Temporal logics and model checking for fairly correct systems

Daniele Varacca*
Imperial College London, UK

Hagen Völzer
Universität zu Lübeck, Germany

Abstract

We motivate and study a generic relaxation of correctness of reactive and concurrent systems with respect to a temporal specification. We define a system to be fairly correct *if there exists a fairness assumption under which it satisfies its specification. Equivalently, a system is fairly correct if the set of runs satisfying the specification is large from a topological point of view, i.e., it is a* co-meager *set.*

We compare topological largeness with its more popular sibling, probabilistic largeness, where a specification is probabilistically large *if the set of runs satisfying the specification has probability 1. We show that topological and probabilistic largeness of ω-regular specifications coincide for bounded Borel measures on finite-state systems. As a corollary, we show that, for specifications expressed in LTL or by Büchi automata, checking that a finite-state system is fairly correct has the same complexity as checking that it is correct.*

Finally we study variants of the logics CTL and CTL, where the 'for all runs' quantifier is replaced by a 'for a large set of runs' quantifier. We show that the model checking complexity for these variants is the same as for the original logics.*

1 Introduction

Sometimes, a model of a concurrent or reactive system does not satisfy a desired linear-time temporal specification but the runs violating the specification seem to be artificial and rare. For example, in Dijkstra's dining philosophers, a philosopher may starve because his two neighbours 'conspire' against him by alternately eating in such a way that the philosopher's two forks are never available at the same time. If a specification prescribed starvation-freedom, such a run would obviously violate it. Although such runs exist, they require special conditions that in practice may not arise.

For this particular example, there are also starvation-free solutions, but for many problems, a system satisfying the actual specification is impossible, too difficult, or too expensive to obtain [13]. In such cases, we could be content with a system where the specification is almost satisfied, i.e., the set of runs satisfying the specification is 'large'.

One natural way to formalise 'large set' is to mean *probabilistically large*, i.e., a set of measure 1 for a given probability measure. This notion however needs a concrete probability measure, which may be hard to justify for a given system. Alternatively, one can define a *fairness assumption* under which the specification is satisfied, with the intuition that 'most' runs are fair. This intuition has a formal counterpart: Ben-Eliyahu and Magidor [7] observe that for many fairness notions from the literature, the set of fair runs is *topologically large*, i.e., a *co-meager* set in the natural topology of runs of a given system. Völzer *et al.* [28] show that this is in fact true for most of the existing fairness notions and they also give more arguments why fairness should be *defined* as co-meagerness in the natural topology. An important consequence of this definition is the following: A linear-time property X is topologically large in a system iff there exists a fairness assumption F such that $(F \Rightarrow X)$ is satisfied in that system.

The notions of probabilistic and topological largeness share many properties. A classic mathematical text book [23] is devoted to study their similarities and differences. Although similar, these notions do not coincide in general—in fact, even for the most straightforward probability measure on the set of runs, there are topologically large sets that have probability 0.

In this paper we propose to call a system *fairly correct* if the set of its runs that satisfy the specification is co-meager or, equivalently, if it is correct under some fairness assumption. We study the problem of verifying when a finite system is fairly correct for a specification expressed in some temporal logic, or via Büchi automata.

We prove that probabilistic and topological largeness of ω-regular specifications coincide for bounded Borel measures on finite-state systems. This allows us to decide fair correctness by using known algorithms for finite Markov chains.

In particular, we show that fair correctness of a finite system is decidable and can be checked with the same com-

*Funded by EPSRC grant GR/T04724/01

plexity as correctness for LTL and Büchi automata specifications (but without the necessity to specify any fairness assumption explicitly). We also show that fair correctness of a system with respect to an LTL+past specification is expressible in CTL+past, strengthening a result of Berwanger *et al.* [8].

Then, we consider variants of the logics CTL and CTL*, where the quantifier 'for all runs' is replaced by 'for a large set of runs'. We show that also for these logics, the model checking complexity is the same as in the standard case: PSPACE-complete for CTL* and linear for CTL.

The path quantifier 'for a large set of runs' also occurs (under a different point of view) in a logic introduced by Pistore and Vardi [24]. We reinterpret their work from a topological point of view, which allows us to derive some basic properties of their logic.

2 Preliminary notions

2.1 Systems and temporal properties

Let Σ be a countable set of *states*. Σ^*, Σ^+ and Σ^ω denote the set of finite, nonempty finite, and infinite sequences over Σ respectively. Finite sequences are denoted α, β and infinite ones by x, y. We set $\alpha\uparrow = \{x \mid \alpha \text{ is a prefix of } x\}$ and $x\downarrow = \{\alpha \mid \alpha \text{ is a prefix of } x\}$. A set $X \subseteq \Sigma^\omega$ is called a *(linear-time temporal) property* and a set $Q \subseteq \Sigma^+$ a *finitary (temporal) property*.

Let AP be a nonempty set of *atomic propositions*. A *temporal structure* $M = (\Sigma, R, L)$ over AP consists of a set Σ of *states*, a total binary relation $R \subseteq \Sigma \times \Sigma$, and a mapping $L : \Sigma \to 2^{AP}$. A pair (M, s_0) of a temporal structure M and a state s_0 of M will be called a *system*. A *path* (*path fragment*) of (M, s_0) is an infinite (finite) sequence $s_0, s_1, \ldots$ that starts in s_0 such that $(s_i, s_{i+1}) \in R$ for all $i \geq 0$. The set of all paths of (M, s_0) is denoted by $M(s_0)$.

2.2 Temporal-logical properties

We consider various temporal logics here. The most expressive one is CTL*+past [16], which is defined by the following syntax rules (S1)-(P1), where a ranges over atomic propositions, p over *state formulas*, h over *history formulas*, and ϕ over *path formulas*:

$$p := a \mid \neg p \mid p \wedge p \tag{S1}$$
$$p := \mathsf{A}\,\phi \mid \mathsf{E}\,\phi \tag{S2}$$
$$h := p \mid \neg h \mid h \wedge h \mid \mathsf{Y}\,h \mid h\ \mathsf{S}\ h \tag{H1}$$
$$h := \phi \tag{H2}$$
$$\phi := h \mid \neg\phi \mid \phi \wedge \phi \mid \mathsf{X}\,\phi \mid \phi\ \mathsf{U}\ \phi \tag{P1}$$

LTL+past is the sublanguage where rule (S2) is removed. CTL+past [16] is the sublanguage where (H2) is removed and (P1) is replaced by (P3) below. Finally, versions without past are defined by replacing (H1) by (H3) below.

