From Verification to Control: Dynamic Programs for Omega-regular Objectives[*][†]

Luca de Alfaro Thomas A. Henzinger Rupak Majumdar

Electrical Engineering and Computer Sciences, University of California, Berkeley
{dealfaro,tah,rupak}@eecs.berkeley.edu

Abstract. Dynamic programs, or fixpoint iteration schemes, are useful for solving many problems on state spaces, including model checking on Kripke structures ("verification"), computing shortest paths on weighted graphs ("optimization"), computing the value of games played on game graphs ("control"). For Kripke structures, a rich fixpoint theory is available in the form of the μ-calculus. Yet few connections have been made between different interpretations of fixpoint algorithms. We study the question of when a particular fixpoint iteration scheme φ for verifying an ω-regular property Ψ on a Kripke structure can be used also for solving a two-player game on a game graph with winning objective Ψ. We provide a sufficient and necessary criterion for the answer to be affirmative in the form of an *extremal-model theorem for games*: under a game interpretation, the dynamic program φ solves the game with objective Ψ if and only if both (1) under an existential interpretation on Kripke structures, φ is equivalent to $\exists\Psi$, and (2) under a universal interpretation on Kripke structures, φ is equivalent to $\forall\Psi$. In other words, φ is correct on all two-player game graphs iff it is correct on all extremal game graphs, where one or the other player has no choice of moves. The theorem generalizes to quantitative interpretations, where it connects two-player games with costs to weighted graphs.

While the standard translations from ω-regular properties to the μ-calculus violate (1) or (2), we give a translation that satisfies both conditions. Our construction, therefore, yields fixpoint iteration schemes that can be uniformly applied on Kripke structures, weighted graphs, game graphs, and game graphs with costs, in order to meet or optimize a given ω-regular objective.

1 Introduction

If Ψ is a property of a Kripke structure, then every μ-calculus formula φ that is equivalent to Ψ prescribes an algorithm for model checking Ψ. This is because the μ-calculus formula φ can be computed by iterative fixpoint approximation. Indeed, the μ-calculus has been called the "assembly language" for model checking.

In control, we are given a two-player game structure and an objective, and we wish to find out if player 1 (the "controller") has a strategy such that for all strategies of player 2 (the "plant")

[*]This research was supported in part by the DARPA SEC grant F33615-C-98-3614, the MARCO GSRC grant 98-DT-660, the AFOSR MURI grant F49620-00-1-0327, the NSF Theory grant CCR-9988172, and the NSF ITR grant CCR-0085949.

[†]An abbreviated version of this paper will appear in the *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science* (LICS), 2001.

the outcome of the game meets the objective. If the outcome of a game is an infinite sequence of states, then objectives are naturally specified as ω-regular properties [15]. A simple but important objective is the reachability property $\Diamond T$, for a set T of states, which asserts that player 1 wins if it can direct the game into the target set T, while player 2 wins if it can prevent the game from entering T forever. We write $\langle\langle 1 \rangle\rangle \Diamond T$ for the reachability game with target T for player 1. A dynamic program for solving the reachability game can be viewed as evaluating a fixpoint equation, namely,

$$\langle\langle 1 \rangle\rangle \Diamond T \;=\; \mu x.(T \vee \mathit{1Pre}(x)),$$

where $\mathit{1Pre}(T)$ is the set of states from which player 1 can force the game into T in a single step. It is not difficult to see that this fixpoint equation is identical to the μ-calculus expression for model checking the reachability property $\exists \Diamond T$, namely,

$$\exists \Diamond T \;=\; \mu x.(T \vee \mathit{EPre}(x)), \tag{1}$$

except for the use of the *predecessor operator* EPre in place of $\mathit{1Pre}$, where $\mathit{EPre}(T)$ is the set of states that have a successor in T.

For every ω-regular property Ψ, it is well-known how to construct an equivalent μ-calculus formula φ [7, 4], which can then be used to model check $\exists\Psi$, i.e., to compute the set of states from which there is a path satisfying Ψ. Now suppose we want to solve the control problem with objective Ψ. The question we set out to answer in this paper is whether φ is of any use for this purpose; more specifically, if we simply replace all EPre operators in φ by $\mathit{1Pre}$ operators, do we obtain an algorithm for solving the game with objective Ψ, i.e., for computing the set of states from which player 1 can ensure that Ψ holds?

In general, the answer is negative. Consider the co-Büchi property $\Diamond\Box T$, which asserts that, eventually, the target T is reached and never left again. The Emerson-Lei translation [7] yields the equivalent μ-calculus formula

$$\exists \Diamond\Box T \;=\; \mu x.(\mathit{EPre}(x) \vee (\nu y.\mathit{EPre}(y) \wedge T)). \tag{2}$$

The Dam translation [4] gives

$$\exists \Diamond\Box T \;=\; \mu x.(\mathit{EPre}(x) \vee (T \wedge \mathit{EPre}(\nu y.(T \wedge \mathit{EPre}(y))))), \tag{3}$$

and Bhat-Cleaveland [2] produce the same result. But neither of these formulas give the correct solution for games. To see this, consider the following game on the state space $\{s_1, s_2, s_3\}$. At s_1, player 2 can play two moves: one of them keeps the game in s_1, the other takes the game to s_2. At s_2, player 1 can play two moves: one of them keeps the game in s_2, the other takes the game to s_3. Once in s_3, the game remains in s_3 forever. The target set is $T = \{s_1, s_3\}$. Then, $\langle\langle 1 \rangle\rangle \Diamond\Box T = \{s_1, s_2, s_3\}$. However, both equations (2) and (3) denote the smaller set $\{s_2, s_3\}$ when EPre is replaced by $\mathit{1Pre}$.

We present an extremal-model theorem which says that the fixpoint formula φ over $\mathit{1Pre}$ solves the game with ω-regular objective Ψ if and only if both of the following conditions are met:

- **E** The EPre version of φ is equivalent to the existential property $\exists\Psi$.
- **A** The APre version of φ is equivalent to the universal property $\forall\Psi$. (Here, $\mathit{APre}(T)$ is the set of states all of whose successors lie in T, and $\forall\Psi$ holds at a state if all paths from the state satisfy Ψ.)

In other words, for a fixpoint formula φ to solve the game with ω-regular objective Ψ, it is not only necessary but also sufficient that φ coincides with Ψ under the two extremal, non-game interpretations. In the co-Büchi example, while the expressions (2) and (3) satisfy condition **E** of the extremal-model theorem, they violate condition **A**. By contrast, in the reachability example, the expression (1) meets also condition **A**, because

$$\forall\Diamond T \;=\; \mu x.(T \vee \mathit{APre}(x)).$$

We show constructively that for every ω-regular objective Ψ there is indeed a fixpoint formula φ which meets both conditions of the extremal-model theorem. The construction is based on the determinization of ω-automata [12, 13], and on the translation from alternating ω-automata to μ-calculus [5]. In particular, for the co-Büchi property we obtain

$$\langle\!\langle 1\rangle\!\rangle\Diamond\Box T \;=\; \mu x.\nu y.(\mathit{1Pre}(x) \vee (\mathit{1Pre}(y) \wedge T)). \tag{4}$$

The reader may check that both

$$\begin{aligned}\exists\Diamond\Box T &= \mu x.\nu y.(\mathit{EPre}(x) \vee (\mathit{EPre}(y) \wedge T)),\\ \forall\Diamond\Box T &= \mu x.\nu y.(\mathit{APre}(x) \vee (\mathit{APre}(y) \wedge T)).\end{aligned}$$

In general, our translation provides optimal algorithms for solving games with ω-regular objectives; in particular, if the objective is given by a formula Ψ of linear temporal logic, then the resulting algorithm has a 2EXPTIME complexity in the length of Ψ [11].

Our results also shed light on a related question: given a "verification" μ-calculus formula φ_v that uses only the predecessor operator *EPre*, what is the relation between φ_v and its "control" version φ_c, obtained by replacing *EPre* with *1Pre*? From [6] we know that if φ_v is *deterministic,* i.e., if every conjunction in φ_v has at least one constant argument, then φ_v specifies an ω-regular language; that is, φ_v is equivalent to $\exists\Psi$ for some ω-regular property Ψ. We introduce the syntactic class of ***strongly deterministic*** μ-calculus formulas, a subclass of the deterministic formulas, and we show that if φ_v is strongly deterministic, then φ_v solves the verification problem for specification $\exists\Psi$ iff φ_c solves the control problem for objective Ψ. This correspondence does not hold in general for deterministic formulas.

We extend the connection between verification and control also to *quantitative* properties. Consider a graph with nonnegative edge weights, which represent costs. By defining an appropriate quantitative predecessor operator $\mathit{Pre}_{\mathbb{R}}$, the dynamic program for reachability, $\mu x.(T \vee \mathit{Pre}_{\mathbb{R}}(x))$, computes the cost of the shortest path to the target T. Similarly, consider a game whose moves incur costs. Then again, for a suitable quantitative predecessor operator $\mathit{1Pre}_{\mathbb{R}}$, the dynamic program $\mu x.(T \vee \mathit{1Pre}_{\mathbb{R}}(x))$ computes the real value of the game, which is defined as the minimal cost for player 1 to reach the target T (or infinity, if player 1 has no strategy to reach T). For general ω-regular objectives, we define the cost of the infinite outcome of a game as the cost of the shortest (possibly finite) prefix that is a witness to the objective. We show that the extremal-model theorem applies to this quantitative setting also. This gives us dynamic programs for solving the real-valued games with respect to all ω-regular objectives. For example, equation (4) with *1Pre* replaced by $\mathit{1Pre}_{\mathbb{R}}$ specifies a dynamic program for the quantitative co-Büchi game, whose value is the minimal cost for player 1 to reach and stay inside the target T (this cost is infinite unless player 1 can enforce an infinite sequence of moves all but finitely many of which have cost 0).

