Available online at www.sciencedirect.com
SCIENCE DIRECT
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 119 (2005) 33–49
www.elsevier.com/locate/entcs

Game Over: The Foci Approach to LTL Satisfiability and Model Checking

Christian Dax Martin Lange

Institut für Informatik, University of Munich

Abstract

Focus games have been shown to yield game-theoretical characterisations for the satisfiability and the model checking problem for various temporal logics. One of the players is given a tool – the focus – that enables him to show the regeneration of temporal operators characterised as least or greatest fixpoints. His strategy usually is build upon a priority list of formulas and, thus, is not positional. This paper defines foci games for satisfiability of LTL formulas. Strategies in these games are trivially positional since they parallelise all of the focus player's choices, thus resulting in a 1-player game in effect. The games are shown to be correct and to yield smaller (counter-)models than the focus games. Finally, foci games for model checking LTL are defined as well.

Keywords: Verification, Temporal Logic, Tool Support

1 Introduction

Verification is strongly linked with logics and in fact is nowadays a broadly accepted and active area of research in logics for computer science. This is because logical formulas express properties and, therefore, can be used to express correctness in particular. The question of whether or not a piece of hardware or software obeys a certain correctness property is often reduced to the satisfiability or model checking problem for a temporal logic.

In both cases, games provide an advantageous mechanism for carrying out the verification task. A game-based algorithm deciding one of these problems computes a winning strategy for a player in a certain 2-player game. This strategy can then be used not only to report to the user of a verification tool that the examined formula is or is not satisfied/satisfiable, but also to show to her *why* this is the case. This is done by letting the user play against the

doi:10.1016/j.entcs.2004.07.007

winning strategy. By its very definition, the user is bound to lose any resulting play. Since the rules of such games usually follow the semantics of formulas closely, this provides the user with insight into where exactly a transition system fails to satisfy the formula at hand for example.
Here we deal with Linear Time Temporal Logic (LTL) [10]. It has been well studied since and enjoys various decision procedures for both its model checking and its satisfiability problem. They include tableau methods [10,8,1,12], resolution [5], reductions [4], automata-theoretic procedures [13,14] as well as symbolic methods [2].

Satisfiability games for LTL have been defined in [6] in terms of *focus games*. They work on sets of formulas but equip the universal player (refuter) with a simple tool that allows him to neatly show the regeneration of least fixpoint constructs – one possible source of unsatisfiability. This makes them suitable for making the user of a verification tool understand unsatisfiability of a given formula as described above.

Focus games were first used to give a game-theoretical characterisation of the model checking problem for the full branching time logic CTL* [7]. It is well known that this is very closely related to LTL's model checking problem. Thus, [7] also provides a game-based approach to LTL model checking.

These focus games have nice theoretical properties. For example, refuter's strategies can explicitly be given in terms of priority lists, and refuter's moves of the focus can be determinised at no further complexity-theoretic costs. However, when it comes to implementing these games a few obstacles have to be dealt with. The main problem is to decide which formula refuter should focus on in case there is no least fixpoint construct in the actual configuration. Other difficulties arise when trying to recognise the right moment a focused formula is fulfilled and the refuter has to react. Furthermore, the priority list needs to be constantly maintained because formulas in the corresponding configurations change shape during a play.

Their main disadvantage however, is the fact that refuter's winning strategies are not positional but *Least Appearance Record* strategies [9]. This means that the tree of plays cannot simply be stored as a directed acyclic graph which would be much better for efficiency.

Here we propose *foci games* to overcome these issues. They extend the focus approach by letting the refuter focus on all possible least fixpoint constructs in parallel.

This has several implications: On the positive side, foci games provide a game-theoretical characterisation of LTL's satisfiability problem which is better suited for tool support. They also improve the small model property gained from focus games by a linear factor, thus yielding shorter models of

satisfiable formulas. On the negative side, foci games are degenerated games – strictly speaking. In (satisfiability) focus games, all that refuter does is set the focus which he is completely deprived of in the foci games. However, we do not propose the use of foci games directly for interacting with the user of a verification tool. Instead we show how (rather simply) winning strategies for focus games can be obtained from those of foci games. Then, focus games can be used as an interface between a user and the foci games. This combines advantages of both approaches in a verification tool.