$$\phi := \mathsf{X}\,h \mid h\ \mathsf{U}\ h \tag{P3}$$
$$h := p \tag{H3}$$

Satisfaction is defined as usual [10, 16]. In particular, we follow the *Ockhamist* interpretation of the past, where each state has a unique past. The semantics of history formulas is as follows:

- $M, x, i \models \mathsf{Y}\,h$ iff $(i > 0)$ and $M, x, i-1 \models h$
- $M, x, i \models h\ \mathsf{S}\ g$ iff $\exists j \leq i : M, x, j \models g$ and $\forall k : j < k \leq i : M, x, k \models h$

Additional operators, such as $\bot$, $\top$, $\mathsf{F}\,\phi$, $\mathsf{G}\,\phi$, $\mathsf{F}^{-1}\,\phi$, $(\phi\ \mathsf{W}\ \psi)$, etc. are also defined as usual. In particular, $(\phi\ \mathsf{W}\ \psi)$ stands for $(\phi\ \mathsf{U}\ \psi) \vee \mathsf{G}\,\phi$ and $\mathsf{F}^{-1}\,\phi$ for $\top\ \mathsf{S}\ \phi$. For an LTL(+past) formula ϕ, we will also use ϕ to denote the set of paths that satisfy ϕ when no confusion arises.

An *ω-regular property* is a property that is accepted by some *Büchi automaton* (see e.g. [26]).

2.3 Path games and the Pistore-Vardi logic

The path quantifiers A ('for all paths') and E ('there exists a path') of CTL* are two extreme notions of satisfaction of a path formula in a system. We can think of them as a hostile player (A) and a friendly player (E) that resolve the nondeterminism in the system. Player A tries to violate the formula and E tries to satisfy the formula. Intermediate notions of satisfaction can be derived through a *path game* [8, 24] where hostile and friendly player alternately resolve the nondeterminism for some time.

Let $\kappa \in \{\mathsf{A}, \mathsf{E}\}^\omega$, $X \subseteq \Sigma^\omega$ a linear-time property, and (M, s_0) be a system. The game $G(\kappa, X, M, s_0)$ is played by the two players A (Alter) and E (Ego) and the state of a play is a path fragment of (M, s_0). A play starts in s_0 and in the i-th move $(i \geq 0)$, player $\kappa(i)$ extends by a finite, possibly empty[1] sequence α_i yielding the path fragment $s_0\alpha_0 \ldots \alpha_i$. The play goes on forever converging either to a path x or a path fragment α of the system. Ego wins if $x \in X$ (resp. $\alpha\uparrow \subseteq X$), otherwise Alter wins.

A *strategy* is a mapping[2] $f : \Sigma^+ \to \Sigma^*$ such that for each path fragment α of (M, s), $\alpha f(\alpha)$ is a path fragment of (M, s). A strategy f is *winning* for player $P \in \{\mathsf{A}, \mathsf{E}\}$ if for each strategy g of the other player, P wins the play that results from P playing f and the other player playing g.

It can be shown [24, 8] that each game $G(\kappa, X, M, s_0)$ is equivalent to a game $G(\kappa', X, M, s_0)$ where κ' is one of the

[1] This version of the game is essentially equivalent with those described in [8, 24].

[2] More general strategies that depend on what moves produced the current path fragment are not more powerful in the game considered here.

following: $\mathsf{A}^\omega, \mathsf{AEA}^\omega, (\mathsf{AE})^\omega, \mathsf{AE}^\omega, \mathsf{EA}^\omega, (\mathsf{EA})^\omega, \mathsf{EAE}^\omega, \mathsf{E}^\omega$, abbreviated A, AEA, $\overline{\mathsf{AE}}$, AE, EA, $\overline{\mathsf{EA}}$, EAE, E respectively.

Pistore and Vardi [24] proposed the following extension of LTL. A formula in the Pistore-Vardi logic is of the form $\kappa.\phi$ where $\kappa \in \{\mathsf{A}, \mathsf{E}\}^\omega$ is a *path quantifier* and ϕ is an LTL formula. Satisfaction is defined by

- $M, s_0 \models \kappa.\phi$ iff Ego has a winning strategy in the game $G(\kappa, \phi, M, s_0)$.

More general, we write $M, s_0 \models \kappa.X$ for any $X \subseteq \Sigma^\omega$ iff Ego has a winning strategy in the game $G(\kappa, X, M, s_0)$. It is known [8] that $G(\kappa, X, M, s_0)$ is *determined*, i.e., either Ego or Alter has a winning strategy if X is ω-regular.

To exemplify the properties that can be expressed in the Pistore-Vardi logic, consider the temporal structure $M = (\Sigma, R, L)$, where $\Sigma = \{a, b\}, R = \Sigma \times \Sigma, AP = \Sigma$, and $L(a) = \{a\}, L(b) = \{b\}$. The system (M, a) generates all infinite sequences on $\{a, b\}$ starting with a. We therefore have: $M, a \models \mathsf{A}.a$, $M, a \models \mathsf{AEA}.\mathsf{F}\,b$, $M, a \models \overline{\mathsf{AE}}.\mathsf{G}\,\mathsf{F}\,b$, $M, a \models \mathsf{AE}.\mathsf{F}\,\mathsf{G}\,b$, $M, a \models \mathsf{E}.\mathsf{G}\,a$.

3 Topological classifications

Different topological classifications of linear-time properties have improved our understanding of the verification problem. In this section, we recall those topological classifications.

3.1 Some topological notions

The natural, i.e., Cantor topology on Σ^ω is defined by the sets of the form $\alpha\uparrow$ for $\alpha \in \Sigma^*$. Such a set is a *basic open set* of that topology. As usual, an *open set* is an arbitrary union of basic open sets, a *closed set* is the complement of some open set, and a *dense set* is a set that has a nonempty intersection with every open set.

The family G of open sets is closed under arbitrary union and finite intersection. By duality, the family F of closed sets is closed under arbitrary intersection and finite union. Given a family $\mathscr{F} \subseteq 2^{\Sigma^\omega}$, the family $\mathscr{F}_\delta$ ($\mathscr{F}_\sigma$) is the family of countable intersections (unions) of members of $\mathscr{F}$. Thus, G_δ is the family of sets that can be represented as the intersection of countably many open sets. The family of *Borel sets* is the smallest family $\mathscr{F} \subseteq 2^{\Sigma^\omega}$ that contains all open and closed sets and is closed under countable union and intersection.

