Report

Model checking almost all paths can be less expensive than checking all paths

Author(s):
Schmalz, Matthias; Völzer, Hagen; Varacca, Daniele

Publication Date:
2011

Permanent Link:

https://doi.org/10.3929/ethz-a-006810140 →

Model Checking Almost All Paths Can Be Less Expensive than Checking All Paths

Matthias Schmalz[1], Hagen Völzer[2], and Daniele Varacca[3] *

[1] ETH Zürich, Switzerland
[2] IBM Zurich Research Laboratory, Switzerland
[3] PPS - CNRS & Univ. Paris 7, France

Abstract. We compare the complexities of the following two model checking problems: checking whether a linear-time formula is satisfied by all paths (which we call *universal* model checking) and checking whether a formula is satisfied by almost all paths (which we call *fair* model checking here). For many interesting classes of linear-time formulas, both problems have the same complexity: for instance, they are PSPACE-complete for LTL.
In this paper, we show that fair model checking can have lower complexity than universal model checking, viz., we prove that fair model checking for $L(\mathrm{F}^\infty)$ can be done in time linear in the size of the formula and of the system, while it is known that universal model checking for $L(\mathrm{F}^\infty)$ is co-NP-complete. $L(\mathrm{F}^\infty)$ denotes the class of LTL formulas in which F^∞ is the only temporal operator. We also present other new results on the complexity of fair and universal model checking. In particular, we prove that fair model checking for RLTL is co-NP-complete.

1 Introduction

A reactive system satisfies a specification expressed by a formula of linear-time temporal logic if *all* its executions satisfy the formula. In this case, we say that a system is *universally correct*, and the problem of verifying universal correctness is called *universal model checking*.

Sometimes a system does not satisfy a specification, but only because of a "small" set of executions that do not satisfy the formula. From a measure-theoretic point of view, "small" means having probability 0. From a topological point of view, it means being a *meager* set. The topological point of view corresponds to the notion of *fairness* [14], i.e., a set of executions Y of a system is meager if and only if there exists some fairness assumption F for the system such that each execution in Y is unfair w. r. t. F.

Varacca and Völzer [11] have shown that, for LTL formulas and finite-state systems, the two notions of smallness coincide. More importantly, they coincide

* [1]Matthias.Schmalz@inf.ethz.ch, [2]hvo@zurich.ibm.com, [3]varacca@pps.jussieu.fr; most of the work was done while the first two authors were affiliated with the University of Lübeck, Germany.

independently of the probability measure chosen (provided it belongs to a very general class of measures).

If the set of executions that do not satisfy the specification is small, we say that the system is *almost correct* or *fairly correct*. The problem of verifying fair correctness is called *fair model checking* in this paper.[4] As indicated above, fair model checking is — for finite systems and LTL specifications — equivalent to *qualitative probabilistic model checking* (i.e., checking a specification for probability 1) (cf. [11]). Fair model checking is an interesting alternative to universal model checking even for non-probabilistic systems that are desired to be universally correct for the following reasons:

- The difference between the two notions of correctness is small; most errors (i.e. violations of the specification) found by universal checking are also found by fair checking. In particular, both notions of correctness coincide for *safety* properties (cf. [11]).
- In fair model checking, there is no need to specify any fairness assumption on the system. (Additional fairness assumptions do not change fair correctness [11].)

It is known that universal and fair model checking for LTL have the same complexity: both are PSPACE-complete and can be solved in time linear in the system and exponential in the formula [9, 6, 12, 3]. In this paper, we compare the complexities of universal and fair model checking for subclasses of LTL. Studying subclasses helps to understand the scope of the PSPACE-completeness results and also helps to develop optimised algorithms for frequently used formulas.

It is known that also for some sub- and superclasses of LTL, universal and fair model checking have the same complexity, e.g. LTL+past [9, 6, 3], Büchi automata [10, 13, 12, 3] and Street constraints [1, 11]. We show that this remains true for some additional subclasses. In particular, fair and universal model checking for $L(\mathrm{F})$ (also known as RLTL: the class of LTL formulas built using only the temporal operator F) are both co-NP-complete.

However, as the main result of the paper, we show that fair (and hence qualitative probabilistic) model checking can be easier than universal model checking. We prove that fair model checking for $L(\mathrm{F}^\infty)$ (LTL restricted to F^∞, where F^∞ is short for $\mathrm{G\,F}$) can be done in time linear in the size of the formula (and linear in the size of the system), whereas universal model checking for $L(\mathrm{F}^\infty)$ is co-NP-complete.

To this end, we define and characterise an interesting subclass of $L(\mathrm{F}^\infty)$, called *Muller formulas*, which already separates the two model checking problems with respect to their complexity. The satisfaction of a Muller formula in an execution depends only on the set of states which are visited infinitely often in that execution. Finally, we clarify the scope of our results by looking at some simple subclasses of RLTL.

[4] Note that in this paper fair model checking is *not* the problem of checking whether a system is correct under some fixed fairness assumption. Instead, it is the problem of checking whether there *exists* some fairness assumption for a system such that the system is correct under this fairness assumption.

2 Preliminaries

2.1 Systems and temporal properties

Let Q be a finite set of *states*. The sets Q^*, Q^+ and Q^ω contain all finite, non-empty finite, and infinite sequences over Q, respectively. Finite sequences are called *path fragments (over Q)* and denoted by α, β, and infinite ones are called *paths (over Q)* and denoted by x, y. The i-th element of a path (or path fragment) x is denoted x_i. We have $x = x_0 x_1 \ldots$ A set $Y \subseteq Q^\omega$ is called a *(linear-time temporal) property (over Q)* or a *specification*. If Q is clear from the context, we write Y^c for the complement of Y in Q^ω.