It it well known that LTL's satisfiability problem is closely related to its model checking problem. We also define foci games for LTL model checking. The same discussion about pros and cons holds for them as well.

The paper is organised as follows. Section 2 recalls the definition of LTL. Section 3 gives a short summary and discussion of focus games for satisfiability of LTL formulas. Section 4 defines foci games for this problem, proves them correct, sketches an algorithm that decides their winners and shows how to use their winning strategies back in focus games. Finally, Section 5 defines model checking foci games for LTL.

2 Preliminaries

Let $\mathcal{P}$ be a set of monadic second-order propositions which is closed under complementation, i.e. for every $q \in \mathcal{P}$ there is a $\overline{q} \in \mathcal{P}$ with $\overline{\overline{q}} = q$. Furthermore let $\mathcal{P}$ contain two distinguished propositions $\mathtt{tt}$ and $\mathtt{ff}$ that are complementary to each other.

A *linear time structure* is a tuple $\pi = (\mathcal{S}, <, \mathcal{I}_\pi)$ where $\mathcal{S}$ is a set $\{s_0, s_1, \ldots\}$ of order type ω. We usually call them *states*. $<$ is a total order on the set of states. $\mathcal{I}_\pi$ is an interpretation function that maps every proposition q to the set of states satisfying q.

Interpretations have to obey complementation as well as true and false, i.e. for every $\mathcal{I}_\pi$, every $q \in \mathcal{P}$ and every $i \in \mathbb{N}$ we have $s_i \in \mathcal{I}_\pi(\mathtt{tt})$, and $s_i \in \mathcal{I}_\pi(q)$ iff $s_i \notin \mathcal{I}_\pi(\overline{q})$.

We write π^k for the k-th suffix of π. Its universe is $s_k s_{k+1} \ldots$ and its interpretation is defined by $s_i \in \mathcal{I}_{\pi^k}(q)$ iff $s_{i+k} \in \mathcal{I}_\pi(q)$ for every $q \in \mathcal{P}$.

Formulas of LTL in positive normal form are given by the following grammar.

$$\varphi \quad ::= \quad q \;\mid\; \varphi \vee \varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \mathtt{X}\varphi \;\mid\; \varphi \mathtt{U} \varphi \;\mid\; \varphi \mathtt{R} \varphi$$

The temporal operators $\mathtt{U}$ and $\mathtt{R}$ can be characterised as fixpoints of certain equivalences. The rules in Sections 3, 4 and 5 will make use of this. In order

to do so, we need to define unfoldings of these operators given by

$$\begin{aligned}\mathtt{unf}(\varphi \mathtt{U} \psi) &:= \{\varphi \mathtt{U} \psi, \mathtt{X}(\varphi \mathtt{U} \psi), \varphi \wedge \mathtt{X}(\varphi \mathtt{U} \psi), \psi \vee (\varphi \wedge \mathtt{X}(\varphi \mathtt{U} \psi))\} \\ \mathtt{unf}(\varphi \mathtt{R} \psi) &:= \{\varphi \mathtt{R} \psi, \mathtt{X}(\varphi \mathtt{R} \psi), \varphi \vee \mathtt{X}(\varphi \mathtt{R} \psi), \psi \wedge (\varphi \vee \mathtt{X}(\varphi \mathtt{R} \psi))\}\end{aligned}$$

The set $Sub(\varphi)$ of subformulas of an LTL formula φ is defined as usual except for the cases

$$\begin{aligned}Sub(\varphi \mathtt{U} \psi) &:= \mathtt{unf}(\varphi \mathtt{U} \psi) \cup Sub(\varphi) \cup Sub(\psi) \\ Sub(\varphi \mathtt{R} \psi) &:= \mathtt{unf}(\varphi \mathtt{R} \psi) \cup Sub(\varphi) \cup Sub(\psi)\end{aligned}$$