Each nonempty set $X \subseteq \Sigma^\omega$ is equipped with the Cantor topology *relative to* X, which is defined by the basic open sets of the form $\alpha\uparrow \cap X$ for each $\alpha \in \Sigma^*$. In particular, a system (M, s_0) defines the Cantor topology relative to $M(s_0)$. A set is *dense in* X if it is dense in the topology relative to X.

A set X is *somewhere dense* if it is dense in some open set. X is *nowhere dense* if it is not somewhere dense (or equivalently, if its complement contains a dense open set). A set is *meager* if it is the countable union of nowhere dense sets. A complement of a meager set is called *co-meager*. In this paper, we also say that a co-meager set is *topologically large* or *T-large*.

3.2 The safety-liveness classification

We say that a property $X \subseteq \Sigma^\omega$ is *live in* $\alpha \in \Sigma^*$ if $\alpha\uparrow \cap X \neq \emptyset$. Following Alpern and Schneider [1], a *safety property* is a property X such that $x \notin X$ implies that x has a finite prefix α where X is not live. X is *live* in a safety property S (also: (S, X) is *machine-closed*) if X is live in every $\alpha \in S\downarrow$, where $S\downarrow := \bigcup_{x \in S} x\downarrow$. X is a *liveness property* if X is live in every $\alpha \in \Sigma^*$.

It is easy to see that safety properties are exactly the closed sets and that liveness properties are exactly the dense sets of the Cantor topology [1]. Moreover, X is live in S iff X is dense in S. Every set in a topological space can be obtained as the intersection of a closed and a dense set, therefore every property is the intersection of a safety and a liveness property [1].

3.3 The safety-progress classification

As an alternative to the safety-liveness classification, Manna and Pnueli [22] propose the safety-progress classification. Let $X \subseteq \Sigma^\omega$.

- X is a safety property iff there exists a finitary property Q such that for each $x \in X$, all finite prefixes of x are in Q,
- X is a *guarantee property* iff there exists a finitary property Q such that for each $x \in X$, there exists a finite prefix of x that is in Q,
- X is an *obligation property* if X is expressible as a positive boolean combination of safety and guarantee properties,
- X is a *recurrence property* if there is a finitary property Q such that for each $x \in X$, there are infinitely many prefixes of x in Q,
- X is a *persistence property* if there is a finitary property Q such that for each $x \in X$, all but finitely many finite prefixes of x are in Q,
- X is a *reactivity property* if X is expressible as a positive boolean combination of recurrence and persistence properties.

Simple examples are: guarantee: $\mathsf{F}\, a$, obligation: $\mathsf{F}\, a \rightarrow \mathsf{F}\, b$, recurrence: $\mathsf{G}\,\mathsf{F}\, a$, persistence: $\mathsf{F}\,\mathsf{G}\, a$, and reactivity: $\mathsf{G}\,\mathsf{F}\, a \rightarrow \mathsf{G}\,\mathsf{F}\, b$. It can be shown [22] that safety, guarantee, recurrence, persistence are exactly closed, open, G_δ, and F_σ sets, respectively. Obligation is exactly $G_\delta \cap F_\sigma$ and reactivity is exactly $F_{\sigma\delta} \cap G_{\delta\sigma}$. Furthermore, each ω-regular property and hence each property expressible in LTL+past is a reactivity property [26, 22]. The classes of the safety-progress classification also have natural characterisations in temporal-logical and automata-theoretical terms [22, 9].

3.4 Fairness

Fairness is defined with respect to a given system (M, s) or, more general, with respect to a safety property S. (Note that $M(s)$ is a safety property.) It has been pointed out by Apt, Francez, and Katz [5] and by Lamport [18] that a fairness property for S should be live (i.e., dense) in S. This requirement alone, however, does not rule out some properties that are intuitively not fairness properties, and it implies that fairness is not closed under (finite) intersection [28]. We propose elsewhere [28] to call a property X a *fairness property for* S iff X is co-meager in S, that is a co-meager set in the topology relative to S. The following statements are equivalent with X being a fairness property for $M(s)$:

- X contains a dense G_δ set (relative to $M(s)$), i.e., a recurrence property that is live in $M(s)$,
- Ego has a winning strategy in the game $G(\overline{\mathsf{AE}}, X, M, s)$.

The first statement intuitively says that fairness requires, possibly under some condition, that some live finitary property is satisfied infinitely often. The latter statement is a classical result by Banach and Mazur. That game is also called *Banach-Mazur game*. We can view Ego here as a scheduler that wants to guarantee that all paths are fair.

It follows that:

- each fairness property for S is dense relative to S,
- fairness for S is closed under countable intersection, and
- fairness for S includes some basic intuitive fairness properties, viz. all recurrence properties that are live in S.

It can be shown [28] that this is, in a strong sense, the most liberal definition of fairness that has all three properties above. Furthermore, most fairness notions from the literature fall into this class, i.e., the usual *fairness notions* such as *strong fairness* map each system (safety property) S to a fairness property for S.

For examples of fairness properties, consider the system (M, a) described in Sect. 2. The property $\mathsf{G}\,\mathsf{F}\, b$ is a fairness property for $M(a)$. Another fairness property is $\mathsf{G}\,\mathsf{F}\, a \rightarrow \mathsf{G}\,\mathsf{F}(a \wedge \mathsf{X}\, b)$. These formulas do not represent fairness *notions*, as for some systems, the set of paths satisfying them may not be large. Note that $\mathsf{F}\,\mathsf{G}\, b$ is live but not a fairness property in $M(a)$.

We say that a system (M, s) is *fairly correct* with respect to a linear time specification X if X is large in (M, s). Equivalently, (M, s) is fairly correct wrt X if there exists a fairness assumption F for $M(s)$ such that $F \cap M(s) \subseteq X$.

4 T- versus P-largeness

In this section, we compare topological and probabilistic largeness and prove that they coincide for ω-regular properties in finite-state systems with bounded Borel measures.