Throughout the entire paper, we fix a nonempty set AP of *atomic propositions*. A *system* $\Sigma = (Q, q_0, \to, v)$ consists of a finite set of states $Q \subseteq AP$, an initial state $q_0 \in Q$, a *state relation* $\to \subseteq Q \times Q$, and a *valuation function* $v : Q \to 2^{AP}$ such that $q \in v(q)$, for each $q \in Q$. The technical assumption $Q \subseteq AP$ allows us later to use states as part of temporal formulas. We require that for each $p \in Q$ there be a $q \in Q$ such that $p \to q$. A *path of Σ* is a path x over Q such that $x_0 = q_0$ and $x_i \to x_{i+1}$ ($i \in \mathbb{N}$). Finite prefixes of paths of Σ are called *path fragments of Σ*.

A set $K \subseteq Q$ is a *strongly connected component of Σ* (s. c. c. for short) if it is a strongly connected component of the directed graph $(Q, \to)$. A *bottom strongly connected component of Σ (b. s. c. c.)* K is an s. c. c. with no outgoing edges, i.e., there is no edge $(p, q) \in \to$ such that $p \in K$ and $q \notin K$.

The *size* of a system $\Sigma = (Q, q_0, \to, v)$ is defined as $|\Sigma| := |Q| + |\to|$.

2.2 Temporal logic

In this paper, we consider several languages of linear-time temporal logic. The most expressive one is LTL+past [4], which is defined by the following syntax rules, where ξ ranges over atomic propositions and Φ over *path formulas*:

$$\Phi := \xi \mid \neg\Phi \mid \Phi \vee \Phi \mid \mathrm{X}\,\Phi \mid \Phi\,\mathrm{U}\,\Phi \mid \mathrm{X}^-\,\Phi \mid \Phi\,\mathrm{U}^-\,\Phi$$

Additional operators such as *true*, *false*, $\wedge$, $\Rightarrow$, F, G, etc. are defined as abbreviations as usual [4]. We will also make use of the operator F^∞, defined as abbreviation for $\mathrm{G\,F}$, and G^∞ the abbreviation for $\mathrm{F\,G}$. Non-boolean operators are called *temporal* operators. If Φ does not contain a temporal operator, it is called a *state formula*. By $L(op_1, \ldots, op_n)$ we denote the set of LTL+past formulas that contain only the temporal operators $op_1, \ldots, op_n$. $L(\mathrm{X}, \mathrm{U})$ is known as LTL, $L(\mathrm{F})$ as RLTL. Note that $L(\mathrm{F}) \subseteq L(\mathrm{X}, \mathrm{U})$ because F can be expressed by U. Likewise, formulas in $L(\mathrm{F})$ can also contain $\mathrm{G}, \mathrm{F}^\infty$ and G^∞.

Satisfaction $x \vDash \Phi$, $x, i \vDash \Phi$ is defined as usual [4]. By $Sat(\Phi)$ we denote the set of all paths of the underlying system that satisfy Φ. The *size* $|\Phi|$ of a formula Φ is given by the number of its temporal and boolean operators.

2.3 Universal and fair correctness

A system is *universally* correct *w. r. t. a specification* Y iff each path of the system belongs to Y. It is *universally* correct *w. r. t. a formula* Φ iff each path of the system satisfies Φ. Fair correctness can be defined equivalently in language-theoretic, game-theoretic, topological, or probability-theoretic terms [11]. In particular, the system underlying a finite-state Markov chain is *fairly correct* w. r. t. a specification given by a formula Φ if and only if $Sat(\Phi)$ has measure 1. This property is independent of the precise probabilities in the Markov chain, and fair correctness can in fact be defined without probability. We give the game-theoretic definition here because that will be the most useful in the sequel.

Let $\Sigma = (Q, q_0, \rightarrow, v)$ be a system and Y a property. The *Banach-Mazur game* $G(\Sigma, Y)$ is played by the two players Alter and Ego, and the state of a play is a path fragment of Σ. Alter moves first by choosing a path fragment α_0 of Σ. The players alternately move, and the player of the i-th move ($i \in \mathbb{N}$) extends the path fragment by a finite, nonempty sequence α_i, yielding the path fragment $\alpha_0 \ldots \alpha_i$ of Σ. The play goes on forever, converging to a path x of Σ. Ego wins if $x \in Y$, otherwise Alter wins. A *strategy* is a mapping $f : Q^* \rightarrow Q^+$ such that, for each path fragment α of Σ, $\alpha f(\alpha)$ is a path fragment of Σ. A strategy f is *winning* for player $P \in \{\text{Alter}, \text{Ego}\}$ 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 is well-known that if Y is given by an LTL-formula, then $G(\Sigma, Y)$ is *determinate* (cf. [2]), i.e., either Ego or Alter has a winning strategy.

The system Σ is *fairly* correct w. r. t. Y iff Ego has a winning strategy in $G(\Sigma, Y)$. For convenience, we say that Σ is *fairly* correct w. r. t. Φ iff Ego has a winning strategy in $G(\Sigma, Sat(\Phi))$. *Universal model checking*, denoted by UMC(L), is the problem of deciding whether a given system is universally correct, and *fair model checking*, denoted by FMC(L), is the problem of deciding whether a given system is fairly correct w. r. t. a specification. In both cases, the specification is given by a formula drawn from the language L.

3 Comparing Universal and Fair Model Checking

3.1 Known results

It is known that both universal and fair model checking of LTL are PSPACE-complete [9, 12, 3]. Both problems can be solved in time linear in the system and exponential in the formula [6, 3]. The same holds for the language LTL+past. For universal model checking, this was shown by Sistla and Clarke [9, 8, 6], and for fair model checking, this was claimed by Courcoubetis and Yannakakis [3], but no proof was published. A formal original proof is given in Schmalz' thesis [7].