2 Reachability and Safety

We define our setting, and in doing so, review some well-known results about iterative solutions for simple verification, optimization, and control problems, where the objective is to reach or avoid a

$$\left\{\begin{matrix} \mathit{1Pre}^G_{\mathbb{B}} \\ \mathit{2Pre}^G_{\mathbb{B}} \end{matrix}\right\}(f)(s) = \left\{\begin{matrix} \exists a \in \Gamma_1(s).\forall b \in \Gamma_2(s) \\ \exists b \in \Gamma_2(s).\forall a \in \Gamma_1(s) \end{matrix}\right\}.\ f(\delta(s,a,b)) \tag{5}$$

$$\left\{\begin{matrix} \mathit{EPre}^G_{\mathbb{B}} \\ \mathit{APre}^G_{\mathbb{B}} \end{matrix}\right\}(f)(s) = \left\{\begin{matrix} \exists a \in \Gamma_1(s).\exists b \in \Gamma_2(s) \\ \forall a \in \Gamma_1(s).\forall b \in \Gamma_2(s) \end{matrix}\right\}.\ f(\delta(s,a,b)) \tag{6}$$

Figure 1: Boolean game and transition predecessor operators

given set of states (expending minimal cost).

2.1 Game structures

We define game structures over a global set A of *actions*, and a global set P of *propositions*. A (two-player) *game structure* $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$ (over A and P) consists of a finite set S of *states*, two action assignments $\Gamma_1, \Gamma_2\colon S \to 2^A \setminus \emptyset$ which define for each state two nonempty, finite sets of actions available to player 1 and player 2, respectively, a *transition function* $\delta\colon S \times A \times A \to S$ which associates with each state s and each pair of actions $a \in \Gamma_1(s)$ and $b \in \Gamma_2(s)$ a successor state, a *weight function* $w\colon S \times A \times A \to \mathbb{R}_{\geq 0}$ which associates with each state s and each pair of actions $a \in \Gamma_1(s)$ and $b \in \Gamma_2(s)$ a nonnegative real, and a proposition assignment $\langle\cdot\rangle\colon S \to 2^P$ which defines for each state s a finite set $\langle s\rangle \subseteq P$ of propositions that are true in s. Intuitively, at state s, player 1 chooses an action a from $\Gamma_1(s)$ and, simultaneously and independently, player 2 chooses an action b from $\Gamma_2(s)$. Then, the game proceeds to the successor state $\delta(s,a,b)$. The nonnegative real $w(s,a,b)$ represents the "cost" of the transition $\delta(s,a,b)$ (if it is to be minimized), or a "reward" (if it is to be maximized). Given a proposition $p \in P$, a state $s \in S$ is called a *p-state* iff $p \in \langle s\rangle$. If S is not given explicitly, then we write S^G to denote the state space of the game structure G.

Game structures are "concurrent" [1]; they subsume "turn-based" game structures (i.e., and-or graphs), where in each state at most one of the two players has a choice of actions. A special case of turn-based games are the one-player structures. A *one-player structure* is either a player-1 structure or a player-2 structure. The game structure G is a *player-1 structure* if $\Gamma_2(s)$ is a singleton for all states $s \in S$; and G is a *player-2 structure* if $\Gamma_1(s)$ is a singleton for all $s \in S$. In player-1 structures, player 2 has no choices, and in player-2 structures, player 1 has no choices. Every game structure G defines an *underlying transition structure* $K^G = (S, \to, \langle\cdot\rangle)$, where for all states $s, t \in S$, we have $s \to t$ iff there exist actions $a \in \Gamma_1(s)$ and $b \in \Gamma_2(S)$ such that $\delta(s,a,b) = t$. Transition structures do not distinguish between individual players.

Restrictions of game structures. A *player-1 restriction* of the game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$ is a game structure of the form $G_1 = (S, \Gamma'_1, \Gamma_2, \delta, \langle\cdot\rangle)$ with $\Gamma'_1(s) \subseteq \Gamma_1(s)$ for all states $s \in S$. Symmetrically, a *player-2 restriction* of G is a game structure of the form $G_2 = (S, \Gamma_1, \Gamma'_2, \delta, \langle\cdot\rangle)$ with $\Gamma'_2(s) \subseteq \Gamma_2(s)$ for all $s \in S$. In other words, for $i = 1, 2$, a player-i restriction of a game structure restricts the action choices that are available to player i.

Strategies and runs. Consider a game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$. A *player-i strategy*, for $i = 1, 2$, is a function $\xi_i\colon S^+ \to A$ that maps every nonempty, finite sequence of states to an action available to player i at the last state of the sequence; that is, $\xi_i(\overline{s}\cdot s) \in \Gamma_i(s)$ for every state sequence $\overline{s} \in S^*$ and every state $s \in S$. Intuitively, $\xi_i(\overline{s}\cdot s)$ indicates the choice taken by player i according to strategy ξ_i if the current state of the game is s, and the history of the game is $\overline{s}$. We write Ξ_i

for the set of player-i strategies. We distinguish the following types of strategies. The strategy ξ_i is *memoryless* if in every state $s \in S$, the choice of player i depends only on s; that is, $\xi_i(\overline{s} \cdot s) = \xi_i(s)$ for all state sequences $\overline{s} \in S^*$. The strategy ξ_i is *finite-memory* if in every state $s \in S$, the choice of player i depends only on s, and on a finite number of bits about the history of the game; the formal definition is standard [5].

A *run* r of the game structure G is a nonempty, finite or infinite sequence $s_0(a_0, b_0)s_1(a_1, b_1)s_2 \ldots$ of alternating states $s_j \in S$ and action pairs $(a_j, b_j) \in \Gamma_1(s_j) \times \Gamma_2(s_j)$ such that $s_{j+1} = \delta(s_j, a_j, b_j)$ for all $j \geq 0$. The first state s_0 is called the *source* of the run. The *weight* of the run is $w(r) = \Sigma_{j \geq 0} w(s_j, a_j, b_j)$; the weight $w(r)$ is either a real number, or infinity (if the sum diverges). Let $\xi_1 \in \Xi_1$ and $\xi_2 \in \Xi_2$ be a pair of strategies for player 1 and player 2, respectively. The *outcome* $R_{\xi_1,\xi_2}(s)$ from state $s \in S$ of the strategies ξ_1 and ξ_2 is a source-s infinite run of G, namely, $R_{\xi_1,\xi_2}(s) = s_0(a_0, b_0)s_1(a_1, b_1)s_2 \ldots$ such that (1) $s_0 = s$ and (2) for all $j \geq 0$, both $a_j = \xi_1(s_0 s_1 \cdots s_j)$ and $b_j = \xi_2(s_0 s_1 \cdots s_j)$.

2.2 Single-step verification and control

Values and valuations. A *value lattice* is a complete lattice $(V, \sqcup, \sqcap, \top, \bot)$ of *values* V with join $\sqcup$, meet $\sqcap$, top element $\top$, and bottom element $\bot$. Given $u, v \in V$, we write $u \sqsubseteq v$ iff $u = u \sqcap v$. Consider a game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$. A *valuation* f for G on the value lattice V is a function from states to values; that is, $f\colon S \to V$. The set $[S \to V]$ of valuations is again a lattice, with the lattice operations $(\sqcup, \sqcap, \top, \bot)$ defined pointwise; for example, for two valuations f_1 and f_2, we have $(f_1 \sqcup f_2)(s) = f_1(s) \sqcup f_2(s)$ for all states $s \in S$. If $f\colon S \to V$ is a valuation such that $f(s) \in \{\top, \bot\}$ for all states $s \in S$, then by $-f$ we denote the "complementary" valuation with $-f(s) = \top$ if $f(s) = \bot$, and $-f(s) = \bot$ if $f(s) = \top$. For a set $T \subseteq S$ of states, we write $[T]\colon S \to V$ for the valuation with $[T](s) = \top$ if $s \in T$, and $[T](s) = \bot$ if $s \notin T$. For a proposition $p \in P$, we write $[p]\colon S \to V$ for the valuation with $[p](s) = \top$ if $p \in \langle s \rangle$, and $[p](s) = \bot$ if $p \notin \langle s \rangle$.

Predecessor operators. Let V be a value lattice. Let *Pre* be a family of functions that contains, for every game structure G, a strict (i.e., bottom-preserving), monotone, and continuous function $Pre^G\colon [S^G \to V] \to [S^G \to V]$. The function family *Pre* is a *predecessor-1 operator on* V if for every game structure G, every player-1 restriction G_1 of G, every player-2 restriction G_2 of G, and every valuation $f\colon S^G \to V$, both $Pre^G(f) \sqsupseteq Pre^{G_1}(f)$ and $Pre^G(f) \sqsubseteq Pre^{G_2}(f)$. Symmetrically, the function family *Pre* is a *predecessor-2 operator on* V if for every game structure G, every player-1 restriction G_1 of G, every player-2 restriction G_2 of G, and every valuation $f\colon S^G \to V$, we have both $Pre^G(f) \sqsubseteq Pre^{G_1}(f)$ and $Pre^G(f) \sqsupseteq Pre^{G_2}(f)$. Intuitively, the more actions are available to player 1 in a game structure, the "better" (i.e., closer to top in the valuation lattice) the result of applying a predecessor-1 operator to a valuation, and the "worse" (i.e., closer to bottom) the result of applying a predecessor-2 operator.