4.1 Probabilistic largeness

Given a system (M, s_0), a probability measure μ on $\Omega = M(s_0)$ over the family of Borel sets of the Cantor topology relative to Ω is called a *Borel measure* over Ω. A Borel measure μ is a *Markov measure* when $\mu(\alpha s s'\uparrow \mid \alpha s\uparrow) = \mu(\beta s s'\uparrow \mid \beta s\uparrow)$ for all $\alpha, \beta \in \Sigma^*$ and $s, s' \in \Sigma$ such that $\alpha s s'$ and $\beta s s'$ are path fragments of (M, s_0). A Borel measure μ is *positive* if $\mu(\alpha\uparrow) > 0$ for each path fragment α of (M, s_0), μ is said to be *bounded* if there exists a $c > 0$ such that $\mu(\alpha s\uparrow \mid \alpha\uparrow) > c$ for each path fragment αs of (M, s_0). A Borel set $X \subseteq M(s_0)$ is *μ-large* (or *probabilistically large* or *P-large* when μ is understood from the context) if $\mu(X) = 1$.

4.2 Similarities

Topological and probabilistic largeness are very similar notions. Oxtoby's classic book [23] is devoted to study this similarity. The following observations, taken from there, are true for both T-largeness and P-largeness and confirm our intuition of largeness.

- If a set is large, its complement is not. (Note that this is not true for density.) Call a set *small* if its complement is large.
- Largeness is closed under superset and countable intersection (, i.e., the family of large sets is a *σ-filter*).
- If A is large and B is not small, then $A \cap B$ is not small.
- Every large set is nonempty. Since we restrict to positive Borel measures, every large set is also dense.
- Every countable set is small, but there are uncountable sets that are small.

Furthermore, there is also a strong duality between the two notions [23, Ch.19].

4.3 Separation

Although similar, the two notions do not coincide in general: there are sets that are T-large but not P-large as well as sets where it is the other way around.

Consider an (unrestricted asymmetric) random walk on the integer line starting at 0 going right with probability $p \neq 1/2$ and going left with probability $1 - p$. The property X_1 = 'The walk returns to 0 infinitely often' has probability 0 but is T-large. (One easily displays a winning strategy for Ego in the Banach-Mazur game.) The complement of X_1 has probability 1 but is not T-large.

A similar set can be displayed in a finite system: Consider an initial state from which one can go to a state a with probability $p \neq 1/2$ and to a state b with probability $1 - p$. From a and b we always go back to the initial state. The set X_2 = 'The number of previous a's equals the number of previous b's infinitely often' has probability 0 but is clearly T-large. Note however that a winning strategy for Ego is unbounded, i.e., the length of the sequences Ego adds is unbounded because it has to be able to compensate for unbounded moves by Alter.

The following proposition says that, under mild assumptions, a set can always be found that is large in one sense but small in the other.

Proposition 1 *Let (M, s_0) be a finite system such that every path $x = s_0, \ldots$ has infinitely many* choices*, i.e., positions i such that s_i has more than one R-successor in M. Let μ be a positive Markov measure on $M(s_0)$. Then $M(s_0)$ can be partitioned into a T-large and a μ-large set.*

4.4 Coincidence

We now prove that for bounded Borel measures on finite systems and ω-regular properties, the two notions of largeness coincide. Note that the property X_2 described in the counterexample in Sect. 4.3. is not accepted by any finite-state automaton.

Proposition 2 *Let (M, s_0) be a finite system, μ a bounded Borel measure on $M(s_0)$, and X an ω-regular property. If X is T-large, then it is also μ-large.*

Proof: If X is T-large, Ego has a winning strategy for X in the Banach-Mazur game. Berwanger, Grädel, and Kreutzer [8] have shown that Ego has then also a *positional* winning strategy, i.e., a strategy f such that $f(\alpha s) = f(\beta s)$ for all $\alpha, \beta \in \Sigma^*$ and $s \in \Sigma$. Since there are only finitely many states, the positional strategy is also *bounded*, i.e., there exists a k such that $|f(\alpha)| < k$ for all α. It follows that, in each path fragment, playing f has a positive probability bounded away from zero and therefore the property $\{x \mid x$ has infinitely many positions where the extension is according to $f\}$ has probability 1 (by application of Borel-Cantelli Lemmas). Hence $\{x \mid x$ is the result of some play of the Banach-Mazur game where Ego plays $f\}$ has probability 1. Because f is winning for X, X is μ-large. □

The converse also holds.

Proposition 3 *Let (M, s_0) be a finite system, μ a bounded Borel measure on $M(s_0)$, and X an ω-regular property. If X is μ-large, then it is also T-large.*

Proof: If X is not T-large, then Alter has, due to determinacy, a winning strategy f in the Banach-Mazur game. Let α_0 be the first move of Alter in that strategy. We have $\mu(\alpha_0\uparrow) > 0$. Since f is a winning strategy for Alter, f is also a winning strategy for Ego in the Banach-Mazur game that starts in α_0 and in which Ego plays for $\Omega \setminus X$. From Prop. 2 now follows that $\mu(\Omega \setminus X \mid \alpha_0\uparrow) = 1$. Hence we conclude $\mu(X) < 1$. □

We obtain:

Theorem 1 *T-largeness and P-largeness of ω-regular properties coincide for bounded Borel measures on finite systems.*

In particular they also coincide for properties expressible in LTL(+past).

4.5 Complete fairness

To prove T-largeness of a property X in a system (M, s), it suffices to show $F \cap M(s) \subseteq X$ for some fairness property F for (M, s). We now ask whether there is a *complete* fairness notion to prove T-largeness.

Definition 1 *Let (M, s) be a system and $\mathscr{F}$ a family of linear-time properties. A fairness property F for (M, s) is* $\mathscr{F}$-complete *with respect to (M, s) if for each property $X \in \mathscr{F}$ that is T-large in (M, s), we have $F \cap M(s) \subseteq X$.*

Note that if F is complete for a family $\mathscr{F}$ then it is also complete for every subfamily of $\mathscr{F}$. Lichtenstein *et al.* [21] introduced *α-fairness* and showed that it is complete for showing P-largeness of ω-regular properties of finite-state systems. Zuck, Pnueli, and Kesten [29] point out that *state fairness* is complete for showing P-largeness of properties that are expressible in LTL without the next- and until-operators.

We show now that completeness w.r.t. ω-regular and LTL expressible properties can be characterised through *word fairness*. Say that a word $\beta \in \Sigma^+$ is *enabled* in a state s of a M if $s\beta$ is a path fragment of (M, s) and say that β is *taken* in a position i of a path $x = s_0, \ldots$ if there exists a position j such that $\beta = s_i, s_{i+1}, \ldots, s_j$. Call a path x of a system (M, s) *fair* w.r.t. β if β is enabled only finitely many times in x or β is taken infinitely many times in x; x is *word fair* if it is fair w.r.t. all $\beta \in \Sigma^+$.