These results can also be transferred to branching-time logics, where the model checking problems for CTL and a fair version of CTL (as well as for CTL* and a fair version of CTL*) have the same complexities (cf. [11]). Finally, fair and universal model checking for specifications given by a Büchi automaton are both PSPACE-complete [12, 3, 10, 13].

3.2 RLTL

Sistla and Clarke [9] have shown that universal model checking for RLTL is co-NP-complete. In this section, we show that this is also the case for the fair model checking problem for RLTL. Indeed, fair satisfaction of an RLTL formula can be expressed by another RLTL formula. In this way, fair model checking for RLTL can be reduced to universal model checking for RLTL. To this end, we need the notion of a *complete property*.

Definition 1. *Let L be a sublanguage of LTL+past and Σ a system that is fairly correct w. r. t. a property Y. We say that Y is* L-complete w. r. t. Σ *iff $Y \subseteq Sat(\Phi)$ for each $\Phi \in L$ such that Σ is fairly correct w. r. t. Φ.*

If Y is L-complete, then we have that Σ is fairly correct w. r. t. Φ iff $Y \subseteq Sat(\Phi)$, provided that $\Phi \in L$ (cf. [11]). This yields an alternative way of proving and disproving fair correctness.

We will use the fact that *state fairness* is complete for RLTL and expressible in RLTL. Let x be a path and p, q states of a system $\Sigma = (Q, q_0, \rightarrow, v)$. We say that q is *enabled* at p iff $p \rightarrow q$; moreover, q is *enabled at some position i* of x iff q is enabled at x_i. We say that q is *taken* at position i of x iff $x_i = q$. The path x is *state fair* w. r. t. Σ iff each state q of Σ that is enabled at infinitely many positions of x is also taken at infinitely many positions of x. The set of all state fair paths of Σ is denoted by SF_Σ.

It is easy to show that Σ is fairly correct w. r. t. SF_Σ. A winning strategy for Ego consists in first going to a b. s. c. c., and then, at each subsequent turn, taking each state of that b. s. c. c. at least once.

Theorem 2. *Let Σ be a finite system. Then, SF_Σ is $L(\mathrm{F})$-complete w. r. t. Σ.*

Proof. See appendix.

The intuitive meaning of Theorem 2 is the following: whenever we want to prove that Σ is fairly correct w. r. t. a formula $\Phi \in L(\mathrm{F})$, this can be accomplished by showing that each state fair path of Σ satisfies Φ. Theorem 2 was observed already by Zuck et al. [15], who also gave a proof sketch. There, we give a detailed alternative proof.

State fairness can easily be expressed by the following formula of $L(\mathrm{F})$:

$$\Psi(\Sigma) := \bigwedge_{q \in Q} (\mathrm{F}^\infty \mathit{enabled}(q) \Rightarrow \mathrm{F}^\infty q),$$

where, for each $q \in Q$, $\mathit{enabled}(q)$ is an atomic proposition that holds exactly at these states of Σ at which q is enabled. As F^∞ is a shorthand for $\mathrm{G}\,\mathrm{F}$, and G can be defined in terms of F, $\Psi(\Sigma) \in L(\mathrm{F})$.

We are now ready to prove the main result of this section.

Theorem 3. *The problem* $\mathrm{FMC}(L(\mathrm{F}))$ *is co-NP-complete.*

Proof. Hardness is a consequence of Theorem 10 stated below or can be shown similar as in the universal case (cf. [9]).

We prove co-NP membership of FMC(L(F)) by a reduction from FMC(L(F)) to UMC(L(F)). Given a system Σ and a formula $\Phi \in L(\mathrm{F})$, the reduction maps (Φ, Σ) to $(\hat{\Phi}, \Sigma)$, where $\hat{\Phi} := (\Psi(\Sigma) \Rightarrow \Phi) \in L(\mathrm{F})$. By Theorem 2, Σ is fairly correct w. r. t. Φ iff Σ is universally correct w. r. t. $\hat{\Phi}$.

We remark here that also FMC(L(X)) and UMC(L(X)) are co-NP-complete. See [8] for the universal case. In the fair case, the assertion follows from the fact that Σ is correct w. r. t. Φ iff Σ is fairly correct w. r. t. Φ, provided that $\Phi \in L(\mathrm{X})$.

4 Fair Model Checking Can Be Less Expensive than Universal Model Checking

In this section, we show that for $L(\mathrm{F}^\infty)$ the complexities of fair and universal model checking differ. It is known that universal model checking for $L(\mathrm{F}^\infty)$ formulas is co-NP-complete [5]. We show that fair model checking can be done in linear time in the size of the formula and the system. For this, we first introduce a natural subclass of $L(\mathrm{F}^\infty)$ for which the two complexities already differ.

4.1 Muller formulas

A *Muller formula* is an LTL formula where F^∞ is the only temporal operator and where every variable is in the scope of some temporal operator:

Definition 4. *The language $L^+(\mathrm{F}^\infty)$ of* Muller formulas *is the smallest set of LTL formulas that satisfies the following two conditions M1 and M2:*

M1: *If $\Psi \in L(\mathrm{F}^\infty)$, then $\mathrm{F}^\infty \Psi \in L^+(\mathrm{F}^\infty)$.*
M2: *If $\Psi, \Phi \in L^+(\mathrm{F}^\infty)$, then $\Psi \vee \Phi, \neg\Psi \in L^+(\mathrm{F}^\infty)$.*

The key property of Muller formulas is that their validity in a path x only depends on the set $inf(x)$, i.e., the set of states that occur infinitely often in x.