Note that the number of subformulas of a formula is at most linear in its syntactical length. Thus, we define the size of φ as $|\varphi| := |Sub(\varphi)|$. Furthermore, we define for arbitrary formulas χ the set of all until/release unfoldings of formulas in χ as

$$\begin{aligned}\mathtt{UNT}(\chi) &:= \bigcup \{\, \mathtt{unf}(\varphi \mathtt{U} \psi) \mid \varphi \mathtt{U} \psi \in Sub(\chi) \,\} \\ \mathtt{REL}(\chi) &:= \bigcup \{\, \mathtt{unf}(\varphi \mathtt{R} \psi) \mid \varphi \mathtt{R} \psi \in Sub(\chi) \,\}\end{aligned}$$

LTL is interpreted over linear time structures π. For any π with universe $s_0 s_1 \ldots$ we have

$\pi \models q$	iff	$s_0 \in \mathcal{I}_\pi q$
$\pi \models \varphi \vee \psi$	iff	$\pi \models \varphi$ or $\pi \models \psi$
$\pi \models \varphi \wedge \psi$	iff	$\pi \models \varphi$ and $\pi \models \psi$
$\pi \models \mathtt{X}\varphi$	iff	$\pi^1 \models \varphi$
$\pi \models \varphi \mathtt{U} \psi$	iff	there is a $k \in \mathbb{N}$, s.t. $\pi^k \models \psi$ and for all $j < k : \pi^j \models \varphi$
$\pi \models \varphi \mathtt{R} \psi$	iff	for all $k \in \mathbb{N} : \pi^k \models \psi$ or there is a $j < k$, s.t. $\pi^j \models \varphi$

Approximants of $\mathtt{U}$ and $\mathtt{R}$ formulas are defined for any $k \in \mathbb{N}$ as

$$\begin{aligned}\varphi \mathtt{U}^0 \psi &:= \mathtt{ff} & \qquad \varphi \mathtt{U}^{k+1} \psi &:= \psi \vee (\varphi \wedge \mathtt{X}(\varphi \mathtt{U}^k \psi)) \\ \varphi \mathtt{R}^0 \psi &:= \mathtt{tt} & \qquad \varphi \mathtt{R}^{k+1} \psi &:= \psi \wedge (\varphi \vee \mathtt{X}(\varphi \mathtt{R}^k \psi))\end{aligned}$$

The next lemma is a standard result about least and greatest fixpoints.

Lemma 2.1 *For any linear time structure π, any LTL formulas φ and ψ we have:*
a) $\pi \models \varphi \mathtt{U} \psi$ iff there is a $k \in \mathbb{N}$ s.t. $\pi \models \varphi \mathtt{U}^k \psi$
b) $\pi \models \varphi \mathtt{R} \psi$ iff for all $k \in \mathbb{N}$: $\pi \models \varphi \mathtt{R}^k \psi$

Since the underlying set of propositions is supposed to be closed under complementation and every LTL operator has a dual one we also get that LTL

is closed under negation. Then we can assume that for any LTL formula φ there is also an LTL formula $\neg\varphi$. Furthermore, we will use the abbreviations $\varphi \to \psi := \neg\varphi \vee \psi$ as well as $\mathtt{F}\varphi := \mathtt{tt}\mathtt{U}\varphi$ and $\mathtt{G}\varphi := \mathtt{ff}\mathtt{R}\varphi$. Finally, we write $\mathtt{X}^k\varphi$ to denote $\underbrace{\mathtt{X}\ldots\mathtt{X}}_{k \text{ times}}\varphi$.

3 Focus Games for Satisfiability of LTL Formulas

Focus game for satisfiability of LTL formulas as presented in [6] work as follows. Two players called $\exists$ and $\forall$ play on sets of subformulas of a given formula φ in order to determine whether or not φ is satisfiable. Player $\exists$ believes that it is whereas player $\forall$ wants to show that it is not.