Proposition 4 *A fairness property F is $\mathscr{F}$-complete w.r.t. a system (M, s) if and only if each run in $F \cap M(s)$ is word fair, where $\mathscr{F}$ denotes the family of ω-regular or LTL expressible properties. In particular, word fairness is complete for ω-regular properties.*

Proof: ($\Leftarrow$) If X is an ω-regular property that is large in $M(s)$, then it follows as in the proof of Prop. 2 that Ego has a positional winning strategy f. However, that means that every word-fair path $x \in M(s)$ is the result of some play where Ego plays f. Therefore x is in X and hence word fairness is complete. ($\Rightarrow$) Follows from the fact that word fairness w.r.t. a particular word can be expressed in LTL. □

Note that Prop. 4 does not assume the system to be finite. However, to use Prop. 4 for P-largeness we need to restrict to finite systems.

Clearly, there is a complete fairness property for every countable family $\mathscr{F}$ that contains at least one fairness property. It is obtained by intersecting all fairness properties for (M, s) in $\mathscr{F}$. However, that intersection is not necessarily a member of $\mathscr{F}$. (Note that in Def. 1, F is not required to be a member of $\mathscr{F}$.) It can be shown that this is in fact the case for ω-regular and LTL-expressible properties:

Proposition 5 *There are finite systems (M, s) such that there is no ω-regular fairness property that is complete for the family of LTL expressible properties w.r.t. (M, s).*

For the proof, we consider a completely connected graph and show that Ego has no positional strategy for word fairness. Prop. 5 shows that largeness of an LTL formula ϕ can in general not be checked by expressing a complete fairness property as LTL formula ψ and then checking the formula $(\psi \rightarrow \phi)$.

5 Checking largeness

Berwanger *et al.* [8] showed that checking largeness of an LTL specification for a finite system is decidable by showing that largeness of an LTL formula can be expressed as satisfaction of a CTL* formula. Their translation however is of non-elementary complexity and hence not suitable for complexity analysis. Pistore and Vardi [24] provide an efficient translation into the logic EGCTL* of Kupferman [15], whose model checking complexity is double exponential time [15]. Kupferman and Vardi [17] show that model checking the Pistore-Vardi logic without $\overline{\mathsf{AE}}$ and $\overline{\mathsf{EA}}$ is EXPSPACE-complete leaving the complexity of checking largeness open.

From Thm. 1, we can immediately conclude that checking largeness is PSPACE-complete for LTL or Büchi automata specifications.

5.1 Büchi automata and LTL specifications

Vardi [27] has shown that checking P-largeness of an ω-regular property given by a Büchi automaton is PSPACE-complete in the size of the automaton. Hence we obtain:

Theorem 2 *The problem of checking T-largeness of a Büchi automata specification against a finite system is PSPACE-complete in the size of the automaton.*

Courcoubetis and Yannakakis [11] have shown that checking P-largeness of an LTL formula is PSPACE-complete in the size of the formula. Therefore:

Theorem 3 *The problem of checking T-largeness of an LTL formula in a finite system is PSPACE-complete in the size of the formula.*

Note that the corresponding algorithms for Thms. 2 and 3 use time linear in the size of the temporal structure.

5.2 Reactivity formulas and Streett constraints

It is interesting to provide an independent algorithm for LTL(+past) formulas which, although less efficient in general, can be efficiently applied to an important class of formulas.

A *reactivity formula* [22] is a formula of the form

$$\phi = \bigwedge_{i=1}^{n} (\mathsf{G}\,\mathsf{F}\,h_i \vee \mathsf{F}\,\mathsf{G}\,g_i)$$

where h_i and g_i are *past formulas*, that is, history formulas that do not contain the future operators X, U, and their derivatives. In case all p_i and q_i are state formulas we call ϕ a *state reactivity formula*. A formula of the form $(\mathsf{G}\,\mathsf{F}\,p \vee \mathsf{F}\,\mathsf{G}\,q)$ is called a *Streett constraint* [3].

Consider the following translation of a reactivity formula into a CTL+past formula:

- $[\![\mathsf{F}\,\mathsf{G}\,h]\!] = \mathsf{AG}\,\mathsf{EF}\,\mathsf{AG}\,h$
- $[\![\mathsf{G}\,\mathsf{F}\,h]\!] = \mathsf{AG}\,\mathsf{EF}\,h$
- $[\![\mathsf{G}\,\mathsf{F}\,h \vee \mathsf{F}\,\mathsf{G}\,g]\!] = \mathsf{AG}(\neg[\![\mathsf{F}\,\mathsf{G}\,\neg h]\!] \vee \neg[\![\mathsf{G}\,\mathsf{F}\,\neg g]\!])$
- $[\![\phi \wedge \psi]\!] = [\![\phi]\!] \wedge [\![\psi]\!]$

Proposition 6 *For every system (M, s), we have that a reactivity formula ϕ is large in (M, s) if and only if $M, s \models [\![\phi]\!]$.*

For the first two clauses, the CTL+past formula essentially describes the winning strategy for Ego. (They can also be seen as applications of Proposition 10.3 and 10.4 below, respectively.) For the last clause, we observe that the intersection of two sets is large if and only if both sets

are large. In the third clause, the union of two sets could be large even if neither of them is. The proof instead uses determinacy. We know that all sets involved are determinate [8]. The translated formula says that in every state, Ego does not have a winning strategy for the negation of one of the disjuncts. By determinacy, this happens if and only if Alter has a winning strategy for one of the disjuncts. But this means that after the first move of Alter, Ego (who has now the first move) has a winning strategy for one of the disjuncts.

To check the largeness of a reactivity formula we check the satisfaction of the corresponding CTL+past formula. The model checking problem for CTL+past is PSPACE-complete [19, 25].

Reactivity formulas encompass many interesting formulas, e.g. safety formulas such as $\mathsf{G}\, p$ or $\mathsf{G}(p \rightarrow \mathsf{F}^{-1} q)$, persistence formulas such as $\mathsf{F}\,\mathsf{G}\, p$ and recurrence formulas such as $\mathsf{G}\,\mathsf{F}\, p$, also forms of response such as $\mathsf{F}\,\mathsf{G}\, p \rightarrow \mathsf{G}\,\mathsf{F}\, q$ and $\mathsf{G}\,\mathsf{F}\, p \rightarrow \mathsf{G}\,\mathsf{F}\, q$.