Definition 5. *Let $\Sigma = (Q, q_0, \rightarrow, v)$ be a system. A property Y over Q is a* Muller property *iff for all paths x, y over Q with $inf(x) = inf(y)$ we have $x \in Y$ iff $y \in Y$.*

Theorem 6. *Let Σ be a system. Then, for each $\Phi \in L^+(\mathrm{F}^\infty)$, $Sat(\Phi)$ is a Muller property.*

Proof. See appendix.

It is easy to see that each Muller property can be expressed by a Muller formula (cf. [7]).

4.2 Fair model checking of Muller formulas

In this subsection, we show that fair model checking of Muller formulas can be done in linear time w. r. t. the formula. We are going to present an algorithm for $\mathrm{FMC}(L^+(\mathrm{F}^\infty))$ based on the fact that, for systems Σ that consist of only one s. c. c. and formulas $\Phi \in L(\mathrm{F}^\infty)$, we have that Σ is either fairly correct w. r. t. Φ or w. r. t. $\neg\Phi$.

We are given a system Σ and a Muller formula Φ. Without loss of generality, we assume that Σ has no *isolated states*, i.e., each state of Σ is eventually taken by some path of Σ. First, the algorithm computes the b. s. c. c.s of Σ. Then, for each subformula Υ of Φ, the algorithm partitions each b. s. c. c. K of Σ into K_Υ and $K_{\neg\Upsilon} := K \setminus K_\Upsilon$ as follows. (The meaning of K_Υ is that whenever a state fair path of Σ takes a state of K_Υ, Υ is satisfied at the same position.)

1. If Υ is a state formula, then exactly these states of K that satisfy Υ belong to K_Υ.
2. If $\Upsilon = \Theta \vee \Psi$, then $K_\Upsilon := K_\Theta \cup K_\Psi$.
3. If $\Upsilon = \neg\Theta$, then $K_\Upsilon := K_{\neg\Theta}$.
4. If $\Upsilon = \mathrm{F}^\infty\,\Theta$, then $K_\Upsilon := K$ if $K_\Theta \neq \varnothing$; otherwise, $K_\Upsilon := \varnothing$.

The algorithm accepts its input iff $K = K_\Phi$ for each b. s. c. c. K of Σ.

Proposition 7. *The above algorithm is correct, i.e., the algorithm always terminates, and accepts if and only if Σ is fairly correct w. r. t. Φ.*

Proof. The algorithm obviously terminates. It can be shown by induction over the structure of Υ that the following applies:

1. We have $q \in K_\Upsilon$ iff $SF_\Sigma \subseteq Sat(\mathrm{G}(q \Rightarrow \Upsilon))$.
2. We have $q \in K_{\neg\Upsilon}$ iff $SF_\Sigma \subseteq Sat(\mathrm{G}(q \Rightarrow \neg\Upsilon))$.

Suppose the algorithm accepts Σ and Φ. As Σ is fairly correct w. r. t. SF_Σ, it suffices to show that $SF_\Sigma \subseteq Sat(\Phi)$. Let $x \in SF_\Sigma$. It can be shown that there is a b. s. c. c. K of Σ and a position $i \in \mathbb{N}$ such that $x_i \in K$. Therefore $x_i \in K_\Phi$. With claim 1, $x \vDash \mathrm{G}(x_i \Rightarrow \Phi)$. Hence, $x, i \vDash \Phi$. With Theorem 6, $x \vDash \Phi$.

Now, suppose the algorithm rejects Σ and Φ. Because of Theorem 2, it suffices to show that $SF_\Sigma \nsubseteq Sat(\Phi)$. Let $x \in SF_\Sigma$ such that, for some $i \in \mathbb{N}$, $x_i \in K_{\neg\Phi}$, where K is a b. s. c. c. of Σ with $K \neq K_\Phi$. With claim 2, $x \vDash \mathrm{G}(x_i \Rightarrow \neg\Phi)$. Hence, $x, i \vDash \neg\Phi$. With Theorem 6, $x \nvDash \Phi$.

The computation of the b. s. c. c.s of Σ can be done in $O(|\Sigma|)$ steps. For a given subformula Υ of Φ, also the partition of the b. s. c. c.s K into K_Υ and $K_{\neg\Upsilon}$ can be accomplished in $O(|\Sigma|)$. As Φ has $O(|\Phi|)$ subformulas, the total running time of the algorithm is in $O(|\Sigma||\Phi|)$. We have thus shown the following:

Theorem 8. *The problem $\mathrm{FMC}(L^+(\mathrm{F}^\infty))$ can be solved in $O(|\Sigma||\Phi|)$, where Σ is the input system and Φ the input formula.*

4.3 Fair model checking of $L(\mathrm{F}^\infty)$

Theorem 8 can be extended from $L^+(\mathrm{F}^\infty)$ to $L(\mathrm{F}^\infty)$.

Theorem 9. *The problem* $\mathrm{FMC}(L(\mathrm{F}^\infty))$ *can be solved in* $O(|\Sigma||\Phi|)$*, where* Σ *is the input system and* Φ *the input formula.*

Proof. The algorithm translates Φ to a formula Φ' by applying the following rules as often as possible:

1. Replace each atomic proposition, which is not in the scope of a temporal operator, by its truth value (*true* or *false*) at the initial state of Σ.
2. Replace $\mathit{true} \vee \Psi$ by *true*.
3. Replace $\mathit{false} \vee \Psi$ by Ψ.
4. Replace $\neg \mathit{true}$ by *false*.
5. Replace $\neg \mathit{false}$ by *true*.

It is straightforward to show that, for each path x of Σ, $x \vDash \Phi$ iff $x \vDash \Phi'$. Recall that the only difference between $L(\mathrm{F}^\infty)$ and $L^+(\mathrm{F}^\infty)$ is that in $L^+(\mathrm{F}^\infty)$ each atomic proposition is in the scope of a temporal operator. Therefore, it is not too difficult to see that Φ' is a Muller formula.