Note that a temporal formula can be unsatisfiable because of two reasons. Either it contains a propositional contradiction or a least fixpoint formula that does not get fulfilled in a possible model. The first case of unsatisfiability is easy to handle since it always breaks down to atomic propositions. The second case is slightly more difficult when configurations are sets of formulas. It is not enough to stop a play or a tableau branch once a repeat of a configuration occurs and to check whether a least fixpoint construct occurs in it. It is not immediately clear whether this construct really has not been fulfilled or actually was regenerated by a greatest fixpoint construct. Therefore player $\forall$ is equipped with a tool called *focus* which highlights one formula in a configuration. He can use it to show that a least fixpoint construct does not get fulfilled along a play.

The set of configurations of the game $\Gamma(\varphi_0)$ is $Sub(\varphi_0) \times 2^{Sub(\varphi_0)}$. A configuration is written $\big[\psi\big], \Phi$ which means that ψ is the one formula that player $\forall$ currently focuses on.

The game rules are presented in Figure 1. They are to be read top-to-bottom. For example, if a configuration contains a disjunction then player $\exists$ has to choose one of the disjuncts that replaces the disjunction. Fixpoint formulas are unfolded. At any moment, player $\forall$ can reset the focus to another formula. If every formula present in the current configuration begins with an $\mathtt{X}$ or is an atomic proposition then all the $\mathtt{X}$'s are stripped off and the atomic propositions are discarded unless one of the following winning conditions holds beforehand.

Player $\forall$ wins the play $C_0, \ldots, C_n$ iff

(i) $\mathtt{ff} \in C_n$ or there is a $q \in \mathcal{P}$ s.t. $q \in C_n$ and $\overline{q} \in C_n$, or

(ii) there is an $i < n$ s.t. $C_i = C_n = \big[\varphi\mathtt{U}\psi\big], \Phi$ for some φ, ψ, Φ and between C_i and C_n player $\forall$ has not used rule (FC).

Player $\exists$ wins the play $C_0, \ldots, C_n$ iff

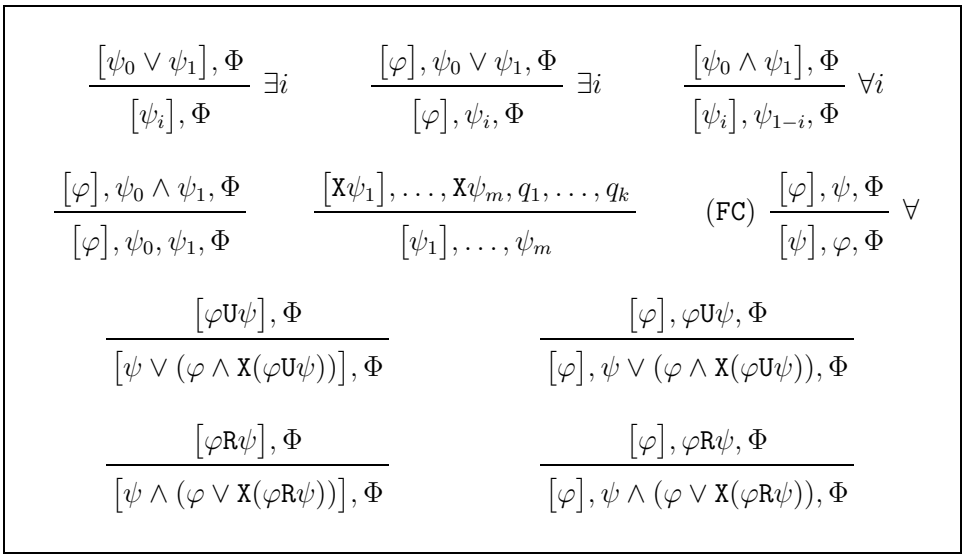

Fig. 1. The rules of the LTL satisfiability focus games.