In fact, every LTL formula can be expressed as a reactivity formula [22]. However, the translation can produce an exponential blowup. Therefore we do not obtain the optimal upper bound of Thm. 3 for the above procedure applied to general LTL formulas.

The translation is also interesting in that it shows that largeness of an LTL+past formula can be expressed in CTL+past, a temporal logic strictly less expressive than CTL* [16], thus strengthening the result of Berwanger *et al.* [8], who showed that largeness of an LTL+past formula can be translated into satisfaction of a CTL* formula.

Checking whether a state reactivity formula is dense in a structure requires time quadratic in the size of the formula [12]. On the other hand, Alur and Henzinger [3] claim that checking whether a state reactivity is large requires linear time. We provide an alternative proof of their result.

For a state reactivity formula, the translation produces a CTL formula without past, whose model checking problem is linear. Thus we have:

Proposition 7 *The problem of checking T-largeness of a state reactivity formula in a finite system can be solved in time linear in the size of the formula.*

In the light of Thm. 1, also checking P-largeness of a state reactivity formula can be done in linear time. We are not aware of any analogous result in the literature.

6 Branching-time largeness

We now study the problem of expressing largeness in a branching time context. We consider logics that are obtained from CTL* and CTL by replacing the universal and existential path quantifiers by path quantifiers expressing largeness and non-smallness respectively.

6.1 The Lehmann-Shelah logic

First consider the logic *T-large CTL**, which is defined as CTL* but where instead of the path quantifiers A and E we have the path quantifiers $\overline{\mathsf{AE}}$ and $\overline{\mathsf{EA}}$ with their meaning defined above. This is essentially the logic studied by Ben-Eliyahu and Magidor [7]. By *P-large CTL** we refer to the logic that is defined as CTL* but where instead of the path quantifiers A and E we have the path quantifiers $\bigtriangledown$ and $\bigtriangleup$, where $\bigtriangledown.\phi$ means ϕ is satisfied with probability 1 and $\bigtriangleup.\phi$ means ϕ is satisfied with probability > 0. This is essentially the logic studied by Lehmann and Shelah [20]. Call τ the bijection between T-large CTL* and P-large CTL* where $\overline{\mathsf{AE}}$ is replaced by $\bigtriangledown$ and $\overline{\mathsf{EA}}$ by $\bigtriangleup$. Using structural induction and Thm. 1 it is easy to prove that:

Theorem 4 *For any T-large-CTL* formula ϕ and finite probabilistic system (M, s), we have $M, s \models \phi$ if and only if $M, s \models \tau(\phi)$.*

Lehmann and Shelah [20] provide sound and complete axiomatic systems for P-large CTL* and different classes of probabilistic systems. Ben-Eliyahu and Magidor [7] show that the axiomatic system for finite systems is sound and complete for T-large CTL* and systems of arbitrary size. This is now a corollary of Lehmann and Shelah's work, Thm. 4, and the finite model property of T-large CTL*, where the latter is shown by Ben-Eliyahu and Magidor [7].

It is straightforward to adapt the model checking algorithm for CTL* [10] to our case, thus obtaining:

Theorem 5 *The model-checking problem for T-large CTL* and P-large CTL* is PSPACE-complete in the size of the formula.*

The procedure is precisely the same as in [10]. For every subformula of the form $\overline{\mathsf{AE}}\,\phi$, where ϕ is a formula without quantifiers, we label the states of the system with a new proposition p, depending on whether $\overline{\mathsf{AE}}\,\phi$ is true or not. This requires polynomial space, as it amounts to check largeness of ϕ for every state. We substitute p for $\overline{\mathsf{AE}}\,\phi$ and we repeat the procedure until there are no more nested quantifiers. Hardness follows from the fact that checking largeness of LTL is PSPACE-hard.

The logic P-large CTL* can be also seen as a restricted version of more expressive probabilistic logics, such as pCTL* [6], which can express all probabilities between 0 and 1. The model checking of pCTL* is also in PSPACE.

One can consider a logic that combines the universal/existential and largeness/non-smallness quantifiers. Again, this does not change the model checking complexity. One could also consider a version of CTL* containing all eight quantifiers of the Pistore-Vardi logics. In the light of the EXPSPACE-completeness of the model checking of the Pistore-Vardi logic [17], the model checking problem for this version would also be EXPSPACE-complete.

6.2 Large CTL

The logic *T-Large CTL* is obtained by restricting T-large CTL* just like CTL is obtained as restriction of CTL*. We now prove that model checking T-large CTL can be done by a simple algorithm in linear time. To this end, we use the following translation into standard CTL[3]:

$$\begin{aligned}
[\![a]\!] &= a \\
[\![\neg p]\!] &= \neg [\![p]\!] \\
[\![p_1 \wedge p_2]\!] &= [\![p_1]\!] \wedge [\![p_2]\!] \\
[\![\overline{\mathsf{AE}}\,\mathsf{X}\, p]\!] &= \mathsf{A}\,\mathsf{X} [\![p]\!] \\
[\![\overline{\mathsf{EA}}\,\mathsf{X}\, p]\!] &= \mathsf{E}\,\mathsf{X} [\![p]\!] \\
[\![\overline{\mathsf{EA}}(p_1 \,\mathsf{U}\, p_2)]\!] &= \mathsf{E}\,([\![p_1]\!]\,\mathsf{U}\,[\![p_2]\!]) \\
[\![\overline{\mathsf{AE}}(p_1 \,\mathsf{U}\, p_2)]\!] &= \\
\mathsf{A}\,([\![p_1]\!]\,\mathsf{W}\,[\![p_2]\!]) &\wedge \neg\, \mathsf{E}\,([\![p_1]\!]\,\mathsf{U}\,\mathsf{AG}\,\neg [\![p_2]\!])
\end{aligned}$$

Proposition 8 *For a T-large CTL formula p we have*

$$M, s \models p \Leftrightarrow M, s \models [\![p]\!] \,.$$

The translation is homomorphic, except for the formula $p = p_1 \,\mathsf{U}\, p_2$. For this case suppose that the translation of p is true. We prove that there is a winning strategy for p. It is easy to see that such strategy is winning in only one move. Indeed if Alter has already produced a run which satisfies p, Ego does nothing. If Alter has produced a run in which p_1 is always true, Ego just needs to produce a continuation where p_1 is always true until p_2 is true.

The first part of the translation makes sure that Alter cannot produce a path that violates p. The second part makes sure that if Alter has not yet validated p, Ego can always get to a place where p_2 is true. In order for p to be validated by such a play, Ego must not have touched a state where p_1 and p_2 are both false. This is again ensured by the first part.