After this translation, the algorithm applies Theorem 8. As the translation can be done in $O(|\Phi|)$, the total running time belongs to $O(|\Sigma||\Phi|)$.

5 Canonical Subclasses of RLTL

In this section, we shed more light on the above results by studying the complexity of some simple subclasses of RLTL. The formulas in these subclasses are 'flat', i.e., there is no nesting of temporal operators.

5.1 Conjunctive formulas

We start by observing that top-level conjunctions are easily dealt with: in order to check $\Phi \wedge \Psi$, it is sufficient to check Φ and Ψ in isolation. This is trivial for universal model checking, but is also easily verified for fair model checking: a system is fairly correct w. r. t. $\Phi \wedge \Psi$ iff it is fairly correct w. r. t. Φ and w. r. t. Ψ (cf. for instance [14]).

Thus, if $\{\Psi_1, \ldots, \Psi_n\}$ is a set of formulas whose length is bounded by some constant k, then $\Phi = \bigwedge_{i=1}^n \Psi_i$ can be checked in time $O(|\Sigma| \cdot n \cdot 2^k)$. This implies, for example, that *Street formulas*, i.e., formulas of the form $\bigwedge_{i=1}^n (\mathrm{F}^\infty \psi_i \vee \mathrm{G}^\infty \xi_i)$ with ψ_i, ξ_i state formulas, can be checked in linear time (i.e. $O(|\Sigma||\Phi|)$).

5.2 Disjunctive formulas of RLTL

Disjunctions are more interesting. In particular, we show that co-NP-hardness of fair and universal model checking of RLTL is implied by the fact that fair and universal model checking for formulas of the form $\bigvee_{i=1}^n (\mathrm{F}\,\psi_i \wedge \mathrm{F}\,\xi_i)$ is already co-NP-hard.

Theorem 10.

1. *Fair and universal model checking a formula* $\Phi = \bigvee_{i=1}^{n}(\mathrm{F}\,\psi_i \wedge \mathrm{F}\,\xi_i)$ *and a system* Σ *are co-NP hard.*
2. *Fair and universal model checking a formula* $\Phi = \bigvee_{i=1}^{n}(\mathrm{G}\,\psi_i \wedge \mathrm{G}\,\xi_i)$ *and a system* Σ *can be done in linear time.*
3. *Fair and universal model checking a formula* $\Phi = \bigvee_{i=1}^{n}\mathrm{F}\,\psi_i$ *and a system* Σ *can be done in linear time.*
4. *Fair and universal model checking a formula* $\Phi = \bigvee_{i=1}^{n}\mathrm{G}\,\psi_i$ *and a system* Σ *can be done in linear time.*

Here ψ_i *and* ξ_i *are state formulas* $(1 \leq i \leq n)$.

Proof. For 1, we define a reduction from the complement of $3-\mathrm{SAT}$ to both fair and universal model checking of formulas $\Phi = \bigvee_{i=1}^{n}(\mathrm{F}\,\psi_i \wedge \mathrm{F}\,\xi_i)$. Let $\phi = \bigwedge_{i=1}^{m}\psi_i$ be a 3-CNF formula, where $\psi_i = \xi_{i,1} \vee \xi_{i,2} \vee \xi_{i,3}$ and $\xi_{i,j} \in \{\zeta_1, \ldots, \zeta_n, \overline{\zeta_1}, \ldots, \overline{\zeta_n}\}$ $(1 \leq i \leq m, 1 \leq j \leq 3)$. Then the reduction maps ϕ to the formula $\Phi := \bigvee_{k=1}^{n}(\mathrm{F}\,\zeta_k \wedge \mathrm{F}\,\overline{\zeta_k})$ and the system $\Sigma = (Q, q_0, \rightarrow, v)$ with the following properties:

- $Q = \{q_0, \ldots, q_m\} \cup \{p_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq 3\}$,
- $\rightarrow$ is the smallest relation such that, for $0 \leq i < m, 1 \leq j \leq 3$,
  - $q_i \rightarrow p_{i+1,j}$,
  - $p_{i+1,j} \rightarrow q_{i+1}$,
  - $q_m \rightarrow q_m$.
- $v(q_i) = \{q_i\}$ $(0 \leq i \leq m)$,
- $v(p_{i,j}) = \{\xi_{i,j}, p_{i,j}\}$ $(1 \leq i \leq m)$.

First, we prove that ϕ is satisfiable iff Σ is not universally correct w. r. t. Φ. Suppose that ϕ is satisfiable. Then there are $j_1, \ldots, j_m \in \{1, 2, 3\}$ such that, for each $i \in \{1, \ldots, m\}$, $\xi_{i,j_i} = \zeta_k$ implies that, for each $i' \in \{1, \ldots, m\}$, $\xi_{i',j_{i'}} \neq \overline{\zeta_k}$. Intuitively, ξ_{i,j_i} is the satisfying literal of the i-th clause. We define $x := q_0 p_{1,j_1} q_1 p_{2,j_2} \ldots q_{m-1} p_{m,j_m} q_m q_m q_m \ldots$ Then x is a path of Σ violating Φ; thus, Σ is not universally correct w. r. t. Φ.

The opposite direction can be shown with similar arguments. For the case of fair model checking, note that Σ is universally correct w. r. t. an arbitrary specification iff it is fairly correct w. r. t. that specification. So the reduction is also valid for fair model checking. Clearly, the reduction can be computed in polynomial time; part 1 of the assertion follows.

