_2 \cap \mathrm{Cpre}(Y_2)) \cup \ldots \cup (C_{2k} \cap \mathrm{Cpre}(Y_{2k})), \tag{2a}$$

where $\mathrm{Cpre}(X)$ denotes the controllable predecessor operator. In the presence of strong transition fairness, the new algorithm becomes

$$\begin{aligned} &\nu Y_{2k}.\ \mu X_{2k-1} \ldots \nu Y_2.\ \mu X_1. \\ &\quad (C_1 \cap \mathrm{Apre}(Y_2, X_1)) \cup (C_2 \cap \mathrm{Cpre}(Y_2)) \cup (C_3 \cap \mathrm{Apre}(Y_4, X_3)) \ldots \cup (C_{2k} \cap \mathrm{Cpre}(Y_{2k})). \end{aligned} \tag{2b}$$

The only syntactic change we make is to substitute every controllable predecessor for each μ variable X_i by a new *almost sure predecessor operator* $\mathrm{Apre}(Y_{i+1}, X_i)$ incorporating also the previous ν variable Y_{i+1}.

In a nutshell, our results show that one can solve games under strong transition fairness assumptions on environment behaviors *while retaining* the algorithmic characteristics of known symbolic fixpoint algorithms when fairness assumptions are not considered. We prove the correctness of our syntactic fixpoint transformation for solving Rabin games [Piterman and Pnueli 2006; Rabin 1969] and *generalized Rabin games*. Further, we also show its correctness for Reachability, Safety, (generalized) Büchi, (generalized) co-Büchi, Rabin-chain, parity [Emerson and Jutla 1991; Maler et al. 1995], and GR(1) games [Piterman et al. 2006] as special cases. While our proofs are subtle, symbolic implementations of our algorithms require very small changes to existing code. Moreover, our empirical evaluation demonstrates that our new algorithm can be orders of magnitude more efficient than previous algorithms based on strong fairness.

Now, let us get into more details.

Recall that *symbolic algorithms* solve two-player games by finding the set of states of the underlying game graph from which the game can be won. They do so by manipulating sets of states and computing fixed points of monotone operators. The benefit of symbolic approaches is that they allow efficient implementations based on manipulations of formulas (often represented using data structures such as BDDs). Such implementations can scale to very large finite state spaces or to infinite, but symbolically representable, state spaces. Indeed, these fixpoint expressions are the cornerstone of many reactive synthesis tools [Brenguier et al. 2014; Ehlers and Raman 2016; Michaud and Colange 2018]. Our approach allows existing symbolic implementations of reactive synthesis to be only slightly modified to incorporate strong transition fairness assumptions.

A symbolic fixpoint algorithm for Rabin games is given by Piterman and Pnueli [2006]. A Rabin game is played between two players Player 0 and Player 1, which move a token along the edges of a directed graph whose vertices are partitioned between them. If the token is in a vertex owned by Player 0, she moves the token along some outgoing edge. If, on the other hand, the vertex is owned by Player 1, he decides the edge. Whether the resulting infinite *play* is winning for Player 0 is decided by the *Rabin winning condition* which is defined using a set of pairs of subsets of the the graph vertices, $\{\langle G_1, R_1\rangle, \ldots, \langle G_k, R_k\rangle\}$. Player 0 wins the Rabin game if there is some $i \in \{1, \ldots, k\}$, such that the set of vertices visited infinitely often intersects G_i and does not intersect R_i. The fixpoint algorithm of Piterman and Pnueli [2006] has alternation depth $2k + 1$ for a Rabin condition with k pairs and runs in time $O(n^{k+1}k!)$.

Rabin conditions form a class of a canonical acceptance condition for all ω-regular objectives, thus, solving a game with any ω-regular objective can be reduced to solving a Rabin game after a product construction with a suitable deterministic automaton. Therefore, our new fixpoint algorithm for Rabin games under strong transition fairness solves the winning condition (1) whenever the environment assumption can be expressed by live edges. Live edges are edges of the game graph originating in Player 1 vertices such that whenever the source vertex of a live edge is visited infinitely often, the edge will be taken infinitely often by Player 1.

A Rabin game under strong transition fairness is a special case of a Rabin game under a *strong fairness (compassion)* assumption [Baier and Katoen 2008, p.364]. A compassion assumption is described by a *Streett winning condition*, which is the dual of a Rabin winning condition. A Streett condition is also specified by a set of pairs of subsets of vertices $\{\langle G_1, R_1\rangle, \ldots, \langle G_l, R_l\rangle\}$. It is satisfied by an infinite play if for each $i \in \{1, \ldots, l\}$, whenever the set of vectices visited infinitely often intersects G_i, it also intersects R_i. Since the dual of a Streett condition is a Rabin condition, a Rabin game with k Rabin pairs under a *compassion assumption* with l Streett pairs is equivalent to a Rabin game (without environment assumptions) with $k + l$ Rabin pairs. Hence, it can be solved by the algorithm of Piterman and Pnueli [2006] with alternation depth $2(k + l) + 1$ that runs in $O(n^{k+l+1}(k + l)!)$ symbolic steps. In a well-defined sense, one cannot expect a general fixpoint solution of lower alternation depth [Bradfield 1998]. In contrast, our algorithm for the special case of strong transition fairness has alternation depth $2(k + 1)$ and runs in $O(n^{k+2}k!)$ symbolic steps. Hence, our algorithm has almost the same complexity as Piterman and Pnueli's algorithm for Rabin games *without* environment assumptions—*independent* of the number of transitions in the strong transition fairness assumption. In many practical cases, including the example of synthesizing mutual exclusion protocols, finding schedulers for concurrent threads, and many other applications, strong transition fairness is sufficient to express the interesting environment assumptions.

The idea to consider strong transition fairness as a tractable fragment of *strong fairness (compassion)* assumptions is inspired by work on the synthesis of supervisory controllers for non-terminating processes by Thistle and Malhamé [1998]. Here, a fixed-point algorithm for the general problem [Thistle 1995] which manipulates Rabin automata, is shown to significantly simplify under a slightly weaker transition fairness assumption. While our algorithm shares the underlying intuition behind the simplification of Thistle and Malhamé [1998], it is syntactically very different due to the symbolic manipulation of sets of states rather then automata.

Next, we consider the GR(1) fragment of LTL introduced by Piterman et al. [2006]. Formulas in the GR(1) fragment consist of environment assumptions expressible as a conjunction of Büchi and safety constraints, and specifications also given as a conjunction of Büchi and safety constraints. GR(1) was introduced as an efficient fragment for synthesis problems in the presence of *weak* fairness assumptions; in particular, Piterman et al. [2006] show a 3-nested fixpoint algorithm . Over the years, the GR(1) fragment has been extensively used as a useful logical fragment of LTL for reactive synthesis, especially in the cyber-physical and robotics domains [Alur et al. 2013; Kress-Gazit et al. 2007, 2009; Maoz and Ringert 2015; Svoreňová et al. 2017]. In fact, there are several reactive synthesis tools which only support the GR(1) fragment for its tractability [Ehlers and Raman 2016; Finucane et al. 2010; Wongpiromsarn et al. 2011]. By applying the same syntactic modification as outlined in (2b), we generalize the 3-nested fixpoint alogirthm for GR(1) objectives to a new 3-nested fixpoint algorithm for GR(1) objectives *with additional* strong *transition* fairness constraints! Recall that the GR(1) fragment is designed explicitly to rule out strong fairness constraints because of the absence of suitable low-depth fixpoint algorithms. Our result shows that, in contrast to full strong fairness, strong *transition* fairness retains algorithmic efficiency while enabling many expressive fairness constraints that go beyond the ones expressible in GR(1).

Another byproduct of our algorithm is a fully symbolic algorithm for qualitative winning in *stochastic* generalized Rabin games. Stochastic two-player games (also known as $2^1/_2$-player games) generalize two-player graph games with an additional category of "random" vertices: whenever the game reaches a random vertex, a random process picks one of the outgoing edges (uniformly at random, w.l.o.g.). The qualitative winning problem asks whether a vertex of the game graph is almost surely winning for Player 0. Stochastic Rabin games were studied by Chatterjee et al. [2005], who showed that the problem remains NP-complete and that winning strategies can be restricted to be pure and memoryless. Moreover, they showed a reduction from qualitative winning in an n-vertex k-pair stochastic Rabin game to an $O\left(n(k+1)\right)$-vertex $(k+1)$-pair (deterministic) Rabin game, resulting in an $O\left((n(k+1))^{k+2}(k+1)!\right)$ algorithm. In contrast, we get a direct $O(n^{k+2}k!)$ symbolic algorithm for the problem.

Our result yields a symbolic algorithm in the following way. We replace the probabilistic transitions with transitions of the environment constrained by *extreme fairness* as described by Pnueli [1983]. Extreme fairness is a special case of strong transition fairness, and is specified via a set of Player 1 vertices. A run is extremely fair if it is strongly transition fair for *every* outgoing edge from these vertices. We show that, to solve a qualitative stochastic generalized Rabin game, we can equivalently solve the generalized Rabin game under extreme fairness. Thus, our algorithm gives a direct symbolic algorithm for this problem.

We have implemented our algorithm in a symbolic reactive synthesis tool called Fairsyn. Fairsyn uses a multi-threaded BDD library [van Dijk and van de Pol 2015] and implements an acceleration technique for the fixpoints [Long et al. 1994]. We show on a number of synthetic benchmarks from the very large transition systems benchmark suite [Garavel and Descoubes 2003] that our algorithm, with the improvements, can scale to large Rabin games and the performance scales with the number of cores. Additionally, we evaluate our tool on two case studies, one from software synthesis [Chatterjee et al. 2013] and the other from stochastic control synthesis [Dutreix et al. 2020]. We show that Fairsyn scales well on these case studies, and outperforms a state-of-the-art stochastic game solver by an order of magnitude. In contrast, a solver that treats transition fairness as Streett fairness does not finish on these case studies.

In conclusion, the contributions of the paper are as follows.

(1) We provide a direct symbolic fixpoint algorithm for Rabin and generalized Rabin games under a strong transition fairness assumption on the environment. The alternation depth of the fixpoint expression depends only on the number of Rabin pairs and not on the number of transition fairness constraints. This is in contrast to strong fairness (Streett) assumptions on the environment.
(2) As special cases, we show that our fixpoint formula generalizes fixpoint algorithms for well-known sub-cases: Reachability, Safety, (generalized) Büchi, (generalized) co-Büchi, Rabin-chain or parity, and GR(1) games—all under strong transition fairness. In all cases, the recipe for the new algorithm is surprisingly simple: it replaces some controllable predecessor operators in the "usual" fixpoint algorithms with an almost sure predecessor operator while possibly including an additional leading ν variable.
(3) Since extreme fairness is a special case, we obtain a direct symbolic algorithm for qualitative generalized Rabin conditions for stochastic two-player games.
(4) We have implemented our algorithm in a BDD-based synthesis tool Fairsyn. We demonstrate on a number of synthetic and real benchmarks that our algorithm scales to large state spaces, whereas a naive encoding into Streett games does not finish on the larger examples.

2 PRELIMINARIES

Notation: We use the notation $\mathbb{N}_0$ to denote the set of natural numbers including "0". Given $a, b \in \mathbb{N}_0$, we use the notation $[a; b]$ to denote the set $\{n \in \mathbb{N}_0 \mid a \leq n \leq b\}$. Observe that, by definition, $[a; b]$ is an empty set if $a > b$. For any set $A \subseteq U$ defined on the universe U, we use the notation $\overline{A}$ to denote the complement of A.

Let A and B be two sets and $R \subseteq A \times B$ be a relation. We use the notation $\mathrm{dom}(R)$ to denote the domain of R, which is the set $\{a \in A \mid \exists b \in B\ .\ (a, b) \in R\}$. For any element $a \in A$, we use the notation $R(a)$ to denote the set $\{b \in B \mid (a, b) \in R\}$, and for any element $b \in B$, we use the notation $R^{-1}(b)$ to denote the set $\{a \in A \mid (a, b) \in R\}$. We generalize $R(\cdot)$ to operate on sets in the following way: for any $A' \subseteq A$, we write $R(A') := \cup_{a \in A'} R(a)$, and for any $B' \subseteq B$, we write $R^{-1}(B') := \cup_{b \in B'} R^{-1}(b)$.

Given an alphabet A, we use the notation A^* and A^ω to denote respectively the set of all finite words and the set of all infinite words formed using the letters of the alphabet A. We use A^∞ to denote the set $A^* \cup A^\omega$. Given two words $a \in A^*$ and $b \in A^\infty$, we use $a \cdot b$ to denote their concatenation.

2.1 Two-Player Games

Game Graphs: We define a *two-player game graph* as a tuple $\mathcal{G} = \langle V, V_0, V_1, E \rangle$, where (i) $V = V_0 \uplus V_1$ is a finite set of vertices[1] that is partitioned into the sets V_0 and V_1; (ii) $E \subseteq (V \times V)$ is a relation denoting the set of edges; The two players are called Player 0 and Player 1, who control the vertices V_0 and V_1 respectively.

Strategies: A strategy of Player 0 is a function $\rho_0 \colon V^* \cdot V_0 \to V$ with the constraint $\rho_0(w \cdot v) \in E(v)$ for every $w \cdot v \in V^* \times V_0$. Likewise, a strategy of Player 1 is a function $\rho_1 \colon V^* \cdot V_1 \to V$ with the constraint $\rho_1(w \cdot v) \in E(v)$ for every $w \cdot v \in V^* \times V_1$. Of special interest is the class of memoryless strategies: a strategy ρ_0 of Player 0 is *memoryless* if for every $w_1 \cdot v, w_2 \cdot v \in V^* \times V_0$, we have $\rho_0(w_1 \cdot v) = \rho_0(w_2 \cdot v)$.

[1]We use the terms 'vertex' and 'state' interchangeably in this paper.

Plays: Consider an infinite sequence of vertices $\pi = v^0 v^1 v^2 \ldots \in V^\omega$. The sequence π is called a *play* over $\mathcal{G}$ starting at the vertex v^0 if for every $i \in \mathbb{N}_0$, we have $v^i \in V$ and $(v^i, v^{i+1}) \in E$. In our convention for denoting vertices, superscripts (ranging over $\mathbb{N}_0$) will denote the position of a vertex within a given play, whereas subscripts, either 0 or 1, will denote the membership of a vertex in the sets V_0 or V_1 respectively. Let ρ_0 and ρ_1 be a given pair of strategies of Player 0 and Player 1, respectively, and let v^0 be a given initial vertex. The play *compliant with* ρ_0 *and* ρ_1 is the unique play $\pi = v^0 v^1 v^2 \ldots$ for which for every $i \in \mathbb{N}_0$, if $v^i \in V_0$ then $v^{i+1} = \rho_0(v^0 \ldots v^i)$, and if $v^i \in V_1$ then $v^{i+1} = \rho_1(v^0 \ldots v^i)$.

Winning Conditions: A *winning condition* φ is a set of infinite plays over $\mathcal{G}$, i.e., $\varphi \subseteq V^\omega$. We adopt Linear Temporal Logic (LTL) notation for describing winning conditions. The atomic propositions for the LTL formulae are sets of vertices, i.e., elements of the set 2^V. We use the standard symbols for the Boolean and the temporal operators: "$\neg$" for *negation*, "$\wedge$" for *conjunction*, "$\vee$" for *disjunction*, "$\rightarrow$" for *implication*, "$\mathcal{U}$" for *until* ($A \,\mathcal{U}\, B$ means "the play remains inside the set A until it moves to the set B"), "$\bigcirc$" for *next* ($\bigcirc A$ means "the next vertex is in the set A"), "$\Diamond$" for *eventually* ($\Diamond A$ means "the play will eventually visit a vertex from the set A"), and "$\Box$" for *always* ($\Box A$ means "the play will only visit vertices from the set A"). The syntax and semantics of LTL can be found in standard textbooks [Baier and Katoen 2008]. By slightly abusing notation, we will use φ interchangeably to denote both the LTL formula and the set of plays satisfying φ. Hence, we write $\pi \in \varphi$ (instead of $\pi \models \varphi$) to denote the satisfaction of the formula φ by the play π.

Winning Regions: Player 0 wins a two-player game over the game graph $\mathcal{G}$ for a winning condition φ from a vertex $v^0 \in V$ if there is a Player 0 strategy ρ_0 such that for all Player 1 strategies ρ_1, the play π from v^0 compliant with ρ_0 and ρ_1 satisfies φ, i.e., $\pi \in \varphi$. The *winning region* $\mathcal{W} \subseteq V$ for Player 0 is the set of vertices from which Player 0 wins the game.

2.2 Fair Adversarial Games

Let $\mathcal{G}$ be a two-player game graph and let $E^\ell \subseteq (V_1 \times V) \cap E$ be a given set of *live* edges. Let $V^\ell := \mathrm{dom}(E^\ell)$ denote the set of Player 1 vertices in the domain of E^ℓ. Intuitively, the edges in E^ℓ represent *fairness assumptions* on Player 1: for *every* edge $(v, v') \in E^\ell$, if v is visited infinitely often along a play, we expect that the edge (v, v') is picked infinitely often by Player 1. I.e., if a vertex v is visited infinitely often, *every* outgoing live edge of v is expected to be taken infinitely often.

We write $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ to denote a game graph with live edges, and extend notions such as plays, strategies, winning conditions, winning region, etc., from game graphs to those with live edges. A play π over $\mathcal{G}^\ell$ is *strongly transition fair* if it satisfies the LTL formula:

$$\alpha := \bigwedge_{(v,v') \in E^\ell} \left(\Box\Diamond v \rightarrow \Box\Diamond(v \wedge \bigcirc v') \right). \tag{3}$$

Given $\mathcal{G}^\ell$ and a winning condition φ, Player 0 wins the *fair adversarial game* over $\mathcal{G}^\ell$ for the winning condition φ from a vertex $v^0 \in V$ if Player 0 wins the game over $\mathcal{G}^\ell$ for the winning condition $\alpha \rightarrow \varphi$ from v^0.

We have two interesting observations about fair adversarial games. First, live edges allow to rule out particular strategies of Player 1, making it easier for Player 0 to win in certain situations. Consider for example a game graph (Fig. 1 (top)) with two vertices p and q. Vertex p (square) is a Player 1 vertex and vertex q is a Player 0 vertex (circle). The edge (p, q) is a live edge (dashed). Suppose the specification for Player 0 is $\varphi = \Box\Diamond q$. In the absence of the live edge, Player 0 does not win for this specification from p, because Player 1 can trap the game in p by always choosing p itself as the successor. In contrast, Player 0 wins from p in the fair adversarial game, because the assumption on the live edge (p, q) forces Player 1 to infinitely often choose the transition to q.

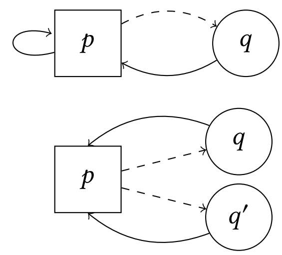

Fig. 1. Two fair adversarial games.

Second, fairness assumptions modeled by live edges restrict the strategy choices of Player 1 less than assuming that Player 1 chooses probabilistically between these edges. Consider for example a fair adversarial game with one Player 1 vertex p (square) which has two outgoing live edges to states q and q' (see Fig. 1 (bottom). If Player 1 chooses randomly between edges (p, q) and (p, q'), *every* finite sequence of visits to states q and q' will happen infinitely often with probability one. This is not true in the fair adversarial game. Here Player 1 is allowed to choose a particular sequence of visits to states q and q' (e.g., only $qq'qq'qq'qq' \ldots$), as long as both are visited infinitely often.

2.3 Symbolic Computations over Game Graphs

Set Transformers: Our goal is to develop symbolic fixpoint algorithms to characterize the winning region of a fair adversarial game over a game graph with live edges. As a first step, given $\mathcal{G}^\ell$, we define the required symbolic transformers of sets of states. We define the existential, universal, and controllable predecessor operators as follows. For $S \subseteq V$, we have

$$\mathrm{Pre}_0^\exists(S) := \{v \in V_0 \mid E(v) \cap S \neq \emptyset\}, \tag{4a}$$

$$\mathrm{Pre}_1^\forall(S) := \{v \in V_1 \mid E(v) \subseteq S\}, \text{ and} \tag{4b}$$

$$\mathrm{Cpre}(S) := \mathrm{Pre}_0^\exists(S) \cup \mathrm{Pre}_1^\forall(S). \tag{4c}$$

Intuitively, the controllable predecessor operator $\mathrm{Cpre}(S)$ computes the set of all states that can be controlled by Player 0 to stay in S after one step regardless of the strategy of Player 1. Additionally, we define two operators which take advantage of the fairness assumption on the live edges. Given two sets $S, T \subseteq V$, we define the live-existential and almost sure predecessor operators:

$$\mathrm{Lpre}^\exists(S) := \{v \in V^\ell \mid E^\ell(v) \cap S \neq \emptyset\}, \text{ and} \tag{5a}$$

$$\mathrm{Apre}(S, T) := \mathrm{Cpre}(T) \cup \left(\mathrm{Lpre}^\exists(T) \cap \mathrm{Pre}_1^\forall(S)\right). \tag{5b}$$

Intuitively, the almost sure predecessor operator[2] $\mathrm{Apre}(S, T)$ computes the set of all states that can be controlled by Player 0 to stay in T (via $\mathrm{Cpre}(T)$) *as well as* all Player 1 states in V^ℓ that (a) will *eventually* make progress towards T if Player 1 obeys its fairness-assumptions encoded in α (via $\mathrm{Lpre}^\exists(T)$) and (b) will never leave S in the "meantime" (via $\mathrm{Pre}_1^\forall(S)$). We see that all set transformers are monotonic with respect to set inclusion. Further, $\mathrm{Cpre}(T) \subseteq \mathrm{Apre}(S, T)$ always holds, $\mathrm{Cpre}(T) = \mathrm{Apre}(S, T)$ if $V^\ell = \emptyset$, and $\mathrm{Apre}(S, T) \subseteq \mathrm{Cpre}(S)$ if $T \subseteq S$ (see Lem. B.1 for a proof).

Fixpoint Algorithms in the μ-calculus: We use the μ-calculus [Kozen 1983] as a convenient logical notation used to define a symbolic algorithm (i.e., an algorithm that manipulates sets of states rather then individual states) for computing a set of states with a particular property over a

[2]We will justify the naming of this operator later in Rem. 1.

given game graph $\mathcal{G}$. The formulas of the μ-calculus, interpreted over a two-player game graph $\mathcal{G}$, are given by the grammar

$$\varphi \ ::= \ p \mid X \mid \varphi \cup \varphi \mid \varphi \cap \varphi \mid pre(\varphi) \mid \mu X.\varphi \mid \nu X.\varphi$$

where p ranges over subsets of V, X ranges over a set of formal variables, *pre* ranges over monotone set transformers in $\{\mathrm{Pre}_0^\exists, \mathrm{Pre}_1^\forall, \mathrm{Cpre}, \mathrm{Lpre}^\exists, \mathrm{Apre}\}$, and μ and ν denote, respectively, the least and the greatest fixed point of the functional defined as $X \mapsto \varphi(X)$. Since the operations $\cup$, $\cap$, and the set transformers *pre* are all monotonic, the fixed points are guaranteed to exist. A μ-calculus formula evaluates to a set of states over $\mathcal{G}$, and the set can be computed by induction over the structure of the formula, where the fixed points are evaluated by iteration. We omit the (standard) semantics of formulas (see [Kozen 1983]).

3 FAIR ADVERSARIAL RABIN GAMES

This section presents the main result of this paper, which is a symbolic fixpoint algorithm that computes the winning region of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ with respect to any ω-regular property formalized as a Rabin winning condition.

Our new fixpoint algorithm has multiple unique features.
(I) It works directly over $\mathcal{G}^\ell$, without requiring any pre-processing step to reduce $\mathcal{G}^\ell$ to a "normal" two-player game. This feature allows us to obtain a direct symbolic algorithm for stochastic games as a by-product (see Sec. 5).
(II) Conceptually, our symbolic algorithm is *not* more complex than the known algorithm solving Rabin games over "normal" two-player game graphs by Piterman and Pnueli [2006] (see Sec. 3.3).
(III) Our new fixpoint algorithm is obtained from the known algorithm of Piterman and Pnueli [2006] by a simple syntactic change (as previewed in (2)). We simply replace all controllable predecessor operators over least fixpoint variables by the almost sure predecessor operator invoking the preceding maximal fixpoint variable. This makes the proof of our new fixpoint algorithm conceptually simple (see Sec. 3.2).

At a higher level, our syntactic change is a very simple yet efficient transformation to incorporate environment assumptions expressible by live edges into reactive synthesis while retaining computational efficiency. Most remarkably, this transformation also works *directly* for fixpoint algorithms solving reachability, safety, Büchi, (generalized) co-Büchi, Rabin-chain and parity games, as these can be formalized as particular instances of a Rabin game (see Sec. 3.4). Moreover, it also works for generalized Büchi and GR(1) games. However, as these games are particular instances of a *generalized* Rabin game, we prove these special cases separately in Sec. 4 after formally introducing generalized Rabin games.

3.1 The Symbolic Algorithm

Fair adversarial Rabin Games: A *Rabin* winning condition is defined by the set $\mathcal{R} = \{\langle G_1, R_1\rangle, \ldots, \langle G_k, R_k\rangle\}$, where $G_i, R_i \subseteq V$ for all $i \in [1;k]$. We say that $\mathcal{R}$ has index set $P = [1;k]$. A play π satisfies the *Rabin condition* $\mathcal{R}$ if π satisfies the LTL formula

$$\varphi := \bigvee_{i \in P} \left(\Diamond\Box\overline{R}_i \wedge \Box\Diamond G_i \right). \tag{6}$$

We now present our new symbolic fixpoint algorithm to compute the winning region of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ with respect to a Rabin winning condition $\mathcal{R}$.

Theorem 3.1. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\mathcal{R}$ be a Rabin condition over $\mathcal{G}$ with index set $P = [1; k]$. Further, let Z^* denote the fixed point*

$$\nu Y_{p_0}.\mu X_{p_0}. \bigcup_{p_1 \in P} \nu Y_{p_1}.\mu X_{p_1}. \bigcup_{p_2 \in P \setminus \{p_1\}} \nu Y_{p_2}.\mu X_{p_2}.\ldots \bigcup_{p_k \in P \setminus \{p_1,\ldots,p_{k-1}\}} \nu Y_{p_k}.\mu X_{p_k}. \left[\bigcup_{j=0}^{k} C_{p_j} \right], \tag{7a}$$

$$\text{where} \quad C_{p_j} := \left(\bigcap_{i=0}^{j} \overline{R}_{p_i} \right) \cap \left[\left(G_{p_j} \cap \mathrm{Cpre}(Y_{p_j}) \right) \cup \left(\mathrm{Apre}(Y_{p_j}, X_{p_j}) \right) \right], \tag{7b}$$

with[3] *$p_0 = 0$, $G_{p_0} := \emptyset$ and $R_{p_0} := \emptyset$. Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (6). Moreover, the fixpoint algorithm runs in $O(n^{k+2}k!)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.*

3.2 Proof Outline

Given a Rabin winning condition over a "normal" two-player game, Piterman and Pnueli [2006] provided a symbolic fixpoint algorithm which computes the winning region for Player 0. The fixpoint algorithm in their paper is almost identical to our fixpoint algorithm in (7): it only differs in the last term of the constructed C-terms in (7b). Piterman and Pnueli [2006] define the term C_{p_j} as

$$\left(\bigcap_{i=0}^{j} \overline{R}_{p_i} \right) \cap \left[\left(G_{p_j} \cap \mathrm{Cpre}(Y_{p_j}) \right) \cup \left(\mathrm{Cpre}(X_{p_j}) \right) \right].$$

Intuitively, a single term C_{p_j} computes the set of states that always remain within $Q_{p_j} := \bigcap_{i=0}^{j} \overline{R}_{p_i}$ while always re-visiting G_{p_j}. I.e, given the simpler (local) winning condition

$$\psi := \Box Q \wedge \Box \Diamond G \tag{8}$$

for two sets $Q, G \subseteq V$, the set

$$\nu Y.\ \mu X.\ Q \cap [(G \cap \mathrm{Cpre}(Y)) \cup (\mathrm{Cpre}(X))] \tag{9}$$

is known to define exactly the states of a "normal" two-player game $\mathcal{G}$ from which Player 0 has a strategy to win the game with winning condition ψ [Maler et al. 1995]. Such games are typically called *Safe Büchi Games*. The key insight in the proof of Thm. 3.1 is to show that the new definition of C-terms in (7b) via the new *almost sure predecessor operator* Apre actually computes the winning state sets of *fair adversarial* safe Büchi games. Subsequently, we generalize this intuition to the fixpoint for the Rabin games.

Fair Adversarial Safe Büchi Games: A fair adversarial safe Büchi game is formalized in the following theorem.

Theorem 3.2. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $Q, G \subseteq V$ be two state sets over $\mathcal{G}$. Further, let*

$$Z^* := \nu Y.\ \mu X.\ Q \cap [(G \cap \mathrm{Cpre}(Y)) \cup (\mathrm{Apre}(Y, X))]\,. \tag{10}$$

Then Z^* is equivalent to the winning region of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition ψ in (8). Moreover, the fixpoint algorithm runs in $O(n^2)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

Intuitively, the fixed points in (9) and (10) consist of two parts: (a) A minimal fixed point over X which computes (for any fixed value of Y) the set of states that can *reach* the "target state set" $T := Q \cap G \cap \mathrm{Cpre}(Y)$ while staying inside the safe set Q, and (b) a maximal fixed point over Y

[3] The Rabin pair $\langle G_{p_0}, R_{p_0} \rangle = \langle \emptyset, \emptyset \rangle$ in (7) is artificially introduced to make the fixpoint representation more compact. It is not part of $\mathcal{R}$.

which ensures that the only states considered in the target T are those that allow to re-visit a state in T while staying in Q.

By comparing (9) and (10) we see that our syntactic transformation only changes part (a). Hence, in order to prove Thm. 3.2 it essentially remains to show that this transformation works for the even simpler *safe reachability games.*

Fair Adversarial Safe Reachability Games: A safe reachability condition is a tuple $\langle T, Q\rangle$ with $T, Q \subseteq V$ and a play π satisfies the *safe reachability condition* $\langle T, Q\rangle$ if π satisfies the LTL formula

$$\psi := Q\,\mathcal{U}\,T. \tag{11}$$

A safe reachability game is often called a *reach-while-avoid* game, where the safe sets are specified by an unsafe set $R := \overline{Q}$ that needs to be avoided. Their fair adversarial version is formalized in the following theorem, proved in App. B.2.1.

Theorem 3.3. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell\rangle$ be a game graph with live edges and $\langle T, Q\rangle$ be a safe reachability winning condition. Further, let*

$$Z^* := \nu Y.\, \mu X.\, T \cup (Q \cap \mathrm{Apre}(Y, X)). \tag{12}$$

Then Z^* is equivalent to the winning region of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition ψ in (11)*. Moreover, the fixpoint algorithm runs in $O(n^2)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.*

To gain some intuition on the correctness of Thm. 3.3, let us recall that the fixed-point for safe reachability games *without* live edges is given by:

$$\mu X.\, T \cup (Q \cap \mathrm{Cpre}(X)). \tag{13}$$

Intuitively, the fixed point in (13) is initialized with $X^0 = \emptyset$ and computes a sequence $X^0, X^1, \ldots, X^k$ of increasing sets until $X^k = X^{k+1}$. We say that v has rank r if $v \in X^r \setminus X^{r-1}$. All states contained in X^r allow Player 0 to force the play to reach T in at most $r-1$ steps while staying in Q. The corresponding Player 0 strategy ρ_0 is known to be winning w.r.t. (11) and along every play π compliant with ρ_0, the path π remains in Q and the rank is always decreasing.

To see why the same strategy is also *sound* in the *fair adversarial* safe reachability game $\mathcal{G}^\ell$, first recall that for vertices $v \notin V^\ell$ of $\mathcal{G}^\ell$, the almost sure pre-operator $\mathrm{Apre}(X, Y)$ simplifies to $\mathrm{Cpre}(X)$. With this, we see that for every $v \notin V^\ell$ a Player 0 winning strategy $\widetilde{\rho}_0$ in $\mathcal{G}^\ell$ can always force plays to stay in Q and to decrease their rank, similar to ρ_0. With this, we see that plays π which are compliant with such a strategy $\widetilde{\rho}_0$ and visit a vertex in V^ℓ only finitely often satisfy (11).

The only interesting case for soundness of Thm. 3.3 are therefore plays π that visits states in V^ℓ infinitely often. However, as the number of vertices is finite, we only have a finite number of ranks and hence a certain vertex $v \in V^\ell$ with a finite rank r needs to get visited by π infinitely often. Due to the definition of Apre we however know that only states $v \in V^\ell$ are contained in X^r if v has an outgoing *live* edge reaching X^k with $k < r$. With this, reaching v infinitely often implies that also a state with rank k s.t. $k < r$ will get visited infinitely often. As $X^1 = T$ we can show by induction that T is eventually visited along π while π always remains in Q until then.

In order to prove *completeness* of Thm. 3.3 we need to show that all states in $V \setminus Z^*$ are loosing for Player 0. Here, again the reasoning is equivalent to the "normal" safe reachability game for $v \notin V^\ell$. For vertices $v \in V^\ell$, we see that v is not added to Z^* via Apre if $v \notin T$ and either (i) all its outgoing live transitions do not make progress towards T or, (ii) it has some outgoing edge (not necessarily a live one) that makes it leave Z^*). One can therefore construct a Player 1 strategy that for (i)-vertexes always chooses a live transition and thereby never makes progress towards T (also if v is visited infinitely often), and for (ii)-vertexes ensures that they are only visited once on plays

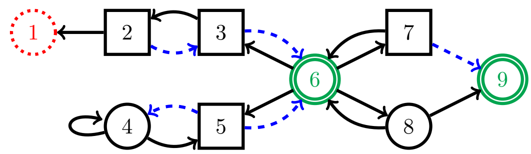

Fig. 2. Fair adversarial game graph discussed in Examples 3.4 and 3.5 with vertex sets $G = \{6, 9\}$ (double cycled, green), $\overline{Q} = \{1\}$ (red,dotted) and live edges $E^\ell = \{(2, 3), (3, 6), (5, 4), (5, 6), (7, 9))\}$ (dashed, blue). Player 0 and Player 1 vertices are indicated by cycles and boxes, respectively.

which remain in Q. This ensures that (ii)-vertexes never make progress towards T via their possibly existing rank-decreasing live edges.

A detailed soundness and completeness proof of Thm. 3.3 along with the respective Player 0 and Player 1 strategy construction is provided in App. B.2.1. In addition, Thm. 3.2 is proven in Sec. B.2.2 by a reduction to Thm. 3.3 for every iteration over Y.

Example 3.4 (Fair adversarial safe reachability game). We consider a fair adversarial safe reachability game over the game graph depicted in Fig. 2 with target vertex set $T = G = \{6, 9\}$ and safe vertex set $Q = V \setminus \{1\}$.