Conversely, suppose the translated formula is false. If the first part is false, then Alter can produce a path that violates p. If the second part is false, then Alter can force the play to a place where Ego can never validate p, as he can never make p_2 true.

In all the other cases it is easy to verify that the homomorphic translation is enough. For instance, since for every state there are only finitely many 'next' states, we have that a large set of runs satisfies $\mathsf{X}\, p$ if and only if all runs satisfy $\mathsf{X}\, p$.

Theorem 6 *The model checking problem for T-large CTL can be solved in linear time.*

[3] A similar translation can be found in [4], where it is used for model checking CTL under *transition fairness*.

The translation produces an exponential blow up, but the model checking algorithm can by-pass this, by a form of dynamic programming. The algorithm proceeds as for CTL, labelling each state with the subformulas that are satisfied in that state. In checking the subformulas, every time we check for a subformula p that appears as second formula within and until operator, we also check for the formula $\mathsf{AG}\,\neg p$. When checking $\mathsf{A}(p_1 \,\mathsf{U}\, p_2)$, we have to run two procedures, one for each part of the translation. This at most doubles the time of the checking, but does not change the asymptotic complexity.

Note that, by Thm. 4, this algorithm can also be used for checking P-large CTL formulas on a finite Markov chain. This provides an alternative to the known linear time algorithm for P-large CTL [2]. Other polynomial model checking algorithms could be derived by viewing P-large CTL as a restricted version of pCTL [6] and PCTL [14].

7 Pistore-Vardi revisited

As indicated in Sect. 2.3, besides $\overline{\mathsf{AE}}$, also the other path quantifiers of the Pistore-Vardi logic could be considered as relaxations of correctness. In this section, we observe that also those other path quantifiers have a perfect topological meaning. We use this to derive some basic properties of the path quantifiers.

Proposition and Definition 9 *Let X be a linear-time property and (M, s) a system.*

- $M, s \models \mathsf{A}$ *.X iff X contains $M(s)$. We say that X is* satisfied *in (M, s) or that X holds in* all paths.
- $M, s \models \mathsf{AEA}$ *.X iff X contains a dense open set in $M(s)$ (or equivalently, it is the complement of a nowhere dense set in $M(s)$). We say that X is* observably large *in $M(s)$.*
- $M, s \models \overline{\mathsf{AE}}$ *.X iff X is co-meager in $M(s)$. We say that X is* large *or* fairly satisfied *in (M, s) or that X holds for* almost all *paths.*
- $M, s \models \mathsf{AE}$ *.X iff X is dense in $M(s)$. We say that X is* live *or* everywhere satisfiable *in (M, s) or that X holds* everywhere for some *path.*
- $M, s \models \mathsf{EA}$ *.X iff X contains a nonempty open subset of $M(s)$ (or equivalently its complement is not dense in $M(s)$). We say that X is* somewhere satisfied *in (M, s) or that X holds* somewhere for all *paths.*
- $M, s \models \overline{\mathsf{EA}}$ *.X iff X is co-meager in some open subset of $M(s)$. We say that X is* somewhere large *in (M, s) or* somewhere fairly satisfied.

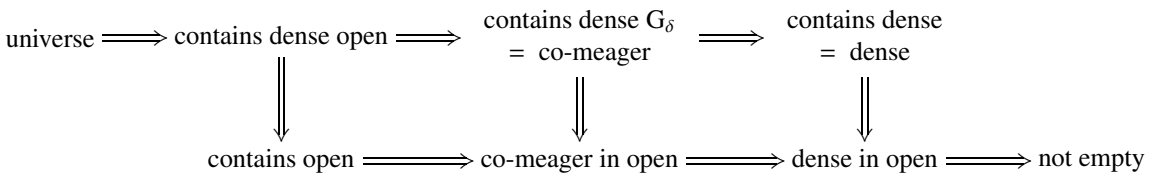


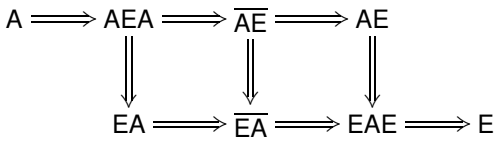
Figure 1. A schema of implications

- $M, s \models \mathsf{EAE}.X$ *iff* X *is dense in some open subset of* $M(s)$. *We say that* X *is* somewhere dense *or* somewhere live *in* (M, s).

- $M, s \models \mathsf{E}.X$ *iff* X *is a nonempty subset of* $M(s)$. *We say* X *is* satisfiable *in* (M, s) *or* X *holds for* some *path.*

Note that density is not a good notion of largeness because, for instance, there are dense sets whose complement is also dense. Nevertheless density is interesting because it formalises that at least the safety property implied by the specification is not violated, i.e., a property is dense in (M, s) iff $\overline{X}$ is satisfied in (M, s), where $\overline{X}$ denotes the smallest safety property that contains X.

Note that all the above classes of properties are upward closed, that is, $M, s \models \kappa.X$ and $X \subseteq Y$ implies $M, s \models \kappa.Y$. The following implications, taken from [24, 8],



can be seen topologically (see Fig. 1). All the implications are trivial there.

Furthermore we observe that for LTL formulas ϕ, the latter four path quantifiers are duals of the former four, that is

$$M, s \models \mathsf{A}.\phi \text{ iff } M, s \not\models \mathsf{E}.\neg\phi$$
$$M, s \models \mathsf{AEA}.\phi \text{ iff } M, s \not\models \mathsf{EAE}.\neg\phi$$
$$M, s \models \overline{\mathsf{AE}}.\phi \text{ iff } M, s \not\models \overline{\mathsf{EA}}.\neg\phi$$
$$M, s \models \mathsf{AE}.\phi \text{ iff } M, s \not\models \mathsf{EA}.\neg\phi$$

Only the proof for $\overline{\mathsf{AE}}$ is not straightforward. There, we must use the fact that ϕ is *determinate* in the Banach-Mazur game.

Recall that checking the quantifiers A and $\overline{\mathsf{AE}}$ and their duals is PSPACE-complete. Checking AE and its dual is EXPSPACE-complete [17]. The complexity of checking AEA and its dual remains open.

Finally, the topological interpretation allows us to prove that in particular situations, different classes collapse.