For 4, we assume, without loss of generality, that Σ has no isolated states. In the case of universal model checking, we propose the following algorithm:

1. Compute the s. c. c. graph of Σ and a topological ordering of the s. c. c.s.
2. Travel through the s. c. c.s in topological order, and compute for each s. c. c. K of Σ:

$$valid(K) = \{i \in \{1, \ldots, n\} \mid \forall q \in K : q \vDash \psi_i\} \cap \bigcap_{K' :\ K' \rightarrow K} valid(K').$$

Given s. c. c.s K_1, K_2 of Σ, $K_1 \rightarrow K_2$ means that there are $p \in K_1, q \in K_2$ such that $p \rightarrow q$.

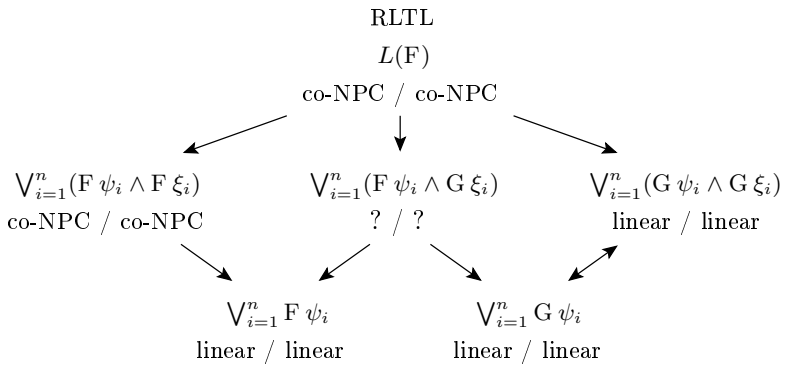

Fig. 1. Results for subclasses of $L(\mathrm{F})$ showing the complexity of universal model checking / fair model checking.

3. The input is accepted iff there is no s. c. c. K of Σ with $\mathit{valid}(K) = \varnothing$.

By induction over the number of s. c. c.s the algorithm has already processed, it can be shown that $i \in \mathit{valid}(K)$ iff each path fragment α of Σ that ends in a state of K at each position satisfies ψ_i. From this, the correctness of the algorithm can be derived:

Let x be a path of Σ with $x \not\models \Phi$. Choose j such that each of the ψ_i is violated at at least one position of $x_0 x_1 \ldots x_j$. Let K be the s. c. c. of Σ such that $x_j \in K$. Then, for each $i \in \{1, \ldots, n\}$, we have $i \notin \mathit{valid}(K)$, because $x_0 x_1 \ldots x_j$ does not satisfy ψ_i at each position. Thus, $\mathit{valid}(K) = \varnothing$.

On the other hand, suppose that $\mathit{valid}(K) = \varnothing$ for some s. c. c. K of Σ. Then there is a path fragment α of Σ such that, for each $i \in \{1, \ldots, n\}$, ψ_i is violated at some position of α. Thus, α can be extended to a path of Σ that violates the specification $\mathit{Sat}(\Phi)$.

In the case of fair model checking, the same algorithm can be applied, because Σ is universally correct w. r. t. Φ iff Σ is fairly correct w. r. t. Φ.

Part 2 of the assertion can be derived from 4, as we have $\mathit{Sat}(\mathrm{G}\,\psi_i \wedge \mathrm{G}\,\xi_i) = \mathit{Sat}(\mathrm{G}(\psi_i \wedge \xi_i))$ for $1 \leq i \leq n$.

For 3, observe that $\mathit{Sat}(\bigvee_{i=1}^{n}\mathrm{F}\,\psi_i) = \mathit{Sat}(\mathrm{F}\bigvee_{i=1}^{n}\psi_i)$. So the problems of 3 can be reduced to the related model checking problems for a formula of the form $\mathrm{F}\,\zeta$, where $\zeta \in AP$. The latter can be solved in linear time (cf. [6, 3]), as the formula has bounded size.

Figure 1 summarises the results for the disjunctive formulas of $L(\mathrm{F})$. An arrow denotes containment, where we also allow trivial translations, e.g., $\mathrm{G}\,\psi_i$ can be written as $\mathrm{G}\,\psi_i \wedge \mathrm{G}\,\mathit{true}$ and $\mathrm{G}\,\psi_i \wedge \mathrm{G}\,\xi_i$ can be written as $\mathrm{G}(\psi_i \wedge \xi_i)$. The complexities of fair and universal model checking of formulas of the form $\bigvee_{i=1}^{n}(\mathrm{F}\,\psi_i \wedge \mathrm{G}\,\xi_i)$ remain open.

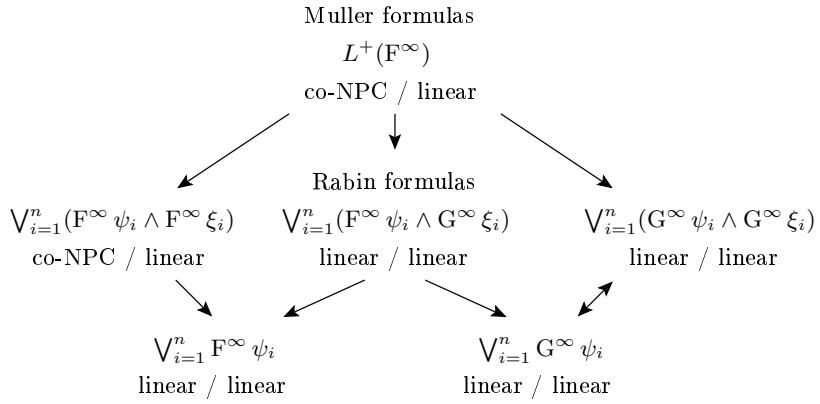

Fig. 2. Results for subclasses of $L^+(\mathrm{F}^\infty)$ showing the complexity of universal model checking / fair model checking.