Example 1: boolean game structures ("control"). Consider the *boolean value lattice* $V_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathtt{T}, \mathtt{F})$, where truth $\mathtt{T}$ is the top element and falsehood $\mathtt{F}$ is the bottom element. The valuations for a game structure G on $V_{\mathbb{B}}$ are called the *boolean valuations* for G; they correspond to the subsets of S^G. Figure 1 defines the predecessor operators $1Pre_{\mathbb{B}}$ and $2Pre_{\mathbb{B}}$, applied to a game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$, boolean valuation $f\colon S \to \mathbb{B}$, and state $s \in S$. For a set $T \subseteq S$ of states, the boolean valuation $1Pre^G_{\mathbb{B}}[T]\colon S \to \mathbb{B}$ of "controllable predecessors" is true at the states from which player 1 can force the game into T in a single step, no matter which action player 2 chooses. The operator $2Pre_{\mathbb{B}}$ behaves symmetrically for player 2, and therefore solves the control problem for the player-2 objective of reaching the target set T in a single step. The operator $1Pre_{\mathbb{B}}$ is a predecessor-1 operator on $V_{\mathbb{B}}$, and $2Pre_{\mathbb{B}}$ is a predecessor-2 operator.

$$\left\{\begin{matrix} \mathit{1Pre}^G_{\mathbb{R}} \\ \mathit{2Pre}^G_{\mathbb{R}} \end{matrix}\right\}(f)(s) \;=\; \left\{\begin{matrix} \min_{a\in\Gamma_1(s)} .\max_{b\in\Gamma_2(s)} \\ \min_{b\in\Gamma_2(s)} .\max_{a\in\Gamma_1(s)} \end{matrix}\right\}.\; w(s,a,b) + f(\delta(s,a,b)) \tag{7}$$

$$\left\{\begin{matrix} \mathit{EPre}^G_{\mathbb{R}} \\ \mathit{APre}^G_{\mathbb{R}} \end{matrix}\right\}(f)(s) \;=\; \left\{\begin{matrix} \min_{a\in\Gamma_1(s)} .\min_{b\in\Gamma_2(s)} \\ \max_{a\in\Gamma_1(s)} .\max_{b\in\Gamma_2(s)} \end{matrix}\right\}.\; w(s,a,b) + f(\delta(s,a,b)) \tag{8}$$

Figure 2: Quantitative game and transition predecessor operators

Example 2: boolean transition structures ("verification"). Consider again the boolean value lattice $V_{\mathbb{B}}$. Figure 1 defines the predecessor operators $\mathit{EPre}_{\mathbb{B}}$ and $\mathit{APre}_{\mathbb{B}}$. For a set $T \subseteq S$ of states, the boolean valuation $\mathit{EPre}^G_{\mathbb{B}}[T]\colon S \to \mathbb{B}$ of "possible predecessors" is true at the states that have some successor in T; the boolean valuation $\mathit{APre}^G_{\mathbb{B}}[T]\colon S \to \mathbb{B}$ of "unavoidable predecessors" is true at the states that have all successors in T. For each game structure G, the functions $\mathit{EPre}^G_{\mathbb{B}}$ and $\mathit{APre}^G_{\mathbb{B}}$ correspond to the branching-time "next" operators $\exists\bigcirc$ and $\forall\bigcirc$, respectively, of temporal logic interpreted over the underlying transition structure K^G. Therefore, $\mathit{EPre}_{\mathbb{B}}$ and $\mathit{APre}_{\mathbb{B}}$ solve the verification problems with the specifications of possibly or certainly reaching the target set T in a single step. The operators $\mathit{EPre}_{\mathbb{B}}$ and $\mathit{APre}_{\mathbb{B}}$ are both predecessor-1 and predecessor-2 operators on $V_{\mathbb{B}}$.

Example 3: quantitative game structures ("optimal control"). Consider the *quantitative value lattice* $V_{\mathbb{R}} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, \max, 0, \infty)$, where 0 is the top element and ∞ is the bottom element. Intuitively, each value represents a cost, and the smaller the cost, the "better." In particular, $u \sqsubseteq v$ iff either $u, v \in \mathbb{R}_{\geq 0}$ and $u \geq v$, or $u = \infty$; that is, the lattice is based on the reverse ordering of the reals. The valuations for a game structure G on $V_{\mathbb{R}}$ are called the *quantitative valuations* for G; they are the functions from S^G to the interval $[0, \infty]$. Figure 2 defines the predecessor operators $\mathit{1Pre}_{\mathbb{R}}$ and $\mathit{2Pre}_{\mathbb{R}}$, applied to a game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$, quantitative valuation $f\colon S \to [0, \infty]$, and state $s \in S$. For a set $T \subseteq S$ of states, the quantitative valuation $\mathit{1Pre}^G_{\mathbb{R}}[T]\colon S \to [0, \infty]$ gives for each state the minimal cost for player 1 of forcing the game into T in a single step (if player 1 cannot force the game into T, then the cost is ∞). The operator $\mathit{2Pre}_{\mathbb{R}}$ behaves symmetrically for player 2, and therefore solves the optimal-control problem with the player-2 objective of reaching the target set T in a single step at minimal cost. The operator $\mathit{1Pre}_{\mathbb{R}}$ is a predecessor-1 operator on $V_{\mathbb{R}}$, and $\mathit{2Pre}_{\mathbb{R}}$ is a predecessor-2 operator.

Example 4: quantitative transition structures ("optimization"). Consider again the quantitative value lattice $V_{\mathbb{R}}$. Figure 2 defines the predecessor operators $\mathit{EPre}_{\mathbb{R}}$ and $\mathit{APre}_{\mathbb{R}}$. For a set $T \subseteq S$ of states, the quantitative valuation $\mathit{EPre}^G_{\mathbb{R}}[T]\colon S \to [0, \infty]$ gives for each state the weight of the minimal transition into T (or ∞, if no such transition exists), and $\mathit{APre}^G_{\mathbb{R}}[T]\colon S \to [0, \infty]$ gives for each state the weight of the maximal transition into T (or ∞, if some transition does not lead into T). These are the single-step shortest-path and single-step longest-path problems on the underlying transition structure K^G. The operators $\mathit{EPre}_{\mathbb{R}}$ and $\mathit{APre}_{\mathbb{R}}$ are both predecessor-1 and predecessor-2 operators on $V_{\mathbb{R}}$.

2.3 Multi-step verification and control

Multi-step verification ("Can a target set be reached in some number of steps?"), optimization ("What is the shortest path to the target?"), and control problems ("Can one player force the game into the target, in some number of steps, no matter what the other player does?") can be

$$\begin{Bmatrix} \langle\!\langle 1 \rangle\!\rangle^G_{\mathbb{B}} \\ \langle\!\langle 2 \rangle\!\rangle^G_{\mathbb{B}} \end{Bmatrix} (\Diamond p)(s) \;=\; \begin{Bmatrix} \exists \xi_1 \in \Xi_1 . \forall \xi_2 \in \Xi_2 \\ \exists \xi_2 \in \Xi_2 . \forall \xi_1 \in \Xi_1 \end{Bmatrix} . \; (R_{\xi_1,\xi_2}(s) \text{ has a prefix in } \Diamond p) \tag{9}$$

$$\begin{Bmatrix} \exists^G_{\mathbb{B}} \\ \forall^G_{\mathbb{B}} \end{Bmatrix} (\Diamond p)(s) \;=\; \begin{Bmatrix} \exists \xi_1 \in \Xi_1 . \exists \xi_2 \in \Xi_2 \\ \forall \xi_2 \in \Xi_2 . \forall \xi_1 \in \Xi_1 \end{Bmatrix} . \; (R_{\xi_1,\xi_2}(s) \text{ has a prefix in } \Diamond p) \tag{10}$$

$$\begin{Bmatrix} \langle\!\langle 1 \rangle\!\rangle^G_{\mathbb{R}} \\ \langle\!\langle 2 \rangle\!\rangle^G_{\mathbb{R}} \end{Bmatrix} (\Diamond p)(s) \;=\; \begin{Bmatrix} \inf_{\xi_1 \in \Xi_1} . \sup_{\xi_2 \in \Xi_2} \\ \inf_{\xi_2 \in \Xi_2} . \sup_{\xi_1 \in \Xi_1} \end{Bmatrix} . \; \{ w(r) \mid r \text{ is a prefix of } R_{\xi_1,\xi_2}(s) \text{ and } r \in \Diamond p \} \tag{11}$$

$$\begin{Bmatrix} \exists^G_{\mathbb{R}} \\ \forall^G_{\mathbb{R}} \end{Bmatrix} (\Diamond p)(s) \;=\; \begin{Bmatrix} \inf_{\xi_1 \in \Xi_1} . \inf_{\xi_2 \in \Xi_2} \\ \sup_{\xi_2 \in \Xi_2} . \sup_{\xi_1 \in \Xi_1} \end{Bmatrix} . \; \{ w(r) \mid r \text{ is a prefix of } R_{\xi_1,\xi_2}(s) \text{ and } r \in \Diamond p \} \tag{12}$$

Figure 3: Boolean and quantitative reachability games

solved by iterating the single-step solutions ("dynamic programming"). Here, we exemplify the solutions for the goals of reachability and safety; more general objectives will be dealt with in Section 4. In the following, consider a game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$ and a proposition $p \in P$.

