Abstract—We introduce a new class of 2/1-player games, the 2/1-player GR(1) games, that allows for solving problems of stochastic nature by adding a probabilistic component to simple 2-player GR(1) games. Further, we present an efficient approach for solving qualitative 2/1-player GR(1) games with polynomial-time complexity. Our approach is based on a reduction from 2/1-player GR(1) games to 2-player GR(1) games that allows for solving the game and constructing, from a sure winning strategy for player $\Box$ (resp. $\Diamond$) in a 2-player GR(1) game, an almost-sure (resp. positively) winning strategy for its corresponding 2/1-player GR(1) game. Key to the effectiveness of the proposed approach is the fact that the reduction generates a 2-player game that is linearly larger than the original 2/1-player game, more precisely, it is linear with respect to the number of probabilistic states in the 2/1-player GR(1) game.

I. INTRODUCTION

Given an environment model and a goal specification, the reactive synthesis problem is to automatically generate a behaviour description, i.e. a controller, such that the controlled environment satisfies the specified goal. The problem of automatically generating controllers that guarantee the goals under assumptions about the behavior of the environment is tackled by extracting the controller from a strategy in a 2-player game modeling the situation and assuming that the controller and the environment are adversaries.

2-player games are turn-based zero-sum perfect information games played by two players (namely $\Box$ and $\Diamond$) in a finite directed game graph $(S, E)$. The game starts in a distinguished state of the graph. Then $\Box$ and $\Diamond$ make moves along the edges, constructing an infinite sequence of states: a play. We restrict our attention to games with objectives (or winning conditions) expressed by arbitrary linear-time temporal logic formulas (or $\omega$-regular conditions) [37].

Although 2-player games have shown to be applicable in many domains, in some cases there is a stochastic component in the problem in which case the 2/1-player game is a natural fit. 2/1-player games are 2-player games with an extra player that behaves probabilistically.

A general way of specifying objectives of players in nonterminating zero-sum games is by providing the set of $\omega$-regular winning plays $W$ for player $\Box$. In 2-player games we consider the problem of finding a $\omega$-winning strategy from the initial state for $\Box$ such that any play formed by applying such a strategy and any possible strategy for $\Diamond$ belong to $W$. In contrast, in 2/1-player games, we restrict our attention to the problem of deciding whether player $\Box$ has an almost-sure winning strategy to beat her opponent (i.e. whether $\Box$ can ensure that against all strategies of the opponent a play consistent with her decisions belongs to $W$ with probability 1 [8]). Such a decision problem is known as the qualitative problem.

Even though 2/1-player games with $\omega$-regular winning conditions allow for very expressive problem specifications, they lead to complex algorithms compromising their practical application. The problem of finding a qualitative solution in 2/1-player games with winning conditions like Rabin, Streett, Muller or Parity are known to be P-complete, coNP-complete, PSPACE-complete and NP $\cap$ coNP respectively [7]. There exists some $\omega$-regular subsets, like safety and reachability, known to allow for polynomial-time qualitative solutions in 2/1-player games, however such winning conditions are still very limiting in practical terms. We propose something in between: A winning condition (subset of $\omega$-regular conditions) with efficient qualitative solutions in 2/1-player games but more expressive than safety and reachability allowing to specify realistic systems.

In 2-player games conditions that allow for polynomial-time solutions but with higher expressiveness than safety or reachability conditions have been studied. One of such cases is that of Generalized Reactivity (1) (GR(1)) conditions [3] for which a qualitative solution can be computed in PTIME. An outstanding characteristic of GR(1) is that it allows not only describing liveness goals that the controller is to achieve but also describing liveness assumptions on the environment that the controller can rely on to achieve its goals. It is well understood (e.g. [20]) that assumptions play an important role in software specifications.

GR(1) games have been applied in numerous settings such as hardware design [4], high-level strategies for self-adaptive systems [15], [5], robot-controller planning [11], [21], [24], [25], [40] and user programming [26], [27]. In addition, many tools for solving GR(1) games have been developed [22], [38], [2], [16] showing that there is real interest in the application of games with GR(1) winning conditions. However, to the best of our knowledge, in 2/1-player games the research for this kind of subsets (i.e. particular cases that allow for efficient solutions but expressive enough from a practical point of view) has been scarcely explored. Furthermore, to the best of our knowledge, there is no work relating explicitly GR(1) conditions with 2/1-player games.