5.3 Disjunctive formulas of $L(\mathrm{F}^\infty)$

The dual of a Streett formula, called a *Rabin formula*, is a formula of the form $\bigvee_{i=1}^n(\mathrm{F}^\infty \psi_i \wedge \mathrm{G}^\infty \xi_i)$. Universal model checking of Rabin formulas can be done in linear time, whereas the proof of co-NP-hardness of $L(\mathrm{F}^\infty)$ uses only formulas of the form $\bigvee_{i=1}^n(\mathrm{F}^\infty \psi_i \wedge \mathrm{F}^\infty \xi_i)$ (cf. [5]). We thus have:

Theorem 11.

1. *Universal model checking a formula $\Phi = \bigvee_{i=1}^n(\mathrm{F}^\infty \psi_i \wedge \mathrm{F}^\infty \xi_i)$ and a system Σ is co-NP hard.*
2. *Fair model checking a formula $\Phi = \bigvee_{i=1}^n(\mathrm{F}^\infty \psi_i \wedge \mathrm{G}^\infty \xi_i)$ and a system Σ can be done in linear time.*

In particular universal model checking for formulas of the form $\bigvee_{i=1}^n \mathrm{F}^\infty \psi_i$ or $\bigvee_{i=1}^n \mathrm{G}^\infty \psi_i$ can be done in linear time.

Figure 2 summarises the results for subclasses of $L^+(\mathrm{F}^\infty)$.

6 Conclusion

We have shown that for formulas in $L(\mathrm{F}^\infty)$ fair model checking can be done more efficiently than universal model checking. We are not aware of any natural sublanguage of LTL for which universal model checking can be done more efficiently than fair model checking. This adds another argument in favour of fair model checking as an interesting alternative or complement to universal model checking, as mentioned in the introduction.

Studying model checking for sublanguages can help to optimise algorithms, as the more general algorithms may not perform optimally for the sublanguage. In fact, the algorithm of Courcoubetis and Yannakakis [3] for fair model checking of LTL can perform exponentially worse on $L(\mathrm{F}^\infty)$ than our algorithm (see

[7]). Moreover, our algorithm for Muller formulas can be integrated with the algorithm of Courcoubetis and Yannakakis [3], which allows us to detect Muller formulas as subformulas of the input LTL formula (or any intermediate formula produced by the algorithm), solve the fair model checking problem for these Muller formulas in linear time and use the result for checking the input formula. The presentation of this integration is beyond the scope of this paper, but it is available in Schmalz' thesis [7]. There it is also shown that, with this optimisation, the algorithm never performs worse but can perform exponentially better than the original.

References

1. R. Alur and T. A. Henzinger. Local liveness for compositional modeling of fair reactive systems. In *CAV*, volume 939 of *Lect. Notes in Comput. Sci.*, pages 166–179. Springer, 1995.
2. D. Berwanger, E. Grädel, and S. Kreutzer. Once upon a time in the west - determinacy, definability, and complexity of path games. In *LPAR*, volume 2850 of *Lect. Notes in Comp. Sci.*, pages 229–243. Springer, 2003.
3. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.
4. E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science, 1990.
5. E. A. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. *Sci. Comput. Program.*, 8(3):275–306, 1987.
6. O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *POPL*, pages 97–107. ACM, 1985.
7. M. Schmalz. Extensions of an algorithm for generalised fair model checking. Diploma Thesis, Technical Report B 07-01, University of Lübeck, Germany, 2007, www.tcs.uni-luebeck.de/Forschung/B0701.pdf.
8. Ph. Schnoebelen. The complexity of temporal logic model checking. In *AiML*, pages 393–436. King's College Publications, 2002.
9. A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985.
10. A. P. Sistla, M. Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. In *ICALP*, volume 194 of *Lect. Notes in Comp. Sci.*, pages 465–474. Springer, 1985.
11. D. Varacca and H. Völzer. Temporal logics and model checking for fairly correct systems. In *LICS*, pages 389–398. IEEE, 2006.
12. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338. IEEE, 1985.
13. M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *LICS*, pages 332–344. IEEE, 1986.
14. H. Völzer, D. Varacca, and E. Kindler. Defining fairness. In *CONCUR*, volume 3653 of *Lect. Notes in Comp. Sci.*, pages 458–472. Springer, 2005.
15. L. D. Zuck, A. Pnueli, and Y. Kesten. Automatic verification of probabilistic free choice. In *VMCAI*, volume 2294 of *Lect. Notes in Comp. Sci.*, pages 208–224. Springer, 2002.

A Proofs

A.1 Semantical properties of RLTL and $L^+(\mathrm{F}^\infty)$

First, we introduce some notations. Let $\Sigma = (Q, q_0, \to, v)$ be a system, $x, y \in Q^\omega$ and Φ an LTL+past formula. Given $T \subseteq Q$, we set $R(T) := \{x \in T^\omega \mid inf(x) = T\}$. Moreover, we write that $x \sim_\Phi y$ iff $x \vDash \Phi \Leftrightarrow y \vDash \Phi$. For a sublanguage L of LTL+past, we write $x \sim_L y$ iff we have $x \sim_\Upsilon y$ for each $\Upsilon \in L$.

Theorem 6. *Let $\Sigma = (Q, q_0, \to, v)$ be a system. Then, for each $\Phi \in L^+(\mathrm{F}^\infty)$, $Sat(\Phi)$ is a Muller property.*

Proof. Let x, y be paths over Q such that $inf(x) = inf(y)$. First, one can show by structural induction over Φ that $x \sim_{\mathrm{F}^\infty \Phi} y$ and $x \sim_{\mathrm{G}^\infty \Phi} y$, for each $\Phi \in L(\mathrm{F}^\infty)$. Then it can be shown by structural induction over Φ that $x \sim_\Phi y$ for each $\Phi \in L^+(\mathrm{F}^\infty)$. See [7] for the details.