Reachability. We define $\Diamond p$ to be the set of minimal finite runs of G that end in a p-state; that is, the run $s_0(a_0, b_0)s_1(a_1, b_1) \ldots s_m$ is in $\Diamond p$ iff (1) $p \in \langle s_m \rangle$ and (2) for all $0 \le j < m$, we have $p \notin \langle s_j \rangle$. Figure 3 defines four boolean valuations in $[S \to \mathbb{B}]$. The valuation $\langle\!\langle 1 \rangle\!\rangle^G_{\mathbb{B}} \Diamond p$ is true at the states from which player 1 can control the game to reach a p-state; the valuation $\langle\!\langle 2 \rangle\!\rangle^G_{\mathbb{B}} \Diamond p$ is true at the states from which player 2 can control the game to reach a p-state; the valuation $\exists^G_{\mathbb{B}} \Diamond p$ is true at the states from which the two players can collaborate to reach a p-state; the valuation $\forall^G_{\mathbb{B}} \Diamond p$ is true at the states from which no matter what the two players do, a p-state will be reached. The first two valuations specify boolean games with the reachability objective $\Diamond p$ for players 1 and 2, respectively; the last two valuations specify the branching-time properties $\exists \Diamond p$ and $\forall \Diamond p$ on the underlying transition structures.

Figure 3 also defines the four corresponding quantitative valuations in $[S \to [0, \infty]]$; we use the convention that the infimum of an empty set of nonnegative reals is ∞, and the supremum is 0. The valuation $\langle\!\langle 1 \rangle\!\rangle^G_{\mathbb{R}} \Diamond p$ gives for each state the minimal cost for player 1 to direct the game to a p-state (or ∞, if player 1 cannot direct the game to a p-state); the valuation $\langle\!\langle 2 \rangle\!\rangle^G_{\mathbb{R}} \Diamond p$ gives for each state the minimal cost for player 2 to direct the game to a p-state; the valuation $\exists^G_{\mathbb{R}} \Diamond p$ gives for each state the minimal cost to reach a p-state if both players collaborate; the valuation $\forall^G_{\mathbb{R}} \Diamond p$ gives for each state the maximal reward achievable, if both players collaborate, before a p-state is reached. The first two valuations specify quantitative games with the reachability objective $\Diamond p$ for players 1 and 2, respectively; the last two valuations specify shortest-path and longest-path problems on the underlying transition structure.

The boolean and quantitative valuations for the reachability objective $\Diamond p$ can be characterized by least-fixpoint expressions on the corresponding valuation lattice:

$$\langle\!\langle 1 \rangle\!\rangle^G_V \Diamond p \;=\; \mu x . \; ([p] \sqcup \mathit{1Pre}^G_V(x)), \tag{13}$$

$$\langle\!\langle 2 \rangle\!\rangle^G_V \Diamond p \;=\; \mu x . \; ([p] \sqcup \mathit{2Pre}^G_V(x)), \tag{14}$$

$$\exists^G_V \Diamond p \;=\; \mu x . \; ([p] \sqcup \mathit{EPre}^G_V(x)), \tag{15}$$

$$\forall^G_V \Diamond p \;=\; \mu x . \; ([p] \sqcup \mathit{APre}^G_V(x)), \tag{16}$$

$$\langle\langle 1\rangle\rangle_{\mathbb{B}}^{G}(\Box p)(s) \;=\; \exists \xi_1 \in \Xi_1 . \forall \xi_2 \in \Xi_2 .\ (R_{\xi_1,\xi_2}(s) \in \Box p) \tag{17}$$

$$\left\{\begin{matrix}\exists_{\mathbb{B}}^{G}\\ \forall_{\mathbb{B}}^{G}\end{matrix}\right\}(\Box p)(s) \;=\; \left\{\begin{matrix}\exists \xi_1 \in \Xi_1 . \exists \xi_2 \in \Xi_2\\ \forall \xi_2 \in \Xi_2 . \forall \xi_1 \in \Xi_1\end{matrix}\right\}.\ (R_{\xi_1,\xi_2}(s) \in \Box p) \tag{18}$$

$$\langle\langle 1\rangle\rangle_{\mathbb{R}}^{G}(\Box p)(s) \;=\; \inf_{\xi_1 \in \Xi_1} . \sup_{\xi_2 \in \Xi_2} . \{w(R_{\xi_1,\xi_2}(s)) \mid R_{\xi_1,\xi_2}(s) \in \Box p\} \tag{19}$$

$$\left\{\begin{matrix}\exists_{\mathbb{R}}^{G}\\ \forall_{\mathbb{R}}^{G}\end{matrix}\right\}(\Box p)(s) \;=\; \left\{\begin{matrix}\inf_{\xi_1 \in \Xi_1} . \inf_{\xi_2 \in \Xi_2}\\ \sup_{\xi_2 \in \Xi_2} . \sup_{\xi_1 \in \Xi_1}\end{matrix}\right\}.\ \{w(R_{\xi_1,\xi_2}(s)) \mid R_{\xi_1,\xi_2}(s) \in \Box p\} \tag{20}$$

Figure 4: Boolean and quantitative safety games

where $V \in \{\mathbb{B}, \mathbb{R}\}$, and the variable x ranges over the boolean valuations in $[S \to \mathbb{B}]$ if $V = \mathbb{B}$, and over the quantitative valuations in $[S \to [0,\infty]]$ if $V = \mathbb{R}$. In other words, a single fixpoint expression (namely, "$\Diamond p = \mu x.(p \vee \mathit{pre}(x))$") suffices for the solution of boolean and quantitative verification and control problems, provided the *pre*-operator is interpreted appropriately.

Fixpoint expressions prescribe algorithms. The solutions to the fixpoint equations (13)–(16) can be computed iteratively on the valuation lattice as the limit of a sequence $x_0, x_1, x_2, \ldots$ of valuations: let $x_0 = \bot$, and for all $k \geq 0$, let $x_{k+1} = [p] \sqcup \mathit{Pre}_V^G(x_k)$, where $\mathit{Pre} \in \{\mathit{1Pre}, \mathit{2Pre}, \mathit{EPre}, \mathit{APre}\}$. For our four examples, the iteration converges in a finite number of steps. This is well-known in the case of boolean game structures and in the case of quantitative transition structures; finite convergence can be shown inductively also for quantitative game structures.

Safety. The complement of a reachability objective is a safety objective. We define $\Box p$ to be the set of infinite runs of the game structure G that never leave p-states; that is, the run $s_0(a_0,b_0)s_1(a_1,b_1)\ldots$ is in $\Box p$ iff $p \in \langle s_j \rangle$ for all $j \geq 0$. Figure 4 defines the boolean and quantitative valuations for the safety objective $\Box p$. For example, the boolean valuation $\langle\langle 1\rangle\rangle_{\mathbb{B}}^{G}\Box p$ is true at the states from which player 1 can control the game to stay within p-states; the quantitative valuation $\exists_{\mathbb{R}}^{G}\Box p$ gives for each state the minimal cost of an infinite path that stays within p-states; the boolean valuation $\forall_{\mathbb{B}}^{G}\Box p$ is true at the states from which p is an invariant.

The boolean and quantitative valuations for the safety objective $\Box p$ can be characterized by greatest-fixpoint expressions on the corresponding valuation lattice:

$$\langle\langle 1\rangle\rangle_{V}^{G}\Box p \;=\; \nu x.\ ([p] \sqcap \mathit{1Pre}_V^G(x)), \tag{21}$$

$$\langle\langle 2\rangle\rangle_{V}^{G}\Box p \;=\; \nu x.\ ([p] \sqcap \mathit{2Pre}_V^G(x)), \tag{22}$$

$$\exists_{V}^{G}\Box p \;=\; \nu x.\ ([p] \sqcap \mathit{EPre}_V^G(x)), \tag{23}$$

$$\forall_{V}^{G}\Box p \;=\; \nu x.\ ([p] \sqcap \mathit{APre}_V^G(x)), \tag{24}$$

where $V \in \{\mathbb{B}, \mathbb{R}\}$. The solutions to these fixpoint equations can again be computed iteratively as the limit of a sequence $x_0, x_1, x_2, \ldots$ of valuations: let $x_0 = \top$, and for all $k \geq 0$, let $x_{k+1} = [p] \sqcap \mathit{Pre}_V^G(x_k)$. This iteration converges for boolean game structures in a finite number of steps, but not necessarily for quantitative game or transition structures, where convergence may require ω many steps.

$$\langle\!\langle 1\rangle\!\rangle_{\mathbb{B}}^{G}\Psi(s) \;=\; \exists\xi_1\in\Xi_1.\forall\xi_2\in\Xi_2.\;(\langle R_{\xi_1,\xi_2}(s)\rangle\in L(\Psi)) \tag{25}$$

$$\begin{Bmatrix}\exists_{\mathbb{B}}^{G}\\ \forall_{\mathbb{B}}^{G}\end{Bmatrix}\Psi(s) \;=\; \begin{Bmatrix}\exists\xi_1\in\Xi_1.\exists\xi_2\in\Xi_2\\ \forall\xi_2\in\Xi_2.\forall\xi_1\in\Xi_1\end{Bmatrix}.\;(\langle R_{\xi_1,\xi_2}(s)\rangle\in L(\Psi)) \tag{26}$$

$$\langle\!\langle 1\rangle\!\rangle_{\mathbb{R}}^{G}\Psi(s) \;=\; \inf_{\xi_1\in\Xi_1}.\sup_{\xi_2\in\Xi_2}.\;\{w(r[0..\kappa(\langle r\rangle,\Psi)])\mid r=R_{\xi_1,\xi_2}(s)\text{ and }\langle r\rangle\in L(\Psi)\} \tag{27}$$

$$\begin{Bmatrix}\exists_{\mathbb{R}}^{G}\\ \forall_{\mathbb{R}}^{G}\end{Bmatrix}\Psi(s) \;=\; \begin{Bmatrix}\inf_{\xi_1\in\Xi_1}.\inf_{\xi_2\in\Xi_2}\\ \sup_{\xi_2\in\Xi_2}.\sup_{\xi_1\in\Xi_1}\end{Bmatrix}.\;\{w(r[0..\kappa(\langle r\rangle,\Psi)])\mid r=R_{\xi_1,\xi_2}(s)\text{ and }\langle r\rangle\in L(\Psi)\} \tag{28}$$

Figure 5: Boolean and quantitative LTL games

3 An Extremal Model Theorem

For verification problems, fixpoint solutions are known for much richer objectives ("specifications") than reachability and safety, and a fixpoint theory —the μ-calculus— is available for this purpose. In the case of reachability and safety, the fixpoint expressions we provided (namely, $\Diamond p = \mu x.\,(p \vee pre(x))$ and $\Box p = \nu x.\,(p \wedge pre(x))$) solve both the verification and control problems. This is not always the case: as we pointed out in the introduction, there are fixpoint expressions that solve a verification problem over transition structures, but do not solve the corresponding control problem over game structures. We now characterize the fixpoint expressions that solve both verification and control problems, provided the predecessor operators are interpreted appropriately.