In this work, we introduce a new class of 2/1-player
games, the $2^1/2$-player GR(1) games, that allows solving problems of stochastic nature by adding a probabilistic component to simple 2-player GR(1) games. Further, we present an efficient approach for solving $2^1/2$-player GR(1) games with polynomial-time complexity. Our approach is based on a reduction from $2^1/2$-player GR(1) games to 2-player GR(1) games that allows solving the game and constructing, from a sure winning strategy for $\lozenge$ (resp. $\diamondsuit$) in a 2-player GR(1) game, an almost-sure (resp. positively) winning strategy for its corresponding in $2^1/2$-player GR(1) game.

The paper is organized as follows. Section II includes the necessary background. Section III presents a formal definition of $2^1/2$-player GR(1) games. Section IV includes a proposal of an efficient (i.e. polynomial-time) solution for $2^1/2$-player GR(1) games and the derived algorithm for constructing strategies as well as a proof of correctness of the approach and its complexity. Finally, we report on the corresponding related work and conclusions.

II. Preliminaries

We consider simple $2^1/2$-player games. That is non-terminating turn-based zero-sum perfect-information finite state games played over a graph $(S, E)$ between three players, $\lozenge$, $\diamondsuit$ and $\circ$, who move a token from state to state along edges of a the graph so that an infinite path is formed. A zero-sum game is one in which, every situation that makes a player lose is such that the other player wins.

In perfect-information games any participant is supposed to have all the required information to force a decision to make the game move.

Definition II.1 [2$^1/2$-player game graph] A $2^1/2$-player game graph is a tuple $G = ((S, E), (S_0, S_0, S))$ where $(S, E)$ is a finite directed graph with state set $S$, edge set $E$ and, $(S_0, S_0, S_0)$ is a partition of $S$.

The partition of states in $S$ is used to define the turns of each player in the game. $\lozenge$ moves along the edges of $G$ from states in $S_0$ and $\diamondsuit$ from those in $S_0$. Both players have full freedom of movement when it is their turn. Player $\circ$, on the other hand, moves by choosing a successor state from states in $S_0$ accordingly to a probabilistic distribution. For simplicity and w.l.o.g, we consider this distribution to be uniformly random. Furthermore, although player $\circ$ is part of the game, he does not look forward to winning or losing. Due to its limited freedom of choice and lack of winning condition we refer to player $\circ$ as a “half player”.

Definition II.2 [2$^1/2$-player game] A $2^1/2$-player game $G$ is a non-terminating turn-based zero-sum perfect-information game played on a game graph $G = ((S, E), (S_0, S_0, S))$ by three players: $\lozenge$, $\diamondsuit$ and $\circ$. In $G$, $\lozenge$ and $\diamondsuit$ chooses the successor of states in $S_0$ and $S_0$ respectively; states in $S_0$ are $\circ$’s states and it always moves by choosing a successor uniformly at random.

The special case where $S_0 = \emptyset$ corresponds to 2-player game graphs.

Definition II.3 [2-player game] A 2-player game is a non-terminating turn-based zero-sum perfect-information game played on the game arena $G = ((S, E), (S_0, S_0))$.

Plays. A play $\rho = s_0, s_1, s_2, \ldots$ over a game graph $G$ is an infinite sequence of states in $S$ such that, for all $i \geq 0$, $(s_i, s_{i+1}) \in E$. We write $\Omega$ for the set of all plays, $\Omega_s$ for the set of all plays starting from the state $s$, and $\Delta_G(s)$ the set of all states $s'$ such that $(s, s') \in E$. We denote $\rho[i]$ the $i$–th state in a play $\rho$. We say that $s_i \leadsto_G s_j$ if there is a play $\rho$ over the game $G$ such that, for some $i \leq j$, $\rho[i] = s_i$ and $\rho[j] = s_j$. And, if $j = i + 1$ we write $s_i \leadsto_G s_j$.