Proposition 10 *Consider a property* $X \subseteq \Sigma^\omega$. *All the following statements are true relative to any fixed system* (M, s)*:*

1. *If* X *is a safety property, then* X *is satisfied iff* X *is live and* X *is somewhere satisfied iff it is somewhere live.*

2. *If* X *is a guarantee property then* X *is observably large iff* X *is live and* X *is somewhere satisfied iff it is satisfiable.*

3. *If* X *is a persistence property then* X *is observably large iff it is fairly satisfied and* X *is somewhere satisfied iff somewhere fairly satisfied.*

4. *If* X *is a recurrence property then* X *is fairly satisfied iff it is live in* M *and* X *is somewhere satisfied iff* X *is somewhere live.*

5. *If* X *is a obligation property then* X *is observably large iff it is live and* X *is somewhere satisfied iff it is somewhere live.*

8 Conclusions

We argued that topological largeness is an interesting notion as it can serve as a natural relaxation of correctness of a system: It has similar properties as probabilistic largeness that confirm our intuitive understanding of largeness, it formalises the intuitive notion of fairness. It is pleasing that topological largeness has various independent characterisations in terms of game-theory, language-theory, automata-theory, and temporal logic.

By showing coincidence of topological and probabilistic largeness, we solved the model checking problem for topological largeness of LTL and ω-regular specifications. Coincidently, this settles the complexity of model checking of the full Pistore-Vardi logic [24] and the complexity of deciding Banach-Mazur games for ω-regular goals [8]. As a side effect,

1. we obtain new characterisations of probabilistic largeness in finite Markov chains, and

2. this shows that any ω-regular fairness property has probability 1 under randomised scheduling.

Checking largeness of a specification maybe useful whenever the specifications is satisfied only under some, possibly strong, fairness assumption and the fairness assumption is either unknown, expensive to specify, or impossible to specify in the temporal logic.

We have shown that the complexity of checking largeness is the same as the complexity of checking satisfaction for the most popular specification formalisms. We have explicitly mentioned only the complexity with respect to the formula, however, as for standard satisfaction algorithms, all algorithms described use time linear in the size of the system.

Our work could be generalised to a situation where fairness is not required for all choices. Some choices would be fair, and some would be completely nondeterministic. This leads us to a model analogous to the concurrent Markov chains of [27], with fair states substituted for probabilistic states. In terms of the Banach-Mazur game, this amounts to not giving Ego access to all transitions. Again we can use Theorem 1 and the results in [11] to get the complexity of model checking for these systems.

References

[1] B. Alpern and F. B. Schneider. Defining liveness. *Inf. Proc. Letters*, 21:181–185, Oct. 1985.

[2] R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking for probabilistic real-time systems (extended abstract). Proc. *ICALP*, *LNCS* 510, pp. 115–126. Springer, 1991.

[3] R. Alur and T. A. Henzinger. Local liveness for compositional modeling of fair reactive systems. In *Proc. CAV*, *LNCS* 939, pp. 166–179. Springer, 1995.

[4] B. Aminof, T. Ball, and O. Kupferman. Reasoning about systems with transition fairness. In *Proc. LPAR*, *LNCS* 3452, pp. 194–208. Springer, 2004.

[5] K. R. Apt, N. Francez, and S. Katz. Appraising fairness in languages for distributed programming. *Distr. Comput.*, 2:226–241, 1988.

[6] A. Aziz, V. Singhal, F. Balarin, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Proc. CAV*, *LNCS* 939, pp. 155–165. Springer, 1995.

[7] R. Ben-Eliyahu and M. Magidor. A temporal logic for proving properties of topologically general executions. *Inf. and Comp.*, 124(2):127–144, 1996.

[8] D. Berwanger, E. Grädel, and S. Kreutzer. Once upon a time in the west - determinacy, definability, and complexity of path games. In *Proc. LPAR*, *LNAI* 2850, pp. 229–243, 2003.

[9] E. Y. Chang, Z. Manna, and A. Pnueli. Characterization of temporal property classes. In *Proc. ICALP*, *LNCS* 623, pp. 474–486. Springer, 1992.

[10] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, 1986.

[11] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

[12] E. A. Emerson and C.-L. Lei. Modalities for model checking: Branching time strikes back. In *Proc. POPL*, pp. 84–96, 1985.

[13] F. E. Fich and E. Ruppert. Hundreds of impossibility results for distributed computing. *Distr. Comput.*, 16(2-3):121–163, 2003.

[14] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput.*, 6(5):512–535, 1994.

[15] O. Kupferman. Augmenting branching temporal logics with existential quantification over atomic propositions. *J. Log. Comput.*, 9(2):135–147, 1999.

[16] O. Kupferman and A. Pnueli. Once and for all. In *Proc. LICS*, pp. 25–35. IEEE Computer Society, 1995.

[17] O. Kupferman and M. Y. Vardi. Memoryful branching-time logic. This volume.

[18] L. Lamport. Fairness and hyperfairness. *Distr. Comput.*, 13(4):239–245, 2000.

[19] F. Laroussinie and P. Schnoebelen. Specification in CTL+past for verification in CTL. *Inf. and Comput.*, 156(1-2):236–263, 2000.

[20] D. Lehmann and S. Shelah. Reasoning with time and chance. *Inf. and Contr.*, 53(3):165–198, 1982.

[21] O. Lichtenstein, A. Pnueli, and L. D. Zuck. The glory of the past. In *Proc. of Logic of Programs*, *LNCS* 193, pp. 196–218. Springer, 1985.

[22] Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *Proc. PODC*, pp. 377–408. ACM, 1990.

[23] J. C. Oxtoby. *Measure and Category. A Survey of the Analogies between Topological and Measure Spaces*. Springer-Verlag, 1971.

[24] M. Pistore and M. Y. Vardi. The planning spectrum - one, two, three, infinity. In *Proc. LICS*, pp. 234–243, 2003.

[25] P. Schnoebelen. The complexity of temporal logic model checking. In *Selected Papers from the 4th Workshop on Advances in Modal Logics (AiML'02)*, pp. 393–436, 2003.

[26] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics. Elsevier, 1990.

[27] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. FOCS*, pp. 327–338, 1985.

[28] H. Völzer, D. Varacca, and E. Kindler. Defining fairness. In *Proc. CONCUR*, *LNCS* 3653, pp. 458–472. Springer, 2005.

[29] L. D. Zuck, A. Pnueli, and Y. Kesten. Automatic verification of probabilistic free choice. In *Proc. VMCAI*, *LNCS* 2294, pp. 208–224. Springer, 2002.