3.1 Linear temporal logic

Consider a game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$. We express winning objectives for the infinite game played on G by formulas of linear temporal logic (LTL). The *LTL formulas* are generated by the grammar

$$\Psi \;::=\; p \mid \neg\Psi \mid \Psi\vee\Psi \mid \bigcirc\Psi \mid \Psi\,\mathcal{U}\,\Psi,$$

where $p \in P$ is a proposition, $\bigcirc$ is the "next" operator, and $\mathcal{U}$ is the "until" operator. Additional constructs such as $\Diamond\Psi = \mathrm{T}\,\mathcal{U}\Psi$ and $\Box\Psi = \neg\Diamond\neg\Psi$ can be defined in the standard way. A *trace* π: $\omega \to 2^P$ is an infinite sequence of sets of propositions. Every LTL formula Ψ has a truth value on each trace. We write $L(\Psi)$ for the set of traces that satisfy Ψ; a formal definition of $L(\Psi)$ can be found in [9].

Boolean LTL games. Every infinite run $r = s_0(a_0, b_0)s_1(a_1, b_1)s_2\ldots$ of the game structure G induces a trace $\langle r\rangle = \langle s_0\rangle\langle s_1\rangle\langle s_2\rangle\ldots$ Consider a state $s \in S$ and an LTL formula Ψ. We say that *player 1 can control state s for objective Ψ in the game structure G* if player 1 has a strategy $\xi_1 \in \Xi_1$ such that for all strategies $\xi_2 \in \Xi_2$ of player 2, the trace induced by the outcome of the game satisfies the formula Ψ; that is, $\langle R_{\xi_1,\xi_2}(s)\rangle \in L(\Psi)$. A suitable strategy ξ_1 is a *winning player-1 strategy for Ψ from s in G*. We write $\langle\!\langle 1\rangle\!\rangle_{\mathbb{B}}^{G}\Psi$: $S \to \mathbb{B}$ for the boolean valuation that is true at the states which can be controlled by player 1 for Ψ in G; see Figure 5. The player-2 winning valuation $\langle\!\langle 2\rangle\!\rangle_{\mathbb{B}}^{G}\Psi$ is defined symmetrically. Figure 5 also defines the boolean valuation $\exists_{\mathbb{B}}^{G}\Psi$: $S \to \mathbb{B}$, which is true at the states that satisfy the existential CTL* formula $\exists\Psi$ in the underlying transition structure K^G; and the boolean valuation $\forall_{\mathbb{B}}^{G}\Psi$: $S \to \mathbb{B}$, which is true at the states that satisfy the universal CTL* formula $\forall\Psi$ in K^G.

Quantitative LTL games. By $\langle\!\langle 1\rangle\!\rangle^G_{\mathbb{R}}\Psi$ we wish to denote the minimal cost for player-1 to achieve the objective Ψ. Recall the previous section. In reachability games, we compute the cost of winning as the weight of a finite run that reaches the target, while in safety games, the cost of winning is the weight of an infinite run. This is because upon reaching the target, we know that the reachability objective is satisfied, while a safety objective can be witnessed only by the entire infinite run generated by a game. We generalize this principle to arbitrary LTL formulas by defining the satisfaction index of a trace with respect to an LTL formula. Given a trace $\pi = \pi_0\pi_1\pi_2\ldots$ and a nonnegative integer k, the trace $\pi' = \pi'_0\pi'_1\pi'_2\ldots$ is a *k-variant* of π iff $\pi_j = \pi'_j$ for all $0 \le j \le k$. Let $\Lambda(\pi, k)$ be the set of k-variants of π. For a trace π and an LTL formula Ψ, the *satisfaction index* $\kappa(\pi,\Psi)$ is the smallest integer $k \ge 0$ such that $\Lambda(\pi, k) \subseteq L(\Psi)$ if such a k exists, and $\kappa(\pi,\Psi) = \infty$ otherwise. Intuitively, $\kappa(\pi,\Psi)$ the minimal number of steps after which we can conclude that the trace π satisfies the formula Ψ.

For an infinite run r and a nonnegative integer k, let $r[0..k]$ be the the prefix of r that contains k states. Given an LTL formula Ψ, the quantitative valuation $\langle\!\langle 1\rangle\!\rangle^G_{\mathbb{R}}\Psi\colon S \to [0,\infty]$ is formally defined in Figure 5. For each state $s \in S$, we say that $\langle\!\langle 1\rangle\!\rangle^G_{\mathbb{R}}\Psi(s)$ is the *player-1 value of the game with objective Ψ at the state s of the game structure G*. A strategy ξ_1 that attains the infimum is an *optimal player-1 strategy for Ψ from s in G*. The player-2 valuation $\langle\!\langle 2\rangle\!\rangle^G_{\mathbb{R}}\Psi$ is defined symmetrically. Figure 5 also defines the quantitative valuation $\exists^G_{\mathbb{R}}\Psi\colon S \to [0,\infty]$, which for each state s gives the minimum cost necessary for determining that some path from s in the underlying transition structure K^G satisfies Ψ (or ∞, if no such path exists). Dually, the valuation $\forall^G_{\mathbb{R}}\Psi\colon S \to [0,\infty]$ gives for each state s the maximal reward attainable along some path from s in K^G until Ψ can no longer be violated.

3.2 Fixpoint calculi for games

We define a family of fixpoint logics on game structures. The *fixpoint formulas* are generated by the grammar

$$\varphi \ ::= \ p \mid \neg p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathit{pre}_1(\varphi) \mid \mathit{pre}_2(\varphi) \mid \mu x.\,\varphi \mid \nu x.\,\varphi,$$

for propositions $p \in P$ and variables x. A fixpoint formula φ is a *one-player formula* if either it contains no pre_2-operator, or it contains no pre_1-operator. In the former case, φ is a *player-1 formula*; in the latter case, a *player-2 formula*. Given a value lattice V, a predecessor-1 operator Pre_1 on V, and a predecessor-2 operator Pre_2 on V, the closed fixpoint formulas form a logic on game structures: for every game structure G, every closed fixpoint formula $\varphi(\mathit{Pre}_1, \mathit{Pre}_2)$ specifies a valuation $[\![\varphi]\!]^G\colon S^G \to V$. The syntactic operator pre_1 is interpreted semantically as the predecessor-1 operator Pre_1, and pre_2 is interpreted as Pre_2. To make the interpretation of the *pre*-operators explicit, we sometimes write $\varphi(\mathit{Pre}_1, \mathit{Pre}_2)$ when naming a fixpoint formula. Then, $\varphi(\mathit{Pre}'_1, \mathit{Pre}'_2)$ describes the syntactically identical fixpoint formula, with the pre_1-operator interpreted as Pre'_1, and pre_2 interpreted as Pre'_2. Likewise, the one-player formulas have only a single predecessor operator as argument.

We now define the semantics of fixpoint formulas formally. Let V be a value lattice V, let Pre_1 be a predecessor-1 operator on V, and let Pre_2 be a predecessor-2 operator on V. Let G be a game structure. A *variable environment* $\mathcal{E}$ for G is a function that maps every variable x to a valuation in $[S^G \to V]$. We write $\mathcal{E}[x \mapsto f]$ for the function that agrees with $\mathcal{E}$ on all variables, except that x is mapped to the valuation f. Given V, Pre_1, Pre_2, G, and a variable environment $\mathcal{E}$ for G, each fixpoint formula φ specifies a valuation $[\![\varphi]\!]^G_{\mathcal{E}}\colon S_G \to V$, which is defined inductively by the following equations:

$$
\begin{array}{rcl}
[\![p]\!]^G_{\mathcal{E}} & = & [p] \\
[\![\neg p]\!]^G_{\mathcal{E}} & = & -[p] \\
[\![x]\!]^G_{\mathcal{E}} & = & \mathcal{E}(x) \\
[\![\varphi_1 \{{}^{\vee}_{\wedge}\} \varphi_2]\!]^G_{\mathcal{E}} & = & [\![\varphi_1]\!]^G_{\mathcal{E}} \, \{{}^{\sqcup}_{\sqcap}\} \, [\![\varphi_2]\!]^G_{\mathcal{E}} \\
[\![\{{}^{pre_1}_{pre_2}\}(\varphi)]\!]^G_{\mathcal{E}} & = & \{{}^{Pre^G_1}_{Pre^G_2}\} [\![\varphi]\!]^G_{\mathcal{E}} \\
[\![\{{}^{\mu}_{\nu}\} x.\ \varphi]\!]^G_{\mathcal{E}} & = & \{{}^{\sqcap}_{\sqcup}\}\{f : S^G \to V \mid f = [\![\varphi]\!]^G_{\mathcal{E}[x \mapsto f]}\}
\end{array}
$$

All right-hand-side (semantic) operations are performed on the valuation lattice $[S^G \to V]$. If φ is a closed formula, then $[\![\varphi]\!]^G = [\![\varphi]\!]^G_{\mathcal{E}}$ for any variable environment $\mathcal{E}$.