Strategies. The choices of players are formalized in the form of strategies. In other words, a strategy is the policy that each player applies to move along the game graph. In general, the strategy that a player uses or follows in the game may depend or not on the history of a play.

A finite-memory strategy $\sigma$ for $\lozenge$ can be encoded by a deterministic transducer $(\mathcal{M}, m_0, \sigma_1, \sigma_1)$ where $i)$ $\mathcal{M}$ is a finite set representing the memory of the strategy, $ii)$ $m_0 \in \mathcal{M}$ is the initial memory value, $iii)$ $\sigma_1 : S_0 \times \mathcal{M} \rightarrow S$ defines a partial transition function, and $iv)$ $\sigma_2 : S \times \mathcal{M} \rightarrow \mathcal{M}$ represents how the memory is updated by $\sigma$. Player $\lozenge$ follows (or uses) a strategy $\sigma$ if in each move, given a current value of memory $m$ at $s \in S_0$ she chooses as successor the state $\sigma_1(s, m)$ and the memory is updated according to $\sigma_2(s, m)$ for every $s \in S$. A play $\rho = s_0, s_1, \ldots$ is consistent with a strategy $\sigma$ if $\sigma_1(s_i, m_i) = s_{i+1}$, where $m_{i+1} = \sigma_2(s_i, m_i)$. Strategies for $\lozenge$ and $\circ$ are defined analogously. We write $\Sigma$ and $\Pi$ for the set of all strategies for $\lozenge$ and $\circ$.

W.l.o.g we consider memory values tracking goals for $\lozenge$ and assumptions for $\lozenge$ in lexicographical order.

Once strategies $\sigma \in \Sigma$ and $\pi \in \Pi$ are fixed, a play starting in state $s \in S$ is determined by a random path (denoted $p^\sigma_\pi$, consistent with $\sigma$ and $\pi$, for which the probabilities of events are uniquely defined, where an event $E \subseteq \Omega$ is a measurable set of paths. We define $G_{\sigma, \pi}$ as the sub-graph obtained by pruning $G$ according to the decisions made by strategies $\sigma$ and $\pi$. We write $P_{p^\sigma_\pi}[E]$ for the probability that a path starting on state $s$ in $G_{\sigma, \pi}$ belongs to the event $E$.

Once strategies $\sigma$ or $\pi$ are fixed a Markov Decision Process (MDP) is obtained. A set $\mathcal{U}$ is an end component in such MDP if the sub-graph induced by $\mathcal{U}$ is strongly connected, and for each probabilistic states in $\mathcal{U}$ every outgoing edges end up in $\mathcal{U}$ (i.e., $\mathcal{U}$ is closed for probabilistic states). The key property about MDPs that is used in our proofs is a result established by [12], [13]: given an MDP, for every strategy, with probability 1 the set of states visited infinitely often is an end component.

Winning conditions. Winning conditions of players in non-terminating $2^1/2$-player games are specified by a set of winning plays $W \subseteq \Omega$ for a player. As we consider zero-sum games if the winning condition of a player is the set $W$ then the winning condition for the other player is $\Omega \setminus W$. Given a
game graph $G$ and a winning condition $W$, we write $G(W)$ for the game played on $G$ with winning condition $W$ for $\square$.

We focus on Generalized Reactivity (1) (GR(1)) conditions [3]. Let $\rho$ be a play in $G$ and $g_1, \ldots, g_n$ and $a_1, \ldots, a_m$ be subsets of $S$. Let $\text{inf}\{\rho\}$ denote the set of states that occur infinitely often in $\rho$. A play $\rho$ is in the GR(1) condition $gr((g_1, \ldots, g_n), (a_1, \ldots, a_m))$ if either for some $i$ we have $\text{inf}\{\rho\} \cap a_i = \emptyset$ or for every $j$ we have $\text{inf}\{\rho\} \cap g_j \neq \emptyset$.