We denote by Y^m the m-th iteration over the fixpoint variable Y in (12), where $Y^0 = V$. Further, we denote by X^{mi} the set computed in the i-th iteration over the fixpoint variable X in (12) during the computation of Y^m where $X^{m0} = \emptyset$. We further have $X^{m1} = T = \{6, 9\}$ as $\mathrm{Apre}(\cdot, \emptyset) = \emptyset$. Now we compute

$$\begin{aligned} X^{12} &= T \cup (Q \cap \mathrm{Apre}(Y^0, X^{11})) \\ &= \{6, 9\} \cup (V \setminus \{1\} \cap [\underbrace{\mathrm{Cpre}(X^{11})}_{\{8\}} \cup \underbrace{(\mathrm{Lpre}^{\exists}(X^{11}) \cap \mathrm{Pre}_1^{\forall}(V))}_{\{3,5,7\}}]) = \{5, 6, 7, 8, 9\} \end{aligned} \tag{14}$$

We observe that the only vertex added to X via the Cpre term is vertex 8. States $\{3, 5, 7\}$ are added due to the existing live edge leading to a target vertex. Here, we note that vertex 7 is added due to its live edge to vertex 9. The additional requirement $\mathrm{Pre}_1^{\forall}(V)$ in $\mathrm{Apre}(Y^0, X^{11})$ is trivially satisfied for all vertices at this point as $Y^0 = V$ and can therefore be ignored. Doing one more iteration over X we see that now vertex 4 gets added via the Cpre term (as it is a Player 0 vertex that allows progress towards 5) and vertex 2 is added via the Apre term (as it allows progress to 3 via a live edge). The iteration over X terminates with $Y^1 = X^{1*} = V \setminus \{1\}$.

Re-iterating over X for Y^1 gives $X^{22} = X^{12} = \{5, 6, 7, 8, 9\}$ as before. However, now vertex 2 does not get added to X^{23} because vertex 2 has an edge leading to $V \setminus Y^1 = \{1\}$. Therefore the iteration over X terminates with $Y^2 = X^{2*} = V \setminus \{1, 2\}$. When we now re-iterate over X for Y^2 we see that vertex 3 is not added to X^{32} any more, as vertex 3 has a transition to $V \setminus Y^2 = \{1, 2\}$. Therefore the iteration over X now terminates with $Y^3 = X^{3*} = V \setminus \{1, 2, 3\}$. Now re-iterating over X does not change the vertex set anymore and the fixed-point terminates with $Y^* = Y^3 = V \setminus \{1, 2, 3\}$.

We note that the fixed-point formula (13) for "normal" safe reachability games terminates after two iterations over X with $X^* = \{6, 8, 9\}$, as vertex 8 is the only vertex added via the Cpre operator in (14). Due to the stricter notion of Cpre requiring that *all* outgoing edges of Player 0 vertices make process towards the target, (13) does not require an outer largest fixed-point over Y to "trap" the play in a set of vertices which allow progress when "waiting long enough". This "trapping" required in (12) via the outer fixed-point over Y actually fails for vertices 2 and 3 (as they are excluded form the winning set of (12)). Here, Player 1 can enforce to "escape" to the unsafe vertex 1 in two steps before 2 and 3 are visited infinitely often (which would imply progress towards 6 via the existing live edges).

We see that the winning region in the "normal" game is significantly smaller than the winning region for the fair adversarial game, as adding live transitions restricts the strategy choices of Player 1, making it easier for Player 0 to win the game.

Example 3.5 (Fair adversarial safe Büchi game). We now consider a fair adversarial safe Büchi game over the game graph depicted in Fig. 2 with sets $G = \{6, 9\}$ and $Q = V \setminus \{1\}$.

We first observe that we can rewrite the fixed-point in (10) as

$$\nu Y.\ \mu X.\ [Q \cap G \cap \mathrm{Cpre}(Y)] \cup [Q \cap (\mathrm{Apre}(Y, X))]\,. \tag{15}$$

Using (15) we see that for $Y^0 = V$ we can define $T^0 := Q \cap G \cap \mathrm{Cpre}(V) = G = \{6, 9\}$. Therefore the first iteration over X is equivalent to (14) and terminates with $Y^1 = X^{1*} = V \setminus \{1\}$.

Now, however, we need to re-compute T for the next iteration over X and obtain $T^1 = Q \cap G \cap \mathrm{Cpre}(Y^1) = V \setminus \{1\} \cap \{6, 9\} \cap V \setminus \{1, 2, 9\} = \{6\}$. This re-computation of T^1 checks which target vertices are re-reachable, as required by the Büchi condition. As vertex 9 has no outgoing edge it is trivially not re-reachable.

With this, we see that for the next iteration over X we only have one target vertex $T^1 = \{6\}$. If we recall that vertex 7 is added to X^{22} due to its live edge to 9, we see that it is now not added anymore. Intuitively, we have to exclude 7 as Player 1 can always decide to take the live edge towards 9 from 7 (also if 7 only gets visited once), and therefore prevents to re-visit a target state.

Now, vertices 2 and 3 get eliminated for the same reason as in the safe reachability game within the second and third iteration over Y. The overall fixed-point computation therefore terminates with $Y^* = Y^3 = \{4, 5, 6, 8\}$.

Proof of Thm. 3.1: With Thm. 3.3 and Thm. 3.2 in place, the proof of Thm. 3.1 is essentially equivalent to the proof of Piterman and Pnueli [2006] while utilizing Thm. 3.3 and Thm. 3.2 at all suitable places. For completeness, we give the full proof of Thm. 3.1, including the memoryless strategy construction, in App. B.3. In addition, we illustrate the steps of the fixed-point algorithm in (7) with a simple fair adversarial Rabin game (depicted in Fig. 9) which has two acceptance pairs in App. A.

Remark 1. *We remark that the fixpoint* (12), *as well as the* Apre *operator, are similar in structure to the solution of almost surely winning states in concurrent reachability games [de Alfaro et al. 1998]. In concurrent games, the fixed point captures the largest set of states in which the game can be trapped while maintaining a positive probability of reaching the target. In our case, the fixed point captures the largest set of states in which Player* 0 *can keep the game while ensuring a visit to the target either directly or through the live edges. The commonality justifies our notation and terminology for* Apre.

Remark 2. *Aminof et al. [2004] studied fair CTL and LTL model checking where the fairness condition is given by a transition fairness with* all *edges of the transition system live. They show that CTL model checking under this all-live fairness condition, can be syntactically transformed to* non-fair *CTL model checking. A similar transformation is possible for fair model checking of Büchi, Rabin, and Streett formulas. The correctness of their transformation is based on reasoning similar to our* Apre *operator. For example, a state satisfies the CTL formula $\forall\Diamond p$ under fairness iff all paths starting from the state either eventually visits p or always visits states from which a visit to p is possible.*

3.3 Complexity

Complexity Analysis of (7)**:** For Rabin games with k Rabin pairs, Piterman and Pnueli [2006] show a fixpoint formula with alternation depth $2k + 1$. Using the accelerated fixpoint computation technique of Long et al. [1994], they deduce a bound of $O(n^{k+1}k!)$ symbolic steps. We show in App. C that this accelerated fixpoint computation can also be applied to (7) yielding a bound of $O(n^{k+2}k!)$

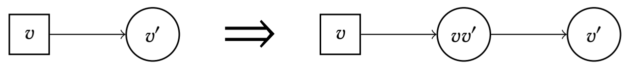

Fig. 3. Left: A live edge (v, v') in $\mathcal{G}^\ell$. Right: The gadget used to replace (v, v') in $\widehat{\mathcal{G}}$. The vertex named vv' is a newly added vertex in $\widehat{\mathcal{G}}$; v belongs to $\widehat{V}_1$, vv' belongs to $\widehat{V}_0$, but v' may belong to either $\widehat{V}_0$ or $\widehat{V}_1$.

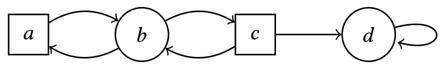

Fig. 4. Counterexample to the equality of strong transition fairness and strong fairness (compassion).

symbolic steps. (The additional complexity is because of an additional outermost ν-fixpoint.) Thus our algorithm is almost as efficient as the original algorithm for Rabin games without environment assumptions—*independent* of the number of strong transition fairness assumptions!

Comparison with a Naïve Solution: We show a naïve reduction from fair adversarial Rabin games to usual Rabin games. Suppose $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ is a game graph with live edges, $\mathcal{R} = \{\langle G_1, R_1 \rangle, \ldots, \langle G_k, R_k \rangle\}$ is a Rabin winning condition defined over $\mathcal{G}^\ell$, and φ is the corresponding LTL specification as defined in (6). Let $\widehat{\mathcal{G}} = \langle \widehat{V}, \widehat{V}_0, \widehat{V}_1, \widehat{E} \rangle$ be a game graph obtained by just replacing every live edge of $\mathcal{G}^\ell$ with a gadget shown in Fig. 3 and explained next. For every live edge $(v, v') \in E^\ell$ we introduce a new intermediate vertex named $vv' \in \widehat{V}$, and without loss of generality we assume that $vv' \in \widehat{V}_0$. (We could have equivalently used the convention that $vv' \in \widehat{V}_1$.) Then we replace the edge (v, v') with a pair of new edges $(v, vv') \in \widehat{E}$ and $(vv', v') \in \widehat{E}$; the rest remains the same as in $\mathcal{G}$. Assuming that $|E^\ell| = l$ and $|V| = n$, the number of vertices of $\widehat{\mathcal{G}}$ is $n + l$.

Intuitively, the event of the newly introduced vertices being reached in $\widehat{\mathcal{G}}$ simulates the event of the corresponding live edge being taken in $\mathcal{G}^\ell$, and vice versa. We are now ready to transfer the specification $\alpha \rightarrow \varphi$ to a new Rabin winning condition $\widehat{\mathcal{R}}$ for $\widehat{\mathcal{G}}$. First observe that $\alpha \rightarrow \varphi$ is equivalent to $\neg\alpha \vee \varphi$, and $\neg\alpha$ can be expressed in LTL as $\bigvee_{(v,v') \in E^\ell} (\Box\Diamond\{v\} \wedge \Diamond\Box\overline{\{vv'\}})$. and is therefore equivalent to the Rabin winning condition $\mathcal{R}^\ell \coloneqq \{\langle \{v\}, \{vv'\} \rangle \mid (v, v') \in E^\ell\}$. Since Rabin winning conditions are closed under union, we obtain the new Rabin condition $\widehat{\mathcal{R}} \coloneqq \mathcal{R} \cup \mathcal{R}^\ell$.

Once $\widehat{\mathcal{G}}$ and $\widehat{\mathcal{R}}$ are obtained, one can use the fixpoint algorithm of Piterman and Pnueli [2006] for "normal" two-player Rabin games. This whole process yields a symbolic algorithm for fair adversarial Rabin games with $2(k + l) + 1$ alternations of fixpoint operators on a set of $(n + l)$ vertices that runs in time $O((n + l)^{k+l+1}(k + l)!)$. In contrast, our main theorem shows that we get a symbolic fixpoint expression with $2(k + 1)$ alternations that runs in $O(n^{k+2}k!)$ symbolic steps. In many applications, we expect $l = \Theta(n)$, for which our algorithm is significantly faster.

Remark 3. *As already mentioned in the introduction, not all strong fairness assumptions (Streett assumptions) can be translated into live edges (see e.g., [Baier and Katoen 2008, p.264]). As an example, consider the two-player game graph depicted in Fig. 4. Player* 0 *and Player* 1 *vertices are indicated by a circle and a box, respectively. Now consider the following one-pair Streett assumption*

$$\varphi_A \coloneqq \Box\Diamond\{a, b, c\} \rightarrow \Box\Diamond\{a\} = \Diamond\Box\{d\} \vee \Box\Diamond\{a\}. \tag{16}$$

This fairness assumption states that it is not possible for a game to infinitely stay inside the set $\{a, b, c\}$ *if Player* 0 *decides to not transition from* b *to* a *anymore from some point onward. We see that we cannot model this behavior by a fair edge leaving a Player* 1 *(square) state. If we mark the edge* (c, d) *live, any fair play will transition to* d *no matter if* a *is visited infinitely often or not. Let us call this fair edge assumption* α_A. *Then we see that* $\alpha_A \rightarrow \varphi_A$ *but not vice versa.*

3.4 Specialized Rabin Games

This section shows that the known fixpoint algorithms for Rabin chain, Parity, and Generalized Co-Büchi winning conditions allow for the same "syntactic transfomation" as in the Rabin case to get the right algoirthm for their fair adversarial version. We prove these claims by reducing the fixed point in (7) to the special cases induced by the aforementioned winning conditions.

We note that the fixpoint algorithm for fair adversarial Rabin games in (7) reduces to the normal fixed point for Rabin games if $E^\ell = \emptyset$. Therefore, our reductions of (7) to fixpoint algorithms for other winning conditions also proves these reductions in the usual case. We are not aware of such reductions proved elsewhere in the literature.

Fair Adversarial Rabin Chain Games: A *Rabin chain* winning condition [Mostowski 1984] is a *Rabin* condition $\mathcal{R} = \{\langle G_1, R_1\rangle, \ldots, \langle G_k, R_k\rangle\}$, with the additional *chain condition*

$$R_1 \supseteq R_2 \supseteq \ldots \supseteq R_k \qquad \text{and} \qquad G_1 \supseteq G_2 \supseteq \ldots \supseteq G_k. \tag{17}$$

Intuitively, the fixpoint algorithm computing Z^* in (7) simplifies to a single permutation sequence, namely $p_1 = k$, $p_2 = k-1$, ..., $p_k = 1$, if (17) holds. This is formalized in the following theorem which is proved in App. B.4.1.

Theorem 3.6. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell\rangle$ be a game graph with live edges and $\mathcal{R}$ be a Rabin condition over $\mathcal{G}$ with k pairs for which the chain condition* (17) *holds. Further, let*

$$Z^* := \nu Y_0.\ \mu X_0.\ \nu Y_k.\ \mu X_k.\ \nu Y_{k-1}.\ \ldots \mu X_1.\ \textstyle\bigcup_{j=0}^{k} \widetilde{C}_j, \tag{18a}$$

$$\text{where} \quad \widetilde{C}_j := \overline{R}_j \cap \left[\left(G_j \cap \mathrm{Cpre}(Y_j)\right) \cup \mathrm{Apre}(Y_j, X_j)\right] \tag{18b}$$

with $G_{p_0} := \emptyset$ and $R_{p_0} := \emptyset$. Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 *in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in* (6)*. Moreover, the fixpoint algorithm runs in $O(n^{k+2})$ symbolic steps, and a memoryless winning strategy for Player* 0 *can be extracted from it.*

Fair Adversarial Parity Games: A *Parity* winning condition [Emerson and Jutla 1989] is defined by a set $\mathcal{C} = \{C_1, C_2, \ldots C_{2k}\}$ of colors, where each $C_i \subseteq V$ is the set of vertices of $\mathcal{G}$ with color i. Further, $\mathcal{C}$ partitions the state space, i.e., $\bigcup_{i\in[1;2k]} C_i = V$ and $C_i \cap C_j = \emptyset$ for all $i, j \in [1; 2k]$ with $i \neq j$. A play π satisfies the *Parity condition* $\mathcal{C}$ if π satisfies the LTL formula

$$\varphi := \textstyle\bigwedge_{i\in[1;k]} \left(\Box\Diamond C_{2i-1} \rightarrow \bigvee_{j\in[i;k]} \Box\Diamond C_{2j}\right). \tag{19}$$

That is, the maximal color visited infinitely often along π is even. A Parity winning condition $\mathcal{C}$ with $2k$ colors corresponds to the Rabin chain winning condition

$$\{\langle F_2, F_3\rangle, \ldots, \langle F_{2k}, \emptyset\rangle\} \qquad \text{s.t. } F_i := \textstyle\bigcup_{j=i}^{2k} C_j, \tag{20}$$

which has k pairs. Due to $\mathcal{C}$ forming a partition of the state space one can further simplify the Rabin chain fixpoint algorithm in (18). Indeed, the resulting fixpoint algorithm coincides with the one obtained from applying our syntactic transfomation to the well-known algorithm for Parity games (see (2)). This is formalized in the next theorem, which is proved in App. B.4.2.

Theorem 3.7. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell\rangle$ be a game graph with live edges and $\mathcal{C}$ be a Parity condition over $\mathcal{G}$ with $2k$ colors. Further, let*

$$\begin{aligned} Z^* := \nu Y_{2k}.\ \mu X_{2k-1} \ldots \nu Y_2.\ \mu X_1.& \\ (C_1 \cap \mathrm{Apre}(Y_2, X_1)) \cup (C_2 \cap \mathrm{Cpre}(Y_2)) \cup (C_3 \cap \mathrm{Apre}(Y_4, X_3)) \ldots \cup (C_{2k} \cap \mathrm{Cpre}(Y_{2k})).& \end{aligned} \tag{21}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 *in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in* (19)*. Moreover, the fixpoint algorithm runs in $O(n^{k+1})$ symbolic steps, and a memoryless winning strategy for Player* 0 *can be extracted from it.*

Fair Adversarial (Generalized) Co-Büchi Games: A *Co-Büchi* winning condition is defined by a subset $A \subseteq V$ of vertices of $\mathcal{G}$. A play π satisfies the *Co-Büchi condition* A if π satisfies

$$\varphi := \Diamond\Box A. \tag{22}$$

A *Generalized Co-Büchi* winning condition is defined by a set $\mathcal{A} = \{A_1, \ldots A_r\}$, where each $A_i \subseteq V$ is a subset of vertices of $\mathcal{G}$. A play π satisfies the *Generalized Co-Büchi condition* $\mathcal{A}$ if π satisfies

$$\varphi := \bigvee_{a\in[1;r]} \Diamond\Box A_a. \tag{23}$$

Generalized Co-Büchi winning conditions correspond to a Rabin condition $\mathcal{R}$ with r pairs s.t.

$$\forall j \in [1;r] \;.\; R_j := \overline{A}_j \quad \text{and} \quad G_j := V. \tag{24}$$

Intuitively, the fact that $G_j := V$ for all j leads to a cancelation of all Apre terms in C_j and all terms become ordered, i.e., we have $C_{p_{j+1}} \subseteq C_{p_j}$ for every permutation sequence used in (7). As we take the union over all C_{p_j}-s in (7a), the term C_{p_1} absorbs all others for every permutation sequence. Hence, for every permutation sequence we only have two terms left, one for $j = 0$ (over the artificially introduced Rabin pairs $G_{p_0} = R_{p_0} = \emptyset$) and one for the first choice p_1 made in this particular permutation. This is formalized in the following theorem which is proved in App. B.4.3.

Theorem 3.8. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\mathcal{A}$ be a generalized Co-Büchi winning condition $\mathcal{G}$ with r pairs. Further, let*

$$Z^* := \nu Y_0.\, \mu X_0.\, \bigcup_{a\in[1;r]} \nu Y_a.\, \mathrm{Apre}(Y_0, X_0) \cup (\overline{A}_a \cap \mathrm{Cpre}(Y_a)). \tag{25}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (23). Moreover, the fixpoint algorithm runs in $O(rn^2)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

4 GENERALIZED RABIN GAMES

In this section, we slightly generalize our main result, Thm. 3.1, to fair adversarial *generalized* Rabin games. That is, for each Rabin pair, we allow the goal set G_i to be a set of goal sets $\mathbf{G}_j = \{\, {}^1G_j, \ldots, {}^{m_j}G_j\}$. Then a play fulfills the winning condition if there exists one generalized Rabin pair $\langle \mathbf{G}_i, R_i\rangle$ such that the play eventually remains in $\overline{R}_i$ and visits *all* sets ${}^lG_i$ infinitely often.

The motivation of this generalization is to show that our syntactic transformation also works for fair adversarial games with a *generalized reactivity winning condition of rank* 1 (GR(1) games for short) [Piterman et al. 2006]. *Generalized* Rabin games allow us to see a GR(1) winning condition as a particularly simple instantiation of a Rabin game as shown in Sec. 4.2.

4.1 Fair Adversarial Generalized Rabin Games

Generalized Rabin Conditions: A *generalized Rabin condition* is defined by a set $\widetilde{\mathcal{R}} = \{\langle \mathbf{G}_1, R_1\rangle, \ldots, \langle \mathbf{G}_k, R_k\rangle\}$ where each $\mathbf{G}_j = \{\, {}^1G_j, \ldots, {}^{m_j}G_j\}$ is a finite set s.t. ${}^lG_j \subseteq V$ for all $j \in [1;k]$ and all $l \in [1;m_j]$. We say that $\widetilde{\mathcal{R}}$ has global index set $P = [1;k]$. A play π satisfies the *generalized Rabin condition* $\widetilde{\mathcal{R}}$ if π satisfies the LTL formula

$$\varphi := \bigvee_{j\in P} \left(\Diamond\Box\overline{R}_j \wedge \bigwedge_{l\in[1;m_j]} \Box\Diamond\, {}^lG_j \right). \tag{26}$$

Recalling the discussion of Sec. 3.1, we know that the proof of Thm. 3.1 fundamentally relies on the correctness of our transformation for safe Büchi (Thm. 3.2) and safe reachability (Thm. 3.3) games. Similarly, one needs to prove correctness of our syntactic transformation for safe *generalized* Büchi games in the case of generalized Rabin games.

Safe Generalized Büchi Games A *safe generalized Büchi condition* is defined by a tuple $\langle \mathcal{F}, Q\rangle$ where $Q \subseteq V$ is a set of safe states and $\mathcal{F} = \{\ {}^1F, \dots,\ {}^sF\}$ is a set of goal sets. A play π satisfies the *safe generalized Büchi condition* $\langle \mathcal{F}, Q\rangle$ if π satisfies the LTL formula

$$\varphi := \Box Q \wedge \textstyle\bigwedge_{l\in[1;s]} \Box\Diamond\ {}^lF. \tag{27}$$

Now we can apply our syntactic transformation to the usual fixpoint algorithm for solving safe generalized Büchi games and prove its correctness for all fair adversarial plays. This is formalized in the next theorem and proved in App. B.5.1.

THEOREM 4.1. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell\rangle$ be a game graph with live edges and $\langle \mathcal{F}, Q\rangle$ with $\mathcal{F} = \{\ {}^1F, \dots,\ {}^sF\}$ a safe generalized Büchi winning condition. Further, let*

$$Z^* := \nu Y.\ \bigcap_{b\in[1;s]} \mu\ {}^bX.\ Q \cap \Big[(\ {}^bF \cap \mathrm{Cpre}(Y)) \cup \mathrm{Apre}(Y,\ {}^bX)\Big]. \tag{28}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (27)*. Moreover, the fixpoint algorithm runs in $O(sn^2)$ symbolic steps, and a finite-memory winning strategy for Player 0 can be extracted from it.*

Intuitively, the proof of Thm. 4.1 reduces to Thm. 3.2 in a similar manner as the proof of Thm. 3.2 reduces to Thm. 3.3. However, the challenge in proving Thm. 4.1 is to show that it is indeed sound to use the fixpoint variable Y which is actually the intersection of fixpoint variables X both within Cpre and Apre. The proof of this correctness essentially requires to show that upon termination we have $Y^* =\ {}^bX^*$ for all $b \in [1;s]$ (see App. B.5.1 for a formal proof).

The Symbolic Algorithm: By knowing that (28) allows to correctly solve safe generalized Büchi games, we can immediately generalize this observation to Rabin games. This is formalized in the following theorem which is an immediate consequence of Thm. 3.1 and Thm. 4.1.

THEOREM 4.2. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell\rangle$ be a game graph with live edges and $\widetilde{\mathcal{R}}$ be a generalized Rabin condition over $\mathcal{G}$ with index set $P = [1;k]$. Further, let*

$$Z^* := \nu Y_0.\ \mu X_0.\ \bigcup_{p_1\in P} \nu Y_{p_1}.\ \bigcap_{l_1\in[1;m_{p_1}]} \mu\ {}^{l_1}X_{p_1}.\ \dots\dots \bigcup_{p_k\in P\setminus\{p_1,\dots,p_{k-1}\}} \nu Y_{p_k}.\ \bigcap_{l_k\in[1;m_{p_k}]} \mu\ {}^{l_k}X_{p_k}.\ \bigcup_{j=0}^{k}\ {}^{l_j}C_{p_j}, \tag{29a}$$

$$\text{where}\quad {}^{l_j}C_{p_j} := \Big(\textstyle\bigcap_{i=0}^{j} \overline{R}_{p_i}\Big) \cap \Big[\Big(\ {}^{l_j}G_{p_j} \cap \mathrm{Cpre}(Y_{p_j})\Big) \cup \mathrm{Apre}(Y_{p_j},\ {}^{l_j}X_{p_j})\Big] \tag{29b}$$

with[4] *$p_0 = 0$, $G_{p_0} := \{\emptyset\}$ and $R_{p_0} := \emptyset$. Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in* (26)*. Moreover, the fixpoint algorithm runs in $O(n^{k+2}k!m_1\dots m_k)$ symbolic steps, and yields a finite-memory winning strategy for Player 0.*

The proof of Thm. 4.2 is almost identical to the proof of Thm. 3.1 in App. B.3, when using Thm. 4.1 instead of Thm. 3.2 in all appropriate places. This, yields a finite memory winning strategy by suitably "stacking" the individual finite-memory strategies constructed in the proof of Thm. 4.1. (See App. B.5.2 for a complete proof of Thm. 4.2.)

[4] Again, the generalized Rabin pair $\langle \mathrm{G}_{p_0}, R_{p_0}\rangle$ in (7) is artificially introduced and not part of $\widetilde{\mathcal{R}}$.

4.2 Fair Adversarial GR(1) Games

Within this section, we show how fair adversarial Rabin games can be reduced to fair adversarial games with GR(1) winning conditions.

GR(1) winning condition: A GR(1) winning condition is defined by two sets $\mathcal{A} = \{A_1, \ldots, A_r\}$ and $\mathcal{F} = \{F_1, \ldots, F_s\}$, where for every $i \in [1;r]$ and $j \in [1;s]$, $A_i, F_j \subseteq V$. A play π satisfies the GR(1) condition $(\mathcal{A}, \mathcal{F})$ if it satisfies the LTL formula

$$\varphi := \left(\bigwedge_{a\in[1;r]} \Box\Diamond A_a\right) \rightarrow \left(\bigwedge_{b\in[1;s]} \Box\Diamond F_b\right) = \left(\bigvee_{a\in[1;r]} \Diamond\Box\overline{A}_a\right) \vee \left(\bigwedge_{b\in[1;s]} \Box\Diamond F_b\right). \tag{30}$$

By comparing φ in (30) with φ in (26), we see that a GR(1) condition $(\mathcal{A}, \mathcal{F})$ can be transformed into a generalized Rabin condition $\widetilde{\mathcal{R}}$ with $k = r + 1$ pairs, such that

$$\forall j \in [1;r]\ .\ R_j := A_j \quad \text{and} \quad \mathbf{G}_j := \{V\}, \quad \text{and} \tag{31a}$$

$$R_k := \emptyset \quad \text{and} \quad \mathbf{G}_k := \mathcal{F}. \tag{31b}$$

Fixpoint Algorithm: We first observe that the first r Rabin pairs with trivial goal sets actually correspond to a generalized Co-Büchi condition (compare (24)) which can be solved by the fixed point in Thm. 3.8 (see Sec. 3.4). Intuitively, the fixed point in Thm. 3.8 only needs to consider single indices form $P = [1;r]$ rather then full permutation sequences as in Thm. 3.1. By adding the last tuple $\langle \mathbf{G}_k, R_k \rangle$ to the winning condition, we essentially need to consider two indices in each conjunct of (18), i.e., p_j (with $j \in [1;r]$) and p_k. In principle, we would need to consider both possible orderings of these two indices (compare (29)). However, by inspecting (31) we see that the sets corresponding to these indices always fulfill a (generalized) chain condition (compare (17)). That is, we have $R_j \supseteq R_k$ and $V = {}^1G_j \supseteq {}^bF$ for any $j \in [1;r]$ and $b \in [1;s]$. Hence, we only need to consider the permutation sequence $p_k p_j$ (compare (18)). Using this insight, along with some additional simplifications, we indeed yield the fixed point that we would obtain by simply applying our transformation to the well-known GR(1) fixed point (compare e.g. [Piterman et al. 2006]). This observation is formalized in the next theorem and proved in App. B.5.3.

Theorem 4.3. *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $(\mathcal{A}, \mathcal{F})$ a GR(1) winning condition. Further, let*

$$Z^* = \nu Y_k.\ \bigcap_{b\in[1;s]} \mu\, {}^bX_k.\ \bigcup_{a\in[1;r]} \nu Y_a.\quad (F_b \cap \mathrm{Cpre}(Y_k)) \cup \mathrm{Apre}(Y_k,\ {}^bX_k) \cup (\overline{A}_a \cap \mathrm{Cpre}(Y_a)). \tag{32}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (30). *Moreover, the fixpoint algorithm runs in $O(n^2 rs)$ symbolic steps, and a finite-memory winning strategy for Player 0 can be extracted from it.*

In particular, the strategy extraction is performed in the same way as by Piterman et al. [2006] for a "normal" GR(1) game.

Remark 4. *Svoreňová et al. [2017] presented a symbolic fixpoint algorithm for stochastic games (which can be modeled using fair adversarial games, see Sec. 5) with respect to GR(1) winning conditions. While one can show that the output of their algorithm coincides with the output of our newly derived fixpoint algorithm in* (32)*, their algorithm is structurally more involved. On a conceptual level, we feel our insight about simply "swapping" predecessor operators in the right manner is insightful even if one can also use their algorithm to find a solution to this problem.*

Fair Adversarial vs. Environmentally-Friendly GR(1) Games: The idea of the simple "predecessor operator swapping trick" shares resemblence with environmentally-friendly GR(1) synthesis, proposed by Majumdar et al. [2019]. There, the authors show a direct symbolic algorithm to compute

Player 0 strategies which do not win a given GR(1) game vacuously, by rendering the assumptions false. More precisely, given a synthesis game for the specification $\varphi := (\varphi_A \rightarrow \varphi_G)$ with φ_A and φ_G being LTL formulas modeling respectively environment assumptions and system guarantees, Player 0 can win by violating φ_A and thereby satisfying φ vacuously. Environmentally-friendly synthesis rules out such undesired strategies by only computing so called non-conflicting winning strategies. Interestingly, the fixpoint algorithm introduced by Majumdar et al. [2019] also swaps Cpre and Apre operators, but in a slightly different way.

The GR(1) fragment considered by Majumdar et al. [2019] corresponds to a specification $\varphi_A \rightarrow \varphi_G$ where both φ_A and φ_G can be realized by a deterministic generalized Büchi automaton. Hence, they provide an algorithm to compute non-conflicting winning strategies in a deterministic generalized Büchi game under deterministic generalized Büchi assumptions. If the used deterministic Büchi assumptions can be translated into live edges over the same game graph, the resulting fair adversarial game is a generalized Büchi game (not a GR(1) game), solvable by the fixed point in (28) for $Q = V$.

By reducing a GR(1) game to a fair adversarial game, one transforms the given assumption into one expressed by fair edges which cannot be falsified by Player 0 and therefore yields a simpler algorithm to compute non-conflicting strategies. However, the direct relationship between deterministic generalized Büchi assumptions and live-edge assumptions is not known, i.e., we do not know if all environmentally-friendly GR(1) games can be reduced to fair adversarial generalized Büchi games.

Finally, we want to point out that fair adversarial GR(1) games compute winning strategies that are only non-conflicting with respect to the environment assumptions encoded in the live edges. Player 0 can still win a fair adversarial GR(1) game vacuously by falsifying φ_A, i.e., never visiting any set A_i in $\mathcal{A}$ (see (30)) infinitely often.

5 STOCHASTIC GENERALIZED RABIN GAMES

We present an important application of our fixpoint algorithm in solving stochastic two-player games, commonly known as $2^1/_2$-player games. $2^1/_2$-player games form an important subclass of stochastic games, and have been studied quite extensively in the literature [Chatterjee et al. 2005; Condon 1992; Zielonka 2004]. They can be seen as a generalization of two-player games by additionally capturing the environmental randomness inside the game. In order to do so, in addition to Player 0 and Player 1 vertices as in a two-player game, they include a new set of vertices called the *random vertices*. Whenever the game reaches a random vertex, one of the outgoing edges is picked uniformly at random. Player 0 is said to win an $2^1/_2$-player game almost surely if she wins the game with probability 1; the respective Player 0 strategy is called an almost sure winning strategy. We only consider stochastic games with a uniform probability distribution over edges which originate from a random vertex. This is indeed without loss of generality since it is known that stochastic games with other probability distributions over random edges have exactly the same almost sure winning sets as $2^1/_2$-player games [Chatterjee et al. 2005].

We present a reduction from the computation of almost sure winning strategies in $2^1/_2$-player generalized Rabin games to the computation of winning strategies in fair adversarial generalized Rabin games. This yields a direct symbolic algorithm for solving $2^1/_2$-player generalized Rabin games.

5.1 Preliminaries: $2^1/_2$-player games

We introduce the basic setup of the $2^1/_2$-player games.

The game graph: We consider usual $2^1/_2$-player games played between Player 0, Player 1, and a third player representing *environmental randomness*. Formally, a $2^1/_2$-player game graph is a tuple

$\mathcal{G} = \langle V, V_0, V_1, V_r, E\rangle$ where (i) V is a finite set of vertices, (ii) V_0, V_1, and V_r are subsets of V which form a partition of V, and (iii) $E \subseteq V \times V$ is the set of directed edges. The vertices in V_r are called *random vertices*, and the edges originating in a random vertex are called *random edges*. The set of all random edges is denoted by $E_r := E(V_r)$.

Strategies and plays: We define strategies for Player 0 and Player 1 in exactly the same way as the strategies in two-player games. While in principle, we could consider randomized strategies, it is known that optimal strategies for ω-regular winning conditions are pure [Chatterjee et al. 2005]. The new part is when the $2^1/_2$-player game reaches a random vertex, the game chooses one of the random edges uniformly at random. A play is, as usual, an infinite sequence of vertices $(v^0, v^1, \ldots)$ that satisfies the edge relation between two consecutive vertices in the sequence. Due to the presence of random edges, given an initial vertex $v^0 \in V$ and given a pair of strategies ρ_0 and ρ_1 of Player 0 and Player 1 respectively, we will obtain a *probability distribution over the set of plays*. We denote the set of strategies of Player 0 and Player 1 by Π_0 and Π_1, respectively.

Almost sure winning: Let φ be any ω-regular specification over V. Let us denote the event that the runs of a $2^1/_2$-player game graph $\mathcal{G}$ satisfies φ using the symbol $\mathcal{G} \models \varphi$ For a given initial vertex $v^0 \in V$ and for a given pair of strategies ρ_0 and ρ_1 of Player 0 and Player 1, we denote the probability of the occurrence of the event $\mathcal{G} \models \varphi$ by $P^{\rho_0,\rho_1}_{v^0}(\mathcal{G} \models \varphi)$. We define the set of almost sure winning states of Player 0 for the specification φ as the set of vertices $\mathcal{W}^{a.s.} \subseteq V$ such that for every $v \in \mathcal{W}^{a.s.}$,

$$\sup_{\rho_0 \in \Pi_0} \inf_{\rho_1 \in \Pi_1} P^{\rho_0,\rho_1}_{v}(\mathcal{G} \models \varphi) = 1. \tag{33}$$

5.2 The reduction

Suppose $\mathcal{G}$ is a $2^1/_2$-player game graph and $\widetilde{\mathcal{R}}$ is a generalized Rabin winning condition. To obtain the reduced two-player game graph, we simply reinterpret the random vertices as Player 1 vertices and the random edges as live edges. Let us first formalize this notion of the reduced game graph.