Provided that the predecessor operators Pre_1 and Pre_2 on V are computable, each formula $\varphi(Pre_1, Pre_2)$ prescribes a dynamic program for computing the valuation $[\![\varphi]\!]^G$ over a game structure G by iterative approximation.

Example: mu-calculus. Choose the boolean value lattice $V_{\mathbb{B}}$, and the predecessor operators $Pre_1 = EPre_{\mathbb{B}}$ and $Pre_2 = APre_{\mathbb{B}}$. The resulting logic on game structures coincides is the μ-calculus [8] on the underlying transition structures.

Example: boolean game calculus. Choose the boolean value lattice $V_{\mathbb{B}}$, and the predecessor operators $Pre_1 = 1Pre_{\mathbb{B}}$ and $Pre_2 = 2Pre_{\mathbb{B}}$. The resulting logic on game structures is the alternating-time μ-calculus of [1]. The player-i fragment, for $i = 1, 2$, is expressive enough to compute the winning states for player i with respect to any LTL objective.

Example: quantitative game calculus. Choose the quantitative value lattice $V_{\mathbb{R}}$, and the predecessor operators $Pre_1 = 1Pre_{\mathbb{R}}$ and $Pre_2 = 2Pre_{\mathbb{R}}$. The resulting logic may be called the *quantitative game calculus*. We shall see that the player-i fragment, for $i = 1, 2$, is expressive enough to compute all player-i values with respect to any LTL objective.

Example: quantitative mu-calculus. Choose the quantitative value lattice $V_{\mathbb{R}}$, and the predecessor operators $Pre_1 = EPre_{\mathbb{R}}$ and $Pre_2 = APre_{\mathbb{R}}$. The resulting logic may be called the *quantitative μ-calculus*. It can be used to compute, for example, the minimal and maximal weights of paths that satisfy LTL formulas in transition structures.

Monotonicity. The following monotonicity property of fixpoint formulas will be useful.

Lemma 1 *For every game structure G, every 1-restriction G_1 of G, every 2-restriction G_2 of G, and every player-1 fixpoint formula φ, we have $[\![\varphi]\!]^G \sqsupseteq [\![\varphi]\!]^{G_1}$ and $[\![\varphi]\!]^G \sqsubseteq [\![\varphi]\!]^{G_2}$. A symmetrical result holds for player-2 formulas.*

Lean fixpoint formulas. We shall use fixpoint formulas as algorithms for computing the values of LTL games. The quantitative interpretation of a fixpoint formula, however, does not take into account the satisfaction index of the corresponding LTL formula, and may compute the cost of a trace even beyond the satisfaction index. For example, the LTL formula $\bigcirc \mathrm{T}$ has the satisfaction index 0, because every state has a successor. Hence $(\exists^G_{\mathbb{R}} \bigcirc \mathrm{T})(s) = 0$ for all game structures G and states $s \in S^G$. While $\exists^G_{\mathbb{B}} \bigcirc \mathrm{T} = [\![EPre_{\mathbb{B}}(\mathrm{T})]\!]^G$ for all game structures G, if $s \in S^G$ is a state all of whose outgoing transitions have positive weights, then $[\![EPre_{\mathbb{R}}(\mathrm{T})]\!]^G(s) > 0$. This motivates the definition of lean fixpoint formulas. A fixpoint formula φ is *valid* if for every game structure G and every state $s \in S^G$, we have $[\![\varphi(1Pre_{\mathbb{B}}, 2Pre_{\mathbb{B}})]\!]^G(s) = \mathrm{T}$. A fixpoint formula is *lean* if no valid subformula contains *pre*-operators.

From now on we will make heavy use of the following convenient notation. If f^G and g^G are two families of valuations, one each for every game structure G, then we write $f = g$ short for "$f^G = g^G$ for all game structures G."

Lemma 2 *Let Ψ be an LTL formula, and let φ be a lean one-player fixpoint formula. Then $\exists_{\mathbb{B}}\Psi = [\![\varphi(EPre_{\mathbb{B}})]\!]$ iff $\exists_{\mathbb{R}}\Psi = [\![\varphi(EPre_{\mathbb{R}})]\!]$, and $\forall_{\mathbb{B}}\Psi = [\![\varphi(APre_{\mathbb{B}})]\!]$ iff $\forall_{\mathbb{R}}\Psi = [\![\varphi(APre_{\mathbb{R}})]\!]$.*

3.3 From verification to control: a semantic criterion

The following theorem characterizes the fixpoint formulas that can be used for solving boolean as well as quantitative games with LTL winning objectives. The characterization reduces problems on two-player structures (control) and on quantitative structures (optimization) to problems on *boolean one-player* structures (verification), which are well-understood.

Theorem 1 *For every LTL formula Ψ and every lean player-i fixpoint formula φ, where $i = 1, 2$, the following four statements are equivalent:*

- $\langle\langle i\rangle\rangle_{\mathbb{R}}\Psi = [\![\varphi(iPre_{\mathbb{R}})]\!]$.
- $\langle\langle i\rangle\rangle_{\mathbb{B}}\Psi = [\![\varphi(iPre_{\mathbb{B}})]\!]$.
- $\exists_{\mathbb{R}}\Psi = [\![\varphi(EPre_{\mathbb{R}})]\!]$ *and* $\forall_{\mathbb{R}}\Psi = [\![\varphi(APre_{\mathbb{R}})]\!]$.
- $\exists_{\mathbb{B}}\Psi = [\![\varphi(EPre_{\mathbb{B}})]\!]$ *and* $\forall_{\mathbb{B}}\Psi = [\![\varphi(APre_{\mathbb{B}})]\!]$.

The theorem can be stated equivalently as follows:

> $\langle\langle i\rangle\rangle^G_{\mathbb{R}}\Psi = [\![\varphi(iPre_{\mathbb{R}})]\!]^G$ for all game structures G iff $\langle\langle i\rangle\rangle^G_{\mathbb{B}}\Psi = [\![\varphi(iPre_{\mathbb{B}})]\!]^G$ for all one-player structures G.

In other words, the fixpoint formula φ prescribes an algorithm for computing the boolean or quantitative values of games with the winning objective Ψ iff it does so on all boolean, extremal game structures, where one or the other player has no choice of actions.

Proof sketch. Clearly, a fixpoint formula φ that solves games with objective Ψ also works over one-player structures, which are special cases of games. For the implication from one-player to game structures, we argue by contradiction. We start with the boolean player-1 interpretation (the proof for player 2 is symmetric). First we notice that given a game structure G for which the two valuations $\langle\langle 1\rangle\rangle^G_{\mathbb{B}}\Psi$ and $[\![\varphi(1Pre_{\mathbb{B}})]\!]^G$ differ, we can construct a turn-based game structure G' for which the valuations differ as well. There are two cases. If $\langle\langle 1\rangle\rangle^{G'}_{\mathbb{B}}\Psi(s) < [\![\varphi(1Pre_{\mathbb{B}})]\!]^{G'}(s)$ for some state $s \in S^{G'}$, then we fix a finite-memory optimal strategy of player 2 and show that in the resulting player-1 structure G_1, there is a state t such that $(\exists^{G_1}_{\mathbb{B}}\Psi)(t) < [\![\varphi(EPre_{\mathbb{B}})]\!]^{G_1}(t)$. Similarly, if $\langle\langle 1\rangle\rangle^{G'}_{\mathbb{B}}\Psi(s) > [\![\varphi(1Pre_{\mathbb{B}})]\!]^{G'}(s)$ for some state $s \in S^{G'}$, then we fix a finite-memory optimal strategy of player 1 and argue on the resulting player-2 structure. The proof for quantitative games follows by a similar argument. Finally, we go from quantitative to boolean structures using Lemma 2. $\square$

Suppose we are given an LTL formula Ψ. For verifying whether some path of a transition structure K^G satisfies Ψ, we can construct a μ-calculus formula $\varphi(EPre_{\mathbb{B}})$ that is equivalent to $\exists_{\mathbb{B}}\Psi$ over all transition structures, and check $\varphi(EPre_{\mathbb{B}})$ over K^G; this is, in fact, a symbolic model checking algorithm for LTL [3]. Now suppose that we want player 1 to control the game structure G for the objective Ψ. Theorem 1 tells us whether we can simply substitute the controllable predecessor operator $1Pre_{\mathbb{B}}$ for the μ-calculus predecessor operator $EPre_{\mathbb{B}}$ in the fixpoint formula φ: the substitution works if and only if by substituting $APre_{\mathbb{B}}$ for $EPre_{\mathbb{B}}$ in φ we obtain a formula that is equivalent to the universal interpretation $\forall_{\mathbb{B}}\Psi$ of the LTL formula over all transition structures.

To see that this property is not trivial (i.e., not satisfied by every μ-calculus formula $\varphi(EPre_{\mathbb{B}})$ that is equivalent to $\exists_{\mathbb{B}}\Psi$), consider the co-Büchi formula $\Psi = \Diamond\Box p$. Over transition structures, $\exists\Diamond\Box p$ is equivalent to $\exists\Diamond\exists\Box p$, which is equivalent to the μ-calculus formula $\mu x.(\nu y.(p \wedge EPre_{\mathbb{B}}(y)) \vee EPre_{\mathbb{B}}(x))$; indeed, this is the result of the standard translation from LTL to the μ-calculus for co-Büchi formulas [7, 4]. However, the corresponding game formula $\mu x.(\nu y.(p \wedge 1Pre_{\mathbb{B}}(y)) \vee 1Pre_{\mathbb{B}}(x))$ does not compute the boolean valuation $\langle\langle 1\rangle\rangle_{\mathbb{B}}^{G}\Diamond\Box p$ for all game structures G: the game structure given in the introduction provides a counterexample. The criterion of Theorem 1 fails, because over transition structures, $\forall_{\mathbb{B}}\Diamond\Box p$ is not equivalent to $\forall\Diamond\forall\Box p$, and therefore $\forall_{\mathbb{B}}\Psi$ is not equivalent to $\mu x.(\nu y.(p \wedge APre_{\mathbb{B}}(y)) \vee APre_{\mathbb{B}}(x))$. This is not surprising, given that the solution of ω-regular games requires deterministic (and hence Rabin chain) ω-automata [15], whereas nondeterministic (and hence Büchi) ω-automata suffice for ω-regular verification. The translations of [7, 4] from LTL to the μ-calculus go via nondeterministic Büchi automata, and thus cannot be used to solve ω-regular games.