A practical interpretation of GR(1) conditions is letting $a_1$ through $a_m$ be the assumptions on the environment and $g_1$ through $g_n$ be the guarantees that the system has to fulfill. If the environment ($\lozenge$) behaves as expected by satisfying always its progress assumptions then goals must be guaranteed by $\boxminus$’s strategy to win (i.e., the underlying controller will always eventually progress towards its goals as long as the environment is live).

A 2-player game with a Generalized Reactivity (1) condition is called 2-player GR(1) game. Formally:

**Definition II.4 [2-player GR(1) game]** Let $G = ((S, E), (S_\square, S_\lozenge))$ be a game graph, and $W$ be a winning condition. We say that $G(W)$ is a GR(1) game iff $W = gr((g_1, \ldots, g_n), (a_1, \ldots, a_m))$, with $g_1, \ldots, g_n$ and $a_1, \ldots, a_m$ subsets of $S$.

A winning strategy for $\square$ in a 2-player GR(1) game can be computed in $O(nmS^2)$ [35] where $m$ and $n$ stand for number of assumptions and goals respectively and $S$ is the set of states in the game graph.

**Qualitatively winning modes.** [23] In this work, we focus on the qualitative decision problem. The qualitative solution of a game amounts to computing the set of states from which a player has the possibility to win.

Let $G(W)$ be a 2\slash 2-player game, we say that a strategy $\sigma \in \Sigma$ is:

- **sure winning** for $\square$ in the game $G(W)$ if, for all $\pi \in \Pi$ we have $\rho^\sigma,\pi \in W$.
- **almost-sure winning** for $\square$ in the game $G(W)$ if, for all $\pi \in \Pi$ we have $\text{Pr}^\sigma,\pi[W] = 1$.
- **positive-probability winning** for $\square$ in the game $G(W)$ if, for all $\pi \in \Pi$ we have $\text{Pr}^\sigma,\pi[W] > 0$.

Note that in 2-player games players can only win surely.

**Determinacy.** Let $\mu \in \{s, a, p\}$ be a winning mode, where “$s$” stands for sure win, “$a$” for almost-sure win and “$p$” for positive-probability win. We say that a game $G(W)$ is $\mu$–determined if for all state in $S$, either $\square$ can $\mu$–win or $\lozenge$ can $\mu$–win from such state, where $\bar{s} = s$, $\bar{a} = p$ and $\bar{p} = a$.

In particular, every 2-player game with GR(1) winning condition for player $\square$ (2-player GR(1) game) is $s$-determined (or simply determined) [3]. Thus, the set of states in a 2-player GR(1) game graph is partitioned in those from which $\square$ has a sure winning strategy (i.e. $W_\square$) are those from which $\lozenge$ has a sure winning strategy ($W_\lozenge$).

A proof that 2\slash 2-player GR(1) games are a-determined can be derived from [32].

III. 2\slash 2-player games

In this section we introduce a new class of 2\slash 2-player games.

**Definition III.1 [2\slash 2-player GR(1) game]** Let $G(W)$ be a 2\slash 2-player game with winning condition $W$ for $\square$. We say that $G$ is a 2\slash 2-player GR(1) game iff $W$ is a Generalized Reactivity (1) winning condition.

More specifically, a 2\slash 2-player GR(1) game $G(W)$ is a turn-based zero-sum perfect-information game played by three players: $\square$, $\lozenge$ and $\diamond$ over a game graph $G = ((S, E), (S_\square, S_\lozenge, S_\diamond))$ with a winning condition $W = gr((g_1, \ldots, g_n), (a_1, \ldots, a_m))$ for $\square$ where $g_1, \ldots, g_n$ and $a_1, \ldots, a_m$ denote subsets of $S$, and $W = gr((g_1, \ldots, g_n), (a_1, \ldots, a_m))$ denotes the set of infinite sequences $\rho$ such that for some $i$ we have $\text{inf}\{\rho\} \cap a_i = \emptyset$ or for all $j$ we have $\text{inf}\{\rho\} \cap g_j \neq \emptyset$.

For simplicity and w.l.o.g we assume that each state in the graph has at least one and at most two outgoing edges and that player $\diamond$ moves from states in $S_\lozenge$ by choosing a successor uniformly at random.