Definition 5.1 (Reduction to two-player game with live edges). Let $\mathcal{G} = \langle V, V_0, V_1, V_r, E\rangle$ be a $2^1/_2$-player game graph. Define $\mathit{Derand}(\mathcal{G}) := \langle\langle \widetilde{V}, \widetilde{V_0}, \widetilde{V_1}, \widetilde{E}\rangle, E^\ell\rangle$ as follows:

- $\widetilde{V} = V$, $\widetilde{V_0} = V_0$, $\widetilde{V_1} = V_1 \cup V_r$, $\widetilde{E} = E$, and $E^\ell = E_r$.

It remains to show that the almost sure winning set of Player 0 in $\mathcal{G}$ for the generalized Rabin winning condition $\widetilde{\mathcal{R}}$ is the same as the winning set of Player 0 in the fair adversarial game over $\mathit{Derand}(\mathcal{G})$ for the winning condition $\widetilde{\mathcal{R}}$. This is formalized in the following theorem, which is proved in App. B.6. The proof essentially shows that the random edges of $\mathcal{G}$ simulate the live edges of $\mathit{Derand}(\mathcal{G})$, and vice versa.

Theorem 5.2. *Let $\mathcal{G}$ be a $2^1/_2$-player game graph, $\widetilde{\mathcal{R}}$ be a generalized Rabin condition, $\varphi \subseteq V^\omega$ be the corresponding LTL specification (Eq. (26)) over the set of vertices V of $\mathcal{G}$, and $\mathit{Derand}(\mathcal{G})$ be the reduced two-player game graph. Let $\mathcal{W} \subseteq \widetilde{V}$ be the set of all the vertices from where Player 0 wins the fair adversarial game over $\mathit{Derand}(\mathcal{G})$ for the winning condition φ, and $\mathcal{W}^{a.s.}$ be the almost sure winning set of Player 0 in the game graph $\mathcal{G}$ for the specification φ. Then, $\mathcal{W} = \mathcal{W}^{a.s.}$. Moreover, a winning strategy in $\mathit{Derand}(\mathcal{G})$ is also a winning strategy in $\mathcal{G}$, and vice versa.*

The above theorem generalizes [Glabbeek and Höfner 2019, Thm. 11.1] from liveness properties to all LTL specifications on $2^1/_2$-player games. Together with our symbolic algorithm for fair adversarial Rabin games, the reduction implies a $O(n^{k+2}k!)$ algorithm for stochastic Rabin games for a game with n states and k Rabin pairs. This improves the previous best algorithm from [Chatterjee et al. 2005], which reduces the problem to a normal two-player game with $O(n(k+1))$ states and $k+1$ Rabin conditions, and therefore has a complexity of $O\left((n(k+1))^{k+2}(k+1)!\right)$.

Remark 5. *The idea underlying this section is to replace random edges with live edges to compute almost sure winning states. We recall again that probabilistic choice is different from (i.e., stronger than) strong transition fairness studied in our paper. See Sec. 2.2 for an illustrative example in Fig. 1.*

6 EXPERIMENTAL EVALUATION

We have developed a C++-based tool Fairsyn, which implements the symbolic fair adversarial Rabin fixpoint from Eq. (7) using BDDs. We developed two versions of Fairsyn: A single-threaded version using the (single-threaded) CUDD library [Somenzi 2019], and a multi-threaded version using the (multi-threaded) Sylvan library [van Dijk and van de Pol 2015].

Our tool implements a well-known acceleration technique for fixpoint computations [Long et al. 1994]. It exploits certain monotonicity properties of the fixpoint variables, and "warm-starts" the inner fixpoint iterations by initializing them with earlier computed values for similar configurations of the leading fixpoint variables' iteration indices (see App. C for a formal explanation). The acceleration procedure trades memory for time; it can avoid computations if all the intermediate values of the fixpoint variables for all possible configurations of the fixpoint iteration indices are stored. In practice, this creates an inordinate amount of overhead on the memory requirement: The original algorithm would already run out of memory when solving the smallest instance of the case study reported in Table 1 (first line) on a computer with 1.5 TB of memory. We have therefore adapted the acceleration technique to achieve a novel (space-)bounded acceleration algorithm that we utilize within Fairsyn. Our new algorithm takes an *acceleration parameter M* as input, which bounds the extent to which intermediate values of fixpoint variables are cached (see App. C for details). Whenever no cached value is available during the computation, our algorithm falls back to the default way of initializing fixpoint variables and recomputations.

To show the effectiveness of our proposed symbolic algorithm for fair adversarial Rabin games, we performed various experiments with Fairsyn which fall into two different categories. First, in Sec. 6.1, we demonstrate the merits of utilizing parallelization and acceleration within Fairsyn. Second, in Sec. 6.2, we show the practical relevance of our algorithm by solving two large practical case-studies stemming from the areas of software engineering and control systems.

The experiments in Sec. 6.1 and Sec. 6.2.1 were performed using Sylvan-based Fairsyn on a computer equipped with a 3 GHz Intel Xeon E7 v2 processor with 48 CPU cores and 1.5 TiB RAM. The experiments in Sec. 6.2.2 were performed using CUDD-based Fairsyn on a Macbook Pro (2015) laptop equipped with a 2.7 GHz Dual-Core Intel Core i5 processor with 16 GiB RAM.

6.1 Performance Evaluation

This section discusses a benchmark suite used to empirically evaluate the merits of the two important aspects of Fairsyn, namely the parallelization and the acceleration. Our benchmark suite is build on transition systems taken from the Very Large Transition Systems (VLTS) benchmark suite [Garavel and Descoubes 2003]. For each chosen transition system, we randomly generated benchmark instances of fair adversarial Rabin games with up to 3 Rabin pairs. To transform a given transition systems into a fair adversarial Rabin game, we labeled (i) 50% of randomly chosen vertices as system vertices, (ii) the remaining vertices as environment vertices, (iii) up to 5% of randomly selected environment edges as live edges, and (iv) for every set in $\mathcal{R} = \{\langle G_1, R_1\rangle, \ldots, \langle G_k, R_k\rangle\}$ we randomly selected up to 5% of all vertices to be contained. We have summarized the relevant details of all the randomly generated instances of the fair adversarial Rabin games in Table 3 and Table 4 in App. D. In these examples, the number of vertices were 289–566,639, the number of BDD variables were 9–20, the number of transitions were 1224–3,984,160, and number of live edges were 1–42,757. For all benchmark instances with more than 4 live edges, the naïve version of Fairsyn which treats live

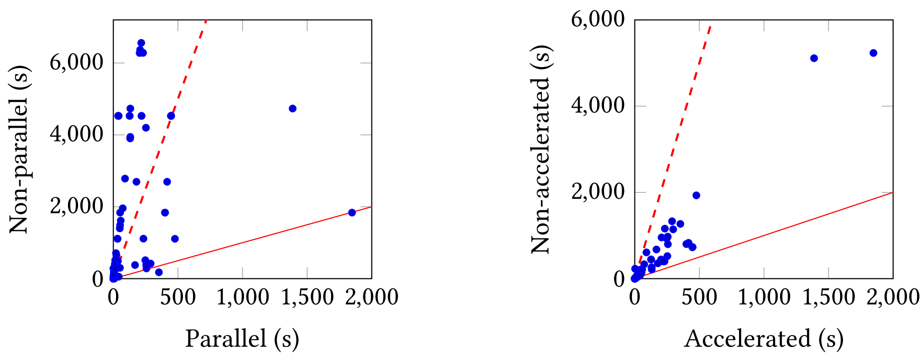

Fig. 5. (Left) Comparison between the computation times for the non-parallel (1 worker thread) and parallel (48 worker threads) version of Fairsyn, with acceleration being enabled in both cases. (Right) Comparison between the computation times for the non-accelerated and the accelerated version of Fairsyn, with parallelization being enabled in both cases. (Both) The points on the solid red line represent the same computation time. The points on the dashed red line represent an order of magnitude improvement.

edges as Streett conditions and transforms them into additional Rabin pairs as discussed in Sec. 3.3, did not terminate after 2 hours.

Merits of parallelization. We ran Fairsyn on 10 different benchmark instances with 1 or 2 Rabin pairs, and varied the number of parallel worker threads used in Fairsyn between 1–48, while keeping the acceleration enabled. The left scatter plot in Fig. 5 plots the computation times with 48 threads (parallel) versus the computation times with 1 thread (non-parallel). Observe that in almost all the experiments, the parallelized version outperforms the non-parallelized version (points above the solid red line). In addition, in many cases the speedup achieved due to the parallelization was more than one order of magnitude (points above the dashed red line).

A more fine-grained analysis of the benefits of parallelization is shown in Fig. 6.(a). Here computation time (in logarithmic scale) is plotted over the number of worker threads used. We observe that the saving due to parallelization is more significant for the curves lying in the top half which correspond to larger examples. This is due to the better utilization of the available pool of worker threads by the larger examples.

Merits of acceleration. We ran Fairsyn on 10 different benchmark instances with 1–3 Rabin pairs, and varied the acceleration parameter M between 2–15, while the number of worker threads was fixed to 48. The right scatter plot in Fig. 5 plots the computation times with $M = 15$ versus the computation times with no acceleration. Observe that in almost all the experiments, the accelerated version outperformed the non-accelerated version (points above the solid red line), and in many cases the achieved speedup is close to an order of magnitude (points near the dashed red line). See Fig. 10 in App. D for a zoomed-in version of Fig. 5.

A more fine-grained analysis of the benefits of acceleration is shown in Fig. 6.(b)–(e). Here we have plotted the total computation time (Plots (b),(d)) and the initialization time (Plots (c),(e)) in logarithmic scale over M for benchmark instances with 2 Rabin pairs (Plots (b),(c)) and 3 Rabin pairs (Plots (d),(e)). Plots for instances with 1 Rabin pair can be found in Fig. 11 in App. D.

The plotted initialization time is needed by the accelerated algorithm for allocating memory to store intermediate fixpoint values. We observe that this initialization time grows exponentially with M, which is due to the $O(M^{k+1}k!)$ space complexity of the acceleration algorithm. As a result, the computational savings due to the use of acceleration get undermined by the high initialization cost

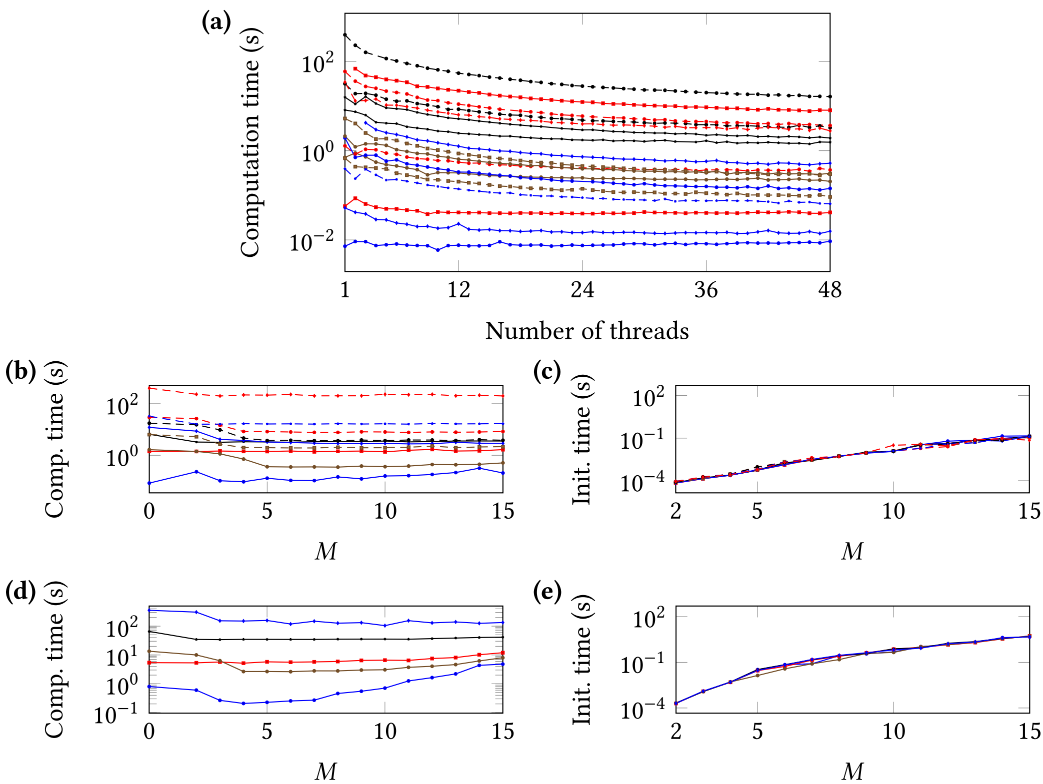

Fig. 6. (a) Effect of parallelization on computation time, with the acceleration enabled. (b,d) Effect of variation of the acceleration parameter M on the total computation time (parallelization being enabled) for 2 and 3 Rabin pairs respectively. (c,e) Effect of variation of the acceleration parameter M on the initialization time for 2 and 3 Rabin pairs respectively. The computation time (Y-axis) is always shown in the logarithmic scale.

for large M. We note that, due to their random generation, the considered benchmark instances are not well structured. This results in low iteration numbers over involved fixed-point variables. Due to this, the allocated memory gets underutilized for large values of M. In the practically relevant examples discussed in Sec. 6.2 the game graph is naturally structured, resulting in a large number of fixpoint iterations and thereby showing superior performance for larger values of M.

6.2 Practical Benchmarks

This section shows that Fairsyn is able to efficiently solve two practical case studies stemming from the areas of software engineering (Sec. 6.2.1) and control systems (Sec. 6.2.2).

6.2.1 Code-Aware Resource Management. We consider a case study introduced by Chatterjee et al. [2013]. It considers the problem of synthesizing a *code-aware resource manager* for a network protocol, i.e., multi-threaded program running on a single CPU. The task of the resource manager is to grant different threads access to different shared synchronization resources (mutexes and counting semaphores). The specification is deadlock freedom across all threads at all time while assuming a *fair scheduler* (scheduling every thread always eventually) and *fair progress* in every thread (i.e., taking every existing execution branch always eventually). By making the resource manager *code-aware*, it can avoid deadlocks by utilizing its knowledge about the require and release characteristics of all treads for different resources.

Chatterjee et al. [2013] showed that the problem of synthesizing a code-aware resource manager can be approximated using a $1^1/_2$-player game[5] generated from the known require and release characteristics of all threads. We used Fairsyn to synthesize a code-aware resource manager for this problem, where the live edges model the aforementioned fairness conditions imposed on the scheduler and the threads.

Motivated by the case study conducted by Chatterjee et al. [2013], we consider a network protocol consisting of 3 threads and 2 queues of bounded capacity, as depicted in Fig. 7. The threads (shown as oval-shaped nodes) are called *generator*, *sender*, and *delay*, and the queues (shown as rectangular nodes) are called *broadcast* and *output*. The generator generates data packets and dispatches them to either the broadcast queue or the output queue. Packets from the broadcast queue are added to the output queue after a random delay, introduced by the delay thread. The purpose of this delay is to avoid packet collisions during broadcasting. The packets in the output queue are in transit and get processed by the sender process. The sender process attempts to transmit packets from the output queue via the network, and when the transmission fails, it adds the respective data packet back to the broadcast queue, so that another transmission attempt can be made after a delay. Access to all queues is protected by mutexes and semaphores. Each queue has one mutex and two semaphores, one for counting the number of empty places and another for counting the number of packets present.

As discussed by Chatterjee et al. [2013], the outlined network protocol may deadlock when both queues are full, a transmission via sender fails, and the sender tries to insert the packet back to the broadcast queue. In this case, due to the output queue being full, the broadcast queue will not be able to make space for the incoming packet, leading to a deadlock situation. The correct strategy for the resource manager to prevent this deadlock is to ensure that the generator never adds packets to the broadcast queue if the output queue is full.

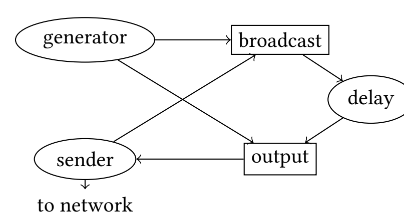

Fig. 7. Structure of network protocol.

We used the parallel and accelerated version of Fairsyn with $M = 15$ to automatically synthesize the resource manager for the outlined network protocol case study. Indeed, Fairsyn was successful in discovering the outlined managing strategy. To showcase Fairsyn's performance on this case study, we report the number of vertices of the problem instance and Fairsyn's computation time to solve it for different queue capacities in Table 1; an extended version of the table with more number of cases has been included in Table 5 in App. D. In all cases, Fairsyn was able to provide expected strategies within a reasonable amount of time. Note that treating the live edges as Streett conditions would result in a game with several million Rabin pairs, making all these examples go far beyond the scope of any synthesis tool for Rabin games.

6.2.2 Controller Synthesis for Stochastically Perturbed Dynamical Systems. Synthesizing verified symbolic controllers for continuous dynamical systems is an active area in cyber-physical systems research [Tabuada 2009]. Recently, it was shown by Majumdar et al. [2021], that the symbolic controller synthesis problem for stochastic continuous dynamical systems can be approximated using a strategy synthesis problem over a (finite) $2^1/_2$-player game graph. This result, together with our reduction in Sec. 5, enables us to use Fairsyn to synthesize a symbolic controller for stochastic continuous dynamical systems. We show in this section, that on different instances of an established case study for this synthesis problem, Fairsyn outperforms state-of-the art synthesis techniques by margins varying between 1 order of magnitude to up to 2.5 orders of magnitude.

[5] A $1^1/_2$-player game is a $2^1/_2$-player game without any Player 1 vertices.

Broadcast Queue Capacity	Output Queue Capacity	Number of Vertices	Number of Transitions	Number of Live edges	Number of BDD variables	Time (seconds)
1	1	5,307,840	10,135,300	5,124,100	25	7.38
2	1	21,231,400	40,541,200	20,496,400	27	24.90
3	1	21,414,100	42,080,300	21,265,900	27	28.98
1	2	21,340,800	40,879,100	20,834,300	27	38.26
1	3	21,559,400	42,756,100	21,772,800	27	51.56
2	2	85,363,200	163,516,000	83,337,200	29	133.20
3	2	86,061,400	169,673,000	86,415,400	29	144.28
2	3	86,237,400	171,024,000	87,091,200	29	163.62
3	3	86,870,100	177,181,000	90,169,300	29	203.15

Table 1. Performance of Fairsyn on the code-aware resource management benchmark experiment.

In the following, we first formalize the case study, which was proposed by Dutreix et al. [2020]. Consider the dynamic model of a bistable switch which is a tuple $\Sigma = (X, U, W, f)$ with a two-dimensional compact state space $X = [0, 4] \times [0, 4] \in \mathbb{R}^2$, a finite input space $U = \{-0.5, 0, 0.5\} \times \{-0.5, 0, 0.5\}$, a two-dimensional bounded disturbance space $W = [-0.4, -0.2] \times [-0.4, -0.2] \in \mathbb{R}^2$, and a transition function $f\colon X \times U \to X$. Suppose $x\colon \mathbb{N} \to X$, $u\colon \mathbb{N} \to U$, and $w\colon \mathbb{N} \to W$ denote the system's state, input, and disturbance trajectories, given as functions of (discrete) time. Note that the functions x, u, w, and f are vector-valued, and we will denote each element of vectors using the element index in the suffix. For instance, x_1, x_2 are the first and the second element of the state trajectory x respectively, and $f_1(x,u), f_2(x,u)$ are the first and the second element of the valuation of the transition function $f(x,u)$ respectively. At each time step k, we assume that $w(k) \in W$ is drawn from a probability distribution with the support W; for our purpose, the shape of the distribution is irrelevant. The state evolution of the system is modeled using a set of difference equations of the following form:

$$x_1(k+1) = f_1(x(k), u(k)) + w_1(k) = x_1(k) + 0.05\,(-1.3x_1(k) + x_2(k)) + u_1(k) + w_1(k), \tag{34}$$
$$x_2(k+1) = f_2(x(k), u(k)) + w_2(k) = x_2(k) + 0.05\left(\frac{(x_1(k))^2}{(x_1(k))^2+1} - 0.25x_2(k)\right) + u_2(k) + w_2(k).$$

A controller for a dynamical system Σ is a function $C\colon X \to U$ that determines the control inputs $u_1(k) := C_1(x(k))$ and $u_2(k) := C_2(x(k))$ in (34) for all time steps k. Recalling that $w(k) \in W$ is drawn from a probability distribution with the support W in every time step, we see that, for a given initial state $x(0) = \mathsf{init} \in X$, a fixed controller C induces a probability measure P^C_{init} over all state trajectories starting at $x(0) = \mathsf{init}$ and evolving in accordance to (34).

In order to formalize a control specification for Σ in (34), the state subsets $A, B, C, D \subseteq X$ whose shape is illustrated in Fig. 8 are considered. Given the LTL formulas over these predicates

$$\varphi_1 := \Box\left((\neg A \wedge \bigcirc A) \to (\bigcirc\bigcirc A \wedge \bigcirc\bigcirc\bigcirc A)\right), \text{ and}$$
$$\varphi_2 := (\Box\Diamond B \to \Diamond C) \wedge (\Diamond D \to \Box\neg C),$$

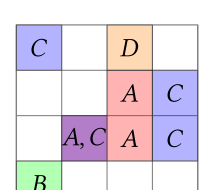

Fig. 8. Predicates over X.

the set $\mathcal{L}(\varphi_i) \subseteq 2^{\mathbb{N}\to X}$ collects all state trajectories of Σ that fulfill φ_i. With this, we define the *almost sure winning region* of Σ for the specification φ as the largest (in term of set inclusion) set of states W_{in} for which there exists a controller C s.t. $P^C_\alpha(\mathcal{L}(\varphi_i)) = 1$

Spec.	# vertices in $2^1/_2$-game abstraction	Total synthesis time			Peak memory footprint		
		Fairsyn	Fairsyn w/o accl.	SS	Fairsyn	Fairsyn w/o accl.	SS
φ_1 (1 Rabin pair)	3.8×10^3	0.02 s	0.02 s	8 s	65 MiB	65 MiB	125 MiB
	2.2×10^4	0.2 s	0.4 s	18 s	68 MiB	68 MiB	1 GiB
	1.1×10^5	1.3 s	3.7 s	9 min 18 s	79 MiB	81 MiB	80 GiB
	6.6×10^5	5.4 s	16.8 s	OoM	128 MiB	126 MiB	127 GiB
	4.3×10^6	35 s	1 min 32 s	OoM	479 MiB	478 MiB	127 GiB
φ_2 (2 Rabin pairs)	3.8×10^3	0.4 s	1 s	30 s	66 MiB	66 MiB	156 MiB
	2.2×10^4	8.2 s	41 s	55 s	72 MiB	69 MiB	1 GiB
	1.1×10^5	1 min 23 s	12 min 38 s	16 min 1 s	108 MiB	102 MiB	81 GiB
	6.6×10^5	5 min 27 s	1 h 1 min	OoM	166 MiB	237 MiB	126 GiB
	4.3×10^6	41 min 7 s	6 h 5 min	OoM	517 MiB	509 MiB	127 GiB

Table 2. Performance comparison between Fairsyn and StochasticSynthesis (abbreviated as SS) [Dutreix et al. 2020] on a comparable implementation of the abstract fair adversarial game (uniform grid-based abstraction). Col. 1 shows the specifications considered and the respective numbers of Rabin pairs, Col. 2 shows the size of the resulting $2^1/_2$-player game graph (computed using the algorithm given in [Majumdar et al. 2021], Col. 3, 4, and 5 compare the total synthesis times and Col. 6, 7, and 8 compare the peak memory footprint (as measured using the "time" command) for Fairsyn, Fairsyn w/o acceleration, and SS respectively. "OoM" stands for out-of-memory.

for every state $\alpha \in W_{\text{in}}$. The synthesis task for this case study then amounts to computing controllers C_1 and C_2 which have the *almost sure winning region* of Σ w.r.t. φ_i and W_{in} as their initial domain.

It was shown by Majumdar et al. [2021] that this synthesis problem can be approximately solved by lifting the system Σ to a finite $2^1/_2$-player game. The almost sure winning region of the resulting controller obtained by solving the abstract $2^1/_2$-player game under-approximates the almost sure winning region of Σ. We employ our fixpoint algorithm for solving this abstract $2^1/_2$-player game, which can be reduced to a fair adversarial game by following the procedure in Sec. 5. In Table 2, we compare both the accelerated and the non-accelerated versions of our fixpoint algorithm against the state-of-the-art algorithm for solving this problem, which is implemented in the tool called StochasticSynthesis (SS) [Dutreix et al. 2020].

7 CONCLUSION

Many practical problems in reactive synthesis give rise to two-player games on graphs with a winning condition of the form

$$\text{Fairness Assumption} \implies \omega\text{−regular Specification}$$

The prevalent way to solve games with fairness assumptions is to either "compile" to a new ω-regular specification for the implication or to identify selected fragments for which a "direct" symbolic algorithm has been devised. The former can handle arbitrary fairness assumptions (e.g., general Streett conditions) but yields an algorithm of high complexity (e.g., adding the number of Streett conditions in the exponent). The latter, exemplified by the GR(1) fragment, can only handle weak fairness (conjunctions of Büchi conditions). Our observation is that many practical fairness assumptions fall into the category of strong transition liveness, and for this class, one can construct a symbolic algorithm that with a slight additional penalty that is independent of the size (number of live edges) of the liveness assumption. As a byproduct, our algorithm improves the state-of-the-art in the symbolic solution of stochastic Rabin games. We experimentally demonstrate that a symbolic implementation of our algorithm based on BDDs can scale to large instances derived from deterministic and stochastic synthesis problems.

REFERENCES

Rajeev Alur, Salar Moarref, and Ufuk Topcu. 2013. Counter-strategy guided refinement of GR(1) temporal logic specifications. In *Formal Methods in Computer-Aided Design, FMCAD 2013, Portland, OR, USA, October 20-23, 2013*. IEEE, 26–33. http://ieeexplore.ieee.org/document/6679387/

Benjamin Aminof, Thomas Ball, and Orna Kupferman. 2004. Reasoning About Systems with Transition Fairness. In *Logic for Programming, Artificial Intelligence, and Reasoning, 11th International Conference, LPAR 2004, Montevideo, Uruguay, March 14-18, 2005, Proceedings (Lecture Notes in Computer Science)*, Franz Baader and Andrei Voronkov (Eds.), Vol. 3452. Springer, 194–208. https://doi.org/10.1007/978-3-540-32275-7_14

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

Julian C. Bradfield. 1998. The Modal μ-Calculus Alternation Hierarchy is Strict. *Theor. Comput. Sci.* 195, 2 (1998), 133–153. https://doi.org/10.1016/S0304-3975(97)00217-X

Romain Brenguier, Guillermo A Pérez, Jean-François Raskin, and Ocan Sankur. 2014. AbsSynthe: abstract synthesis from succinct safety specifications. *arXiv preprint arXiv:1407.5961* (2014).

J. Richard Buchi and Lawrence H. Landweber. 1969. Solving Sequential Conditions by Finite-State Strategies. *Trans. Amer. Math. Soc.* 138 (1969), 295–311.

Krishnendu Chatterjee, Luca De Alfaro, Marco Faella, Rupak Majumdar, and Vishwanath Raman. 2013. Code aware resource management. *Formal Methods in System Design* 42, 2 (2013), 146–174.

Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. 2005. The Complexity of Stochastic Rabin and Streett Games. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP) (Lecture Notes in Computer Science)*, Vol. 3580. Springer, 878–890.

Alonzo Church. 1963. Logic, arithmetic, and automata. *Proceedings of the International Congress of Mathematicians, 1962* (1963), 23–35.

Anne Condon. 1992. The complexity of stochastic games. *Information and Computation* 96, 2 (1992), 203–224.

Luca De Alfaro. 1997. *Formal verification of probabilistic systems*. Number 1601. Citeseer.

Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. 1998. Concurrent Reachability Games. In *39th Annual Symposium on Foundations of Computer Science, FOCS*. IEEE Computer Society, 564–575.

Maxence Dutreix, Jeongmin Huh, and Samuel Coogan. 2020. Abstraction-based synthesis for stochastic systems with omega-regular objectives. *arXiv preprint arXiv:2001.09236* (2020).

Rüdiger Ehlers and Vasumathi Raman. 2016. Slugs: Extensible GR(1) synthesis. In *International Conference on Computer Aided Verification*. Springer, 333–339.

E Allen Emerson and Charanjit S Jutla. 1988. The complexity of tree automata and logics of programs. In *FoCS*, Vol. 88. 328–337.

E. Allen Emerson and Charanjit S. Jutla. 1989. On Simultaneously Determinizing and Complementing omega-Automata (Extended Abstract). In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science*. IEEE Computer Society, 333–342.

E Allen Emerson and Charanjit S Jutla. 1991. Tree automata, mu-calculus and determinacy. In *FoCS*, Vol. 91. 368–377.

Cameron Finucane, Gangyuan Jing, and Hadas Kress-Gazit. 2010. LTLMoP: Experimenting with language, temporal logic and robot control. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1988–1993.

Nissim Francez. 1986. *Fairness*. Springer, Berlin.

Hubert Garavel and Nicolas Descoubes. 2003. Very large transition systems. (2003). http://tinyurl.com/yuroxx

Rob Van Glabbeek and Peter Höfner. 2019. Progress, Justness, and Fairness. *ACM Comput. Surv.* 52, 4, Article 69 (2019), 38 pages.

Yuri Gurevich and Leo Harrington. 1982. Trees, automata, and games. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*. 60–65.

Dexter Kozen. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science* 27, 3 (1983), 333 – 354. International Colloquium on Automata, Languages and Programming (ICALP).

Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. 2007. Where's waldo? sensor-based temporal logic motion planning. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 3116–3121.

Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. 2009. Temporal-logic-based reactive mission and motion planning. *IEEE transactions on robotics* 25, 6 (2009), 1370–1381.

Orna Kupferman and Moshe Y Vardi. 2005. Safraless decision procedures. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. IEEE, 531–540.

David E Long, Anca Browne, Edmund M Clarke, Somesh Jha, and Wilfredo R Marrero. 1994. An improved algorithm for the evaluation of fixpoint expressions. In *International Conference on Computer Aided Verification*. Springer, 338–350.

Rupak Majumdar, Kaushik Mallik, Anne-Kathrin Schmuck, and Sadegh Soudjani. 2021. Symbolic Qualitative Control for Stochastic Systems via Finite Parity Games. In *ADHS 2021*.

Rupak Majumdar, Nir Piterman, and Anne-Kathrin Schmuck. 2019. Environmentally-Friendly GR(1) Synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems*. Springer International Publishing, Cham, 229–246.

Oded Maler, Amir Pnueli, and Joseph Sifakis. 1995. On the synthesis of discrete controllers for timed systems. In *Annual Symposium on Theoretical Aspects of Computer Science*. Springer Berlin Heidelberg, 229–242.

Shahar Maoz and Jan Oliver Ringert. 2015. Synthesizing a Lego Forklift Controller in GR (1): A Case Study. *SYNT 2015* (2015), 5.

Thibaud Michaud and Maximilien Colange. 2018. Reactive synthesis from LTL specification with Spot. In *Proceedings of the 7th Workshop on Synthesis, SYNT@ CAV*.

Andrzej Wlodzimierz Mostowski. 1984. Regular expressions for infinite trees and a standard form of automata. In *Symposium on computation theory*. Springer, 157–168.

N. Piterman and A. Pnueli. 2006. Faster Solutions of Rabin and Streett Games. In *21st Annual IEEE Symposium on Logic in Computer Science (LICS'06)*. 275–284.

Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. 2006. Synthesis of reactive (1) designs. In *International Workshop on Verification, Model Checking, and Abstract Interpretation*. Springer, 364–380.

Amir Pnueli. 1983. On the extremely fair treatment of probabilistic algorithms. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*. 278–290.

Amir Pnueli and Roni Rosner. 1988. A Framework for the Synthesis of Reactive Modules. In *International Conference on Concurrency, Proceedings (LNCS)*, Friedrich H. Vogt (Ed.), Vol. 335. Springer, 4–17.

Amir Pnueli and Roni Rosner. 1989. On the Synthesis of a Reactive Module. In *Annual ACM Symposium on Principles of Programming Languages*. ACM Press, 179–190.

Jean-Pierre Queille and Joseph Sifakis. 1983. Fairness and related properties in transition systems–a temporal logic to deal with fairness. *Acta Informatica* 19, 3 (1983), 195–220.

Michael O Rabin. 1969. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* 141 (1969), 1–35.

Fabio Somenzi. 2019. CUDD 3.0. 0. *URL http://vlsi. colorado. edu/˜ fabio/CUDD/html/. Also available at https://github. com/ivmai/cudd* (2019).

Mária Svoreňová, Jan Křetínskỳ, Martin Chmelík, Krishnendu Chatterjee, Ivana Černá, and Calin Belta. 2017. Temporal logic control for stochastic linear systems using abstraction refinement of probabilistic games. *Nonlinear Analysis: Hybrid Systems* 23 (2017), 230–253.

Paulo Tabuada. 2009. *Verification and control of hybrid systems: a symbolic approach.* Springer Science & Business Media.

John G Thistle. 1995. On control of systems modelled as deterministic Rabin automata. *Discrete Event Dyn. Systems* 5, 4 (1995), 357–381.

John G Thistle and RP Malhamé. 1998. Control of ω-automata under state fairness assumptions. *Systems & control letters* 33, 4 (1998), 265–274.

Tom van Dijk and Jaco van de Pol. 2015. Sylvan: Multi-core decision diagrams. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 677–691.

Tichakorn Wongpiromsarn, Ufuk Topcu, Necmiye Ozay, Huan Xu, and Richard M. Murray. 2011. TuLiP: a software toolbox for receding horizon temporal logic planning. In *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011*, Marco Caccamo, Emilio Frazzoli, and Radu Grosu (Eds.). ACM, 313–314. https://doi.org/10.1145/1967701.1967747

Wieslaw Zielonka. 1998. Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees. *Theor. Comput. Sci.* 200, 1-2 (1998), 135–183.

Wieslaw Zielonka. 2004. Perfect-Information Stochastic Parity Games. In *International Conference on Foundations of Software Science and Computation Structures (FOSSACS) (LNCS)*, Vol. 2987. Springer, 499–513.

A EXAMPLE-COMPUTATION OF THE RABIN FIXED-POINT

Consider the game graph depicted in Fig. 9, where circles and squares denote Player 0 and Player 1 vertices, respectively. We are given a Rabin condition with two pairs $\mathcal{R} = \{\langle G_1, R_1\rangle, \langle G_2, R_2\rangle\}$ s.t.

$$\overline{R_1} = \{q_1, q_3, q_4, q_6, q_7\} \qquad G_1 = \{q_1, q_4\} \qquad \overline{R_2} = \{q_2, q_3, q_5, q_6\} \qquad G_2 = \{q_3\}$$

which are indicated in green and orange, respectively, in Fig. 9. The only live edge in the game graph is indicated in dashed blue from q_2 to q_3. We assert that Player 0 wins from every vertex. However, in the absence of the live edge, she wins only from $\{q_3, q_4, q_5, q_6, q_7\}$. (This is because Player 1 can force the game to stay forever in q_2 from the remaining states.)