The following theorem characterizes the cost of checking the criterion given in Theorem 1. There is a gap between the lower and upper bounds, which is due to the gap between the best known lower and upper bounds for the equivalence problem between an LTL formula and a μ-calculus formula.

Theorem 2 *Let Ψ be an LTL formula, and let φ be a one-player fixpoint formula. The complexity of checking whether $\exists_{\mathbb{B}}\Psi = [\![\varphi]\!]$ is in 2EXPTIME and PSPACE-hard in the size of Ψ, and in EXPTIME in the size of φ. The complexity of checking whether $\forall_{\mathbb{B}}\Psi = [\![\varphi]\!]$ is the same.*

3.4 From verification to control: a syntactic criterion

Not all fixpoint formulas correspond to verification or control problems with respect to linear-time objectives. This is always the case, however, for the deterministic fixpoint formulas. The *deterministic* fixpoint formulas are generated by the grammar

$$\varphi \ ::= \ p \mid \neg p \mid x \mid \varphi \vee \varphi \mid p \wedge \varphi \mid \neg p \wedge \varphi \mid pre_1(\varphi) \mid pre_2(\varphi) \mid \mu x.\,\varphi \mid \nu x.\,\varphi.$$

From [6] we know that if $\varphi(EPre_{\mathbb{B}})$ is a one-player deterministic fixpoint formula, then there is an ω-regular language Θ such that $\exists_{\mathbb{B}}\Theta = [\![\varphi(EPre_{\mathbb{B}})]\!]$. However, the examples (2) and (3) in the introduction illustrate that for such a formula $\varphi(EPre_{\mathbb{B}})$, in general it is not the case that $\langle\langle 1\rangle\rangle_{\mathbb{B}}\Theta = [\![\varphi(1Pre_{\mathbb{B}})]\!]$. In other words, the correspondence between the deterministic fixpoint formula and the ω-regular language does not necessarily carry over from verification to control. It is then natural to ask what other conditions we need, in addition to determinism, for a one-player fixpoint formula to have related meanings in verification and control. We answer this question by introducing a subclass of the deterministic formulas. A fixpoint formula φ is ***strongly deterministic*** iff φ consists of a string of fixpoint quantifiers followed by a quantifier-free part ψ, which is generated by the grammar

$$\begin{aligned} \psi \ &::= \ p \mid \neg p \mid \psi \vee \psi \mid p \wedge \psi \mid \neg p \wedge \psi \mid pre_1(\chi) \mid pre_2(\chi), \\ \chi \ &::= \ x \mid \chi \vee \chi. \end{aligned}$$

Note that every strongly deterministic fixpoint formula is lean. The following theorem shows that the one-player strongly deterministic fixpoint formulas provide a syntactic class of fixpoint formulas for which the criterion of Theorem 1 applies. In particular, it follows that for every LTL formula Ψ, every one-player strongly deterministic fixpoint formula φ, and $i = 1, 2$, we have $\langle\langle i \rangle\rangle_{\mathbb{R}} \Psi = [\![\varphi(i Pre_{\mathbb{R}})]\!]$.

Theorem 3 *For every LTL formula Ψ and every one-player strongly deterministic fixpoint formula φ, we have $\exists_{\mathbb{B}} \Psi = [\![\varphi(EPre_{\mathbb{B}})]\!]$ iff $\forall_{\mathbb{B}} \Psi = [\![\varphi(APre_{\mathbb{B}})]\!]$.*

Proof sketch. A strongly deterministic formula starts with a quantifier prefix. In the sequence $\mu x_1.\nu x_2 \ldots \nu x_{2k}$ of alternating fixpoints, the "evaluation order" is $x_2 \succ x_4 \succ \cdots \succ x_{2k} \succ x_{2k-1} \succ \cdots \succ x_1$ (this reflects the extension of the variables when the expression is being evaluated). Using this evaluation order, every one-player strongly deterministic fixpoint formula $\varphi(EPre_{\mathbb{B}})$ can be brought into the normal form $\mu x_1.\nu x_2 \ldots \nu x_{2k}.(d_0 \vee \bigvee_{j=1}^{2k}(d_j \wedge EPre_{\mathbb{B}}(x_j)))$, for some $k > 0$ and some mutually exclusive boolean combinations $d_0, d_1, \ldots, d_{2k}$ of propositions. The theorem follows from the fact that this formula has essentially the same structure as the solution formula of a Rabin-chain game (cf. [5] and Section 4). $\square$

While the one-player strongly deterministic fixpoint formulas obey strict syntactic conditions, the proof of Theorem 3 shows that they suffice for solving all control problems with Rabin-chain objectives. In turn, every ω-regular property can be specified by a deterministic Rabin-chain automaton [10, 15]. We can therefore transform every control problem with an ω-regular objective into a control problem with a Rabin-chain objective that is to be solved on the automata-theoretic product of the given game structure and a Rabin-chain automaton. Hence, at the cost of possibly enlarging the game structure, the one-player strongly deterministic fixpoint formulas suffice for the solution of games with arbitrary ω-regular objectives.

4 Dynamic Programs for LTL

We show that for every LTL formula Ψ we can construct an equivalent fixpoint formula φ_Ψ that meets the criterion of Theorem 1. The formula φ_Ψ has the following properties: it solves both the verification problem (on transition structures) for specification Ψ and the control problem (on game structures) for objective Ψ, both under boolean and quantitative interpretations. The construction of φ_Ψ is optimal for the boolean case, in that the 2EXPTIME complexity of the resulting algorithm for solving boolean games with LTL objectives matches the hardness of the problem [11].

4.1 (Co)Büchi and Rabin-chain games

The objective of a *Büchi game* is an LTL formula of the form $\Box\Diamond p$, for a proposition $p \in P$, and the objective of a *co-Büchi game* is an LTL formula of the form $\Diamond\Box p$. For $V = \{\mathbb{B}, \mathbb{R}\}$ and $i = 1, 2$, the Büchi and co-Büchi valuations can be computed by the fixpoint formulas

$$\begin{aligned}\langle\langle i \rangle\rangle_V \Box\Diamond p &= [\![\nu y.\mu x.(i Pre_V(x) \vee (p \wedge i Pre_V(y)))]\!],\\ \langle\langle i \rangle\rangle_V \Diamond\Box p &= [\![\mu x.\nu y.(i Pre_V(x) \vee (p \wedge i Pre_V(y)))]\!].\end{aligned}$$

The objective a *Rabin-chain game* is an LTL formula of the form $\Phi = \bigvee_{j=0}^{k-1}(\Box\Diamond d_{2j} \wedge \neg\Box\Diamond d_{2j+1})$, where $k > 0$ is called the *index* of Φ, and $d_0, \ldots, d_{2k}$ are boolean combinations of propositions such that $\emptyset = [d_{2k}] \subseteq [d_{2k-1}] \subseteq \cdots \subseteq [d_0] = S^G$ for all game structures G. An alternative characterization of Rabin-chain games with objective Φ can be obtained by defining a family Ω_Φ^G: $S^G \to \{0, 1, \ldots, 2k-1\}$ of index functions, one for every game structure G, such that $\Omega_\Phi^G(s) = j$

for all states $s \in [d_j] \setminus [d_{j+1}]$. Given an infinite run r of G, let $\mathit{Inf}(r) \subseteq S^G$ be the set of states that occur infinitely often along r, and let $\mathit{MaxIndex}(\Omega_\Phi, r) = \max\{\Omega^G_\Phi(s) \mid s \in \mathit{Inf}(r)\}$ be the largest index of such a state. Then, the run r satisfies the objective Φ iff $\mathit{MaxIndex}(\Omega_\Phi, r)$ is even. For $V \in \mathbb{B}, \mathbb{R}$ and $i = 1, 2$, the Rabin-chain valuation can be computed by the fixpoint formula

$$\langle\langle i \rangle\rangle_V \Phi \;=\; [\![\lambda_{2k-1} x_{2k-1} ... \mu x_1 . \nu x_0 . \bigvee_{j=0}^{2k-1} (d_j \wedge \neg d_{j+1} \wedge i\mathit{Pre}_V(x_j))]\!],$$

where $\lambda_j = \nu$ if j is even, and $\lambda_j = \mu$ if j is odd (cf. [5]). Note that the fixpoint solutions for Büchi, co-Büchi, and Rabin-chain games are all one-player strongly deterministic fixpoint formulas.