IV. Solving qualitatively 2\slash 2-player GR(1) games

We present an efficient qualitative solution for 2\slash 2-player GR(1) games that allows low polynomial-time complexity.

A. Reduction

The notion of reduction between games is an important aspect in the study of games as it allows to understand which classes of games are subsumed by others. The main result of this section is a proposal for solving 2\slash 2-player GR(1) games via an efficient reduction to 2\slash 2-player GR(1) games. We translate a 2\slash 2-player game to a 2\slash 2-player GR(1) game where we know the solution has low polynomial-time complexity. Key to the effectiveness of the proposed approach is the fact that the reduction generates a 2-player game that is linearly larger than the original 2\slash 2-player game, more precisely, it is linear with respect to the number of random states. The reduction abstracts away probabilistic choices by allowing the other players to make the choice while balancing the power of each player.

**Definition IV.1 [2\slash 2-player GR(1) to 2\slash 2-player GR(1) games reduction]** Given a 2\slash 2-player GR(1) game $G = ((S, E), (S_\square, S_\lozenge, S_\diamond))$ with winning condition $W = gr(G, A)$ where $G = (g_1, \ldots, g_n)$ and $A = (a_1, \ldots, a_m)$ with $g_1, \ldots, g_n$ and $a_1, \ldots, a_m$ subsets of $S$, we define the 2\slash 2-player GR(1) game $G = ((S, E), (S_\square, S_\lozenge, S_\diamond))$ with winning condition $W$ as follows:

\[
\begin{align*}
S_\square &= S_\square \cup \{s_1^{1,1}, s_1^{1,2}, s_2^{1,2} | s \in S_\square\} \\
S_\lozenge &= S_\lozenge \cup \{s_2^{1,2}, s_3^{3,3} | s \in S_\lozenge\} \\
S_\diamond &= S_\diamond \cup S_\diamond \\
E &= \{(s, s') \in S_\square \cup S_\lozenge | (s, s') \in E\} \cup \\
&\quad \{(s, s_1^{1,1}), (s_1^{1,1}, s_1^{1,2}) | s \in S_\square\} \cup \\
&\quad \{(s, s_2^{1,2}), (s_2^{1,2}, s_2^{1,2}) | s \in S_\lozenge\} \cup \\
&\quad \{(s_1^{1,2}, s'), (s_2^{3,3}, s') | s \in S_\lozenge, s' \in \Delta_G(s)\}
\end{align*}
\]

$W = gr(A, G)$ where
...tasksolving a 2/2-player game efficiently by computing a solution to a 2-player one. More specifically, based on a
sure winning strategy for □ (resp. ◇) found in the 2-player GR(1) game, an almost-sure (resp. positive-probability) win-
ning strategy for its corresponding in the 2/2-player GR(1) game can be constructed. In this section, we define such a
construction and we present a proof of its correctness.

Definition IV.2 [2-player GR(1) to 2/2-player GR(1) games solution] Let \( \pi \) (resp. \( \sigma \)) be a sure winning strategy for □
(resp. ◇) in \( G \) then a strategy \( \sigma \) (resp. \( \pi \)) for □ (resp. ◇) in \( G \) can be constructed as follows:

• For every pair \((s, m)\):
  \[\sigma(s, m) = \pi_i(s, m), \text{ for every } s \in S_\sigma \text{ and, } \]
  \[\sigma_u(s, m) = \begin{cases} \pi_u(s, m) & s \in S_\sigma \cup S_\sigma \\ m & s \in S_o \end{cases}\]

• Analogously, for every pair \((s, m)\):
  \[\pi(s, m) = \pi_i(s, m), \text{ for every } s \in S_\sigma \text{ and, } \]
  \[\pi_u(s, m) = \begin{cases} \sigma_u(s, m) & s \in S_\sigma \cup S_\sigma \\ m & s \in S_o \end{cases}\]

W.l.o.g. our approach relies on the existence of a sure
winning strategy that propagates the memory through the
reduction. The existence of such a strategy is warranted
with the existence of a generic winning one by considering
memory values tracking visited goals (resp. assumptions)
for □ (resp. ◇) in lexicographical order. Even more, such
assumption does not compromise the correctness of the
proposed technique but it may implies a slight adaptation of
the algorithm applied to solve the 2-player game constructed.