We first flatten the algorithm in (7) for two Rabin pairs. This yields the following algorithm:

$$\nu Y_0.\ \mu X_0. \tag{35a}$$

$$\Big\{\nu Y_1.\mu X_1.\ \nu Y_2.\mu X_2. \tag{35b}$$

$$\mathrm{Apre}(Y_0, X_0)$$

$$\cup\ \Big(\overline{R}_1 \cap [(G_1 \cap \mathrm{Cpre}(Y_1)) \cup (\mathrm{Apre}(Y_1, X_1))]\Big)$$

$$\cup\ \Big(\overline{R}_1 \cap \overline{R}_2 \cap [(G_2 \cap \mathrm{Cpre}(Y_2)) \cup (\mathrm{Apre}(Y_2, X_2))]\Big)$$

$$\cup\ \nu Y_2'.\mu X_2'.\ \nu Y_1'.\mu X_1'. \tag{35c}$$

$$\mathrm{Apre}(Y_0, X_0)$$

$$\cup\ \Big(\overline{R}_2 \cap \big[(G_2 \cap \mathrm{Cpre}(Y_2')) \cup (\mathrm{Apre}(Y_2', X_2'))\big]\Big)$$

$$\cup\ \Big(\overline{R}_1 \cap \overline{R}_2 \cap \big[(G_1 \cap \mathrm{Cpre}(Y_1')) \cup (\mathrm{Apre}(Y_1', X_1'))\big]\Big)\ \Big\}$$

We first consider the upper part of (35), i.e., the permutation sequence $\delta = 012$ (labeled by (35b)). We first recall that the computation is initialized with $Y_i^0 = V$ and $X_i^0 = \emptyset$ and we see from the structure of the game graph that $\mathrm{Cpre}(V) = V$. Further, we see from the definition of Apre that $\mathrm{Apre}(\cdot, \emptyset) = \emptyset$. So, we have

$$X_2^1 = (\overline{R}_1 \cap G_1) \cup (\overline{R}_1 \cap \overline{R}_2 \cap G_2) = \{q_1, q_4\} \cup \{q_3\} = \{q_1, q_3, q_4\}.$$

As q_6 is the only other state in $\overline{R}_1 \cap \overline{R}_2 \cap G_2$ and q_6 does not have an edge to $\{q_1, q_3, q_4\}$ the iteration over X_2 terminates and we get $Y_2^1 = \{q_1, q_3, q_4\}$. As $q_3 \notin \mathrm{Cpre}(Y_2^1)$ the last line of the upper part

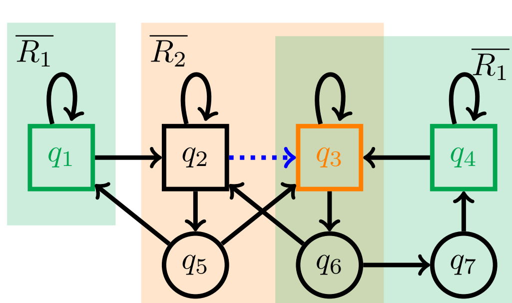

Fig. 9. Example of a fair adversarial Rabin game with two pairs $\langle G_1, R_1\rangle = \langle\{q_1, q_4\}, \{q_2, q_5\}\rangle$ (G_1 and $\overline{R_1}$ are indicated in green) and $\langle G_2, R_2\rangle = \langle\{q_3\}, \{q_1, q_4, q_7\}\rangle$ (G_2 and $\overline{R_2}$ are indicated in orange), and one live edge $E^\ell = \{(q_2, q_3)\}$ (dashed blue).

of (35) becomes the empty set and we terminate with $Y_2^* = X_2^* = (\overline{R}_1 \cap G_1) = \{q_1, q_4\}$. This gives $X_1^1 = \{q_1, q_4\}$ and resets Y_2 and X_2 to V and $\emptyset$, respectively. Therefore, we now get

$$X_2^1 = (\overline{R}_1 \cap G_1) \cup \mathrm{Apre}(Q, X_1^1) \cup (\overline{R}_1 \cap \overline{R}_2 \cap G_2) = \{q_1, q_4\} \cup \{q_7\} \cup \{q_3\}.$$

Now, as $q_7 \in X_2^1$, also q_6 is added before X_2 terminates. This now gives $Y_2^1 = \{q_1, q_3, q_4, q_6, q_7\}$ and hence $q_3 \in \mathrm{Cpre}(Y_2^1)$. As there are no other states in $\overline{R}_1 \cap \overline{R}_2 \cap G_2$ that can be added to this set, the iteration over X_2 terminates and we get $Y_2^2 = \{q_1, q_3, q_4, q_6, q_7\}$, which also terminates the iteration over Y_2, resulting in $X_1^2 = \{q_1, q_3, q_4, q_6, q_7\}$. As there are again no other states inside $\overline{R}_1$ that could be added, this iteration over X_1 terminates, giving $Y_1^1 = \{q_1, q_3, q_4, q_6, q_7\}$. Now we see that $\mathrm{Cpre}(Y_1^1) = \{q_3, q_4, q_6, q_7\}$. As the exclusion of q_1 from Y_1 does not influence the reasoning about $\{q_3, q_4, q_6, q_7\}$ the iteration terminates with $Y_1^* = \{q_3, q_4, q_6, q_7\}$.

Now we consider the lower part of (35), i.e., the permutation sequence $\delta = 021$ (labeled by (35c)). Here, we get

$$X_1'^1 = (\overline{R}_2 \cap G_2) \cup (\overline{R}_1 \cap \overline{R}_2 \cap G_1) = \{q_3\} \cup \emptyset = \{q_3\}.$$

For the same reason as before we see again that the last line of the lower part of (35) becomes the empty set and we terminate with $Y_1'^* = X_1'^* = (\overline{R}_2 \cap G_2) = \{q_3\}$. This gives $X_2'^1 = \{q_3\}$ and resets Y_1' and X_1' to V and $\emptyset$, respectively. With this, we now get

$$X_1'^2 = (\overline{R}_2 \cap G_2) \cup \mathrm{Apre}(Q, X_2'^1) \cup (\overline{R}_1 \cap \overline{R}_2 \cap G_1) = \{q_3\} \cup \{q_2, q_5\} \cup \emptyset.$$

Here, for the first time, the live edge from q_2 to q_3 comes into play. If this would not be a live edge, q_2 would not be added to X_1', as in this case the environment could trap the game in q_2, and thereby prevent the second Rabin pair to hold. However, due to the edge from q_2 to q_3 being live, we know that the environment will always eventually transition from q_2 to q_3. With this, now also q_6 is added to X_1', finally leading to a termination of the iteration over X_2' with $\{q_2, q_3, q_5, q_6\}$ and hence $Y_2'^1 = \{q_2, q_3, q_5, q_6\}$. As $q_3 \in \mathrm{Cpre}(Y_2'^1)$ the iteration over Y_2' terminates with $Y_2'^* = \{q_2, q_3, q_5, q_6\}$.

With both the upper and the lower part of (35) terminated, we can now take the union of $Y_1^* = \{q_3, q_4, q_6, q_7\}$ and $Y_2'^* = \{q_2, q_3, q_5, q_6\}$ to get $X_0^1 = \{q_2 \dots q_7\}$ (reaching the part of the formula labeled with (35a)). After this update of X_0 all inner fixpoint variables (in (35b) and (35c)) are reset, and the upper and lower expressions in (35) are re-evaluated. As $\mathrm{Apre}(Q, X_0^1) = \{q_2 \dots q_7\}$, we see that every iteration over X_i in (35b) and (35c) is essentially initialized with a set containing $\{q_2 \dots q_7\}$. This implies that q_1 will actually remain within Y_1, leading to $Y_1^* = V$, and with this $X_0^2 = V$. As this implies $Y_0^1 = V = Y_0^0$, the computation terminates with $Z^* = V$.

Despite all states being winning, we see that Player 0 has to play appropriately to enforce winning. Intuitively, from state q_5 she must go to q_3 and from q_6 she has to consistently either (i) always go to q_2 or (ii) always go to q_7. If she picks option (i), the play is won by satisfying the second Rabin pair, i.e., always eventually visiting q_3 while remaining within $\overline{R_2}$. If she picks option (ii), it is up to the environment whether the game is won by satisfying the first or the second Rabin pair. Intuitively, if the environment plays such that either (a) the game eventually remains in q_4 or (b) the edges (q_4, q_3) and (q_3, q_6) are taken infinitely often, the game fulfills the first Rabin condition. If, however, (c), the environment decides to trap the game in q_3, the game is won by satisfying the second Rabin pair. This influence of the environment on the selection of the satisfied Rabin pair intuitively requires the evaluation of all possible permutation sequences in the evaluation of the fixpoint algorithm. We will see later that for Rabin pairs which are ordered by inclusion (corresponding to the special case of a Rabin-chain condition), no permutation is required.

We comment that the strategy construction outlined in Thm. B.7 provided in App. B.3 chooses to enforce a transition from q_6 to q_7 (see Example B.8 in App. B.3 for a detailed discussion).

B DETAILED PROOFS

CONTENTS

B.1 General Lemmas

We first introduce some useful general lemmas.

Lemma B.1. *If* $Y \supseteq X$ *then* $\mathrm{Cpre}(Y) \cup \mathrm{Apre}(Y, X) = \mathrm{Cpre}(Y)$.

Proof. The claim follows from the following derivation

$$\begin{aligned}
\mathrm{Cpre}(Y) \cup \mathrm{Apre}(Y, X) &= \mathrm{Cpre}(Y) \cup \mathrm{Cpre}(X) \cup \left(\mathrm{Lpre}^{\exists}(X) \cap \mathrm{Pre}_1^{\forall}(Y)\right) \\
&= \mathrm{Cpre}(Y) \cup \left(\mathrm{Lpre}^{\exists}(X) \cap \mathrm{Pre}_1^{\forall}(Y)\right) \\
&= \left(\mathrm{Cpre}(Y) \cup \mathrm{Lpre}^{\exists}(X)\right) \cap \left(\mathrm{Cpre}(Y) \cup \mathrm{Pre}_1^{\forall}(Y)\right) \\
&= \left(\mathrm{Cpre}(Y) \cup \mathrm{Lpre}^{\exists}(X)\right) \cap \mathrm{Cpre}(Y) \\
&= \mathrm{Cpre}(Y)
\end{aligned}$$

where the second line follows from $\mathrm{Cpre}(X) \subseteq \mathrm{Cpre}(Y)$ (as $X \subseteq Y$) and the forth line follows as $\mathrm{Cpre}(Y) = \mathrm{Pre}_0^{\exists}(Y) \cup \mathrm{Pre}_1^{\forall}(Y) \supseteq \mathrm{Pre}_1^{\forall}(Y)$. □

Lemma B.2. *If* $Y \subseteq X$ *then* $\mathrm{Apre}(Y, X) = \mathrm{Cpre}(X)$.

Proof. The claim follows from the following derivation

$$\begin{aligned}\mathrm{Apre}(Y,X) &= \mathrm{Cpre}(X) \cup \Big(\mathrm{Lpre}^{\exists}(X) \cap \mathrm{Pre}_1^{\forall}(Y)\Big)\\ &= \Big(\mathrm{Cpre}(X) \cup \mathrm{Lpre}^{\exists}(X)\Big) \cap \Big(\mathrm{Cpre}(X) \cup \mathrm{Pre}_1^{\forall}(Y)\Big)\\ &= \Big(\mathrm{Cpre}(X) \cup \mathrm{Lpre}^{\exists}(X)\Big) \cap \mathrm{Cpre}(X)\\ &= \mathrm{Cpre}(X)\end{aligned}$$

where the fourth line follows as $\mathrm{Cpre}(X) = \mathrm{Pre}_0^{\exists}(X) \cup \mathrm{Pre}_1^{\forall}(X) \supseteq \mathrm{Pre}_1^{\forall}(Y)$ as $Y \subseteq X$. □

Lemma B.3. *Let $f(X,Y)$, $g(X,Y)$, $h_a(X,Y)$ and $h_b(X,Y)$ be functions which are monotone in both $X \subseteq V$ and $Y \subseteq V$. Further, let*

$$\begin{aligned} Z &:= \nu Y_a.\ \mu X_a.\ \nu Y_b.\ \mu X_b.\ (h_a(X_a,Y_a) \cup f(X_a,Y_a)) \cup (h_b(X_b,Y_b) \cup g(X_b,Y_b))\\ \widetilde{Z} &:= \nu \widetilde{Y}_a.\ \mu \widetilde{X}_a.\ \nu \widetilde{Y}_b.\ \mu \widetilde{X}_b.\ \Big(h_a(\widetilde{X}_a,\widetilde{Y}_a) \cup f(\widetilde{X}_a,\widetilde{Y}_a)\Big) \cup g(\widetilde{X}_b,\widetilde{Y}_b)\\ \check{Z} &:= \nu \check{Y}_a.\ \mu \check{X}_a.\ \nu \check{Y}_b.\ \mu \check{X}_b.\ f(\check{X}_a,\check{Y}_a) \cup \Big(h_b(\check{X}_b,\check{Y}_b) \cup g(\check{X}_b,\check{Y}_b)\Big)\end{aligned}$$

Then
(i) $Z = \widetilde{Z}$ if $h_b(X,Y) \subseteq h_a(X,Y)$ for all $X, Y \subseteq V$,
(ii) $Z = \check{Z}$ if $h_a(X,Y) \subseteq h_b(X,Y)$ for all $X, Y \subseteq V$, and
(iii) $Z = \widetilde{Z} = \check{Z}$ if $h_a(X,Y) = h_b(X,Y)$ for all $X, Y \subseteq V$.

Proof. We first observe that (iii) is a direct consequence of (i) and (ii). We prove (i) in step 1-2 and (ii) in step 3-4 below.

Within these proofs we denote by Y_a^l the set computed in the l-th iteration over the fixpoint variable Y_a, where $Y_a^0 = V$. Further, we denote by X_a^{lj} the set computed in the j-th iteration over the fixpoint variable X_a during the computation of Y_a^l where $X_a^{l0} = \emptyset$. In a very similar way Y_b^{ljm} and X_b^{ljmi} are defined, as well as their tilded and checked versions in (ii) and (iii).

► **Step 1:** First we show that if $h_b(X,Y) \subseteq h_a(X,Y)$ it holds for all $l > 0$ that

$$Y_a^{l+1} = h_a(Y_a^{l+1}, Y_a^l) \cup f(Y_a^{l+1}, Y_a^l) \cup g(Y_a^{l+1}, Y_a^{l+1}). \tag{36}$$

For this purpose observe that

$$X_b^{ljm1} = h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup h_b(\emptyset, Y_b^{ljm}) \cup g(\emptyset, Y_b^{ljm}).$$

Now recall that $X_b^{ljm0} = \emptyset \subseteq X_b^{ljm1}$. We can generalize this inclusion to arbitrary $i > 1$ by utilizing the monotonicity of f, g, h to observe the following derivation:

$$\begin{aligned} X_b^{ljmi+1} =& h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup h_b(X_b^{ljmi}, Y_b^{ljm}) \cup g(X_b^{ljmi}, Y_b^{ljm})\\ =& h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup h_b(X_b^{ljmi-1}, Y_b^{ljm}) \cup g(X_b^{ljmi-1}, Y_b^{ljm}) \cup h_b(X_b^{ljmi}, Y_b^{ljm}) \cup g(X_b^{ljmi}, Y_b^{ljm})\\ =& X_b^{ljmi} \cup h_b(X_b^{ljmi}, Y_b^{ljm}) \cup g(X_b^{ljmi}, Y_b^{ljm})\end{aligned}$$

obviously implying $X_b^{ljmi} \subseteq X_b^{ljmi+1}$ for all $i \geq 0$.

With this we have

$$Y_b^{ljm+1} = \bigcup_{i \geq 0} X_b^{ljmi} = X_b^{ljm*} = h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup h_b(X_b^{ljm*}, Y_b^{ljm}) \cup g(X_b^{ljm*}, Y_b^{ljm}). \tag{37}$$

Here, $X_b^{ljm*} := X_b^{ljmi^{\uparrow}}$ where $i^{\uparrow}$ is the iteration in which the fixed point over X_b^{ljmi} is attained.

Now recall that $Y_b^{lj0} = V$ and therefore $Y_b^{lj1} \subseteq Y_b^{lj0}$. Hence, we can assume that $Y_b^{ljm} \subseteq Y_b^{ljm-1}$ to see that $X_b^{ljm1} \subseteq X_b^{ljm-11}$ and therefore, subsequently, that $Y_b^{ljm+1} \subseteq Y_b^{ljm}$ (again due to the monotonicity of h, f, g). With this we see that

$$X_a^{lj+1} = \bigcap_{m \geq 0} Y_b^{ljm} = Y_b^{lj*} = h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup \left(h_b(Y_b^{lj*}, Y_b^{lj*}) \cup g(Y_b^{lj*}, Y_b^{lj*})\right).$$

Again, $Y_b^{lj*} = Y_b^{ljm^\uparrow}$ where $m^\uparrow$ is the iteration in which the fixed point over Y_b^{ljm} is attained. As we know that $Y_b^{ljm^\uparrow} = X_b^{ljm^\uparrow *}$ we can replace $X_b^{ljm^\uparrow *}$ with Y_b^{lj*} above. With this, we proved that

$$X_a^{lj+1} = h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup h_b(X_a^{lj+1}, X_a^{lj+1}) \cup g(X_a^{lj+1}, X_a^{lj+1}) \tag{38}$$

for any $j \geq 0$. Using the same monotonicity argument as before we again get $X_a^{lj+1} \supseteq X_a^{lj}$ and therefore

$$Y_a^{l+1} = \bigcup_{j>0} X_a^{lj+1} = X_a^{l*} = h_a(Y_a^{l+1}, Y_a^l) \cup f(Y_a^{l+1}, Y_a^l) \cup h_b(Y_a^{l+1}, Y_a^{l+1}) \cup g(Y_a^{l+1}, Y_a^{l+1}). \tag{39}$$

Utilizing the monotonicity argument one more time we get that $Y_a^l \supseteq Y_a^{l+1}$. With this it follows from the monotonicity of h_b that

$$h_b(Y_a^{l+1}, Y_a^{l+1}) \subseteq h_b(Y_a^{l+1}, Y_a^l) \subseteq h_a(Y_a^{l+1}, Y_a^l)$$

and therefore

$$Y_a^{l+1} = h_a(Y_a^{l+1}, Y_a^l) \cup f(Y_a^{l+1}, Y_a^l) \cup g(Y_a^{l+1}, Y_a^{l+1}), \tag{40}$$

which proves the claim.

► **Step 2:** By utilizing (40), we now show that $Z = \widetilde{Z}$.
As $\widetilde{Z} = Y_b^*$, it follows from the fixpoint equation defining $\widetilde{Z}$ that $\widetilde{Z}$ is the unique *largest* set of states s.t. $\widetilde{Y}_a^l = \widetilde{Y}_a^{l+1}$ and

$$\widetilde{Z} = h(\widetilde{Z}, \widetilde{Z}) \cup f(\widetilde{Z}, \widetilde{Z}) \cup g(\widetilde{Z}, \widetilde{Z}). \tag{41}$$

It now follows from (40) and the fact that $Z = Y_a^*$, that Z is also the unique *largest* set s.t. $Y_a^l = Y_a^{l+1}$ and thereby fulfills equation (41). Hence Z and $\widetilde{Z}$ must be equivalent.

► **Step 3:** We now show that if $h_a(X, Y) \subseteq h_b(X, Y)$, it holds for $l = l^\uparrow$, the corresponding $j = j^\uparrow$ and any $m \geq 0$ that

$$Y_b^{l^\uparrow j^\uparrow m+1} = f(X_a^{l^\uparrow j^\uparrow}, Y_a^{l^\uparrow}) \cup h_b(Y_b^{l^\uparrow j^\uparrow m+1}, Y_b^{l^\uparrow j^\uparrow m}) \cup g(Y_b^{l^\uparrow j^\uparrow m+1}, Y_b^{l^\uparrow j^\uparrow m}). \tag{42}$$

First observe that for arbitrary $l, j, m, i \geq 0$ it holds that

$$X_b^{ljmi+1} = h_a(X_a^{lj}, Y_a^l) \cup f(X_a^{lj}, Y_a^l) \cup h_b(X_b^{ljmi}, Y_b^{ljm}) \cup g(X_b^{ljmi}, Y_b^{ljm}).$$

When re-initializing the inner FP, we have $X_a^{lj} \supseteq X_b^{l000} = \emptyset$ and $Y_a^l \subseteq Y_b^{lj0} = V$. Hence, the two h-terms are incomparable. However, we see that whenever $X_b^{ljmi} \supseteq X_a^{lj}$ (while still $Y_a^l \subseteq Y_b^{lj0}$) we have

$$h_a(X_a^{lj}, Y_a^l) \subseteq h_b(X_a^{lj}, Y_a^l) \subseteq h_b(X_b^{ljmi}, Y_b^{ljm})$$

and we get

$$X_b^{ljmi+1} = f(X_a^{lj}, Y_a^l) \cup h_b(X_b^{ljmi}, Y_b^{ljm}) \cup g(X_b^{ljmi}, Y_b^{ljm}).$$

Now we know from the structure of this fixed point that we keep increasing X_b until

$$X_b^{ljm*} = f(X_a^{lj}, Y_a^l) \cup h_b(X_b^{ljm*}, Y_b^{ljm}) \cup g(X_b^{ljm*}, Y_b^{ljm}),$$

where $Y_b^{ljm+1} = X_b^{ljm*} \subseteq Y_b^{ljm}$. Further, we know that the fixed point is attained over Y_b if equality holds. It remains to show that for $l = l^{\uparrow}$ and $j = j^{\uparrow}$ the set Y_b will never get smaller then Y_a (as this would render the two h terms incomparable again). I.e., we need to show that for all m we have $Y_a^{l^{\uparrow}} \subseteq Y_b^{l^{\uparrow}j^{\uparrow}m}$. To see this, recall that l and j are such that the fixed point over X_a and Y_a is already attained. That is, we know that $X_a^{l^{\uparrow}j^{\uparrow}+1} = X_a^{l^{\uparrow}j^{\uparrow}}$. It further follows from the structure of the fixed point that $Y_b^{l^{\uparrow}j^{\uparrow}*} = X_a^{l^{\uparrow}j^{\uparrow}+1} = X_a^{l^{\uparrow}j^{\uparrow}}$. With this, it follows from the monotonicity of the fixed point that $Y_a^{l^{\uparrow}} \subseteq Y_b^{l^{\uparrow}j^{\uparrow}m}$ for every $m \geq 0$.

► **Step 4:** By utilizing (42), we now show that $Z = \check{Z}$.

It immediately follows from the structure of the fixpoint equation defining $\check{Z}$ that we similarly have

$$\check{Y}_b^{l^{\uparrow}j^{\uparrow}m+1} = f(\check{X}_a^{l^{\uparrow}j^{\uparrow}}, \check{Y}_a^{l^{\uparrow}}) \cup h(\check{Y}_b^{l^{\uparrow}j^{\uparrow}m+1}, \check{Y}_b^{l^{\uparrow}j^{\uparrow}m}) \cup g(\check{Y}_b^{l^{\uparrow}j^{\uparrow}m+1}, \check{Y}_b^{l^{\uparrow}j^{\uparrow}m}). \tag{43}$$

By further observing that $Z = Y_a^{l^{\uparrow}} = X_a^{l^{\uparrow}j^{\uparrow}}$ and $\check{Z} = \check{Y}_a^{l^{\uparrow}} = \check{X}_a^{l^{\uparrow}j^{\uparrow}}$ we see that both Z and $\check{Z}$ are the unique largest sets s.t. the inner fixpoint computations over X_b and Y_b (resp. $\check{X}_b$ and $\check{Y}_b$) converge to Z (resp. $\check{Z}$) via the same fix-point equation in (42) (resp. (43)). This proves that $Z = \check{Z}$. □

Lemma B.4. *Let $f(X,Y)$ and $g(X,Y)$ be two functions which are monotone in both $X \subseteq V$ and $Y \subseteq V$. Further, let*

$$\begin{aligned} Z_a &:= \nu Y_a.\ \mu X_a.\ \nu Y_b.\ \mu X_b.\ f(X_a, Y_a) \cup g(X_b, Y_b) \\ Z_b &:= \nu Y_a.\ \mu X_a.\ \nu Y_b.\ \mu X_b.\ g(X_a, Y_a) \cup f(X_b, Y_b) \\ Z_c &:= \nu Y_c.\ \mu X_c.\ f(X_c, Y_c) \end{aligned}$$

Then it holds that
(i) $Z_c \subseteq Z_a$ and
(ii) $Z_c \subseteq Z_b$.
If, in addition, $g(X,Y) \subseteq f(X,Y)$ for all $X, Y \subseteq V$, then it holds that
(iii) $Z_a = Z_c$ and
(iv) $Z_b = Z_c$.

Proof. We prove all claims separately:

► *(i)* "$Z_c \subseteq Z_a$" : First, consider a stage of the fixed point evaluation where Y_a and X_a have their initialization value $Y_a^0 := V$ and $X_a^{00} := \emptyset$ (here, the notation X_a^{lk} refers to the value of X_a computed in the k'th iteration over X_a using the value for Y_a computed in the l'th iteration over Y_a). Then we see that $X_a^{01} = Y_b^{00*}$ where $Y_b^{00*} = f(\emptyset, V) \cup g(Y_b^{00*}, Y_b^{00*})$. We therefore see that $X_a^{01} \supseteq X_c^{01} = f(\emptyset, V)$. With this, it follows from the monotonicity of f and g that $Y_a^{01} = X_a^{0*} \supseteq X_c^{0*} = Y_c^1$. With this, we see that $X_a^{m1} \supseteq X_c^{m1}$ for all $m > 0$ and therefore $Z_a = Y_a^* \supseteq Y_c^* = Z_c$.

► *(ii)* "$Z_c \subseteq Z_b$" : Consider arbitrary values Y_a^m and X_a^{mn} and assume that Y_b and X_b have their initialisation value, i.e., $Y_b^{mn0} := V$ and $X_b^{mn00} := \emptyset$. Then we have

$$X_b^{mn01} = g(X_a^{mn}, Y_a^m) \cup f(\emptyset, V) \supseteq X_c^{01}.$$

Using the same reasoning as in the previous part, we see that this implies $Y_b^{mn*} \supseteq Y_c^* = Z_c$. As this holds for any m and n it also holds when the fixed point over Y_a and X_a is obtained, i.e., when we have $Z_a = Y_a^* = Y_b^{***}$, which proves the statement.

► *(iii)-(iv)* This is a simple consequence of Lem. B.3 (iii). It follows by choosing $f(X,Y) = \emptyset$ and $g(X,Y) = \emptyset$ in Lem. B.3 and interpreting h_a as f and h_b as g to show (iii) and h_a as g and h_b as f to show (iv). □

B.2 Additional Proofs for Sec. 3

B.2.1 Proof of Thm. 3.3.

Theorem (Thm. 3.3 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\langle T, Q \rangle$ be a safe reachability winning condition. Further, let*

$$Z^* := \nu Y.\ \mu X.\ T \cup (Q \cap \mathrm{Apre}(Y, X)). \tag{44}$$

Then Z^* is equivalent to the winning region of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition ψ in (11). Moreover, the fixpoint algorithm runs in $O(n^2)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

We denote by Y^m the m-th iteration over the fixpoint variable Y in (44), where $Y^0 = V$. Further, we denote by X^{mi} the set computed in the i-th iteration over the fixpoint variable X in (44) during the computation of Y^m where $X^{m0} = \emptyset$. Then it follows form (44) that

$$X^{m1} = X^{m0} \cup T \cup (Q \cap \mathrm{Apre}(Y^{m-1}, X^{m0})) = \emptyset \cup T \cup (Q \cap \mathrm{Apre}(Y^m, \emptyset)) = T,$$
$$X^{m2} = X^{m1} \cup T \cup (Q \cap \mathrm{Apre}(Y^{m-1}, X^{m1})) = T \cup (Q \cap \mathrm{Apre}(Y^{m-1}, X^{m1})) \supseteq X^{m1},$$

and therefore, in general,

$$X^{mi+1} = T \cup (Q \cap \mathrm{Apre}(Y^{m-1}, X^{mi})) \supseteq X^{mi}.$$

With this, the fixed point over X corresponds to the set $X^{m*} = \bigcup_{i>0} X^{mi} = X^{mi^\uparrow}$, where $i^\uparrow$ is the iteration where the fixed point over X^{mi} is attained.

Now consider the computation of Y. Here we have $Y^0 = V$ and $Y^m = Y^{m-1} \cap X^{m*} \subseteq Y^{m-1}$ where equality holds when a fixed point is reached. Hence, in particular we have $Y^* = X^{**} = Z^*$. For simplicity we denote X^{*i} by X^i.

Strategy construction. In order to construct a winning strategy for Player 0 from (44), we construct a ranking over V by choosing

$$\mathrm{rank}(v) = i \ \Leftrightarrow\ v \in X^i \setminus X^{i-1} \quad \text{and} \quad \mathrm{rank}(v) = \infty \ \Leftrightarrow v \notin Z^*. \tag{45}$$

As $X^0 = \emptyset$, $X^1 = T$ (from above) and $Z^* = \bigcup_{i>0} X^i$, it follows that $\mathrm{rank}(v) = 1$ iff $v \in T$ and $1 < \mathrm{rank}(v) < \infty$ iff $v \in Z^* \setminus T$. Using this ranking we define a Player 0 strategy $\rho_0 : V_0 \to V$ s.t.

$$\rho_0(v) = \min_{(v,w) \in E} \mathrm{rank}(w)\,. \tag{46}$$

We next show that this player 0 strategy is actually winning w.r.t. ψ (in (11)) in every fair adversarial play over $\mathcal{G}^\ell$.

Soundness. To prove soundness, we need to show $Z^* \subseteq \mathcal{W}$. That is, we need to show that for all $v \in Z^*$ there exists a strategy for player 0 s.t. the goal set T is eventually reached along all live compliant plays π starting at v while staying in Q. We choose ρ_0 in (46) and show that the claim holds.

First, it follows from the definition of Apre that for a vertex $v \in Z^*$ exactly one of the following cases holds:
(a) $v \in T$ and hence $\mathrm{rank}(v) = 1$,
(b) $v \in (V_0 \cap Z^*) \setminus T$, i.e., $1 < \mathrm{rank}(v) < \infty$ and $v \in Q$ and there exists a $v' \in E(v)$ with $\mathrm{rank}(v') < \mathrm{rank}(v)$,
(c) $v \in ((V_1 \setminus V^\ell) \cap Z^*)) \setminus T$, i.e., $1 < \mathrm{rank}(v) < \infty$ and $v \in Q$ and for all $v' \in E(v)$ it holds that $\mathrm{rank}(v') < \mathrm{rank}(v)$, or
(ℓ) $v \in (V^\ell \cap Z^*) \setminus T$, i.e., $1 < \mathrm{rank}(v) < \infty$ and $v \in Q$ and there exists a $v' \in E^\ell(v)$ with $\mathrm{rank}(v') < \mathrm{rank}(v)$ *and* $E(v) \subseteq Z^*$.

We see that $\rho_0(v)$ chooses one existentially quantified edge in (b) vertices. In all other cases player 1 chooses the successor.

Further, we see that any play π which starts in $\pi(0) = v \in Z^*$ and obeys ρ_0 has the property that $\pi(k) \in Z^* \setminus T$ implies $\pi(k) \in Q$ and $\pi(k+1) \in Z^*$ for all $k \geq 0$. This, in turn, means that for any such state $v = \pi(k) \in Z^* \setminus T$ as well as for its successor $\pi(k+1)$ a rank is defined, i.e., $\pi(k) \in X^i$ for some $0 < i < \infty$ and exactly one of the cases (b)-(ℓ) applies. We call a vertex for which case (α) applies, an (α) vertex.

Now observe that the above reasoning implies that whenever an (a) vertex is hit along a play π the claim holds. We therefore need to show that any play starting in $v \in Z^*$ eventually reaches an (a) vertex. First, consider a play in which no (ℓ) vertex occurs. Then constantly hitting (b) and (c) vertices always reduces the rank of visited states (as we assume that π obays ρ_0 in (46)). As the maximal rank is finite, we see that we must eventually hit a state with rank 1, which is an (a) state.

Note that the same argument holds when only a finite number of (ℓ) vertices is visited along π. In this case we know that from some time onward no more (ℓ) vertex occurs. As the last (ℓ) vertex has a finite rank, there can only be a finite sequence of (b) and (c) vertices afterwards until finally an (a) vertex is reached.

We are therefore left with showing that on every path with an infinite number of (ℓ) vertices, eventually an (a) vertex will be reached. We prove this claim by contradiction. I.e., we show that there cannot exist a path with infinitely many (ℓ) vertices and no (a) vertex.

We first show that infinitely many (ℓ) vertices and no (a) vertices in π imply that vertices with rank 2 can only occur finitely often along π.

► Recall that the construction of ρ_0 ensures that whenever we visit a state $v \in V_0 \cap Z^*$ with $\mathrm{rank}(v) = 2$ we will surely visit a state with rank 1 afterwards, implying the occurrence of a vertex labeled (a). As no (a) labeled vertices are assumed to occur along π, no (b) vertices with $\mathrm{rank}(v) = 2$ occur along π.

► Now assume that $v \in V_1 \cap Z^*$ with $\mathrm{rank}(v) = 2$. If v is a (c) vertex all successor states will have rank 1. With the same reasoning as before, this cannot occur.

► Now assume that $v \in V_1 \cap Z^*$ with $\mathrm{rank}(v) = 2$ is labeled with (ℓ). In this case there surely exists a successor v' of v s.t. $(v, v') \in E^\ell$ and $\mathrm{rank}(v') = 1$. But there *might* also exist another successor v'' of v (i.e., ($v'' \in E(v)$) s.t. $\mathrm{rank}(v'') > 1$. If there does not exists such a successor v'', all successors have rank 1 and we again cannot visit v.

► Now assume that $v \in V_1 \cap Z^*$ with $\mathrm{rank}(v) = 2$, labeled with (ℓ) and there exists a successor $v'' \in E(v)$ s.t. $\mathrm{rank}(v'') > 1$. Now let us assume that such a state v is visited infinitely often along π. As π is a fair adversarial play over G we know that visiting v infinitely often along π implies that v' with $(v, v') \in E^\ell$ and $\mathrm{rank}(v') = 1$ (which surely exists by the definition of Apre) will also be visited infinitely often along π. This is again a contradiction to the above hypothesis and implies that such v's can only be visited finitely often.

► As V is a finite set, the set of states with rank 2 is finite. Hence, the occurrence of infinitely many states with rank 2 along π implies that one of the above cases must occur infinitely often, which gives a contradiction to the above hypothesis. Using the same arguments, we can inductively show that states with any fixed rank can only occur finitely often if states with rank 1 (i.e., (a)-labeled vertices) never occur. As the maximal rank is finite (due to the finiteness of V) this contradicts the assumption that π is an infinite play.

We therefore conclude that along any *infinite* fair adversarial play π with infinitely many vertices labeled by (ℓ) we will eventually see a vertex labeled by (a).

Completeness. We now show that the fixpoint in (44) is complete, i.e., that every state in $\overline{Z}^* := V \setminus Z^*$ is loosing for Player 0. In particular, we show that from every vertex $v \in \overline{Z}^*$ Player 1 has a memoryless strategy ρ_1 s.t. all fair adversarial plays compliant with ρ_1 satisfy

$$\overline{\psi} := \neg\psi = \neg(Q\mathcal{U}T) = \Box\neg T \vee \neg T\mathcal{U}\neg Q \tag{47}$$

and are hence loosing for Player 0.