Lemma 12. *Let $\Sigma = (Q, q_0, \to, v)$ be a system and x, y be paths over Q. Then $x \sim_{L(\mathrm{F})} y$ if*

1. *$x_0 = y_0$,*
2. *$inf(x) = inf(y)$, and*
3. *there are $n \in \mathbb{N}$, $\alpha \in Q^n$ such that*

$$x, y \in \{\alpha_0\}^+ \dots \{\alpha_{n-1}\}^+ R(inf(x)).$$

Proof. For ease of notation, we define $x[i, \infty] := x_i x_{i+1} \dots$ $(i \in \mathbb{N})$.

We show by structural induction over Φ that 1 - 3 imply

$$x \vDash \Phi \Leftrightarrow y \vDash \Phi \tag{1}$$

for each $\Phi \in L(\mathrm{F})$. Note that it suffices to show that $x \vDash \Phi \Rightarrow y \vDash \Phi$. The reverse direction follows from symmetry arguments.

Let Φ be a state formula. Assertion (1) follows from $x_0 = y_0$.

Let $\Phi = \mathrm{G}\,\Psi$ for some $\Psi \in L(\mathrm{F})$, and suppose the left-hand side of (1) holds. Hence $x[j, \infty] \vDash \Psi$ for each $j \in \mathbb{N}$. Let $i \in \mathbb{N}$. By assumption, we have

$$y[i, \infty] \in \{\alpha_\ell\}^+ \dots \{\alpha_{n-1}\}^+ R(inf(x)),$$

for some ℓ with $0 \leq \ell \leq n$. We choose $j \in \mathbb{N}$ such that $x_j = y_i$ and also

$$x[j, \infty] \in \{\alpha_\ell\}^+ \dots \{\alpha_{n-1}\}^+ R(inf(x)).$$

Note that 1 - 3 also hold if we replace x by $x[j, \infty]$ and y by $y[i, \infty]$. Thus, the induction hypothesis yields $y[i, \infty] \vDash \Psi$. As i was arbitrarily chosen, the right-hand side of (1) holds.

In the cases $\Phi = \neg\Upsilon$ and $\Phi = \Upsilon \vee \Psi$, for some $\Upsilon, \Psi \in L(\mathrm{F})$, Equation (1) immediately follows from the induction hypothesis.

Schmalz [7] showed that 1 - 3 of Lemma 12 are also *necessary* for $x \sim_{L(\mathrm{F})} y$.

A.2 State fairness is complete for RLTL

Lemma 13. *Let Σ be a system and $x \in SF_\Sigma$. Then $x = \alpha y$ such that, for a b. s. c. c. K of Σ, we have $y \in R(K)$.*

Proof. Let q be a state that is visited infinitely often by x. We fix $i \in \mathbb{N}$ such that $x_i = q$. Now choose β such that $x_0 \dots x_i\beta$ is a path fragment of Σ that ends in a b. s. c. c., say K. It can be shown by induction over the length of β that each state on β is taken infinitely often by x. As the last state of β may be arbitrarily chosen, each state in K is taken infinitely often by x. The assertion follows.

Lemma 14. *Let Σ be a system, Z a property such that Alter has a winning strategy for $G(\Sigma, Z^c)$, and Y a property such that Ego has a winning strategy for $G(\Sigma, Y)$. If $(Z \cap Y)^c$ is determinate, then Alter has a winning strategy for $G(\Sigma, (Z \cap Y)^c)$.*

Proof. Suppose Ego has a winning strategy for $G(\Sigma, (Z \cap Y)^c)$. As fair correctness is preserved under intersection of the specification [14], Ego has a winning strategy for $G(\Sigma, (Z \cap Y)^c \cap Y)$. Because $(Z \cap Y)^c \cap Y \subseteq Z^c$, Ego has a winning strategy for $G(\Sigma, Z^c)$ – a contradiction. Because of determinacy, Alter has a winning strategy for $G(\Sigma, (Z \cap Y)^c)$.

Theorem 2. *Let Σ be a finite system. Then, SF_Σ is $L(\mathrm{F})$-complete w. r. t. Σ.*

Proof. Let $\Phi \in L(\mathrm{F})$ such that Σ is fairly correct w. r. t. Φ and x be a state fair path of Σ. We have to show that $x \vDash \Phi$.

With Lemma 13, we choose $\alpha \in Q^*, q \in Q, y \in Q^\omega$ such that $x = \alpha qy$ and $qy \in R(K)$, where K is a b. s. c. c. of Σ. For ease of notation, we set $\beta\uparrow := \{x \in Q^\omega \mid \beta \text{ is a prefix of } x\}$. Note that Alter has a winning strategy for $G(\Sigma, (\alpha q\uparrow)^c)$ and Ego one for $G(\Sigma, SF_\Sigma)$. With Lemma 14, Alter has a winning strategy g for $G(\Sigma, (\alpha q\uparrow \cap SF_\Sigma)^c)$. (Note that $(\alpha q\uparrow \cap SF_\Sigma)^c$ is determinate, as it can be expressed by an LTL formula [2].) Moreover, f is a winning strategy for Ego in $G(\Sigma, Sat(\Phi))$. If Ego plays according to f and Alter according to g, then they create a path $u = \alpha qz$ of Σ that is state fair *and* satisfies Φ. Clearly, $x_0 = u_0$. As both x and u are state fair and eventually take a state of K, we have $inf(x) = K = inf(u)$. Moreover, $x, u \in \alpha R(K)$. With Lemma 12, $x \sim_{L(\mathrm{F})} u$. As $\Phi \in L(\mathrm{F})$ and $u \vDash \Phi$, we conclude that $x \vDash \Phi$.