We now present a proof sketch of correctness for the
proposed solution mapping from Definition [IV.2].

Theorem 1 [Correctness] Given a 2/2-player GR(1) game
G and a 2-player GR(1) game \( \bar{G} \). Let \( \pi \) (resp. \( \sigma \)) be a sure
winning strategy for □ (resp. ◇) in \( \bar{G} \) and \( \sigma \) (resp. \( \pi \)) the
corresponding ones in \( G \) constructed as in Definition [IV.2].
If \( \bar{G} \) is the game obtained by applying the reduction in
Definition [IV.1] to \( G \), the following holds:

For every state \( s \) in a 2/2-player GR(1) game \( G \), \( \sigma \) (resp. \( \pi \))
is an almost-sure (resp. positive-probability) winning
strategy from \( s \) in \( G \), if and only if \( \pi \) (resp. \( \sigma \)) is a sure
winning strategy from state \( s \) in \( \bar{G} \).

Proof: [Sketch] (\( \Rightarrow \)) For □. The proof proceeds by
contradiction assuming that there exists a positive-probability
winning strategy \( \pi \in \Pi \) for ◇. We argue that if such a \( \pi \)
exists then, it is possible to define a \( \pi \in \bar{\Pi} \) adequately
that makes a “mimic” of \( \pi \) decisions in \( G \) so that a path
leading to “the same” loosing end component is formed in \( \bar{G} \)
(which is a contradiction since \( \bar{\pi} \) was supposed to be a sure
winning strategy in \( \bar{G} \) from \( s \)).

(\( \Rightarrow \)) For ◇. The proof for ◇ is very similar but, now
assuming that exists an almost-sure strategy \( \sigma \in \Sigma \) for □ and
defining adequately a \( \sigma \in \bar{\Sigma} \) leading to the contradiction.

Theorem 1 states that for every state \( s \) in a 2/2-player
GR(1) game \( G \), □ (resp. ◇) has an almost-sure (resp.
positive-probability) winning strategy from \( s \) in \( G \), then if
\[ \square \text{ (resp. } \Diamond \text{)} \text{ has a sure winning strategy from state } s \text{ in } G \].
Proof of \( \Rightarrow \) for both players follows from the fact that both games are determined.

Correctness states that the problem of finding a \( \mu \)-winning strategy for a player in a GR(1) game \( G \) is equivalent to the problem of finding a sure winning for the same player in \( G \), constructed following Definition IV.1.

C. Algorithm

A simple algorithm can be derived directly from the definitions above by following the steps described below:

1) Pre-processing input: Given the 2\(1/2 \)-player GR(1) game \( G \), build \( \overline{G} \) by applying the reduction in Definition IV.1.

2) Solving the 2-player GR(1) game: Find a sure (memory preserving) winning strategy \( \sigma \) (resp. \( \overline{\sigma} \)) for \( \square \) (resp. \( \Diamond \)) in \( G \) using some existing algorithm.

3) Post-processing output: Build an almost-sure \( \sigma \) (resp. positive-probability \( \pi \)) winning strategy for \( \square \) (resp. \( \Diamond \)) in \( G \) by applying the mapping in Definition IV.2.

The growth in the number of states of the reduction is linear (with constant 5) in the number of random states in \( G \) since for each state \( s \in S_\text{c} \), we add five additional states in the construction of \( \overline{G} \). Since \( \overline{G} \) state space is at most five times larger than \( G \) and 2-player GR(1) games have polynomial-time complexity we show that qualitatively solving 2\(1/2 \)-player GR(1) games is also polynomial.

More precisely, the qualitative decision problem to a 2\(1/2 \)-player GR(1) game with the proposed solution is in \( O(mnS^2) \) with \( m \), \( n \) the number of assumptions and goals respectively, and \( S \) is the set of states in the game graph.