In order to prove the latter claim we fist compute $\overline{Z}^* := V \setminus Z^*$ by negating the fixed-point formula in (44). For this, we define $\overline{X}^* := V \setminus X$, $\overline{Y}^* := V \setminus Y$ and use the negation rule of the μ-calculus, i.e., $\neg(\mu X.f(X)) = \nu\overline{X}.V \setminus f(X)$ along with common De-Morgan laws. This results in the following derivation.

$$\overline{Z}^* = \mu\overline{Y}.\ \nu\overline{X}.\ \overline{T} \cap (\overline{Q} \cup V \setminus \mathrm{Apre}(Y, X))$$

where

$$\begin{aligned}
&V \setminus \mathrm{Apre}(Y, X)\\
&= V \setminus \left[\mathrm{Cpre}(X) \cup \left(\mathrm{Lpre}^{\exists}(X) \cap \mathrm{Pre}_1^{\forall}(Y)\right)\right]\\
&= [V \setminus \mathrm{Cpre}(X)] \cap \left[V \setminus \left(\mathrm{Lpre}^{\exists}(X) \cap \mathrm{Pre}_1^{\forall}(Y)\right)\right]\\
&= \left[\mathrm{Pre}_1^{\exists}(\overline{X}) \cup \mathrm{Pre}_0^{\forall}(\overline{X})\right] \cap \left[V_0 \cup (V_1 \setminus V^{\ell}) \cup \left(V^{\ell} \setminus \left(\mathrm{Lpre}^{\exists}(X) \cap \mathrm{Pre}_{\ell}^{\forall}(Y)\right)\right)\right]\\
&= \left[\mathrm{Pre}_1^{\exists}(\overline{X}) \cup \mathrm{Pre}_0^{\forall}(\overline{X})\right] \cap \left[V_0 \cup (V_1 \setminus V^{\ell}) \cup \left(\mathrm{Lpre}^{\forall}(\overline{X}) \cup \mathrm{Pre}_{\ell}^{\exists}(\overline{Y})\right)\right]\\
&= \mathrm{Pre}_0^{\forall}(\overline{X}) \cup \mathrm{Pre}_{1\setminus\ell}^{\exists}(\overline{X}) \cup \left[\mathrm{Pre}_1^{\exists}(\overline{X}) \cap \left(\mathrm{Lpre}^{\forall}(\overline{X}) \cup \mathrm{Pre}_{\ell}^{\exists}(\overline{Y})\right)\right]\\
&= \mathrm{Pre}_0^{\forall}(\overline{X}) \cup \mathrm{Pre}_{1\setminus\ell}^{\exists}(\overline{X}) \cup \left[\mathrm{Pre}_{\ell}^{\exists}(\overline{X}) \cap \left(\mathrm{Lpre}^{\forall}(\overline{X}) \cup \mathrm{Pre}_{\ell}^{\exists}(\overline{Y})\right)\right]\\
&= \mathrm{Pre}_0^{\forall}(\overline{X}) \cup \mathrm{Pre}_{1\setminus\ell}^{\exists}(\overline{X}) \cup \mathrm{Lpre}^{\forall}(\overline{X}) \cup \mathrm{Pre}_{\ell}^{\exists}(\overline{Y}).
\end{aligned}$$

The last line in the above derivation follows from the observation that $\mathrm{Lpre}^{\forall}(\overline{X}) \subseteq \mathrm{Pre}_l^{\exists}(\overline{X})$ and $\overline{Y} \subseteq \overline{X}$ for all iterations of the fixed-point. The additionally introduced pre-operators are defined in close analogy to (4) and (5) as follows:

$$\begin{aligned}
\mathrm{Pre}_1^{\exists}(S) &:= \{v \in V_1 \mid E(v) \cap S \neq \emptyset\},\\
\mathrm{Pre}_0^{\forall}(S) &:= \{v \in V_0 \mid E(v) \subseteq S\},\\
\mathrm{Pre}_{1\setminus\ell}^{\exists}(S) &:= \{v \in V_1 \setminus V^{\ell} \mid E(v) \cap S \neq \emptyset\},\\
\mathrm{Pre}_{\ell}^{\exists}(S) &:= \{v \in V^{\ell} \mid E(v) \cap S \neq \emptyset\},\\
\mathrm{Pre}_{\ell}^{\forall}(S) &:= \{v \in V^{\ell} \mid E(v) \subseteq S\},\\
\mathrm{Lpre}^{\forall}(S) &:= \{v \in V^{\ell} \mid E^{\ell}(v) \subseteq S\}.
\end{aligned}$$

With this, we can conclude that

$$\overline{Z}^* = \mu\overline{Y}.\ \nu\overline{X}.\ \overline{T} \cap \left(\overline{Q} \cup \mathrm{Pre}_0^{\forall}(\overline{X}) \cup \mathrm{Pre}_{1\setminus\ell}^{\exists}(\overline{X}) \cup \mathrm{Lpre}^{\forall}(\overline{X}) \cup \mathrm{Pre}_l^{\exists}(\overline{Y})\right). \tag{48}$$

where $\overline{T} = V \setminus T$ and $\overline{Q} = V \setminus Q$.

Now denote by $\overline{Y}^m$ the m-th iteration over the fixpoint variable $\overline{Y}$ in (48), where $\overline{Y}^0 = \emptyset$. Further, we denote by $\overline{X}^{mi}$ the set computed in the i-th iteration over the fixpoint variable $\overline{X}$ in (48) during the computation of $\overline{Y}^m$ where $\overline{X}^{m0} = V$. After termination of the inner fixed point over $\overline{X}^{mi}$ we have by construction that $\overline{Y}^m = \overline{X}^{m*}$ and therefore

$$\overline{Y}^m = \overline{T} \cap \left(\overline{Q} \cup \mathrm{Pre}_0^{\forall}(\overline{Y}^m) \cup \mathrm{Pre}_{1\setminus\ell}^{\exists}(\overline{Y}^m) \cup \mathrm{Lpre}^{\forall}(\overline{Y}^m) \cup \mathrm{Lpre}^{\exists}(\overline{Y}^{m-1}) \right). \tag{49}$$

Similar to the soundness proof, we define a ranking over V induced by the iterations of the smallest fixed-point, which now is $\overline{Y}$:

$$\overline{\mathrm{rank}}(v) = m \leftrightarrow v \in \overline{Y}^m \setminus \overline{Y}^{m-1} \quad \text{and} \quad \overline{\mathrm{rank}}(v) = \infty \;\leftrightarrow\; v \notin \overline{Z}^*.$$

This ranking can now be used to define a memoryless Player 1 strategy $\rho_1 : V_1 \to V$ s.t.

$$\rho_1(v) = \min_{(v,w)\in E} \overline{\mathrm{rank}}(w). \tag{50}$$

Towards proving that ρ_1 is winning for $\overline{\psi}$ in (47) we first observe that for every vertex $v \in \overline{Z}^*$ exactly one of the following cases holds:
(a) $v \in (V_0 \cap \overline{Z}^* \cap \overline{T})$, i.e., $\overline{\mathrm{rank}}(v) < \infty$ and $v \in \overline{Q}$ or for all $v' \in E(v)$ it holds that $\overline{\mathrm{rank}}(v') \leq \overline{\mathrm{rank}}(v)$,
(b) $v \in ((V_1 \setminus V^\ell) \cap \overline{Z}^* \cap \overline{T}))$, i.e., $\overline{\mathrm{rank}}(v) < \infty$ and $v \in \overline{Q}$ or there exists $v' \in E(v)$ s.t. $\overline{\mathrm{rank}}(v') \leq \overline{\mathrm{rank}}(v)$, or
($\ell_\forall$) $v \in (V^\ell \cap Z^* \cap \overline{T})$ and $\overline{\mathrm{rank}(v)} < \infty$ and $v \in \overline{Q}$ or for all $v' \in E^\ell(v)$ holds that $\overline{\mathrm{rank}}(v') \leq \overline{\mathrm{rank}}(v)$
($\ell_\exists$) $v \in (V^\ell \cap Z^* \cap \overline{T})$ and $\overline{\mathrm{rank}}(v) > 1$ (and $\overline{\mathrm{rank}}(v) < \infty$), and ($\ell_\forall$) does not hold, but there exists a $v' \in E(v)$ s.t. $\overline{\mathrm{rank}}(v') < \overline{\mathrm{rank}}(v)$.

Using this observation, we now show that every fair adversarial play π compliant with ρ_1 satisfies $\overline{\psi}$ in (47), that is, either stays in $\overline{T}$ forever, or eventually visits $\overline{Q}$ before visiting T.

First, observe that for every node $v \in \overline{Z}^*$ one of the cases (a),(b),($\ell_\forall$), or ($\ell_\exists$) holds. If v is an (a) vertex, we see that either $v \in \overline{Q}$ or for all choices of Player 0 (i.e., for any Player 0 strategy), the play remains in $\overline{Z}^* \subseteq \overline{T}$. Further, it is obvious that ρ_1 ensures, that whenever a (b) vertex is seen, the play remains in $\overline{Z}^* \subseteq \overline{T}$ if we do not already have $v \in \overline{Q}$. The same is true for ($\ell_\forall$) vertexes.

Now consider a fair adversarial play π that is compliant with ρ_1 and $\pi(0) \in \overline{Z}^* \subseteq \overline{T}$. Then it follows from the above intuition that for all visits to (a),(b),($\ell_\forall$) we have two cases: (i) Either $\overline{\psi}$ is immediately true on π by visiting $\overline{Q}$ (and having been in $\overline{Z}^* \subseteq \overline{T}$ in all previous time steps). In this case the suffix of π is irrelevant, because Player 0 has already lost (by visiting $\overline{Q}$ without seeing T). Or (ii) the play remains in $\overline{Z}^* \subseteq \overline{T}$. Now observe that this is also true for infinite visits to (a),(b),($\ell_\forall$) vertexes. As π is fair adversarial, visiting a ($\ell_\forall$) vertex infinitely often, implies that all live edges are taking infinitely often, which all ensure that the play remains in $\overline{Z}^* \subseteq \overline{T}$ or is immediately lost by visiting $\overline{Q}$. Therefore, the only interesting case occurs if π visits ($\ell_\exists$) vertexes. If such a vertex is visited finitely often, ρ_1 ensures that the play stays in $\overline{Z}^* \subseteq \overline{T}$. However, if they are visited infinitely often, a live edge that leaves $\overline{Z}^*$ will also be taken infinitely often. Hence, in order to ensure that π is loosing for Player 0, we need to show that ρ_1 enforces that ($\ell_\exists$) vertexes are only visited finitely often.

To see this, let v be an ($\ell_\exists$) vertex and observe that $\overline{\mathrm{rank}}(v)$ is finite and larger than 1. At the first visit of π to v, ρ_1 decreases the rank as it chooses by definition one of the existentially quantified successors $v' \in E^\ell(v)$ with $\overline{\mathrm{rank}}(v') < \overline{\mathrm{rank}}(v)$. Now observe that for all other cases (a),(b),($\ell_\forall$) either $\overline{Q}$ is visited and the play is immediately loosing for Player 0 or the play is kept in $\overline{Z}^* \subseteq \overline{T}$ and the strategy ρ_1 never increases the rank. As every vertex has a unique rank, ρ_1 ensures that every ($\ell_\exists$) vertex is visited at most once along every compliant fair adversarial play that remains in $\overline{Z}^* \subseteq \overline{T}$. This proves the claim.

B.2.2 Proof of Thm. 3.2.

Theorem (Thm. 3.2 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $Q, G \subseteq V$ be two state sets over $\mathcal{G}$. Further, let*

$$Z^* := \nu Y.\ \mu X.\ Q \cap [(G \cap \mathrm{Cpre}(Y)) \cup (\mathrm{Apre}(Y, X))]\,. \tag{51}$$

Then Z^* is equivalent to the winning region of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition ψ in (8). Moreover, the fixpoint algorithm runs in $O(n^2)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

In order to simplify the proof of Prop. B.2.2, we first prove the following lemma.

Lemma B.5. *Let $Q, G \subseteq V$ and*

$$Z^* := \nu Y.\mu X.Q \cap [(G \cap \mathrm{Cpre}(Y)) \cup \mathrm{Apre}(Y, X)] \tag{52a}$$

$$\widetilde{Z}^* := \nu \widetilde{Y}.\nu Y.\mu X.Q \cap \left[\left(G \cap \mathrm{Cpre}(\widetilde{Y})\right) \cup \mathrm{Apre}(Y, X)\right]. \tag{52b}$$

Then $Z^* = \widetilde{Z}^*$.

Proof. We prove this lemma by a reduction to Lem. B.3 (iii). For this purpose, we define

$$f(\widetilde{X}, \widetilde{Y}) := \emptyset, \qquad h_a(\widetilde{X}, \widetilde{Y}) := Q \cap G \cap \mathrm{Cpre}(\widetilde{Y}),$$
$$g(X, Y) := Q \cap \mathrm{Apre}(Y, X), \text{ and} \qquad h_b(X, Y) := Q \cap G \cap \mathrm{Cpre}(Y).$$

With this we see that (52a) can be equivalently written as

$$\nu\widetilde{Y}.\ \nu\widetilde{X}.\ \nu Y.\ \mu X.\ f(\widetilde{X}, \widetilde{Y}) \cup h_b(X, Y) \cup g(X, Y)$$

while (52b) can be written as

$$\nu\widetilde{Y}.\ \nu\widetilde{X}.\ \nu Y.\ \mu X.\ h_a(\widetilde{X}, \widetilde{Y}) \cup f(\widetilde{X}, \widetilde{Y}) \cup g(X, Y).$$

With this, it follows from Lem. B.3 (iii) that both equations are equivalent. □

With Lem. B.5 in place, we can use (52b) instead of (51) to prove Thm. 3.2. Further, let us define $Z^*(\langle T, Q \rangle)$ to be the set of states computed by the fixpoint algorithm in (12). Then we know that upon termination we have

$$\widetilde{Z}^* = \widetilde{Y}^* = Z^*(\langle Q \cap G \cap \mathrm{Cpre}(\widetilde{Y}^*), Q \rangle). \tag{53}$$

Now we will use (53) to prove soundness and completeness of Thm. 3.2.

Soundness Let us now define $T := Q \cap G \cap \mathrm{Cpre}(\widetilde{Y}^*))$. Pick any state $v \in \widetilde{Z}^*$ and the strategy ρ_0 defined as in (46) over the sets X^i computed in the last iteration over X when computing $Z^*(\langle T, Q \rangle)$. Further, let π be an arbitrary fair adversarial play starting in v and being compliant with ρ_0. Then we need to show that π fulfills ψ in (8).

Using (53) and the fact that $v \in \widetilde{Z}^*$ we know from Thm. 3.3 that π fulfills $Q\mathcal{U}T$. That is, there exists a $k \in \mathbb{N}$ s.t. $\pi(i) \in Q$ for all $i < k$ and $\pi(k) \in T = Q \cap G \cap \mathrm{Cpre}(\widetilde{Y}^*))$. With this we know that (a) $\pi(k) \in Q$, (b) $\pi(k) \in G$ and (c) $v \in \mathrm{Cpre}(\widetilde{Y}^*)$. Now we have two cases: (c.1) If $\pi(k) \in V^1$, then it follows from the definition of Cpre that $E(\pi(k)) \subseteq \widetilde{Y}^*$. As $\widetilde{Y}^* = \widetilde{Z}^*$, we know $\pi(k+1) \in \widetilde{Z}^*$. (c.2) If $\pi(k) \in V^0$ we know that $\mathrm{rank}(\pi(k)) = min_{v' \in E(\pi(k))} \mathrm{rank}(v')$. Now recall that $\widetilde{Z}^* = \widetilde{Y}^* = Y^* = \bigcup_{i>0} X^i$. Hence, any state with rank $0 < n < \infty$ is contained in $\widetilde{Z}^*$ and hence, we have $\pi(k+1) \in \widetilde{Z}^*$. With this, we can successively re-apply Thm. 3.3 to $\pi(k+1)$. This shows that G is visited infinitely often along π while π always remains within Q.

Completeness Let $\mathcal{W} \subseteq V$ be the set of states from which Player 0 has a winning strategy w.r.t. ψ in (8). In order to prove completeness, we need to show that $\mathcal{W} \subseteq Z^*$.

Recall, that for all states $v \in \mathcal{W}$ there exists a strategy ρ_0 s.t. all compliant fair adversarial plays π fulfill ψ. Now consider the weaker LTL formula $\widetilde{\psi} := Q\mathcal{U}(Q \cap G)$ and let $\widetilde{\mathcal{W}}$ be the winning state set for $\widetilde{\psi}$. Then we know by construction that $\widetilde{\psi}$ holds for $\pi(0)$ and for every $\pi(k) \subseteq Q \cap G$ while π always remains in Q. We can therefore strengthen $\widetilde{\psi}$ to $\widetilde{\psi} := Q\mathcal{U}(Q \cap G \cap \mathrm{Cpre}(\widetilde{\mathcal{W}}))$ and see that still $\psi \to \widetilde{\psi}$ and therefore $\mathcal{W} \subseteq \widetilde{\mathcal{W}}$.

Now observe that it follows from Thm. 3.3 that $\widetilde{\mathcal{W}} = Z^*(\langle Q \cap G \cap \mathrm{Cpre}(\widetilde{\mathcal{W}}), Q \rangle)$. It further follows from the monotonicity of the fixed-point that $\widetilde{Z}^*$ is the *largest* set of states s.t. equality holds in (53). We therefore have to conclude that $\widetilde{\mathcal{W}} \subseteq \widetilde{Z}^*$. As we have shown that $\mathcal{W} \subseteq \widetilde{\mathcal{W}}$, the claim is proved.

B.3 Proof of Thm. 3.1

Theorem (Thm. 3.1 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\mathcal{R}$ be a Rabin condition over $\mathcal{G}$ with index set $P = [1; k]$. Further, let*

$$\begin{aligned} Z^* := \nu Y_{p_0}.\mu X_{p_0}. \textstyle\bigcup_{p_1 \in P} \nu Y_{p_1}.\mu X_{p_1}. \\ \textstyle\bigcup_{p_2 \in P \setminus \{p_1\}} \nu Y_{p_2}.\mu X_{p_2}. \\ \vdots \\ \textstyle\bigcup_{p_k \in P \setminus \{p_1, \ldots, p_{k-1}\}} \nu Y_{p_k}.\mu X_{p_k}. \left[\textstyle\bigcup_{j=0}^{k} C_{p_j} \right], \end{aligned}$$

where

$$C_{p_j} := \bigcap_{i=0}^{j} \overline{R}_{p_i} \cap \left[\left(G_{p_j} \cap \mathrm{Cpre}(Y_{p_j}) \right) \cup \left(\mathrm{Apre}(Y_{p_j}, X_{p_j}) \right) \right],$$

with $p_0 = 0$, $G_{p_0} := \emptyset$ and $R_{p_0} := \emptyset$. Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (6). Moreover, the fixpoint algorithm runs in $O(n^{k+2}k!)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

This section contains the proof of Thm. 3.1 which is inspired by the proof of Piterman and Pnueli [2006] for "normal" Rabin games. We first give a construction of a ranking induced by the fixpoint algorithm in (7) in Sec. B.3.1, and use this ranking to define a memoryless Player 0 strategy. As part of the soundness proof for Thm. 3.1 in Sec. B.3.2, we then show that this extracted strategy is indeed a winning strategy of Player 0 in the *fair adversarial game* over $\mathcal{G}^\ell$ w.r.t. φ. Further, we show in Sec. B.3.3 that the fixpoint algorithm in (7) is also complete, that is $\mathcal{W} \subseteq Z^*$. Intuitively, completeness shows that if Z^* is empty, there indeed exists no live-sufficient winning strategy (with arbitrary memory) for the given fair adversarial Rabin game. Additional lemmas and proofs can be found in Appendix B.3.4. The time complexity of the algorithm is proven separately in App. C.

B.3.1 Strategy Extraction. Our strategy extraction is adapted from the ranking in [Piterman and Pnueli 2006, Sec. 3.1]. Recall, that we consider the set of Rabin pairs $\mathcal{R} = \{\langle G_1, R_1 \rangle, \ldots, \langle G_k, R_k \rangle\}$ with index set $P = \{1, \ldots, k\}$ and the artificial Rabin pair $\langle G_0, R_0 \rangle$ s.t. $G_0 = R_0 = \emptyset$. A *permutation* of the index set P is an one-to-one and onto function from P to P; as usual, we write $p_1 \ldots p_k$ to denote the permutation mapping i to p_i, for $i = 1, \ldots, k$. We define $\Pi(P)$ to be the set of all permutations over P. The *configuration domain* of the Rabin condition $\mathcal{R}$ is defined as

$$D(\mathcal{R}) := \left\{ p_0 i_0 p_1 i_1 \ldots p_k i_k \;\middle|\; i_j \in [0; n],\ p_0 = 0,\ p_1 \ldots p_k \in \Pi(P) \right\} \cup \{\infty\} \tag{55}$$

where $n < \infty$ is a natural number which is larger then the maximal number of iterations needed in any instance of the fixed point computation in (7) which is known to be finite. If $\mathcal{R}$ is clear from the context, we write D instead of $D(\mathcal{R})$.

Intuition: We first explain the intuition behind the chosen ranking. For this we consider the definition of ranks for states $v \in Z^*$ in an iterative fashion. First, consider the last iteration over X_{p_0} converging to the fixed point $Z^* = Y^*_{p_0} = \bigcup_{i_0>0} X^{i_0}_{p_0}$ where $X^0_{p_0} := \emptyset$. By flattening (7) we see that for all $i_0 > 0$ we have

$$X^{i_0}_{p_0} = \mathrm{Apre}(Y^*_{p_0}, X^{i_0-1}_{p_0}) \cup \mathcal{A}_{p_0 i_0} \tag{56a}$$

where $\mathcal{A}_{p_0 i_0}$ collects all remaining terms of the fixpoint algorithm in (7) and will be specified later. For now, we want to assign a "minimal rank" to all states added to Z^* via the first term in (56a). Let us assume that the right "minimal rank" for these states is

$$d = p_0 i_0 p_1 0 \ldots p_k 0 \quad \text{with} \quad p_1 < p_2 < \ldots < p_k \text{ and } i_0 > 0.$$

We assign this rank to v iff $v \in \mathrm{Apre}(Y^*_{p_0}, X^{i_0-1}_{p_0}) \setminus X^{i_0-1}_{p_0}$, i.e., if v is not already added to the fixed point in a previous iteration. The intuition behind this rank choice is that we want to remember that we have added v to Z^* in the i_0's computation over X_{p_0}, which sets the counter for p_0 in d to i_0. We keep all other counters at 0 because there is no actual contribution of terms involving variables X_{p_i} for $p_i \in P$ for the "adding" of v.

Now recall that

$$X^{i_0}_{p_0} = \bigcup_{p_1 \in P} Y^*_{p_1} = \bigcup_{p_1 \in P} \bigcup_{i_1>0} X^{i_1}_{p_1}.$$

Further, we know that

$$\mathrm{Apre}(Y^*_{p_0}, X^{i_0-1}_{p_0}) \subseteq X^{i_1}_{p_1} \quad \text{for all} \quad p_1 \in P \text{ and } i_1 > 0. \tag{56b}$$

Hence, any state added to the fixed point via $X^{i_0}_{p_0}$ (which is not contained in $X^{i_0-1}_{p_0}$) is either added via $\mathrm{Apre}(Y^*_{p_0}, X^{i_0}_{p_0})$ or via any other remaining term within $X^{i_1}_{p_1}$ for at least one p_1 and $i_1 > 0$. So let us explore the ranking in the latter case.

For this, let us proceed by going over all $X^{i_1}_{p_1}$ in increasing order over P, i.e, we start with selecting $p_1 = 1$. Further, we remember that we compute the next iteration over X_{p_1} (i.e., $X^{i_1}_{p_1}$ given $X^{i_1-1}_{p_1}$) as part of computing the set $X^{i_0}_{p_0}$. I.e., we remember the *computation-prefix* $\delta = p_0 i_0$ in the computation of $X^{i_1}_{p_1}$. To make δ explicit, we denote $X^{i_1}_{p_1}$ by $X^{i_1}_{\delta p_1}$. Now, we again consider the last iteration over $X_{\delta p_1}$ converging to the fixed point $Y^*_{\delta p_1}$ (for the currently considered computation-prefix δ). Then we have

$$X^{i_1}_{\delta p_1} = \underbrace{\mathrm{Apre}(Y^*_{p_0}, X^{i_0-1}_{p_0})}_{=:S_\delta} \cup \underbrace{\overline{R}_{p_1} \cap \left[\left(G_{p_1} \cap \mathrm{Cpre}(Y^*_{\delta p_1})\right) \cup \mathrm{Apre}(Y^*_{\delta p_1}, X^{i_1-1}_{\delta p_1})\right]}_{=:C_{\delta p_1 i_1}} \cup \mathcal{A}_{\delta p_1 i_1}.$$

We now want to assign the "minimal rank" to all states that are added to the fixed point via $C_{\delta p_1 i_1}$. The immediate choice of this rank is

$$d = p_0 i_0 p_1 i_1 p_2 0 \ldots p_k 0 = \delta p_1 i_1 p_2 0 \ldots p_k 0 \quad \text{with } p_2 < \ldots < p_k \quad \text{and } i_0, i_1 > 0. \tag{56c}$$

(Note that we do not necessarily have $p_1 < p_2$!)

We only want to assign this rank to states that are actually added to the fixed point via $C_{\delta p_1 i_1}$, i.e., do not already have a rank assigned. First, all states $v \in S_\delta$ already have an assigned rank (as discussed before). Second, for $i_1 > 1$ all states in $C_{\delta p_1 i_1 - 1}$ have already an assigned rank. But, third, also all states that have been added by considering a different $X_{\tilde{p}_1}$ with $\tilde{p}_1 \in P$ being smaller then the currently considered p_1 also have an already assigned rank.

Now consider the ranking choices suggested in (56b) and (56c). Then we see that all already assigned ranks are *smaller* (in terms of the lexicographic order over D) than the one in (56c). To see this, first consider a state $v \in S_\delta$. Either, $v \in X^{i_0-1}_{p_0}$ in which case its 0'th counter is smaller then

i_0 (i.e., $i_0 - 1 < i_0$) or v has been added via S_δ, in which case the 0'th counter is equivalent but the first counter is 0 and therefore smaller then i_1 in (56c) (as, $i_1 > 0$). Now consider a state $v \in X_{\tilde{p}_1}$ with $\tilde{p}_1 < p_1$. In this case we see that 0'th counter is equivalent but the first permutation index is smaller (as $\tilde{p}_1 < p_1$).

We can therefore avoid specifying exactly in which set v should *not* be contained to be a newly added state. We can simply collect all possible rank assignments for every state and then, post-process this set to select the smallest rank in this set. Let us now generalize this idea to all possible configuration prefixes.

Proposition B.1. *Let $\delta = p_0 i_0 \dots p_{j-1} i_{j-1}$ be a* configuration prefix, *$p_j \in P \setminus \{p_1, \dots, p_{j-1}\}$ the next permutation index and $i_j > 0$ a counter for p_j. Then the flattening of* (7) *for this configuration prefix is given by*

$$X^{i_j}_{\delta p_j} = \underbrace{S_\delta \cup C_{\delta p_j i_j}}_{S_{\delta p_j i_j}} \cup \mathcal{A}_{\delta p_j i_j} \tag{57a}$$

where

$$Q_{p_0 \dots p_a} := \bigcap_{b=0}^{a} \overline{R}_{p_b}, \tag{57b}$$

$$C_{\delta p_a i_a} := \left(Q_{\delta p_a} \cap G_{p_a} \cap \mathrm{Cpre}(Y^*_{\delta p_a})\right) \cup \left(Q_{\delta p_a} \cap \mathrm{Apre}(Y^*_{\delta p_a}, X^{i_a - 1}_{\delta p_a})\right), \tag{57c}$$

$$S_{p_0 i_0 \dots p_a i_a} := \bigcup_{b=0}^{a} C_{p_0 i_0 \dots p_b i_b}, \tag{57d}$$

$$\mathcal{A}_{\delta p_j i_j} := \bigcup_{p_{j+1} \in P \setminus \{p_1, \dots, p_j\}} \bigcup_{i_{j+1} > 0} \left(X^{i_{j+1}}_{\delta p_j i_j p_{j+1}} \setminus S_{\delta p_j i_j}\right) \tag{57e}$$

As this flattening follows directly from the structure of the fixpoint algorithm in (7) and the definition of C_{p_j} in (7b), the proof is omitted.

Using the flattening of (7) in (57) we can define a ranking function induced by (7) as follows.

Definition B.6. Given the premises of Prop. B.1, we define $\underline{\gamma} := p_{j+1} 0 p_{j+2} 0 \dots p_k 0$ with $p_{j+1} < p_{j+2} < \dots < p_k$ to be the minimal configuration post-fix. Then we define the rank-set $R : V \longrightarrow 2^D$ s.t. (i) $\infty \in R(v)$ for all $v \in V$, and (ii) $\delta p_j i_j \underline{\gamma} \in R(v)$ iff $v \in S_{\delta p_j i_j}$. The ranking function rank $: V \longrightarrow D$ is defined s.t. rank $: v \mapsto \min\{R(v)\}$.

Based on the ranking in Def. B.6 we define a memory-less player 0 strategy ρ_0, s.t. $\rho_0(v)$ forces progress to a state reachable from v which has minimal rank compared to all other successors of v. We prove Thm. B.7 in Sec. B.3.2.

Theorem B.7. *Given the premises of Prop. B.1, the memory-less player* 0 *strategy $\rho_0 : V^0 \cap Z^* \longrightarrow V^1$ s.t.*

$$\rho_0(v) := \min_{(v,w) \in E} (\mathrm{rank}(w)), \tag{58}$$

is a winning strategy for player 0 *in the* fair adversarial game *over $\mathcal{G}^\ell$ w.r.t. φ.*

Example B.8. Consider the Rabin game depicted in Fig. 9 and discussed in App. A. Here, the strategy construction outlined in Thm. B.7 enforces a transition from q_6 to q_7 and a transition from q_5 to q_3. This is observed by noting that rank$(q_2) = 002012$ and rank$(q_7) = 001121$ where rank$(q_7) <$ rank(q_2). In addition, rank$(q_1) = 011021$ and rank$(q_3) = 001121$, where rank$(q_3) <$ rank(q_1).

B.3.2 Soundness. We now show why the fixpoint algorithm in (7) is *sound*, i.e., why $Z^* \subseteq \mathcal{W}$ in Thm. 3.1 holds. In addition, we also show that Thm. B.7 holds.

We prove soundness by an induction over the nesting of fixed points in (7) from inside to outside. In particular, we iteratively consider instances of the flattening in (57), starting with $j = k$ as the base case, and doing an induction from "$j + 1$" to "j". To this end, we consider a local winning condition which refers to the current configuration-prefix $\delta = p_0 i_0 \ldots p_{j-1} i_{j-1}$ in (57), namely

$$\psi_{\delta p_j} := \left(\begin{array}{ll} & Q_{\delta p_j} \mathcal{U} S_\delta \\ \vee & \Box Q_{\delta p_j} \wedge \Box\Diamond G_{p_j} \\ \vee & \Box Q_{\delta p_j} \wedge \left(\bigvee_{i \in P \setminus \{p_0, \ldots, p_j\}} \left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right) \right) \end{array} \right). \tag{59}$$

Further, we denote by $\mathcal{W}_{\delta p_j}$ the set of states for which player 0 wins the *fair adversarial game* over $\mathcal{G}^\ell$ w.r.t. $\psi_{\delta p_j}$ in (59).

By recalling that for $p_j = p_0 = 0$ we have $Q_{p_0} = V$, $S_\varepsilon = \emptyset$ and $G_{p_0} = \emptyset$, we see that for $j = 0$ the condition in (59) simplifies to

$$\psi_{p_0} = \bigvee_{i \in P} \left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right).$$

This implies that ψ_{p_0} is equivalent to φ in (6). Given this observation, the proof of soundness in Thm. 3.1 proceeds by inductively showing that

$$X^{i_j}_{\delta p_j} \subseteq \mathcal{W}_{\delta p_j} \tag{60}$$

for any configuration prefix δ, next permutation index p_j and counter $i_j > 0$. Thereby, we ultimately also prove this claim for $p_j = p_0 = 0$ where δ is the empty string and $Y^*_{p_0} = \bigcup_{i_0 > 0} X^{i_0}_{p_0}$ coincides with Z^* in (7), which proves the statement.

With this insight the proof of Thm. B.7 as well as the soundness part of Thm. 3.1 reduce to the following proposition.

Proposition B.2. *For all $j \in [0, k]$, computation-prefixes $\delta = p_0 i_0 \ldots p_{j-1} i_{j-1}$, next permutation index $p_j \in P \setminus \{p_0, \ldots, p_{j-1}\}$, counter $i_j > 0$ and state $v \in X^{i_j}_{\delta p_j}$ the strategy ρ_0 in* (58) *wins the fair adversarial game over $\mathcal{G}^\ell$ w.r.t. $\psi_{\delta p_j}$ in* (59).

To see why Prop. B.2 holds, we consider the computation of $X^{i_j+1}_{\delta p_j}$ in (57a) and observe that the states in $X^{i_j+1}_{\delta p_j}$ can be clustered based on their rank induced via Def. B.6 as follows (see Sec. B.3.5 for a full proof).

Proposition B.3. *Given the premisses of Prop. B.2, let*

$$\begin{array}{lll} \underline{\gamma} = p_{j+1} 0 p_{j+2} 0 \ldots p_k 0 & \text{with} \quad p_{j+1} < p_{j+2} < \ldots < p_k, & \text{and} \\ \overline{\gamma} = p_{j+1} n p_{j+2} n \ldots p_k n & \text{with} \quad p_k < p_{k-1} < \ldots < p_{j+1} & \end{array}$$

be the minimal *and* maximal *post-fix, respectively. Then, for all $v \in X^i_{\delta p_j}$ exactly one of the following cases holds:*
(a) $v \in S_\delta$ and $\mathrm{rank}(v) \leq \delta p_j 0 \overline{\gamma}$,
(b) $v \in Q_{\delta p_j} \cap G_{p_j} \cap \mathrm{Cpre}(Y^*_{\delta p_j})$ and $\mathrm{rank}(v) = \delta p_j 1 \underline{\gamma}$,
(c) $v \in Q_{\delta p_j} \cap \mathrm{Apre}(Y^*_{\delta p_j}, X^{i_j-1}_{\delta p_j})$ and $\mathrm{rank}(v) = \delta p_j i_j \underline{\gamma}$ s.t. $i_j > 1$, or
(d) $v \in \mathcal{A}_{\delta p_j i_j}$ and there exists $\underline{\gamma} < \gamma' \leq \overline{\gamma}$ s.t. $\mathrm{rank}(v) = \delta p_j i_j \gamma'$.

Using Prop. B.3 we prove Prop. B.2 by an induction over j.