4.2 LTL games

Given an LTL formula Ψ, we construct a lean one-player fixpoint formula φ_Ψ such that

$$\langle\langle i \rangle\rangle_V \Psi \;=\; [\![\varphi_\Psi(i\mathit{Pre}_V)]\!] \tag{29}$$

for $V \in \{\mathbb{B}, \mathbb{R}\}$ and $i = 1, 2$. Following [5, 10], our construction is based on deterministic Rabin-chain automata (also called *parity automata* [14]). A *Rabin-chain automaton of index k* over the input alphabet 2^P is a tuple $\mathcal{C} = (Q, Q_0, \Delta, \langle\cdot\rangle, \Omega)$, where Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, Δ: $Q \to 2^Q$ is the transition relation, $\langle\cdot\rangle$: $Q \to 2^P$ assigns propositions to states, and Ω: $Q \to \{0, \ldots, 2k-1\}$ is the acceptance condition. An *execution* of $\mathcal{C}$ from a source state $q_0 \in Q$ is an infinite sequence $q_0 q_1 q_2 \cdots$ of automaton states such that $q_{j+1} \in \Delta(q_j)$ for all $j \geq 0$; if $q_0 \in Q_0$, then the execution is *initialized.* The execution $e = q_0 q_1 q_2 \ldots$ is *generated* by the trace $\langle e \rangle = \langle q_0 \rangle \langle q_1 \rangle \langle q_2 \rangle \ldots$ The execution e is *accepting* if $\mathit{MaxIndex}(\Omega, e)$ is even. The *language* $L(\mathcal{C})$ is the set of traces π such that $\mathcal{C}$ has an initialized accepting execution e generated by π. The automaton $\mathcal{C}$ is *deterministic and total* if (1a) for all states $q', q'' \in Q_0$, if $q' \neq q''$, then $\langle q' \rangle \neq \langle q'' \rangle$; (1b) for all proposition sets $P' \subseteq P$, there is a state $q' \in Q_0$ such that $\langle q' \rangle = P'$; (2a) for all states $q \in Q$ and $q', q'' \in \Delta(q)$, if $q' \neq q''$, then $\langle q' \rangle \neq \langle q'' \rangle$; (2b) for all states $q \in Q$ and all proposition sets $P' \subseteq P$, there is a state $q' \in \Delta(q)$ such that $\langle q' \rangle = P'$. If $\mathcal{C}$ is deterministic and total, then we write $\Delta(q, P')$ for the unique state $q' \in \Delta(q)$ with $\langle q' \rangle = P'$.

From the LTL formula Ψ, we construct a deterministic, total Rabin-chain automaton $\mathcal{C}_\Psi$ such that $L(\Psi) = L(\mathcal{C}_\Psi)$, by first building a nondeterministic Büchi automaton with the language $L(\Psi)$ [16], and then determinizing it [12, 13]. Let $\mathcal{C}_\Psi = (Q, Q_0, \Delta, \langle\cdot\rangle, \Omega)$. In order to obtain a lean fixpoint formula φ_Ψ, we need to compute the set $F \subseteq Q$ of automaton states q such that all executions with source q are accepting. To this end, it suffices to compute the set $Q \setminus F$ of states q' such that there is an execution e with source q' and $\mathit{MaxIndex}(\Omega', e)$ is even, where Ω' is the complementary acceptance condition with $\Omega'(q) = (2k-1) - \Omega(q)$ for all states $q \in Q$. This corresponds to checking the nonemptiness of a Rabin-chain automaton [5].

We derive the fixpoint formula φ_Ψ that satisfies (29) in two steps. First, we build a fixpoint formula φ' that solves the game with objective Ψ on the product structure $G \times \mathcal{C}$, for all game structures G. From φ', we then construct the formula φ_Ψ that solves the game directly on G, for all G. Consider an arbitrary game structure $G = (S, \Gamma_1, \Gamma_2, \delta, \langle\cdot\rangle)$. Define $G \times \mathcal{C} = (S', \Gamma'_1, \Gamma'_2, \delta', \langle\cdot\rangle)$, where $S' = \{(s, q) \in S \times Q \mid \langle s \rangle = \langle q \rangle\}$, where $\Gamma'_i(s, q) = \Gamma_i(s)$ for $i = 1, 2$, where $\delta'((s, q), a_1, a_2) = (\delta(s, a_1, a_2), \Delta(q, \langle \delta(s, a_1, a_2) \rangle))$. Finally, for $q \notin F$ let $\langle s, q \rangle = \langle s \rangle \cup \{c_{\Omega(q)}\}$, and for $q \in F$ let $\langle s, q \rangle = \langle s \rangle \cup \{p_F, c_{\Omega(q)}\}$, where $p_F, c_0, \ldots, c_{2k-1}$ are new propositions.

We construct φ' by proceeding similarly to [2]. We give the fixpoint formula φ' in equational form; it can then be unfolded into a nested fixpoint formula in the standard way. The formula φ' is composed of blocks $B'_0, \dots, B'_{2k-1}$, where B'_0 is the innermost block and B'_{2k-1} the outermost block. The block B'_0 is a ν-block which consists of the single equation $x_0 = f \vee \bigvee_{j=0}^{2k-1}(c_j \wedge \mathit{pre}_1(x_j)))$. For $0 < \ell \leq 2k-1$, the block B'_ℓ is a μ-block if ℓ is odd, and a ν-block if ℓ is even; in either case it consists of the single equation $x_\ell = x_{\ell-1}$. The output variable is x_{2k-1}. Then, $(\langle\langle 1\rangle\rangle_{\mathbb{B}}^G \Psi)(s) = [\![\varphi'(\mathit{1Pre}_{\mathbb{B}})]\!]^{G\times\mathcal{C}}(s,q)$ for all states $s \in S$ and for the unique $q \in Q_0$ such that $(s,q) \in S'$.

The formula φ_Ψ mimics on G the evaluation of φ' on $G \times \mathcal{C}$. For each variable x_ℓ of φ', for $0 \leq \ell \leq 2k-1$, the formula φ_Ψ contains the set $\{x_\ell^q \mid q \in Q\}$ of variables: the value of x_ℓ^q at s keeps track of the value of x_ℓ at (s,q). The formula φ_Ψ is composed of the blocks $B_0, \dots, B_{2k-1}$: for $0 \leq \ell \leq 2k-1$, the block B_ℓ consists of the set $\{E_\ell^q \mid q \in Q\}$ of equations. The equation E_ℓ^q is derived from the equation for x_ℓ in φ' by replacing the variable x_ℓ on the left-hand side with the variable x_ℓ^q, by replacing c_j with $\mathtt{T}$ if $\Omega(q) = j$ and $\mathtt{F}$ otherwise, by replacing p_F with $\mathtt{T}$ if $q \in F$ and $\mathtt{F}$ otherwise, and by replacing $\mathit{pre}_1(x_j)$ with $\mathit{pre}_1(\bigvee_{q'\in\Delta(q)} x_j^{q'})$; the right-hand side is then conjuncted with the propositions in $\langle q\rangle$. The block B_{2k-1} contains the extra equation $x_{out} = \bigvee_{q\in Q_0} x_{2k-1}^q$, which defines the output variable x_{out}. Note that φ_Ψ is independent of the game structure G, and contains no propositions other than those in Ψ.

Theorem 4 *For every LTL formula Ψ and $i = 1, 2$, we have $\langle\langle i\rangle\rangle_{\mathbb{B}}\Psi = [\![\varphi_\Psi(\mathit{iPre}_{\mathbb{B}})]\!]$. Moreover, the fixpoint formula φ_Ψ is lean and its size is doubly exponential in the size of Ψ.*

Since φ_Ψ is lean, by Theorem 1 it follows that $\langle\langle i\rangle\rangle_{\mathbb{R}}\Psi = [\![\varphi_\Psi(\mathit{iPre}_{\mathbb{R}})]\!]$. The doubly exponential size of φ_Ψ is optimal, because boolean games with LTL objectives are 2EXPTIME-hard [11].

References

[1] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proc. 38th Symp. Foundations of Computer Science*, pp. 100–109. IEEE Computer Society, 1997.

[2] G. Bhat and R. Cleaveland. Efficient model checking via the equational μ-calculus. In *Proc. 11th Symp. Logic in Computer Science*, pp. 304–312. IEEE Computer Society, 1996.

[3] E.M. Clarke, O. Grumberg, and D.E. Long. Verification tools for finite-state concurrent systems. In *A Decade of Concurrency: Reflections and Perspectives*, LNCS 803, pp. 124–175. Springer, 1994.

[4] M. Dam. CTL* and ECTL* as fragments of the modal μ-calculus. *Theoretical Computer Science*, 126:77–96, 1994.

[5] E.A. Emerson and C. Jutla. Tree automata, mu-calculus, and determinacy. In *Proc. 32th Symp. Foundations of Computer Science*, pp. 368–377. IEEE Computer Society, 1991.

[6] E.A. Emerson, C.S. Jutla, and A.P. Sistla. On model checking for fragments of μ-calculus. In *CAV 93: Computer-aided Verification*, LNCS 697, pp. 385–396. Springer, 1993.

[7] E.A. Emerson and C. Lei. Efficient model checking in fragments of the propositional μ-calculus. In *Proc. First Symp. Logic in Computer Science*, pp. 267–278. IEEE Computer Society, 1986.

[8] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[9] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer, 1992.

[10] A.W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In *Proc. 5th Symp. Computation Theory*, LNCS 208, pp. 157–168. Springer, 1984.

[11] R. Rosner. *Modular Synthesis of Reactive Systems.* PhD Thesis, Weizmann Institute of Science, Rehovot, Israel, 1992.

[12] S. Safra. On the complexity of ω-automata. In *Proc. 29th Symp. Foundations of Computer Science*, pp. 319–327. IEEE Computer Society, 1988.

[13] S. Safra. Exponential determinization for ω-automata with strong-fairness acceptance condition. In *Proc. 24th Symp. Theory of Computing*, pp. 275–282. ACM, 1992.

[14] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, vol. B, pp. 133–191. Elsevier, 1990.

[15] W. Thomas. On the synthesis of strategies in infinite games. In *STACS 95: Theoretical Aspects of Computer Science*, LNCS 900, pp. 1–13. Springer, 1995.

[16] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1–37, 1994.