V. RELATED WORK

2\(1/2 \)-player games have recently gained attention, mainly due to their possible applications and open problems. The complexity of solving 2\(1/2 \)-player games has been the center of attention in the literature, e.g. \[10, 6, 9, 8, 7, 19, 39, 17\]. Based on the known complexities for the standard \( \omega \)-regular conditions \[7\], central to the applicability of 2\(1/2 \)-player games is to find efficient, yet expressive, settings that allow modeling interesting problems in practice.

Since the complexity of the synthesis problem with linear-time temporal logics \[37\] specifications was established as 2\text{EXPTIME}-complete \[36\], a number of games with restricted winning conditions have been explored \[1, 30\]. In \[3\], games with GR(1) conditions are introduced. GR(1) games allow for polynomial-time solutions and have been widely applied in practice \[4, 26, 40\]. By extending 2-player GR(1) games with a stochastic component introducing 2\(1/2 \)-player GR(1) games, our approach expands the applicability of 2-player GR(1) games in domains such as robotics and autonomous systems where stochastic models are a more natural modeling approach.

Central to our approach is the definition of a reduction. We are not the first in obtaining results in game theory by means of a reduction from one game to another. Examples of this approach are \[8, 7, 23, 18\]. In particular, reductions such as those used for Parity \[8\] Rabin \[7\] and Muller \[33\] games may be adapted to efficiently solve qualitatively 2\(1/2 \)-player GR(1) games. One possible approach can be to combine the split tree \[41\] of the GR(1) condition with a reduction of a 2\(1/2 \)-player GR(1) game to a 2\(1/2 \)-player Parity of three priorities that are known to allow for polynomial-time solution \[8\]. Combining such results suggests a polynomial-time complexity result in the same order as ours. However, no previous work was found dealing explicitly with 2\(1/2 \)-player games. Furthermore, a naive combination of these results leads to a concrete complexity that is beyond the linear complexity (with constant 5) of the algorithm provided in this paper. In addition, the alternative reduction ends up in a Parity game while our proposal reduces to GR(1) games which are well studied and with several tools enabling applications to concrete problems in practice.

Another naive solution for trying to reduce the complexity of 2\(1/2 \)-player games can be to replace the stochastic component by non-determinism in 2-player games. However, this approach may result in oversimplified representations of the environment leading to synthesis problems where the environment is simply too powerful. For instance, in a reactive systems scenario where some of the system’s components may fail, considering failures as purely adversarial (i.e. controlled by the environment) may lead to unrealizable problems even when solutions to such problems may exist. The setting presented in \[14\] solves this problem by considering a very restricted notion of failure and imposing fairness condition on failures. Our approach allows for a generalization of \[14\] permitting to naturally model failures with probabilistic behaviour but requires a slight adaptation of the standard GR(1) algorithm (see Section IV.C). Supervisory control community has also shown interest in defining frameworks for dealing with plants and properties based on probabilistic languages (e.g. \[29, 28, 34\]) or stochastic process-theoretic approaches (e.g. \[31\]). Those approaches aim at building a (randomized) controller that try to match the behavior of a controlled plant with a given probabilistic specification language as a way to relax hard constraints imposed by the classical supervisory frameworks or optimize quality attributes. Note that our goal -of qualitative nature- is different: define a strategy that almost-sure satisfies progress specifications in the presence of, or resorting to, random behavior in the plant (the half player).

VI. CONCLUSIONS

2-player as well as 2\(1/2 \)-player games have several applications. We introduced an expressive family of 2\(1/2 \)-player games and presented a specific solution for qualitative decision problem, proven to be correct and efficient since it conforms to the theoretical complexity of the problem (PTIME). Our approach is based on a reduction from 2\(1/2 \)-player GR(1) games to 2-player GR(1) games that are known to have polynomial-time solution. We showed that the reduction preserves the number of non-random states and adds only five states for each random state, and that the exact complexity of solving qualitatively 2\(1/2 \)-player GR(1) games
with the proposed technique is $O(mnS^2)$ where $m$ and $n$ stand for number of assumptions and goals respectively, and $S$ is the set of states in the game graph. In addition, we presented how to construct an strategy in the 2/12-player game from the corresponding one in the reduction.

REFERENCES