Proof of Prop. B.2. **Base case:** First, for $j = k$ the last line of (59) disappears. Then the proof reduces to Thm. 3.3 and Thm. 3.2 in the following way. First, we fix all fixpoint variables $Y^*_{p_0 \ldots p_l}$ and $X^{i_l}_{p_0 \ldots p_l}$ for $l < j$ as well as $Y^*_{\delta p_j}$. With this, we see that $T := S_\delta \cup (Q_{\delta p_j} \cap G_{p_j} \cap \text{Cpre}(Y^*_{\delta p_j}))$ becomes a fixed set of states and (57a) reduces to

$$X^{i_j}_{\delta p_j} = T \cup (Q_{\delta p_j} \cap \text{Apre}(Y^*_{\delta p_j}, X^{i_j - 1}_{\delta p_j}))$$

where we know that $X^{i_j}_{\delta p_j} \subseteq Y^*_{\delta p_j}$. Further, it follows form Prop. B.3 that for all $X^{i_j}_{\delta p_j}$ the ranking only differs by the i_j count. Hence, we can replace ρ_0 in (58) by the simpler strategy ρ_0 in (46) that only consideres the i_j count as the rank of states in $Y^*_{\delta p_j} = \bigcup_{i_j > 0} X^{i_j}_{\delta p_j}$. With this it follows from Thm. 3.3 that for any fair adversarial play π compliant with ρ_0 in (58) and starting in $X^{i_j}_{\delta p_j}$ for some $i_j \geq 0$ it holds that $Q_{\delta p_j} \mathcal{U} T$. This implies that whenever such a play π eventually reaches a state in $S_\delta \subseteq T$ the first line of (59) holds.

Now assume that π does not reach a state in $S_\delta \subseteq T$. Then it reaches a state in $Q_{\delta p_j} \cap G_{p_j} \cap \text{Cpre}(Y^*_{\delta p_j})$ and therefore has a successor state $v' \in Y^*_{\delta p_j} = \bigcup_{i_j > 0} X^{i_j}_{\delta p_j}$. Hence, $v' \in X^{i_j}_{\delta p_j}$ for some $i_j \geq 0$. By repeatedly applying this argument we see that π either eventually reaches a state in $S_\delta \subseteq T$ or it remains infinitely in $C_{\delta p_j}$.. In the latter case, it follows from Thm. 3.2 that the second line of (59) holds.

Induction step: For the induction step (from "$j + 1$" to "j") we first analyze the assumption. I.e., we know that for the longer computation prefix $\delta' = \delta p_j i_j$ and any next permutation index p_{j+1} we have that $Y^*_{\delta' p_{j+1}} \subseteq \mathcal{W}_{\delta' p_{j+1}}$ for all $p_{j+1} \in P \setminus \{p_1, \ldots, p_j\}$. Now recall that (57e) implies

$$\mathcal{A}_{\delta p_j i_j} = \bigcup_{p_{j+1} \in P \setminus \{p_1, \ldots, p_j\}} Y^*_{\delta' p_{j+1}} \setminus S_{\delta p_j i_j}$$

and therefore, we know that for all $v \in \mathcal{A}_{\delta p_j i_j}$ there exists a p_{j+1} s.t. $v \in \mathcal{W}_{\delta' p_{j+1}}$. That is, any fair adversarial play starting in v that is compliant with ρ_0 in (58) fulfills (59).

Therefore, whenever a fair adversarial play π starting in $X^{i_j}_{\delta p_j}$ visits a vertex $v \in \mathcal{A}_{\delta p_j i_j}$ (i.e., case (d) holds), we know that π could possibly come back to a state $v \in S_{\delta' p_{j+1}} = S_\delta \cup C_{\delta p_j i_j}$ (via the first line of $\psi_{\delta' p_{j+1}}$).

In this case, Prop. B.3 ensures that the i_j count of the rank of states always stays constant while the play stays in $\mathcal{A}_{\delta p_j i_j}$. Therefore, one can ignore these finite sequences of (d) vertices in π while applying the ranking arguments of Thm. 3.3 and Thm. 3.2. I.e., we can conclude that in this case either the first or the second line of (59) holds for π. It remains to show that π fulfills the last line of (59) if π eventually stays within $\mathcal{A}_{\delta p_j i_j}$ forever. First, observe that this is only possible if S_δ is not visited along π. Hence, we know that $Q_{\delta p_j}$ holds along π until $\mathcal{A}_{\delta p_j i_j}$ is entered and never left. Further, as $\mathcal{A}_{\delta p_j i_j}$ is assumed to be never left after some time $k > 0$, we know that from that time onward there exists no p_{j+1} s.t. $S_{\delta' p_{j+1}}$ is visited again by π. This implies that for all vertices $\pi(k')$ with $k' > k$ the last two lines of $\psi_{\delta' p_{j+1}}$ (denoted $\psi'_{\delta' p_{j+1}}$) must be true for at lease one p_{j+1}. Hence, π fulfills the property

$$\Psi_{\delta p_j} := \square Q_{\delta p_j} \wedge \diamond \underbrace{\left(\bigvee_{p_{j+1} \in P \setminus \{p_1, \ldots, p_j\}} \psi'_{\delta' p_{j+1}} \right)}_{\Psi'_{\delta p_j}} \tag{61a}$$

With this, it remains to show that $\Psi_{\delta p_j}$ implies that the last line of (59) is true for π. In particular, we can show that both statements are equivalent, i.e.,

$$\Psi_{\delta p_j} = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in P \setminus \{p_1,\ldots,p_j\}} \left(\Diamond\Box \overline{R}_{p_{j+1}} \wedge \Box\Diamond G_{p_{j+1}} \right) \tag{61b}$$

Equation (61) is proved in Sec. B.3.6. This conclues the proof. □

B.3.3 Completeness. We now show why the fixpoint algorithm in (7) is complete, i.e., why $\mathcal{W} \subseteq Z^*$ in Thm. 3.1 holds.

We also prove completeness by an induction over the nesting of fixed points in (7) from inside to outside. In particular, we iteratively consider the fixed points $Y^*_{\delta p_j}$ and show that $Y^*_{\delta p_j} \subseteq \mathcal{W}_{\delta p_j}$. As $\psi_{\delta p_j}$ simplifies to φ in (6) for $p_j = p_0 = 0$, we ultimately show that $\mathcal{W} \subseteq Z^*$ in Thm. 3.1. With this insight the proof of the completness part of Thm. 3.1 reduces to the following proposition.

Proposition B.4. *For all $j \in [0, k]$, computation-prefixes $\delta = p_0 i_0 \ldots p_{j-1} i_{j-1}$ and next permutation index $p_j \in P \setminus \{p_0, \ldots, p_{j-1}\}$ it holds that $\mathcal{W}_{\delta p_j} \subseteq Y^*_{\delta p_j}$.*

Proof. The proof proceeds by a nested induction over j starting with $j = k$.

Base case: Recall that for $j = k$ the last line of (59) disappears. Hence, for any state $v \in \mathcal{W}_{\delta p_j}$ either the first or the second line of (59) holds. Then the proof reduces to Thm. 3.3 and Thm. 3.2 in the following way.

First, we fix all fixpoint variables $Y^*_{p_0 \ldots p_l}$ and $X^{i_l}_{p_0 \ldots p_l}$ for $l < j$ as well as $Y^*_{\delta p_j}$. With this, we see that $T := S_\delta \cup (Q_{\delta p_j} \cap G_{p_j} \cap \mathrm{Cpre}(Y^*_{\delta p_j}))$ becomes a fixed set of states and (57a) reduces to

$$Y^*_{\delta p_j} = Z^*(\langle T, Q_{\delta p_j} \rangle)$$

where $Z^*(\langle T, Q \rangle)$ is the set of states computed by the fixpoint algorithm in (12).

Then it follows from Thm. 3.3 that any state $v \in V$ for which there exists a fair adversarial play π that is winning for the winning condition $Q_{\delta p_j} \mathcal{U} T$ is contained in $Y^*_{\delta p_j}$. If, indeed the first line of (59) holds for π, this ensures that the claim holds.

Now assume that $Q_{\delta p_j} \mathcal{U} T$ holds for π but S_δ is never reached. Hence, $Q_{\delta p_j} \mathcal{U} (Q_{\delta p_j} \cap G_{p_j} \cap \mathrm{Cpre}(Y^*_{\delta p_j}))$ holds for π. With this, it follows form Thm. 3.2 that any state $v \in V$ for which there exists a fair adversarial play π for which the second line of (59) holds is contained in $Y^*_{\delta p_j}$, proving the claim in this case.

Induction Step: For the induction from "$j+1$" to "j" we first analyze the assumption. I.e., we know that for the longer computation prefix $\delta' = \delta p_j$ and any next permutation index p_{j+1} we have that $\mathcal{W}_{\delta' p_{j+1}} \subseteq Y^*_{\delta' p_{j+1}}$. Further, observe that $\Psi'_{\delta p_j} \subseteq \bigcup_{p_{j+1} \in P \setminus \{p_1,\ldots,p_j\}} \mathcal{W}_{\delta' p_{j+1}} \setminus S_{\delta p_j i_j}$ by construction. We therefore have

$$\Psi'_{\delta p_j} \subseteq \bigcup_{p_{j+1} \in P \setminus \{p_1,\ldots,p_j\}} Y^*_{\delta' p_{j+1}} \setminus S_{\delta p_j i_j} = \mathcal{A}_{\delta p_j i_j}.$$

With this observation, we see that any fair adversarial play π which fulfills the last line of (59) also fulfills the weaker condition $Q_{\delta p_j} \mathcal{U} \mathcal{A}_{\delta p_j i_j}$. Therefore, the claim follows from the same reasoning as in the base case by re-defining T to $T := S_\delta \cup (Q_{\delta p_j} \cap G_{p_j} \cap \mathrm{Cpre}(Y^*_{\delta p_j})) \cup \mathcal{A}_{\delta p_j i_j}$. □

B.3.4 Additional Lemmas and Proofs. In this section we provide additional lemmas and proofs to support the proof of Thm. 3.1 and Thm. B.7.

B.3.5 *Proof of Prop. B.3.*

Lemma B.9. *Given the premisses of Prop. B.3, it holds for all* $v \in X^{i_j}_{\delta p_j}$ *that*

(i) $v \in S_\delta$ *iff* $\mathrm{rank}(v) \leq \delta p_j 0 \underline{\gamma}$

(ii) $v \in X^{i_j}_{\delta p_j}$ *iff* $\mathrm{rank}(v) \leq \delta p_j i_j \overline{\gamma}$

(iii) $v \in Y^*_{\delta p_j}$ *iff* $\mathrm{rank}(v) \leq \delta p_j n \overline{\gamma}$

(iv) $v \in \mathcal{A}_{\delta p_j i_j}$ *iff there exists* $\underline{\gamma} < \gamma' \leq \overline{\gamma}$ *s.t.* $\mathrm{rank}(v) = \delta p_j i_j \gamma'$

Proof of Lem. B.9. We prove all claims separately.
(i) It immediately follows from Def. B.6 (i) that $\delta p_j 0 \underline{\gamma} \in R(v)$ iff $v \in S_\delta$. If it is the minimal element in $R(v)$ then $\mathrm{rank}(v) = \delta p_j 0 \underline{\gamma}$, if not, there exists a smaller element in $R(v)$, and then $\mathrm{rank}(v) < \delta p_j 0 \underline{\gamma}$ from the definition of rank.
(ii) First, observe, that for $j = k$ it follows from (57a) that $X^{i_k}_{\delta p_k} = S_\delta p_k i_k$ and therefore from (i) that $v \in X^{i_k}_{\delta p_k}$ iff $\mathrm{rank}(v) \leq \delta p_k i_k$. Now we do an induction, assuming that for any $p_{j+1} \in P \setminus \{p_0, \ldots, p_j\}$ and $0 < i_{j+1} \leq n$ it holds that $v \in X^{i_{j+1}}_{\delta p_{j+1}}$ iff $\mathrm{rank}(v) \leq \delta' p_{j+1} i_{j+1} \overline{\gamma'}$ (where δ' goes up to index j and γ' starts only at index $j + 2$. Now recall that

$$X^{i_j}_{\delta p_j} = \bigcup_{p_{j+1} \in P \setminus \{p_0, \ldots, p_j\}} Y^*_{\delta p_{j+1}} = \bigcup_{p_{j+1} \in P \setminus \{p_0, \ldots, p_j\}} \bigcup_{i_{j+1} > 0} X^{i_{j+1}}_{\delta p_j i_j p_{j+1}}.$$

Hence, $v \in X^{i_j}_{\delta p_j}$ iff there exists $p_{j+1} \in P \setminus \{p_0, \ldots, p_j\}$ and $0 < i_{j+1} \leq n$ s.t. $v \in X^{i_{j+1}}_{\delta p_j i_j p_{j+1}}$. Now we know that for any choice of p_{j+1} and i_{j+1} we have $\mathrm{rank}(v) \leq \delta' p_j i_j p_{j+1} i_{j+1} \overline{\gamma'}$. Now the worst case, in terms of the lexicographic ordering over D is that $p_{j+1} = \max(P \setminus \{p_0, \ldots, p_j\})$ and $i_{j+1} = n$. Hence, we know that $\mathrm{rank}(v) \leq \delta p_j i_j \overline{\gamma}$.
(iii) As $Y^*_{\delta p_j} = \bigcup_{i_j > 0} X^{i_j}_{\delta p_j}$ it follows that there exists $0 < i_j \leq n$ s.t. $v \in X^{i_j}_{\delta p_j}$ and (from (ii)) therefore $\mathrm{rank}(v) \leq \delta p_j i_j \overline{\gamma}$. Again, the worst case is $i_j = n$, giving $\mathrm{rank}(v) \leq \delta p_j n \overline{\gamma}$.
(iv) It follows from (57a) that $v \in \mathcal{A}_{\delta p_j i_j}$ iff $v \in X^{i_j}_{\delta p_j} \setminus S_{\delta p_j i_j}$. Hence, it follows from (i) and (ii) that $\mathrm{rank}(v) > \delta p_j 0 \underline{\gamma}$ and $\mathrm{rank}(v) \leq \delta p_j i_j \overline{\gamma}$ which is true iff there exists $\underline{\gamma} < \gamma' \leq \overline{\gamma}$ s.t. $\mathrm{rank}(v) = \delta p_j i_j \gamma'$, which proves the statement. □

Given these properties of the ranking function, we are ready to prove the suggested case split in Prop. B.3.

Proof of Prop. B.3. We call a vertex $v \in V$ that fulfills cases (α) in either Lem. B.9 or Prop. B.3 an (α)-vertex. First, observe that cases (i) and (iv) in Lem. B.9 coincide with cases (a) and (d), respectively, in Prop. B.3. Further, recall that $X^1_{\delta p_j} = \emptyset$. Therefore, $X^1_{\delta p_j}$ only contains (a)-,(b)- and (d)-vertices, as $\mathrm{Apre}(\cdot, \emptyset) = \emptyset$. Now we know from (ii) that for any $v \in X^1_{\delta p_j}$ we have $\mathrm{rank}(v) \leq \delta p_j 1 \overline{\gamma}$. Now excluding the rankings for (a)- and (d)-vertices we obtain that (b)-vertices must have rank $\mathrm{rank}(v) \leq \delta p_j 1 \underline{\gamma}$. Similarly, for every $i_j > 1$ we know that $X^{i_j}_{\delta p_j}$ contains (a)-, (b)-, (c)- and (d)-vertices. Now excluding (a)-, (b)- and (d)- vertices yields $\mathrm{rank}(v) \leq \delta p_j i_j \underline{\gamma}$ for all (c)-vertices. □

B.3.6 *Proof of* (61).

Given the notation in Sec. B.3.2 we prove that the equality in (61) holds.
First recall that

$$\Psi'_{\delta' p_{j+1}} := \left(\begin{array}{cc} & \Box Q_{\delta' p_{j+1}} \wedge \Box\Diamond G_{p_{j+1}} \\ \vee & \Box Q_{\delta' p_{j+1}} \wedge \left(\bigvee_{i \in \widetilde{P}_{\setminus j+1}} \left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right) \right) \end{array} \right), \tag{62}$$

where $\widetilde{P}_{\setminus j+1} := P \setminus \{p_1, \ldots, p_{j+1}\}$.

For the insertion of (62) into (61a) we have the following observations. First, observe that $\Diamond(B \vee C) = \Diamond B \vee \Diamond C$, i.e., we can distribute the eventuality operator preceding $\Psi'_{\delta' p_{j+1}}$ over both lines. Second, we can re-order the preceeding disjunction over p_{j+1} in (61a) and the disjunction between the two lines of (62). This yields to the following condition

$$\begin{aligned} \Psi_{\delta p_j} =& \Box Q_{\delta p_j} \wedge \left(\bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} (\Diamond \lambda_1) \vee \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} (\Diamond \lambda_2) \right) \\ =& \underbrace{\left(\Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} (\Diamond \lambda_1) \right)}_{=:\Psi_1} \vee \underbrace{\left(\Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} (\Diamond \lambda_2) \right)}_{=:\Psi_2}, \end{aligned} \tag{63}$$

where λ_i denotes the i-th line of the conjunction in (62).

Now let us investigate the terms Ψ_1 and Ψ_2 in (63) separately. For Ψ_1, observe that $\Diamond\Box\Diamond A = \Box\Diamond A$ and $\Diamond(\Box A \wedge \Box B) = \Diamond\Box A \wedge \Diamond\Box B$. Further we have $Q_{\delta' p_{j+1}} = Q_{\delta p_j} \wedge \overline{R}_{j+1} \subseteq Q_{\delta p_j}$ and hence

$$\Psi_1 = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\Diamond\Box (Q_{\delta p_j} \wedge \overline{R}_{p_{j+1}}) \wedge \Box\Diamond G_{p_{j+1}} \right)$$

By using the equatlity $\Diamond\Box(A \wedge B) = \Diamond\Box A \wedge \Diamond\Box B$ and the fact that $Q_{\delta p_j}$ is independent of the choice of p_{j+1} we get

$$\begin{aligned} \Psi_1 =& \Box Q_{\delta p_j} \wedge \Diamond\Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\Diamond\Box \overline{R}_{p_{j+1}} \wedge \Box\Diamond G_{p_{j+1}} \right) \\ =& \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\Diamond\Box \overline{R}_{p_{j+1}} \wedge \Box\Diamond G_{p_{j+1}} \right). \end{aligned} \tag{64}$$

To analyze Ψ_2 in (63), recall that the eventuality operator $\Diamond$ distributes over disjunctions. We can therefore move the inner disjunction over i outside and get

$$\Psi_2 = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\bigvee_{i \in \widetilde{P}_{\setminus j+1}} \left[\Diamond \left(\Box Q_{\delta' p_{j+1}} \wedge \left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right) \right) \right] \right)$$

Now observe that $\left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right) = \Diamond \left(\Box \overline{R}_i \wedge \Box\Diamond G_i \right)$ and $\Diamond(\Box A \wedge \Diamond B) = \Diamond\Box A \wedge \Diamond B$. Additionally using $Q_{\delta' p_{j+1}} = Q_{\delta p_j} \wedge \overline{R}_{p_{j+1}} \subseteq Q_{\delta p_j}$ we get

$$\Psi_2 = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\bigvee_{i \in \widetilde{P}_{\setminus j+1}} \left[\Diamond\Box (Q_{\delta p_j} \wedge \overline{R}_{p_{j+1}}) \wedge \left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right) \right] \right)$$

Now we can do the same trick as in the simplification of Ψ (see (64)) to remove the $Q_{\delta p_j}$ term inside the disjunction and get

$$\Psi_2 = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\bigvee_{i \in \widetilde{P}_{\setminus j+1}} \left[\Diamond\Box \overline{R}_{p_{j+1}} \wedge \left(\Diamond\Box \overline{R}_i \wedge \Box\Diamond G_i \right) \right] \right) \tag{65}$$

To see how we can simplify (65), let us assume that the set $\widetilde{P}_{\setminus j}$ contains three elements, e.g., $\{a, b, c\}$. Then we can expand (65) to

$$\begin{aligned}
&\Diamond\Box\overline{R}_a \wedge \left(\Diamond\Box\overline{R}_b \wedge \Box\Diamond G_b\right)\\
\vee\ &\Diamond\Box\overline{R}_a \wedge \left(\Diamond\Box\overline{R}_c \wedge \Box\Diamond G_c\right)\\
\vee\ &\Diamond\Box\overline{R}_b \wedge \left(\Diamond\Box\overline{R}_a \wedge \Box\Diamond G_a\right)\\
\vee\ &\Diamond\Box\overline{R}_b \wedge \left(\Diamond\Box\overline{R}_c \wedge \Box\Diamond G_c\right)\\
\vee\ &\Diamond\Box\overline{R}_c \wedge \left(\Diamond\Box\overline{R}_b \wedge \Box\Diamond G_b\right)\\
\vee\ &\Diamond\Box\overline{R}_c \wedge \left(\Diamond\Box\overline{R}_a \wedge \Box\Diamond G_a\right)
\end{aligned}$$

Now, we can re-order terms and get

$$\begin{aligned}
&\left(\Diamond\Box\overline{R}_b \wedge \Box\Diamond G_b\right) \wedge \left(\Diamond\Box\overline{R}_a \vee \Diamond\Box\overline{R}_c\right)\\
\vee\ &\left(\Diamond\Box\overline{R}_c \wedge \Box\Diamond G_c\right) \wedge \left(\Diamond\Box\overline{R}_a \vee \Diamond\Box\overline{R}_b\right)\\
\vee\ &\left(\Diamond\Box\overline{R}_a \wedge \Box\Diamond G_a\right) \wedge \left(\Diamond\Box\overline{R}_b \vee \Diamond\Box\overline{R}_c\right)
\end{aligned}$$

Generalizing this observation, we get the following formula equivalent to (65)

$$\Psi_2 = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\left(\Diamond\Box\overline{R}_{p_{j+1}} \wedge \Box\Diamond G_{p_{j+1}})\right) \wedge \bigvee_{j \in \widetilde{P}_{\setminus j+1}} \Diamond\Box\overline{R}_j \right) \tag{66}$$

Now recall that $A \wedge B \Rightarrow A$ for any choice of A and B. With this one can verify that $\Psi_2 \Rightarrow \Psi_1$ as the term after the disjuction over p_{j+1} in (66) implies the term after the disjuction over p_{j+1} in (64). Hence, the set of states which fulfill Ψ_1 in (64) is always larger then the set of states which fulfill Ψ_2 (66)). As both terms are connected by a conjunction in (63), we can ignore Ψ_2 in (63) and obtain

$$\Psi_{\delta p_j} = \Psi_1 = \Box Q_{\delta p_j} \wedge \bigvee_{p_{j+1} \in \widetilde{P}_{\setminus j}} \left(\Diamond\Box\overline{R}_{p_{j+1}} \wedge \Box\Diamond G_{p_{j+1}}\right). \tag{67}$$

This concludes the proof of (61) as (67) coincides with (61b).

B.4 Additional Proofs for Sec. 3.4

B.4.1 Fair Adversarial Rabin Chain Games.

Theorem (Thm. 3.6 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\mathcal{R}$ be a Rabin condition over $\mathcal{G}$ with k pairs for which the chain condition* (17) *holds. Further, let*

$$Z^* := \nu Y_0.\ \mu X_0.\ \nu Y_k.\ \mu X_k.\ \nu Y_{k-1}.\ \ldots \mu X_1.\ \textstyle\bigcup_{j=0}^{k} \widetilde{C}_j, \tag{68a}$$

$$\text{where } \widetilde{C}_j := \overline{R}_j \cap \left[\left(G_j \cap \mathrm{Cpre}(Y_j)\right) \cup \mathrm{Apre}(Y_j, X_j) \right]$$

with $G_{p_0} := \emptyset$ and $R_{p_0} := \emptyset$.

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (6). *Moreover, the fixpoint algorithm runs in $O(n^{k+2})$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.*

In this section we prove Thm. 3.6. That is, we prove that for Rabin Chain conditions, the fixpoint computing Z^* in (7) simplifies to the one in (68). This is formalized in the next proposition.

Proposition B.5. *Given the premisses of Thm. 3.6 let Z^* be the fixed point computed by* (7) *and $\widetilde{Z}^*$ the fixed point computed by* (68). *Then $Z^* = \widetilde{Z}^*$.*

If Prop. B.5 holds, we imediately see that Thm. 3.6 directly follows from Thm. 3.1. It therefore remains to prove Prop. B.5.

Similar to the soundness and completeness proof for Thm. 3.1 we prove Prop. B.5 by an induction over the nesting of fixpoints in (7) form inside to outside. Here, however we do not need to explicitly refer to counters i_j as in Prop. 3.6. Hence, we can look at permutation prefixes instead of configuration prefixes. We have the following proposition.

Proposition B.6. *Let P be the index set of the Rabin chain condition $\mathcal{R}$ in Thm. 3.6. Further, for any $j \in [0;k]$ let $\delta := p_0 p_1 \dots p_{j-1}$ be a* permutation prefix, *$\widetilde{P}_{\setminus\delta} := P \setminus \{p_0, \dots, p_{j-1}\}$ the reduced index set and $q_0 := p_j \in \widetilde{P}_{\setminus\delta}$ the current permutation index. Further, define*[6]

$$\begin{aligned} Z^*_{\delta p_j} := &\nu Y_{q_0}.\ \mu X_{q_0}.\\ &\qquad \textstyle\bigcup_{q_1 \in \widetilde{P}_{\setminus\delta p_j}} \nu Y_{q_1}.\ \mu X_{q_1}.\\ &\qquad \vdots\\ &\qquad\qquad \textstyle\bigcup_{q_n \in \widetilde{P}_{\setminus\delta p_j}\setminus\{q_1,\dots,q_{n-1}\}} \nu Y_{q_n}.\ \mu X_{q_n}.\ S_\delta \cup \left[\bigcup_{\ell=0}^{n} C_{\delta q_\ell}\right] \end{aligned} \tag{69a}$$

where $n := k - j$,

$$C_{\delta q_j} := Q_\delta \cap \bigcap_{i=0}^{\ell} \overline{R}_{q_i} \cap \left[\left(G_{q_\ell} \cap \mathrm{Cpre}(Y_{q_\ell})\right) \cup \left(\mathrm{Apre}(Y_{q_\ell}, X_{q_\ell})\right)\right], \tag{69b}$$

$Q_\delta := \bigcap_{i=0}^{j} \overline{R}_{p_i}$ and $S_{p_0 \dots p_{j-1}} := \bigcup_{b=0}^{j-1} C_{p_0 \dots p_b}$.
Then it holds that

$$Z^*_{\delta p_j} = \nu Y_{r_0}.\ \mu X_{r_0}.\ \nu Y_{r_1}.\ \mu X_{r_1}.\ \dots \nu Y_{r_n}.\ \mu X_{r_n}.\ S_\delta \cup \left[\bigcup_{\ell=0}^{n} \widetilde{C}_{\delta r_\ell}\right], \tag{70a}$$

where

$$\widetilde{C}_{\delta r_\ell} := Q_{\delta p_j} \cap \overline{R}_{r_\ell} \cap \left[\left(G_{r_\ell} \cap \mathrm{Cpre}(Y_{r_\ell})\right) \cup \left(\mathrm{Apre}(Y_{r_\ell}, X_{r_\ell})\right)\right] \tag{70b}$$

with $r_i \in \widetilde{P}_{\setminus\delta p_j}$ for all $i \in [1;n]$ s.t. $r_1 > r_2 > \dots > r_n$ and $r_0 = q_0 = p_j$.

It should be noted that Prop. B.6 needs to hold for any choice of j and δ. Further, we have slightly abused notation by not specifying the values of the fixpoint parameters used within S_δ. This is, however, not relevant for the proof of Prop. B.6 and we should interpret S_δ as a term computed by an arbitrary choice of the involved fixpoint parameters.

Now, it should be obvious that for the choice $j = 0$ we get $\delta = \varepsilon$ and $S_\delta = \emptyset$. Further, we see that in this case, we have $\widetilde{P}_{\setminus\delta p_0} = P$ which implies that $Z^*_{p_0}$ in (69) coincides with Z^* in (7). Further, as $\widetilde{P}_{\setminus\delta p_0} = P$ we must have $r_1 = k$, $r_2 = k - 1, \dots, r_k = 1$ and $r_0 = p_0 = 0$ to fulfill the requirements on r. Further $Q_{p_0} = \overline{R_0} = Q$. Therefore $Z^*_{p_0}$ in (70) coincides with Z^* in (68) in this case. Hence, proving Prop. B.6 for any j (including $j = 0$), immediately proves Prop. B.5.

In the remainder of this section we prove Prop. B.6 by an induction over j, starting with $j = k$ as the base case. Now observe that for $j = k$ we have $\widetilde{P}_{\setminus\delta p_j} = \emptyset$ and hence both (69) and (70) reduce to

[6] Observe that $\delta p_j = p_0 \dots p_{j-1} p_j$ is itself a permutation prefix.

a two-nested fixed point over the variables Y_{q_0}, X_{q_0} and Y_{r_0}, X_{r_0}, respectively, where $r_0 = q_0 = p_k$ by definition. Further, we see that $C_{\delta q_0} = \widetilde{C}_{\delta r_0}$ by definition, which immediately proves the claim of Prop. B.6 for the base case.

In the remainder of this section we prove the induction step from "j" to "$j-1$" in a series of definitions and lemmas.

Definition B.10. Let $\tilde{P} \subseteq \mathbb{N}$ be a set of n indices and $\beta = q_1 \ldots q_n$ with $q_i \in \tilde{P}$ and $q_i \neq q_j$ for all $j \neq i$ a full permutation sequence of the elements from $\tilde{P}$. For $1 \leq j \leq l \leq n$ we call $\beta_{jl} = q_j q_{j+1} \ldots q_j$ a *maximal decreasing sub-sequence* of β if (i) $q_j < q_{j+1} < \ldots < q_l$, (ii) $q_{j-1} > q_j$ or $j = 1$, and (iii) $q_l > q_{l+1}$ or $l = n$.

We see that, by definition, the first maximally decreasing sub-sequences of a permutation sequence β starts with q_1. Intuitively, decreasing sub-sequences allow to immediately utilize the properties in (17) to simplify the fixpoint expression.

Lemma B.11. *Let δ, $\widetilde{P}_{\setminus\delta}$ and $q_0 = p_j$ as in Prop. B.6, $\beta = q_1 \ldots q_n$ a full permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$ and $\beta_{jl} = q_j q_{j+1} \ldots q_j$ a* maximal decreasing sub-sequence *of β. Then*

$$\nu Y_{q_j}.\ \mu X_{q_j}.\ \ldots \nu Y_{q_l}.\ \mu X_{q_l}.\ \textstyle\bigcup_{i=j}^{l} C_{\delta q_i} = \nu Y_{q_j}.\ \mu X_{q_j}.\ C_{\delta q_j} \tag{71}$$

Proof. Let $\alpha := q_0 \ldots q_{j-1}$ and observe that

$$\begin{aligned}
C_{\delta q_j} &= Q_{\delta\alpha} \cap \left[\left(\overline{R}_j \cap G_{q_j} \cap \mathrm{Cpre}(Y_{q_j})\right) \cup \left(\overline{R}_j \cap \mathrm{Apre}(Y_{q_j}, X_{q_j})\right)\right] \\
C_{\delta q_{j+1}} &= Q_{\delta\alpha} \cap \left[\left(\overline{R}_j \cap \overline{R}_{j+1} \cap G_{q_{j+1}} \cap \mathrm{Cpre}(Y_{q_j})\right) \cup \left(\overline{R}_j \cap \overline{R}_{j+1} \cap \mathrm{Apre}(Y_{q_j}, X_{q_j})\right)\right] \\
&= Q_{\delta\alpha} \cap \left[\left(\overline{R}_j \cap G_{q_{j+1}} \cap \mathrm{Cpre}(Y_{q_j})\right) \cup \left(\overline{R}_j \cap \mathrm{Apre}(Y_{q_j}, X_{q_j})\right)\right],
\end{aligned}$$

where the simplification of $C_{\delta q_{j+1}}$ follows from $\overline{R}_j \subseteq \overline{R}_{j+1}$ (see (17)). So $C_{\delta q_j}$ and $C_{\delta q_{j+1}}$ really only differ by the G_{q_j} (resp. $G_{q_{j+1}}$) term in the first term of the disjunct. As $G_{q_j} \supseteq G_{q_{j+1}}$ (see (17)) and all terms in the first part of the disjunct are intersected, we see that $C_{\delta q_j} \supseteq C_{\delta q_{j+1}}$. With this it follows from case (iii) in Lem. B.4 that

$$\nu Y_{q_j}.\ \mu X_{q_j}. \nu Y_{q_{j+1}}.\ \mu X_{q_{j+1}}.\ C_{\delta q_j} \cup C_{\delta q_{j+1}} = \nu Y_{q_j}.\ \mu X_{q_j}.\ C_{\delta q_j}.$$

Applying this argument to all $i \in [j; l]$ proves the claim. □

Definition B.12. We say that a permutation sequence β has *chain index m* if it contains m *maximal decreasing* sub-sequences. For $\beta = q_1 \ldots q_n$ with chain index m we define its reduction $\beta_\downarrow$ as $\beta_\downarrow := r_1 ... r_m$ s.t. $r_m = q_j$ if β_{jl} is the m'th maximally decreasing sub-sequence of β.

Lemma B.13. *Let δ, $\widetilde{P}_{\setminus\delta}$ and $q_0 = p_j$ as in Prop. B.6, $\beta = q_1 \ldots q_n$ a full permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$ with chain index m and $\beta_\downarrow := r_1 ... r_m$. Then*

$$\begin{aligned}
&\nu Y_{q_0}.\ \mu X_{q_0}.\ \nu Y_{q_1}.\ \mu X_{q_1}.\ \ldots \nu Y_{q_n}.\ \mu X_{q_n} \bigcup_{j=0}^{n} C_{\delta q_j} \\
&\qquad = \nu Y_{r_0}.\ \mu X_{r_0}.\ \nu Y_{r_1}.\ \mu X_{r_1}.\ \ldots \nu Y_{r_m}.\ \mu X_{r_m} \bigcup_{l=0}^{m} C_{\delta q_l}
\end{aligned} \tag{72}$$

where $q_0 = r_0 = p_j$.

Proof. First, observe that by construction we always have $r_1 = q_1$. Hence, $Q_{\delta\alpha}$ in the proof of Lem. B.11 reduces to $Q_{\delta q_1}$ in this case. Further, consider $r_2 = q_j$ and observe that in this case $Q_{\delta\alpha} = Q_\delta \cap \bigcap_{i=0}^{j-1} \overline{R}_{q_i} = Q_{\delta q_0} \cap \overline{R}_{q_1} = Q_{\delta p_j} \cap \overline{R}_{r_1}$ as $q_1 \dots q_{j-1}$ is a maximal decreasing sub-sequence by construction. Iteratively re-applying this argument along with Lem. B.11 for every $l \in [1, m]$ therefore proves the claim. □

Now observe that we can re-apply Lem. B.13 to $\beta_\downarrow$ and reduce it even more. That means, $\beta_\downarrow$ could now again have maximal decreasing sub-sequences and we therefore can reduce it to $(\beta_\downarrow)_\downarrow$. This might again be reduceable and so forth. We therefore define the *maximal reduced permutation sequence* $\beta_\Downarrow = (((\beta_\downarrow)_\downarrow)\dots)_\downarrow = r_1 \dots r_n$ s.t. $r_1 > r_2 > \dots r_n$, i.e. the chain index of $\beta_\Downarrow$ is equivalent to its length. With this, we have the following result.

Lemma B.14. *Let δ, $\widetilde{P}_{\setminus\delta}$ and $q_0 = p_j$ as in Prop. B.6, $\beta = q_1 \dots q_n$ a full permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$ and $\beta_\Downarrow := r_1 ... r_m$ its maximal reduced permutation sequence. Then*

$$\nu Y_{q_0}.\, \mu X_{q_0}.\, \nu Y_{q_1}.\, \mu X_{q_1}.\, \dots \nu Y_{q_n}.\, \mu X_{q_n} \bigcup_{j=0}^{n} C_{\delta q_j}$$

$$= \nu Y_{r_0}.\, \mu X_{r_0}.\, \nu Y_{r_1}.\, \mu X_{r_1}.\, \dots \nu Y_{r_m}.\, \mu X_{r_m} \bigcup_{l=0}^{m} \widetilde{C}_{\delta q_l} \tag{73}$$

Proof. It follows from the definition of $\beta_\Downarrow$ and repeatably applying Lem. B.13 that

$$\nu Y_{q_0}.\, \mu X_{q_0}.\, \nu Y_{q_1}.\, \mu X_{q_1}.\, \dots \nu Y_{q_n}.\, \mu X_{q_n} \bigcup_{j=0}^{n} C_{\delta q_j}$$

$$= \nu Y_{r_0}.\, \mu X_{r_0}.\, \nu Y_{r_1}.\, \mu X_{r_1}.\, \dots \nu Y_{r_m}.\, \mu X_{r_m} \bigcup_{l=0}^{m} C_{\delta r_l}$$

Now we have by definition that $r_0 = q_0$ and $r_1 = q_1$ and therefore $C_{\delta r_0} = \widetilde{C}_{\delta r_0}$ and $C_{\delta r_1} = \widetilde{C}_{\delta r_1}$ by definition. Now recall that $r_1 > r_2$, hence $\overline{R}_{r_1} \cap \overline{R}_{r_2} = \overline{R}_{r_2}$. Iteratively applying this argument gives $C_{\delta r_l} = \widetilde{C}_{\delta r_l}$ for all $l \in [1, n]$, what proves the claim. □

Note that the only full permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$ with *chain index n* is the one where $q_1 > q_2 > \dots > q_n$, giving $\beta_\downarrow = \beta_\Downarrow = \beta$. Hence, the sequence $r_1 \dots r_n$ used in (70) is actually the *maximal permutation sequence* of $\widetilde{P}_{\setminus\delta p_j}$. We see that all other full permutation sequences γ of $\widetilde{P}_{\setminus\delta p_j}$ have *chain index m* s.t. $1 \le m < n$. As the $\widetilde{C}$ terms in (18b) do not depend on the history of permutation sequences from $\widetilde{P}_{\setminus\delta p_j}$, we see that any term constructed for a non-maximal permutation sequence is contained in the term constructed for the maximal permutation sequence. This is formalized in the next lemma.

Lemma B.15. *Let δ, $\widetilde{P}_{\setminus\delta}$ and $q_0 = p_j$ as in Prop. B.6 and let $\beta = r_1 ... r_n$ be the maximal permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$, that its $\beta = \beta_\Downarrow$. Further, let $\gamma \neq \beta$ be a full permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$ s.t. $\gamma_\Downarrow = s_1 \dots s_m$ with $m < n$. Then*

$$\nu Y_{r_1}.\, \mu X_{r_1}.\, \dots \nu Y_{r_n}.\, \mu X_{r_n} \bigcup_{l=1}^{n} \widetilde{C}_{\delta r_l} \tag{74}$$

$$\subseteq \nu Y_{s_1}.\, \mu X_{s_1}.\, \dots \nu Y_{s_m}.\, \mu X_{s_m} \bigcup_{l=1}^{m} \widetilde{C}_{\delta s_l} \tag{75}$$

Proof. As β is a full permutation sequence of $\widetilde{P}_{\setminus\delta p_j}$ we know that for any $i \in [1;m]$ there exists one $j \in [1;n]$ s.t. $s_i = r_j$. Further, as $\widetilde{C}$ does not depend on the history of the permutation sequence β and γ we see that $\widetilde{C}_{\delta s_i} = \widetilde{C}_{\delta r_j}$ in this case. As $m < n$ we see that the first line of (75) contains the fixpoint variables and $\widetilde{C}$ terms of the second line of (75). We can therefore apply Lem. B.4 (i) and (ii) which immediately proves the claim. □

Using this result, we are finally ready to prove the induction step of Prop. B.6.

Proof of Prop. B.6. Recall that Prop. B.6 trivially holds for $j = k$ which constitutes the base case of an induction over j. Now let us prove the induction step. Hence, let us assume that Prop. B.6 holds for j. Now consider "$j-1$", i.e., consider the permutation prefix $\delta' = p_0 \dots p_{j-2}$ and pick any $p_{j-1} \in P_{\delta'}$. By the induction hypothesis, we know that Prop. B.6 holds for $\delta = p_0 \dots p_{j-1}$ and any choice of $p_j \in \widetilde{P}_{\setminus\delta}$. That is, $Z^*_{\delta p_j}$ can be computed using (70). With this, the fixpoint algorithm in (69) for δ' and p_{j-1} simplifies to

$$Z^*_{\delta' p_{j-1}} = Z^*_{\delta} = \nu Y_{p_{j-1}}.\ \mu X_{p_{j-1}}.\ \bigcup_{p_j \in \widetilde{P}_{\setminus\delta}} Z^*_{\delta p_j}.$$

Here, for any choice $p_j \in \widetilde{P}_{\setminus\delta}$, the term $Z^*_{\delta p_j}$ is given by (70) where $r_0 = p_j$ and $\beta_{p_j} = r_1 \dots r_n$ being the *maximal permutation sequence* of $\widetilde{P}_{\setminus\delta p_j}$. Now observe that for $j > 0$ and any choice of p_j we see that $\gamma = r_0 \dots r_n$ is actually a permutation sequence of $\widetilde{P}_{\setminus\delta}$, but not necessarily the maximal one. However, observe that the maximal permutation sequence β of $\widetilde{P}_{\setminus\delta}$ (that is $\beta = \beta_{\Downarrow}$) is actually defined by $\beta = \tilde{p}_j \beta_{\tilde{p}_j}$ for $\tilde{p}_j := \max(\widetilde{P}_{\setminus\delta})$. With this, we can apply Lem. B.15 to see that $Z^*_{\delta p_j} \subseteq Z^*_{\delta \tilde{p}_j}$ for all $p_j \in \widetilde{P}_{\setminus\delta}$. With this we obtain

$$Z^*_{\delta' p_{j-1}} = Z^*_{\delta} = \nu Y_{p_{j-1}}.\ \mu X_{p_{j-1}}.\ Z^*_{\delta \tilde{p}_j}.$$

One can now verify that this allows us to choose $r_0 = p_{j-1}$, $r_1 = \tilde{p}_j$ and $r_2 \dots r_{n+1} = \beta_{\tilde{p}_j}$ and have $r_1 > r_2 > \dots r_{n+1}$. Hence, $Z^*_{\delta' p_{j-1}}$ can be written in the form of (70), which proves the statement. □

B.4.2 Fair Adversarial Parity Games. We now consider a *Parity* winning condition $C = \{C_1, C_2, \dots C_{2k}\}$ of colors, where each $C_i \subseteq V$ is the set of states of $\mathcal{G}$ with color i. Further, C partition's the state space, i.e., $\bigcup_{i \in [1,2k]} C_i = V$ and $C_i \cap C_j = \emptyset$ for all $i, j \in [0, 2k-1]$ s.t. $i \neq j$.

Theorem (Thm. 3.7 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and C be a Parity condition over $\mathcal{G}$ with $2k$ colors. Further, let*

$$Z^* := \nu Y_{2k}.\ \mu X_{2k-1} \dots \nu Y_2.\ \mu X_1. \tag{76}$$

$$(C_1 \cap \mathrm{Apre}(Y_2, X_1)) \cup (C_2 \cap \mathrm{Cpre}(Y_2)) \cup (C_3 \cap \mathrm{Apre}(Y_4, X_3)) \dots \cup (C_{2k} \cap \mathrm{Cpre}(Y_{2k})). \tag{77}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (19). Moreover, the fixpoint algorithm runs in $O(n^{k+1})$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

A Parity winning condition C with $2k$ colors corresponds to the Rabin chain winning condition

$$\{\langle F_2, F_3 \rangle, \dots, \langle F_{2k}, \emptyset \rangle\} \quad \text{s.t.} \quad F_i := \bigcup_{j=i}^{2k} C_j, \tag{78}$$

which has k pairs. Translating the Rabin Chain condition induced by C in (78) into a Rabin condition as in Thm. 3.1 we get the tuple $\mathcal{R} = \{\langle G_1, R_1\rangle, \ldots, \langle G_k, R_k\rangle\}$ s.t.

$$R_i = F_{2i+1} = \bigcup_{j=2i+1}^{2k} C_j \tag{79a}$$

$$\overline{R}_i = \bigcup_{j=1}^{2i} C_j \tag{79b}$$

$$G_i = F_{2i} = \bigcup_{j=2i}^{2k} C_j \tag{79c}$$

$$\overline{R}_i \cap G_i = C_{2i} \tag{79d}$$

Using these properties, the fixpoint algorithm in (68) simplifies further to the fixpoint algorithm for Parity winning conditions in (76). This is formalized in the following proposition.

Proposition B.7. *Let $\mathcal{R} = \{\langle G_1, R_1\rangle, \ldots, \langle G_k, R_k\rangle\}$ be a Rabin chain condition s.t.* (79) *holds. Further let Z^* be the fixed point computed by* (7) *and $\widetilde{Z}^*$ the the fixed point computed by* (76). *Then $Z^* = \widetilde{Z}^*$.*

Proof. Recall the fixpoint algorithm for Rabin chain games in (68), i.e.,

$$Z^* := \nu Y_0.\ \mu X_0.\ \nu Y_k.\ \mu X_k.\ \nu Y_{k-1}.\ \ldots \nu Y_1.\ \mu X_1.\ \bigcup_{j=0}^{k} \widetilde{C}_j,$$

First, observe that $R_0 = G_0 = \emptyset$ have been artificially introduced, and result in $\widetilde{C}_0 = \mathrm{Apre}(Y_0, X_0)$. Further, as we have assumed that C is such that $\bigcup_{i\in[1,2k]} C_i = V$. We can equivalently write

$$\widetilde{C}_0 = \left(\bigcup_{j=1}^{2k} C_j\right) \cup \mathrm{Apre}(Y_0, X_0) = ((C_1 \cup \ldots \cup C_{2k}) \cap \mathrm{Apre}(X_0, Y_0))$$

For $j > 0$, by using (79) we observe that the definition of $\widetilde{C}_j$ in (18b) can be written as

$$\begin{aligned}\widetilde{C}_j &= \big(C_{2j} \cap \mathrm{Cpre}(Y_j)\big) \cup \Big(\Big(\textstyle\bigcup_{l=1}^{2j} C_l\Big) \cap \mathrm{Apre}(Y_j, X_j)\Big)\\ &= \big(C_{2j} \cap \mathrm{Cpre}(Y_j)\big) \cup \big(C_1 \cap \mathrm{Apre}(Y_j, X_j)\big) \cup \ldots \cup \big(C_{2j} \cap \mathrm{Apre}(Y_j, X_j)\big)\,.\end{aligned}$$

With this, we obtain the following fixpoint equation

$$\begin{aligned}Z^* :=& \nu Y_0.\ \mu X_0.\ \nu Y_k.\ \mu X_k. \ldots \nu Y_1.\ \mu X_1.\\ & ((C_1 \cup \ldots \cup C_{2k}) \cap \mathrm{Apre}(X_0, Y_0))\\ & \cup (C_{2k} \cap \mathrm{Cpre}(Y_k)) \cup ((C_1 \cup \ldots \cup C_{2k}) \cap \mathrm{Apre}(X_k, Y_k))\\ & \cup\ \ldots\\ & \cup (C_2 \cap \mathrm{Cpre}(Y_1)) \cup ((C_1 \cup C_2) \cap \mathrm{Apre}(X_1, Y_1))\end{aligned} \tag{80}$$

Now consider Lem. B.3 and let us define

$$\begin{aligned}f(X_0, Y_0) :=& \emptyset\\ h_0(X_0, Y_0) :=& ((C_1 \cup \ldots \cup C_{2k}) \cap \mathrm{Apre}(X_0, Y_0))\\ g(X_k, Y_k) :=& (C_{2k} \cap \mathrm{Cpre}(Y_k))\\ h_1(X_k, Y_k) :=& ((C_1 \cup \ldots \cup C_{2k}) \cap \mathrm{Apre}(X_k, Y_k)).\end{aligned}$$

Then the result of the first part of the fixpoint in (80) over Y_0, X_0, Y_k, X_k corresponds to the fixed point defining $\check{Z}$ in Lem. B.3. It therefore follows from Lem. B.3 (ii) that this computation remains unchanged if we remove the fixpoint varibales X_0 and Y_0. Now we consider any j s.t. $k \geq j > 1$ and define

$$\begin{aligned}
f_j(X_j, Y_j) &:= \big(C_{2j} \cap \mathrm{Cpre}(Y_j)\big) \cup (C_{2j-1} \cap \mathrm{Apre}(Y_j, X_j)) \cup (C_{2j} \cap \mathrm{Apre}(Y_j, X_j))\\
h_j(X_j, Y_j) &:=(C_1 \cap \mathrm{Apre}(Y_j, X_j)) \cup \ldots \cup (C_{2j-1} \cap \mathrm{Apre}(Y_j, X_j))\\
g_{j-1}(X_{j-1}, Y_{j-1}) &:= \big(C_{2(j-1)} \cap \mathrm{Cpre}(Y_{j-1})\big)\\
h_{j-1}(X_{j-1}, Y_{j-1}) &:=(C_1 \cap \mathrm{Apre}(Y_{j-1}, X_{j-1})) \cup \ldots \cup (C_{2(j-1)-1} \cap \mathrm{Apre}(Y_{j-1}, X_{j-1})).
\end{aligned}$$

It is easy to see that indeed $h_j(X, Y) = h_{j-1}(X, Y)$ for all $X, Y \subseteq V$. We can therefore start with $j = k$ and iteratively remove the h_j terms from the fixpoint equation in (80). This yields

$$\begin{aligned}
Z^* :=&\nu Y_k.\ \mu X_k.\ \ldots\ \nu Y_1.\ \mu X_1.\\
&\cup (C_{2k} \cap \mathrm{Cpre}(Y_k)) \cup (C_{2k-1} \cap \mathrm{Apre}(Y_k, X_k)) \cup (C_{2k} \cap \mathrm{Apre}(Y_k, X_k))\\
&\cup\ \ldots\\
&\cup (C_2 \cap \mathrm{Cpre}(Y_1)) \cup (C_1 \cap \mathrm{Apre}(Y_1, X_1)) \cup (C_2 \cap \mathrm{Apre}(Y_1, X_1))
\end{aligned} \tag{81}$$

Now we recall from Lem. B.1 and Lem. B.2 that for all j s.t. $k \geq j \geq 1$ we have

$$(C_{2j} \cap \mathrm{Cpre}(Y_j)) \cup (C_{2j} \cap \mathrm{Apre}(Y_j, X_j)) = (C_{2j} \cap \mathrm{Cpre}(Y_j)).$$

With this, we obtain

$$\begin{aligned}
Z^* :=&\nu Y_k.\ \mu X_k.\ \ldots\ \nu Y_1.\ \mu X_1.\\
&\cup (C_{2k} \cap \mathrm{Cpre}(Y_k)) \cup (C_{2k-1} \cap \mathrm{Apre}(Y_k, X_k))\\
&\cup\ \ldots\\
&\cup (C_2 \cap \mathrm{Cpre}(Y_1)) \cup (C_1 \cap \mathrm{Apre}(Y_1, X_1))
\end{aligned} \tag{82}$$

With this the claim follows by renaming the fixpoint variables accordingly. □

B.4.3 Fair Adversarial Generalized Co-Büchi Games.

Theorem (Thm. 3.8 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\mathcal{A}$ be a generalized Co-Büchi winning condition $\mathcal{G}$ with r pairs. Further, let*

$$Z^* := \nu Y_0.\ \mu X_0.\ \bigcup_{a \in [1;r]} \nu Y_a.\ \mathrm{Apre}(Y_0, X_0) \cup (\overline{A}_a \cap \mathrm{Cpre}(Y_a)). \tag{83}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (23). Moreover, the fixpoint algorithm runs in $O(rn^2)$ symbolic steps, and a memoryless winning strategy for Player 0 can be extracted from it.

In this section we prove Thm. 3.8. That is, we prove that for generalized Co-Büchi conditions, the fixpoint computing Z^* in (7) simplifies to the one in (83). This is formalized in the next proposition.

Proposition B.8. *Let $\mathcal{R} = \{\langle G_1, R_1 \rangle, \ldots, \langle G_k, R_k \rangle\}$ be a Rabin condition s.t. (24) holds. Further let Z^* be the fixed point computed by (7) and $\widetilde{Z}^*$ the fixed point computed by (83). Then $Z^* = \widetilde{Z}^*$.*

Proof. Now consider the flattening of (7) in (57) for $\widetilde{\mathcal{R}}$. Then we see that for all $j > 0$ we have

$$\begin{aligned}
C_{\delta p_j i_j} &:= \left(Q_{\delta p_j} \cap \mathrm{Cpre}(Y^*_{\delta p_j})\right) \cup \left(Q_{\delta p_j} \cap \mathrm{Apre}(Y^*_{\delta p_j}, X^{i_j - 1}_{\delta p_j})\right)\\
&= Q_{\delta p_j} \cap \left(\mathrm{Cpre}(Y^*_{\delta p_j}) \cup \mathrm{Apre}(Y^*_{\delta p_j}, X^{i_j - 1}_{\delta p_j})\right)
\end{aligned}$$

and we always have $X^{i_j-1}_{\delta p_j} \subseteq Y^*_{\delta p_j}$. With this, it follows from Lem. B.1 that

$$C_{\delta p_j i_j} = Q_{\delta p_j} \cap \mathrm{Cpre}(Y^*_{\delta p_j}) \tag{84}$$

for all δ, p_j and i_j with $j > 0$.

Now observe that for $\delta' = \delta p_j i_j$ and all $p_{j+1} \in P \setminus \{p_0, \ldots, p_j\}$ we have

$$Q_{\delta' p_{j+1}} = Q_{\delta p_j} \cap \overline{R}_{p_{j+1}} \subseteq Q_{\delta p_j}.$$

It further follows from the structure of the fixed point in (7) that

$$Y^*_{\delta p_j} = \bigcup_{i_j>0} X^{i_j}_{\delta p_j} = \bigcup_{i_j>0} \bigcup_{p_{j+1} \in P \setminus p_0,\ldots,p_j} Y^*_{\delta' p_{j+1}}$$

and therefore

$$Y^*_{\delta' p_{j+1}} \subseteq Y^*_{\delta p_j}.$$

With this we get

$$C_{\delta' p_{j+1} i_{j+1}} \subseteq C_{\delta p_j i_j}$$

for all δ, p_j and i_j with $j > 0$. Then it follows from Lem. B.4 (iii) that for every permutation sequence $\delta = p_0 p_1 \ldots p_k$ the union over all $C's$ terms simplifies to two terms, one for $j = 0$ and one for $j = 1$. Using this insight, we see that for the particular Rabin condition $\widetilde{\mathcal{R}}$ the fixpoint algorithm in (7) simplifies to

$$\nu Y_0.\ \mu X_0.\ \bigcup_{p_1 \in P} \nu Y_{p_1}.\ \mu X_{p_1}.\ C_{p_0} \cup C_{p_1}. \tag{85}$$

Now recalling that C_{p_1} simplifies to $\overline{A}_a \cap \mathrm{Cpre}(Y_a)$ for $a = p_1$ (see (84)) if (24) holds, and that $C_{p_0} = \mathrm{Apre}(Y_0, X_0)$ as $R_0 = Q_0 = \emptyset$, we see that (85) coincides with (83). □

B.5 Additional Proofs for Sec. 4

B.5.1 Proof of Thm. 4.1.

Theorem (Thm. 4.1 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\langle \mathcal{F}, Q \rangle$ with $\mathcal{F} = \{\ ^1F, \ldots, \ ^sF\}$ a safe generalized Büchi winning condition. Further, let*

$$Z^* := \nu Y.\ \bigcap_{b \in [1;s]} \mu\ ^bX.\ Q \cap \Big[(\ ^bF \cap \mathrm{Cpre}(Y)) \cup \mathrm{Apre}(Y,\ ^bX) \Big]. \tag{86}$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (27). Moreover, the fixpoint algorithm runs in $O(sn^2)$ symbolic steps, and a finite-memory winning strategy for Player 0 can be extracted from it.

Our goal is to prove Thm. 4.1 by a reduction to Thm. 3.2 and Thm. 3.3. We therefore first show that a similar construction of an extended fixed point $\widetilde{Z}$ as in (52) within the proof of Thm. 3.2 also works for the generalized case. This is formalized in the following proposition.

Proposition B.9. *Given the premisses of Thm. 4.1, let*

$$Z^* := \nu Y.\ \bigcap_{b \in [1;s]} \mu\ ^bX.\ (\ ^bF \cap \mathrm{Cpre}(Y)) \cup \mathrm{Apre}(Y,\ ^bX) \tag{87a}$$

and

$$\widetilde{Z}^* := \nu Y.\ \bigcap_{b \in [1;s]} \nu\ ^bY.\ \mu\ ^bX.\ (\ ^bF \cap \mathrm{Cpre}(Y)) \cup \mathrm{Apre}(\ ^bY,\ ^bX). \tag{87b}$$

Then $\widetilde{Z}^* = Z^*$.

However, as in (87) a conjunction is used to update Y, the proof is not as straight forward as for (52). We therefore separately show for both equations (87a) and (87b) that, upon termination, we have $Y^* = {}^bX^*$ for all $b \in [1;s]$. Both claims are formalized in Lem. B.16 and Lem. B.17, respectively.

Lemma B.16. *Given the premises of Prop. B.9, let ${}^bX^i$ be the set computed in the i-th iteration over the fixpoint variable ${}^bX$ in* (87a) *during the last iteration over Y, i.e., $Y = Z^*$ already. Further, we define ${}^bX^0 = \emptyset$ and ${}^bX^* := \bigcup_{i>0} {}^bX^i$. Then it holds that $Z^* = {}^bX^*$ for all $b \in [1;s]$.*

Proof. We fix $Y = Z^*$ and $b \subseteq [1;s]$ and observe from (87a) that

$${}^bX^0 = ({}^bF \cap \mathrm{Cpre}(Z^*))$$

and therefore

$$\begin{aligned}{}^bX^1 &= {}^bX^0 \cup ({}^bF \cap \mathrm{Cpre}(Z^*)) \cup \mathrm{Apre}(Z^*, {}^bX^0)\\ &= ({}^bF \cap \mathrm{Cpre}(Z^*)) \cup \mathrm{Apre}(Z^*, {}^bX^0) \supseteq {}^bX^0\end{aligned}$$

With this, we have in general that

$$\begin{aligned}{}^bX^{i+1} &= {}^bX^i \cup ({}^bF \cap \mathrm{Cpre}(Z^*)) \cup \mathrm{Apre}(Z^*, {}^bX^i)\\ &=({}^bF \cap \mathrm{Cpre}(Z^*)) \cup \mathrm{Apre}(Z^*, {}^bX^i)\end{aligned}$$

which implies ${}^bX^{i+1} \supseteq {}^bX^i$. Hence, ${}^bX^* := \bigcup_{i\in[0,i_{max}]} {}^bX^i = {}^bX^{i_{max}}$, and therefore, in particular

$${}^bX^* = ({}^bF \cap \mathrm{Cpre}(Z^*)) \cup \mathrm{Apre}(Z^*, {}^bX^*). \tag{88}$$

By recalling that $Z^* = \bigcap_b {}^bX^*$ we see that $Z^* \subseteq {}^bX^*$.

For the inverse direction, we use the observation $Z^* \subseteq {}^bX^*$ together with Lem. B.2 to see that $\mathrm{Apre}(Z^*, {}^bX^*) = \mathrm{Cpre}({}^bX^*)$. With this $({}^bF \cap \mathrm{Cpre}(Z^*)) \subseteq \mathrm{Cpre}(Z^*) \subseteq \mathrm{Cpre}({}^bX^*) = \mathrm{Apre}(Z^*, {}^bX^*)$ and hence (88) reduces to

$${}^bX^* = \mathrm{Cpre}({}^bX^*) \supseteq \mathrm{Cpre}(Z^*).$$

As the last equality holds for all $b \subseteq [1;s]$ we see that

$$Z^* = \bigcap_b {}^bX^* = \bigcap_b \mathrm{Cpre}({}^bX^*) \supseteq \mathrm{Cpre}(Z^*). \tag{89}$$

We can now use (89) to proof that $Z^* \supseteq {}^bX^*$ also holds. To show this, we pick a vertex $v \in {}^bX^*$ and prove that $v \in Z^*$. To that end, observe that either (i) $v \in ({}^bF \cap \mathrm{Cpre}(Z^*)) \subseteq \mathrm{Cpre}(Z^*) \subseteq Z^*$ which immediately proves the statement, or (ii) $v \in \mathrm{Apre}(Z^*, {}^bX^*)$. If (ii) holds we again have two cases. Either (a) $v \in \mathrm{Cpre}({}^bX^*)$ which implies that there exists a finite sequence $\mathrm{Cpre}(\mathrm{Cpre}(\ldots \mathrm{Cpre}({}^bX^1)\ldots))$ where ${}^bX^1 = {}^bF \cap \mathrm{Cpre}(Z^*) \subseteq \mathrm{Cpre}(Z^*) \subseteq Z^*$ and therefore $v \in \mathrm{Cpre}(\mathrm{Cpre}(\ldots \mathrm{Cpre}(Z^*)\ldots)) \subseteq Z^*$. Finally we could have (b) that $v \in \mathrm{Pre}^{\exists}_l({}^bX^*) \cap \mathrm{Pre}^{\forall}_1(Z^*) \subseteq \mathrm{Pre}^{\forall}_1(Z^*) \subseteq \mathrm{Cpre}(Z^*) \subseteq Z^*$, which again proves the statement. □

Lemma B.17. *Given the premises of Prop. B.9, let ${}^bY^i$ be the set computed in the i-th iteration over the fixpoint variable ${}^bY$ in* (87b) *during the last iteration over Y, i.e., $Y = \widetilde{Z}^*$ already. Further, we define ${}^bY^0 = V$ and ${}^bY^* := \bigcap_{i>0} {}^bY^i$. Then it holds that $\widetilde{Z}^* = {}^bY^*$ for all $b \in [1;s]$.*

Proof. Recall that $\widetilde{Z}^* = \bigcap_b {}^bY^*$ from the structure of the fixpoint algorithm in (87b). To prove $\widetilde{Z}^* = {}^bY^*$ for all $b \in [1;s]$ it therefore suffices to show that ${}^bY^* = {}^{b'}Y^*$ for any two $b, b' \in [1;s]$ s.t. $b \neq b'$.

Towards this goal, recall from Thm. 3.3 that ${}^bY^*$ is exactly the set of states from which player 0 can win a fair adversarial reachability game with target ${}^bT := {}^bF \cap \mathrm{Cpre}(\widetilde{Z}^*)$. However, every state

$v \in {}^{b}T$ allows player 0 to force the game to a state $v' \in \widetilde{Z}^* = \bigcap_{b'} {}^{b'}Y^*$. Therefore, by definition player 0 has a strategy to reach a state $v' \in {}^{b'}Y^*$ from any state $v \in {}^{b}Y^*$ for any $b' \in [1;s]$ s.t. $b \neq b'$. As, however ${}^{b'}Y^*$ is defined as the winning region of player 0 w.r.t. the goal set ${}^{b'}T := {}^{b'}F \cap \mathrm{Cpre}(\widetilde{Z}^*)$, we know that there actually exists a player 0 strategy to drive the game from any $v \in {}^{b}Y^*$ to ${}^{b'}T$, and therefore, by definition ${}^{b}Y^* \subseteq {}^{b'}Y^*$. As this inclusion holds mutually for all $b, b' \in [1;s]$ s.t. $b \neq b'$ we have that ${}^{b}Y^* = {}^{b'}Y^*$. With this, it immediately follows that $\widetilde{Z}^* = {}^{b}Y^*$ for all $b \in [1;s]$. □

With Lem. B.16 and Lem. B.17 in place, we see that for every update of Y the structure of the fixed point over ${}^{b}Y$ and ${}^{b}X$ upon termination of ${}^{b}Y$ coincides with the one in (52). With this, Prop. B.9 immediately follows from Lem. B.5.

Using Prop. B.9 we know that (87a) and (87b) compute the same set. Hence, we can use (87b) instead of (86) to prove Thm. 4.1. This allows us to simply reduce the proof of Thm. 4.1 to Thm. 3.2 and Thm. 3.3 as formalized below.

Proof of Thm. 4.1. **Soundness & Completeness:** Let us define $Z^*(\langle T, Q\rangle)$ to be the set of states computed by the fixpoint algorithm in (12). Then it follows from (87b) that

$$\widetilde{Z}^* = \nu Y. \bigcap_{b\in[1;s]} Z^*(\langle Q \cap {}^{b}F \cap \mathrm{Cpre}(Y), Q\rangle).$$

In particular, it follows from Lem. B.17 that

$$\widetilde{Z}^* = Z^*(\langle Q \cap {}^{b}F \cap \mathrm{Cpre}(\widetilde{Z}^*), Q\rangle)\ \forall b \in [1;s].$$

Now let us define ${}^{b}\mathcal{W}$ to be the fair adversarial winning state set for

$${}^{b}\psi = \Box Q \wedge \Box\Diamond\, {}^{b}F.$$

With this, it follows from Thm. 3.2 that $\widetilde{Z}^* = {}^{b}\mathcal{W}$ for all $b \in [1;s]$. Therefore, we obviously have $\bigcap_{b\in[1;s]} {}^{b}\mathcal{W} = \widetilde{Z}^*$. Now let $\mathcal{W}$ be the fair adversarial winning set w.r.t.

$$\psi = \Box Q \wedge \bigwedge_{b\in[1;s]} \Box\Diamond({}^{b}F).$$

(compare (26)). Then we always have $\mathcal{W} \subseteq \bigcap_{b\in[1;s]} {}^{b}\mathcal{W}$ which immediately implies $\mathcal{W} \subseteq \widetilde{Z}^*$. However, as ${}^{a}\mathcal{W} = {}^{b}\mathcal{W}$ for all $a, b \in [1;s]$, we know that ψ holds for all $v \in \widetilde{Z}^*$, hence $Z^* \subseteq \mathcal{W}$.

Strategy construction: We can define a rank function for every b as in (46) within the proof of Thm. 3.3 (see App. B.2.1), i.e.,

$${}^{b}\mathrm{rank}(v) = i \quad \text{iff} \quad v \in {}^{b}X^i \setminus {}^{b}X^{i-1}. \tag{90}$$

Then, we have a different strategy, ${}^{b}\rho_0$, which is defined via (46) (see App. B.2.1) using the corresponding ^{b}rank function. With this, we define a new strategy ρ which circles through all possible goal sets in a pre-defined order. That is

$$\rho_0(v, b) = \begin{cases} {}^{b}\rho_0(v) & v \notin {}^{b}F \\ {}^{b^+}\rho_0(v) & v \in {}^{b}F \end{cases} \tag{91}$$

where $b^+ = b + 1$ if $b < s$ and $b^+ = 1$ if $b = s$.

The strategy in (91) is obviously winning for ψ in (26) as every ${}^{b}\rho_0$ is a winning strategy for ${}^{b}\psi$ (from Thm. 3.2) and upon reaching ${}^{b}F$ we know that the respective state v is also contained in $\mathrm{Cpre}(\widetilde{Z}^*)$ where $\widetilde{Z}^* = {}^{b^+}Y^*$. Now it follows from the definition of Cpre that $\mathrm{Cpre}({}^{b^+}Y^*) \subseteq {}^{b^+}Y^*$, hence, allowing to apply ${}^{b^+}\rho_0$ upon reaching ${}^{b}F$. □

B.5.2 Proof for Thm. 4.2.

Theorem (Thm. 4.2 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $\widetilde{\mathcal{R}}$ be a generalized Rabin condition over $\mathcal{G}$ with index set $P = [1;k]$. Further, let*

$$\begin{aligned} Z^* :=&\nu Y_0.\ \mu X_0. \\ &\bigcup_{p_1 \in P} \nu Y_{p_1}.\ \bigcap_{l_1 \in [1;m_{p_1}]} \mu\ {}^{l_1}X_{p_1}. \\ &\quad\ddots \\ &\qquad \bigcup_{p_k \in P\setminus\{p_1,\ldots,p_{k-1}\}} \nu Y_{p_k}.\ \bigcap_{l_k \in [1;m_{p_k}]} \mu\ {}^{l_k}X_{p_k}.\ \bigcup_{j=0}^{k} {}^{l_j}C_{p_j}, \end{aligned} \tag{92a}$$

where

$${}^{l_j}C_{p_j} := \left(\bigcap_{i=0}^{j} \overline{R}_{p_i}\right) \cap \left[\left({}^{l_j}G_{p_j} \cap \mathrm{Cpre}(Y_{p_j})\right) \cup \mathrm{Apre}(Y_{p_j},\ {}^{l_j}X_{p_j})\right]$$

with $p_0 = 0$, $G_{p_0} := \{\emptyset\}$ and $R_{p_0} := \emptyset$. Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (26). Moreover, the fixpoint algorithm runs in $O(n^{k+2}k!m_1 \ldots m_k)$ symbolic steps, and a finite-memory winning strategy for Player 0 can be extracted from it.

We show how the proof of Thm. 3.1 in App. B.3 needs to be adapted in order to prove the generalized version of Thm. 3.1, namely Thm. 4.2, instead.

Strategy Construction: Similar to the finite-memory strategy constructed for generalized Büchi games in App. B.5.1, the strategy for generalized Rabin games needs to remember the index of all the goal sets currently "chaised" for each permutation index up to p_j. To formalize this, we define the set of full goal chain sequences for a given generalized Rabin specification $\widetilde{\mathcal{R}}$ by

$$\Phi(\widetilde{\mathcal{R}}) := \{\ell_0\ell_1 \ldots \ell_k \mid \ell_0 = 1,\ \ell_j \in [0; m_j]\}. \tag{93}$$

If $\widetilde{\mathcal{R}}$ is clear from the context we simply write Φ. Given a goal chain prefix $\phi := \ell_0\ell_1 \ldots \ell_{j-1}$ we can now construct a ranking for each such prefix, using the flattening of (92) instead of (7). This yields the following proposition which follows from Prop. B.1 by simply annotating all terms with the goal chain prefix ϕ.

Proposition B.10. *Let $\delta = p_0 i_0 \ldots p_{j-1} i_{j-1}$ be a* configuration prefix, *$\phi := \ell_0\ell_1 \ldots \ell_{j-1}$ a* goal chain prefix, *$p_j \in P \setminus \{p_1, \ldots, p_{j-1}\}$ the next permutation index, $\ell_j \in [1; m_{p_j}]$ the next goal set and $i_j > 0$ a counter for p_j. Then the flattening of (92) for this configuration and goal prefix is given by*

$${}^{\phi\ell_j}X^{i_j}_{\widetilde{\delta p_j}} = \underbrace{{}^{\phi}S_\delta \cup {}^{\ell_j}C_{\delta p_j i_j}}_{{}^{\phi\ell_j}S_{\delta p_j i_j}} \cup {}^{\phi\ell_j}\mathcal{A}_{\delta p_j i_j} \tag{94a}$$

where

$$Q_{p_0 \dots p_a} := \bigcap_{b=0}^{a} \overline{R}_{p_b}, \tag{94b}$$

$$^{\ell_a}C_{\delta p_a i_a} := \left(Q_{\delta p_a} \cap {}^{\ell_a}G_{p_a} \cap \mathrm{Cpre}(Y^*_{\delta p_a})\right) \cup \left(Q_{\delta p_a} \cap \mathrm{Apre}(Y^*_{\delta p_a}, {}^{\ell_a}X^{i_a-1}_{\delta p_a})\right)$$

$$^{\ell_0 \dots \ell_a}S_{p_0 i_0 \dots p_a i_a} := \bigcup_{b=0}^{a} {}^{\ell_b}C_{p_0 i_0 \dots p_b i_b}, \tag{94c}$$

$$^{\phi \ell_i}A_{\delta p_j i_j} := \bigcup_{p_{j+1} \in P \setminus \{p_1, \dots, p_j\}} \left(\bigcap_{\ell_{j+1} \in [1; m_{p_{j+1}}]} \left(\bigcup_{i_{j+1} > 0} \left({}^{\phi \ell_j \ell_{j+1}}X^{i_{j+1}}_{\delta p_j i_j p_{j+1}} \setminus {}^{\phi \ell_i}S_{\delta p_j i_j}\right)\right)\right). \tag{94d}$$

Again we see that this flattening follows directly from the structure of the fixpoint algorithm in (92) and the definition of ${}^{l_j}C_{p_j}$ in (29b). Using the flattening of (92) in (94) we can define a ranking function for each goal chain prefix ϕ identical to Def. B.6. That is, given the premises of Prop. B.10, we define ${}^{\phi \ell_j}R : V \to 2^{\widetilde{D}}$ s.t. (i) $\infty \in {}^{\phi \ell_j}R(v)$ for all $v \in V$, and (ii) $\delta p_j i_j \underline{\gamma} \in {}^{\phi \ell_j}R(v)$ iff $v \in {}^{\phi \ell_j}S_{\delta p_j i_j}$. The ranking function ${}^{\phi}\mathrm{rank} : V \to D$ is then again defined as in Def. B.6 s.t. ${}^{\phi}\mathrm{rank} : v \mapsto \min\{{}^{\phi}R(v)\}$. Similarly, we can define a memoryless winning strategy for every fixed goal sequence ϕ as in (58). That is,

$$^{\phi}\rho_0(v) := \min_{(v,w) \in E} ({}^{\phi}\mathrm{rank}(w)). \tag{95}$$

Now, similar to the proof of Thm. 4.1 (see Sec. 4.1) we can "stack" these memoryless winning strategies to define a new strategy with finite memory which circles through all possible goal sets in a pre-defined order. That is

$$\rho_0(v, \phi \ell_j) := \begin{cases} {}^{\phi \ell_j}\rho_0(v) & v \notin {}^{\ell_j}F \\ {}^{\phi \ell_j^+}\rho_0(v) & v \in {}^{\ell_j}F \end{cases} \tag{96}$$

where $\ell_j^+ := \ell_j + 1$ if $\ell_j < m_{p_j}$ and $\ell_j^+ := 1$ if $\ell_j = m_{p_j}$.

Using this goal chain dependent ranking function, the proof of soundness and completeness of (92) along with the proof that ρ_0 in (96) is indeed a winning strategy for player 0 in the fair adversarial generalized Rabin game, follows exactly the same lines as the proof in App. B.3. That is, we iteratively consider instances of the flattening in (94), starting with $j = k$ as the base case, and doing an induction from "$j + 1$" to "j". To this end, we consider a *generalized* local winning condition which refers not only to the current configuration-prefix $\delta = p_0 i_0 \dots p_{j-1} i_{j-1}$ but also to the current goal chain prefix $\phi := \ell_0 \dots \ell_{j-1}$. Hence, (59) gets modified to

$$^{\phi}\psi_{\delta p_j} := \left(\begin{array}{ll} & Q_{\delta p_j} \,\mathcal{U}\, {}^{\phi}S_{\delta} \\ \vee & \Box Q_{\delta p_j} \wedge \bigwedge_{\ell_j \in [1; m_{p_j}]} \Box\Diamond\, {}^{\ell_j}G_{p_j} \\ \vee & \Box Q_{\delta p_j} \wedge \left(\bigvee_{i \in \widetilde{P}_{\setminus j}} \left(\Diamond\Box \overline{R}_i \wedge \bigwedge_{b \in [1; m_i]} \Box\Diamond\, {}^{b}G_i \right)\right) \end{array} \right) \tag{97}$$

where $\widetilde{P}_{\setminus j} = P \setminus \{p_0, \dots, p_j\}$. With this, it becomes obvious that the proof of soundness, completness and the winning strategy for Thm. 4.2 follows exactly the same reasoning as in App. B.3 while additionally using Thm. 4.1 to reason about the conjunction over goal sets.

The only remaining part to be shown concerns the last line of ${}^{\phi}\psi_{\delta p_j}$. For this, we recall from App. B.3.2 that the induction step from "$j + 1$" to "j" relies on the fact that

$$^{\phi \ell_j}\Psi_{\delta p_j} := \Box Q_{\delta p_j} \wedge \Diamond \left(\bigvee_{p_{j+1} \in P \setminus \{p_1, \dots, p_j\}} {}^{\phi'}\psi'_{\delta' p_{j+1}} \right) \tag{98}$$

is indeed equivalent to the last line of ${}^{\phi}\psi_{\delta p_j}$, where ${}^{\phi'}\psi'_{\delta' p_{j+1}}$ denotes the last two lines of ${}^{\phi'}\psi_{\delta' p_{j+1}}$ with $\phi' := \phi\ell_j$ and $\delta' := \delta p_j$.

For (non-generalized) Rabin games this equivalence is proved in App. B.3.6. It can be seen by inspection within this proof, that using a conjunction over goal sets instead of a single goal set within the second and third line of ${}^{\phi}\psi_{\delta p_j}$ does not change any step in the derivation. Therefore, the same derivation can be used in the generalized case and is therefore omitted. This concludes the proof of Thm. 4.2.

B.5.3 Proof of Thm. 4.3.

Theorem (Thm. 4.3 restated for convenience). *Let $\mathcal{G}^\ell = \langle \mathcal{G}, E^\ell \rangle$ be a game graph with live edges and $(\mathcal{A}, \mathcal{F})$ a GR(1) winning condition. Further, let*

$$Z^* = \nu Y_k.\ \bigcap_{b\in[1;s]} \mu\, {}^bX_k.\ \bigcup_{a\in[1;r]} \nu Y_a.\ (F_b \cap \mathrm{Cpre}(Y_k)) \cup \mathrm{Apre}(Y_k,\ {}^bX_k) \cup (\overline{A}_a \cap \mathrm{Cpre}(Y_a)).$$

Then Z^* is equivalent to the winning region $\mathcal{W}$ of Player 0 in the fair adversarial game over $\mathcal{G}^\ell$ for the winning condition φ in (30). Moreover, the fixpoint algorithm runs in $O(n^2rs)$ symbolic steps, and a finite-memory winning strategy for Player 0 can be extracted from it.

Within this section we proof Thm. 4.3. That is, we prove that for GR(1) winning conditions, the fixpoint computing Z^* in (92) simplifies to the one in (32). This is formalized in the next proposition.

Proposition B.11. *Let $\widetilde{\mathcal{R}}$ be a generalized Rabin condition with k pairs s.t.* (31) *holds for $r := k-1$. Further let Z^* be the fixed point computed by* (92) *and $\tilde{Z}^*$ the fixed point computed by* (32). *Then $Z^* = \tilde{Z}^*$.*

If Prop. B.11 holds, we immediately see that Thm. 4.3 directly follows from Thm. 4.2. It therefore remains to prove that Prop. B.11 holds.

Proof. First, consider an arbitrary permutation sequence $\delta = p_0 \ldots p_k$. Then we know that there exists exactly one $j > 0$ s.t. $p_j = k$ and all other indices come from the set $[1;r]$. We can therefore define $\gamma' = p_1 \ldots p_{j+1}$ and $\gamma'' = p_{j+1} \ldots p_k$ s.t. $p_i \in [1;r]$ for all $i \neq j$. We note that $\gamma' = \varepsilon$ if $j = 1$ and $\gamma'' = \varepsilon$ if $j = k$. With this we have $\delta = p_0\gamma' p_j \gamma''$.

By inspecting (31) we see that the first r pairs of the generalized Rabin condition induced by the GR(1) specification actually form a Generalized Co-Büchi condition (compare (24) in Sec. 3.4). Hence, given a permutation sequence $\delta = p_0\gamma' p_j\gamma''$ we can use the same reasoning as in the proof of Thm. 3.8 in App. B.4.3 to see that

$$C_{p_1} \supseteq \ldots \supseteq C_{p_{j-1}} \text{ and } C_{p_{j+1}} \supseteq \ldots \supseteq C_{p_k}. \tag{99}$$

Now recall from the proof of Thm. 3.6 in App. B.4.1 that these inclusions allow to recursively apply Lem. B.4 to delete all C terms which are included in either C_{p_1} or $C_{p_{j+1}}$ along with the fixpoint variables used within these terms (compare Lem. B.11 where now γ' and γ'' are interpreted as decreasing sub-sequences). Applying these simplifications to (92) (in exactly the same manner as these simplifications where applied to (7) in the proof of Thm. 3.6) results in a simpler fixpoint algorithm where all permutation sequences have the form $\delta = 0q_1kq_2$ with $q_1 \neq q_2$ and $q_1, q_2 \in [1;r]$ (here q_1 and q_2 correspond to p_1 and p_{j+1} in (99), and k corresponds to p_j).

Now we can inspect (31) again to see that $R_i \supseteq R_k$ and $G_i \supseteq {}^bG_{p_j}$ for all $i \in [1;r]$ and $b \in [1;s]$. This can be understood as a "generalized Rabin chain condition" (compare (17) in Sec. 3.4). Hence, we can apply Lem. B.11 one more time, now to the "decreasing sub-sequence" q_1k within every permutation sequence. Again, utilizing this argument iteratively in (92) yields a simpler fixpoint algorithm which only contains permutation sequences $\delta = 0ka$ with $a \in [1;r]$. This proves that Z^* is equivalent to the set

$$\nu Y_0.\ \mu X_0.\ \nu Y_k.\ \bigcap_{b\in[1;s]} \mu\ {}^{b}X_0.\ \bigcup_{a\in[1;r]} \nu Y_a.\ \mu X_a.\quad C_{p_0} \cup {}^{b}C_k \cup C_a.$$

Now inserting the simplifications for terms from the generalized Co-Büchi part (see (84) in App. B.4.3) and using $R_0 = G_0 = \emptyset$, we obtain

$$\nu Y_0.\ \mu X_0.\ \nu Y_k.\ \bigcap_{b\in[1;s]} \mu\ {}^{b}X_0.\ \bigcup_{a\in[1;r]} \nu Y_a.$$
$$\mathrm{Apre}(Y_0, X_0) \cup (\ {}^{b}F \cap \mathrm{Cpre}(Y_k)) \cup \mathrm{Apre}(Y_k,\ {}^{b}X_k) \cup (\overline{A}_a \cap \mathrm{Cpre}(Y_a)).$$

Now we can apply Lem. B.4 (iii) again to remove the first occurrence of the Apre term to obtain the same expression as in (32). This concludes the proof. □

B.6 Additional Proofs for Sec. 5

B.6.1 Preliminaries. $1^1/_2$-**player game:** A special case of $2^1/_2$-player game graphs is a *Markov Decision Process* (MDP) or $1^1/_2$-*player game*, which is obtained by assuming that every Player 0 vertex in V_0 has only one outgoing edge.[7] Analogously to the $2^1/_2$-player games, for a given $1^1/_2$-player game graph $\mathcal{G}$, we use the notation $P_{v^0}^{\rho_1}(\mathcal{G} \models \varphi)$ to denote the probability of occurrence of the event $\mathcal{G} \models \varphi$ when the runs initiate at v^0 and when Player 1 uses the strategy ρ_1.

Role of end components in $1^1/_2$-**player game:** Limiting behaviors in a $1^1/_2$-player game can be characterized using the structure of the underlying game graph. We summarize one key technical argument in the following.

Let $\mathcal{G} = \langle V, V_0, V_1, V_r, E\rangle$ be a $1^1/_2$-player game graph. A set of vertices $U \subseteq V$ is called *closed* if (1) for every $v \in U \cap V_r$, $E(v) \subseteq U$, and (2) for every $v \in U \cap (V_0 \cup V_1)$, $E(v) \cap U \neq \emptyset$. A closed set of vertices U induces a *subgame graph* $(V', V_0', V_1', V_r', E')$, denoted by $\mathcal{G} \downarrow U$, which is itself a $1^1/_2$-player game graph and is defined as follows:

- $V' = U$,
- $V_0' = U \cap V_0$,
- $V_1' = U \cap V_1$,
- $V_r' = U \cap V_r$, and
- $E' = E \cap (U \times U)$.

A set of vertices $U \subset V$ of a $1^1/_2$-player game graph $\mathcal{G}$ is an *end component* if (a) U is closed, and (b) the subgame graph $\mathcal{G} \downarrow U$ is strongly connected.

Denote the set of all end components of $\mathcal{G}$ by $\mathcal{E} \subset 2^V$. The next lemma states that under every strategy ρ_1 (being memoryless or not) of Player 1 in the $1^1/_2$-player game, the set of states visited infinitely often along a play is an end component with probability one.

Lemma B.18. *[De Alfaro 1997, Thm. 3.2] For every $1^1/_2$-player game graph, for every vertex $v \in V$, and every Player 1 strategy ρ_1,*

$$P_v^{\rho_1}\left(\mathcal{G} \models \bigvee_{U\in\mathcal{E}} \left(\Diamond\Box U \wedge \bigwedge_{u\in U} \Box\Diamond u\right)\right) = 1. \tag{100}$$

This lemma implies the following corollary, which is motivated by similar claim for Rabin winning conditions in the literature [Chatterjee et al. 2005].

[7] Alternatively, we could also define $1^1/_2$-player game graphs by restricting the outgoing edges from the Player 1 vertices; our choice is actually tailored for the content of the rest of the section.

Corollary 1. *For a given* $1^1/_2$*-player game, for a given vertex* $v \in V$*, and for a given Player* 1 *strategy* ρ_1*, a generalized Rabin condition* $\widetilde{\mathcal{R}} = \{\langle \mathrm{G}_1, R_1\rangle, \ldots, \langle \mathrm{G}_k, R_k\rangle\}$ *is satisfied almost surely if and only if for every end component* U *reachable from* v^0*, there is a* $j \in \{1, 2, \ldots, k\}$ *such that* $U \cap R_j = \emptyset$ *and for every* $l \in [1; m_j]$*,* $U \cap {}^l G_j \neq \emptyset$*.*

B.6.2 Proof of Thm. 5.2.

Theorem (Thm. 5.2 restated for convenience). *Let* $\mathcal{G}$ *be a* $2^1/_2$*-player game graph,* $\widetilde{\mathcal{R}}$ *be a generalized Rabin condition,* $\varphi \subseteq V^\omega$ *be the corresponding LTL specification (Eq. (26)) over the set of vertices* V *of* $\mathcal{G}$*, and* $Derand(\mathcal{G})$ *be the reduced two-player game graph. Let* $\mathcal{W} \subseteq \widetilde{V}$ *be the set of all the vertices from where Player* 0 *wins the fair adversarial game over* $Derand(\mathcal{G})$ *for the winning condition* φ*, and* $\mathcal{W}^{a.s.}$ *be the almost sure winning set of Player* 0 *in the game graph* $\mathcal{G}$ *for the specification* φ*. Then,* $\mathcal{W} = \mathcal{W}^{a.s.}$*. Moreover, a winning strategy in* $Derand(\mathcal{G})$ *is also a winning strategy in* $\mathcal{G}$*, and vice versa.*

We define the fairness constraint on the random edges of $\mathcal{G}$ as per Eq. (3):

$$\varphi^\ell := \wedge_{(v,v')\in E_r} \Box\Diamond v \rightarrow \Box\Diamond(v \wedge \bigcirc v').$$

We first show that $\mathcal{W} \subseteq \mathcal{W}^{a.s.}$. Consider an arbitrary initial vertex $v^0 \in \mathcal{W}$ and an arbitrary strategy ρ_1 of Player 1 in $\mathcal{G}$. Let ρ_0^* be a corresponding winning strategy for Player 0 from v^0 for the fair adversarial game over *Derand*$(\mathcal{G})$ for the winning condition φ. By definition, ρ_0^* realizes the specification φ, whenever the adversary satisfies the strong fairness condition on the live edges in *Derand*$(\mathcal{G})$. On the other hand, the live edges in *Derand*$(\mathcal{G})$ are exactly the random edges in $\mathcal{G}$. In other words, we already know that if we apply the *same* strategy ρ_0^* to $\mathcal{G}$, then $\inf_{\rho_1\in R_1} P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi^\ell \rightarrow \varphi) = 1$.

We first show that the random edges E_r also satisfy the strong fairness condition φ^ℓ *almost surely*; actually we show that the probability of violation of φ^ℓ in $\mathcal{G}$ is 0. Consider the following:

$$\begin{aligned} P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \neg\varphi^\ell\right) &= P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \neg \bigwedge_{(v,v')\in E_r} \Box\Diamond v \rightarrow \Box\Diamond(v \wedge \bigcirc v')\right) \\ &= P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \bigvee_{(v,v')\in E_r} \Box\Diamond v \wedge \Diamond\Box\neg(v \wedge \bigcirc v')\right) \\ &\leq \sum_{(v,v')\in E_r} P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \Box\Diamond v \wedge \Diamond\Box\neg(v \wedge \bigcirc v')\right). \end{aligned}$$

We show that the right-hand side of the last inequality equals to 0 by proving that for every $(v, v') \in E_r$,

$$P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \Box\Diamond v \wedge \Diamond\Box\neg(v \wedge \bigcirc v')\right) = 0.$$

Consider any arbitrary $(v, v') \in E_r$ and assume that the probability of taking the edge (v, v') from v is p_1. Let π be a play on $\mathcal{G}$ and $(i_0, i_1, i_2, \ldots)$ be the infinite sequence of time indices when the vertex v is visited. For every i_k, the probability of *not* visiting v' for the next l time steps $(i_{k+1}+1, \ldots, i_{k+l}+1)$ is given by $(1-p)^l$, which converges to 0 as l approaches ∞. This proves that for every i_k, eventually there will be a v' at $(i_k + 1)$ with probability 1; in other words v' will be visited infinitely often with probability 1. Hence, it follows that $\sum_{(v,v')\in E_r} P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \Box\Diamond v \wedge \Diamond\Box\neg(v \wedge \bigcirc v')\right) = 0$, which in turn establishes that $P_{v^0}^{\rho_0^*,\rho_1}\left(\mathcal{G} \models \neg\varphi^\ell\right) = 0$.

Now consider the following derivation:

$$P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi^\ell \to \varphi) = P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \neg\varphi^\ell \vee \varphi) \le P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \neg\varphi^\ell) + P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi)$$
$$= 0 + P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi) = P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi).$$

Since we know that $P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi^\ell \to \varphi) = 1$, hence it follows that $P_{v^0}^{\rho_0^*,\rho_1}(\mathcal{G} \models \varphi) = 1$.

Next, we show that $\mathcal{W} \supseteq \mathcal{W}^{a.s.}$. Consider an arbitrary initial vertex $v^0 \in \mathcal{W}^{a.s.}$. Let ρ_0^* be a corresponding almost sure winning strategy for Player 0 from v^0 in the $2^1/_2$-player game $\mathcal{G}$ with the specification φ. We show that Player 0 wins the fair adversarial game over *Derand*$(\mathcal{G})$ for the winning condition φ from vertex v^0 using the strategy ρ_0^*.

Let $\rho_1 \in R_1$ be any arbitrary Player 1 strategy in the game *Derand*$(\mathcal{G})$ such that the unique resultant play $\pi = (v^0, v^1, \dots)$ due to ρ_0^* and ρ_1 satisfies the fairness assumption. We use the notation $\mathrm{Inf}(\pi)$ to denote the set of infinitely occurring vertices along the play π, i.e., $\mathrm{Inf}(\pi) \coloneqq \{w \in V \mid \forall m \in \mathbb{N}_0 \,.\, \exists n > m \,.\, v^n = w\}$. First we show that (i) the set of vertices $\mathrm{Inf}(\pi)$ forms an end component in $\mathcal{G}$, and moreover (ii) there exists a Player 1 strategy ρ_1' in the game $\mathcal{G}$ such that $P_{v^0}^{\rho_0^*,\rho_1'}(\mathcal{G} \models \mathrm{Inf}(\pi)) > 0$. Claim (i) follows by observing the following:

- For all $v \in \mathrm{Inf}(\pi) \cap V_r$, $V_r(v) \subseteq \mathrm{Inf}(\pi)$, as otherwise in *Derand*$(\mathcal{G})$ there would be a vertex in $E^\ell(v)$ and outside $\mathrm{Inf}(\pi)$ which would be visited infinitely many times due to infinitely many visits to v.
- For every $v \in \mathrm{Inf}(\pi) \cap (V_0 \cup V_1)$, $E(v) \neq \emptyset$, as otherwise in *Derand*$(\mathcal{G})$ the play π would reach a dead-end.
- The subgame graph $\mathcal{G} \downarrow \mathrm{Inf}(\pi)$ is strongly connected, as otherwise in *Derand*$(\mathcal{G})$ there would be two vertices $u, v \in \mathrm{Inf}(\pi)$ so that v would not be reachable from u, contradicting the assumption that both u and v are visited infinitely often by π.

Claim (ii) follows by defining a strategy $\rho_1' \equiv \rho_1$ on $\mathcal{G}$. Now observe that for every edge (v, v') chosen by Player 1 from a vertex $v \in \mathrm{dom}(E^\ell)$ in *Derand*$(\mathcal{G})$, there exists a corresponding positive probability edge (v, v') in $\mathcal{G}$. Since $\mathrm{Inf}(\pi)$ is entered by π after finite time steps, hence the Claim (ii) follows.

Now, from Cor. 1 it follows that there is a $j \in \{1, 2, \dots, k\}$ such that $\mathrm{Inf}(\pi) \cap R_j = \emptyset$ and for every $l \in \{1, \dots, m_j\}$, $\mathrm{Inf}(\pi) \cap {}^lG_j \neq \emptyset$. Thus the play π satisfies the generalized Rabin condition $\widetilde{\mathcal{R}}$. Since this holds for any arbitrary Player 1 strategy, hence $\mathcal{W} \supseteq \mathcal{W}^{a.s.}$ and ρ^* is the corresponding winning strategy for Player 0.

C THE ACCELERATED FIXPOINT ALGORITHM

Consider the fixpoint algorithm in (7). In the correctness proof of Thm. 3.1 discussed in App. B.3, we have been remembering so called configuration prefixes $\delta = p_0 i_0 \dots p_{j-1} i_{j-1}$ for some $j \leq k$ for every fixed-point variable X (see Eq. (55)). We denoted by $X^{i_j}_{\delta p_j}$ the set of states computed in the i_j'th iteration of the fixed-point over X_{p_j} after the fixed-point over Y_{p_j} has already terminated within the i_{j-1}th iteration over $X_{p_{j-1}}$ after the fixed-point over $Y_{p_{j-1}}$ has terminated in the i_{j-2}th iteration over $X_{p_{j-2}}$ and so forth.

In order to describe the accelerated implementation of (7), we do not assume that the fixed-points over Y-variables have already terminated, but additionally remember their counters m. This leads to configuration prefixes $\delta = p_0 m_0 i_0 \dots p_{j-1} m_{j-1} i_{j-1}$ and lets us define that $X^{m_j i_j}_{\delta p_j}$ is the set of states computed in the i_jth iteration of the fixed-point over X_{p_j} during the m_jth iteration over Y_{p_j}, computing the set $Y^{m_j}_{\delta p_j}$ and so forth.

Given two configuration prefixes $\delta = p_0 m_0 i_0 \dots p_{j-1} m_{j-1} i_{j-1}$ and $\delta' = p'_0 m'_0 i'_0 \dots p'_{j-1} m'_{j-1} i'_{j-1}$ we define $\delta <_m \delta'$ if $p_0 \dots p_{j-1} = p'_0 \dots p'_{j-1}$, $m_0 \dots m_{j-1} < m'_0 \dots m'_{j-1}$ (using the induced lexicographic order) and $i_0 \dots i_{j-1} = i'_0 \dots i'_{j-1}$. We define $\delta <_i \delta'$ similarly.

Now Piterman and Pnueli [2006] showed, based on a result of Long et al. [1994], that for every configuration prefix $\delta = p_0 m_0 i_0 \dots p_{j-1} m_{j-1} i_{j-1}$ the computation of $Y^0_{\delta p_j}$ can start from the *minimal* set $Y^{m_j}_{\delta' p_j}$ (instead of the entire set of vertices V) such that $\delta' p_j m_j <_m \delta p_j 0$. Dually, for every configuration prefix $\delta = p_0 m_0 i_0 \dots p_{j-1} m_{j-1} i_{j-1}$ the computation of $X^{m_j 0}_{\delta p_j}$ can start from the *maximal* set $X^{m_j i_j}_{\delta' p_j}$ (instead of the empty set) such that $\delta' p_j m_j i_j <_i \delta p_j m_j 0$.

Further, we see that for the innermost fixpoint, i.e. when $j = k$, it follows that for every computation prefix δ , there can be at most n iterations over both Y_{p_k} and X_{p_k}, where n is the total number of vertices. I.e., n different sets $Y^{m_k}_{\delta p_k}$ and $X^{m_k i_k}_{\delta p_k}$ have to be freshly computed for each δp_k and $\delta p_k m_k$ respectively. We see that there are $O(n^{k+1}k!)$ different such permutation sequences. As the computation of the innermost fixpoint dominates the computation time, it is shown by Long et al. [1994] that this results in an overall worst-case computation time of $O(n^{(k+1)+1}k!) = O(n^{k+2}k!)$ (where n is the total number of vertices and k is the number of Rabin pairs).

Unfortunately, the memory requirement of this acceleration algorithm is enormous. To see this, observe that in order to warm-start the computation of $Y^0_{\delta p_j}$ with $\delta = p_0 m_0 i_0 \dots p_{j-1} m_{j-1} i_{j-1}$ we need to store the current minimal set w.r.t. the m-prefix for every combination of p- and i-prefixes that can occur in δ, which are $O(n^{k+1}k!)$ many. Similarly, to warm-start the computation of $X^{m_j i_j}_{\delta p_j}$ we need to store the current minimal set w.r.t. the i-prefix for every combination of p- and m-prefixes that can occur in δ. This means that the memory required by the algorithm is $O(n^{k+1}k!)$, which is prohibitively large for large values of n and k.

We implemented a *space-bounded* version of the acceleration algorithm, where for any given parameter M (chosen by the user), we stored only up to M values for each counter. Whenever the values of all the counters are less than M, we use the regular acceleration algorithm as outlined above. Otherwise, if any of the counters exceeds M, then we fall back to the regular initialization procedure of fixpoint algorithms, i.e. depending on whether it is an Y or an X variable, initialize it with V or $\emptyset$ respectively. As a result, the memory requirement of our accelerated fixpoint algorithm is given by $O(M^{k+1}k!)$. This space-bounded acceleration algorithm made our implementation much faster and yet practically feasible, as has been demonstrated in Sec. 6.

D SUPPLEMENTARY RESULTS FOR THE EXPERIMENTS

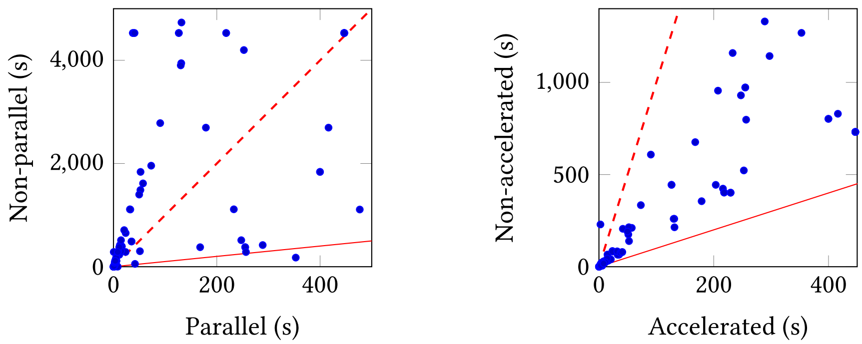

Fig. 10. Zoomed-in version of Fig. 5. (Left) Comparison between the computation times for the non-parallel (1 worker thread) and parallel (48 worker threads) version of Fairsyn, with acceleration being enabled in both cases. (Right) Comparison between the computation times for the non-accelerated and the accelerated version of Fairsyn, with parallelization being enabled in both cases. (Both) The points on the solid red line represent the same computation time. The points on the dashed red line represent an order of magnitude improvement.

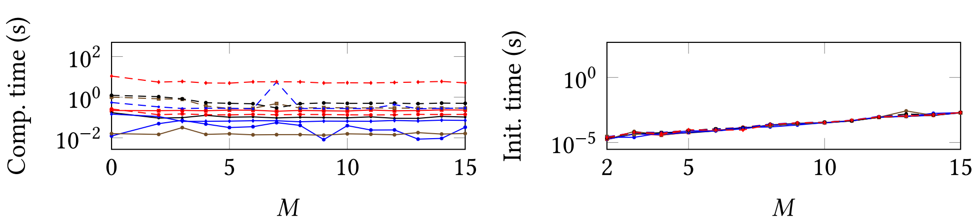

Fig. 11. (Left) Effect of variation of the acceleration parameter M on the total computation time (parallelization being enabled) for the VLTS benchmark examples with 1 Rabin pair. (Right) Effect of variation of the acceleration parameter M on the initialization time for the VLTS benchmark examples with 1 Rabin pair. The computation time (Y-axis) in both the plots are shown in the logarithmic scale.

Number of Vertices	Number of Transitions	Number of Live Edges	Number of BDD Variables
289	1224	17	9
289	1224	25	9
289	1224	13	9
1952	2387	1	11
1952	2387	5	11
1952	2387	25	11
1183	4464	16	11
1183	4464	49	11
1183	4464	9	11
3995	14,552	39	12
3995	14,552	139	12
3995	14,552	153	12
5121	9392	1	13
5121	9392	54	13
5121	9392	73	13
8879	24,411	473	14
8879	24,411	397	14
7119	38,424	626	14
7119	38,424	835	14
7119	38,424	597	14
10,849	56,156	241	14
10,849	56,156	482	14
18,746	73,043	1585	15
18,746	73,043	1729	15
18,746	73,043	575	15
25,216	25,216	137	15
25,216	25,216	595	15
25,216	25,216	373	15
40,006	60,007	1130	16
40,006	60,007	865	16
52,268	292,823	107	16
52,268	292,823	3254	16
65,537	524,293	13,727	17
65,537	524,293	25,229	17
66,929	569,322	23,290	17
66,929	569,322	13,698	17
69,753	359,575	11,071	17
69,753	359,575	5058	17
83,435	259,488	1682	17
83,435	259,488	2707	17
96,878	282,880	6225	18
96,878	282,880	585	18

Table 3. Details of the fair adversarial Rabin games randomly generated from the VLTS benchmark suite. Continued to Table 4.

Number of Vertices	Number of Transitions	Number of Live Edges	Number of BDD Variables
116,456	364,596	8316	17
116,456	364,596	7774	17
142,471	925,429	19,259	18
142,471	925,429	3304	18
164,865	1,619,200	13,407	18
164,865	1,619,200	24,868	18
166,463	518,976	13,633	18
166,463	518,976	4155	18
214,140	683,205	13,588	18
214,140	683,205	12,113	18
371,804	641,565	3413	19
371,804	641,565	12,151	19
386,496	1,171,870	26,247	19
386,496	1,171,870	17,823	19
566,639	3,984,160	7109	20
566,639	3,984,160	42,757	20

Table 4. Continued from Table 3. Details of the fair adversarial Rabin games randomly generated from the VLTS benchmark suite.

Broadcast Queue Capacity	Output Queue Capacity	Number of Vertices	Number of Transitions	Number of Live Edges	Number of BDD Variables	Time (seconds)
1	1	5,307,840	10,135,300	5,124,100	25	7.37
2	1	21,231,400	40,541,200	20,496,400	27	24.90
3	1	21,414,100	42,080,300	21,265,900	27	28.97
1	2	21,340,800	40,879,100	20,834,300	27	38.25
1	3	21,559,400	42,756,100	21,772,800	27	51.55
4	1	84,925,400	162,165,000	81,985,500	29	57.70
5	1	85,295,700	165,243,000	83,524,600	29	65.01
6	1	85,656,300	168,321,000	85,063,700	29	73.19
7	1	86,007,400	171,399,000	86,602,800	29	77.97
1	4	85,363,200	163,516,000	83,337,200	29	92.56
1	5	85,808,000	167,270,000	85,214,200	29	113.18
2	2	85,363,200	163,516,000	83,337,200	29	133.20
1	6	86,237,400	171,024,000	87,091,200	29	135.67
3	2	86,061,400	169,673,000	86,415,400	29	144.27
1	7	86,651,500	174,778,000	88,968,200	29	145.76
8	1	339,702,000	648,659,000	327,942,000	31	149.68
2	3	86,237,400	171,024,000	87,091,200	29	163.62
9	1	340,447,000	654,815,000	331,020,000	31	174.29
10	1	341,183,000	660,972,000	334,098,000	31	197.02
3	3	86,870,100	177,181,000	90,169,300	29	203.15
1	8	341,453,000	654,066,000	333,349,000	31	248.38
1	9	342,350,000	661,574,000	337,103,000	31	283.85
1	10	343,232,000	669,082,000	340,857,000	31	331.78
7	2	345,587,000	691,003,000	351,818,000	31	567.26
4	2	341,453,000	654,066,000	333,349,000	31	710.78
2	4	341,453,000	654,066,000	333,349,000	31	806.74
5	2	342,868,000	666,378,000	339,505,000	31	852.37
6	2	344,246,000	678,691,000	345,661,000	31	936.04
2	5	343,232,000	669,082,000	340,857,000	31	1034.57
4	3	344,950,000	684,098,000	348,365,000	31	1071.52
2	7	346,606,000	699,113,000	355,873,000	31	1111.64
7	3	348,693,000	721,035,000	366,834,000	31	1312.88
2	6	344,950,000	684,098,000	348,365,000	31	1336.35
5	3	346,233,000	696,410,000	354,521,000	31	1351.31
3	4	344,246,000	678,691,000	345,661,000	31	1632.63
6	3	347,480,000	708,723,000	360,677,000	31	1667.54
8	2	1,365,810,000	2,616,260,000	1,333,400,000	33	2478.13
9	2	1,368,660,000	2,640,890,000	1,345,710,000	33	2783.77

Table 5. Experimental evaluation for the code-aware resource management case study (extended table).