(iii) $C_n = [q_1], \ldots, q_k$ and for all $i = 1, \ldots, k$: $\overline{q_i} \notin C_n$,

(iv) there is an $i < n$ s.t. $C_i = C_n = [\varphi \mathtt{R} \psi], \Phi$ for some φ, ψ, Φ, and between C_i and C_n player $\forall$ has not used rule (FC),

(v) there is an $i < n$ s.t. $C_i = C_n = [\varphi], \Phi$ for some φ, Φ, and between C_i and C_n player $\forall$ has used rule (FC).

Lemma 3.1 [6] *Every play of the game $\Gamma(\varphi)$ has length at most $O(|\varphi| \cdot 2^{|\varphi|})$.*

This is mainly because there are only $|\varphi| \cdot 2^{|\varphi|}$ many different configurations in the game $\Gamma(\varphi)$.

Theorem 3.2 [6] *Player $\exists$ has a winning strategy for the game $\Gamma(\varphi)$ iff φ is satisfiable.*

The winning strategies can be given explicitly: player $\exists$ has to preserve satisfiability whenever she chooses disjuncts. A winning strategy for player $\forall$ is conceptually more difficult but easier from a complexity-theoretic point of view. Note that all he does is to set the focus in these games. Thus, a winning strategy must tell him how to do this sensibly. Also note that because of the way the winning conditions are defined, "sensibly" means that he must try to minimise focus changes with rule (FC). This is accomplished by maintaining a priority list of U formulas s.t. he

- always focuses on the first one in the list that is present in the current configuration,

- only changes focus when player ∃ discards an unfolding of an `U` with the rule for a disjunction,
- moves formulas that were previously focused on to the end of the list.

Positional strategies. It is not hard to see that player ∃'s winning strategies are positional, i.e. her moves only depend on the actual configuration. This is not the case for player ∀. At any moment in a play, the maintained list is a compact representation of the part of the play's history that matters to player ∀.

Complexity. In order to measure the asymptotic complexity of deciding the winner of a game it is often helpful to design an alternating algorithm and then to employ results about alternating complexity classes from [3]. This is because an alternating algorithm only follows a single play. Here, a single play can be played using polynomial space only, see [6] for details. This would naturally lead to an EXPTIME procedure for deciding the winner because EXPTIME equals alternating PSPACE. However, player ∀'s moves can be determinised using the above described strategy at no further (complexity-theoretic) costs. This leaves a game in which only player ∃ makes choices which is nothing more than a nondeterministic procedure. Luckily, NPSPACE equals PSPACE according to [11]. Hence, deciding the winner of these games is in PSPACE.

Small model property. The soundness part of Theorem 3.2 can be proved by constructing a model for φ from a winning strategy for player ∃ and the game $\Gamma(\varphi)$. Then, the small model property for LTL is a consequence of Lemma 3.1 and yields an upper bound of $O(|\varphi| \cdot 2^{|\varphi|})$ for the size of a model for φ.

4 Foci Games

Given a $\varphi \in$ LTL, the satisfiability foci game $\mathcal{G}(\varphi)$ parallelises and, hence, eliminates all possible choices to be made by player ∀ in the game $\Gamma(\varphi)$. Therefore, it is rather a tableau than a game. These tableaux (for unsatisfiability) are nothing more than representations of winning strategies for player ∀ in the games of the previous section. They are also similar to non-deterministic automata for LTL. It is the winning conditions which makes the difference between a game and an automaton. Here, the winner is determined by what happens between two repeating configurations which is not exactly the same as a Büchi acceptance condition.

Despite the similarities to tableaux and automata we will continue to call $\mathcal{G}(\varphi)$ a game even though one of the players in it is left with nothing to do.

$$\frac{\psi_0 \vee \psi_1, \Phi; c}{\psi_0, \Phi; c \qquad \psi_1, \Phi; c}\ \exists \qquad\qquad \frac{\left[\psi_0 \vee \psi_1\right]^i, \Phi; c}{\psi_0, \Phi; c \qquad \left[\psi_1\right]^k, \Phi; c}\ \exists$$

$$\frac{\psi_0 \wedge \psi_1, \Phi; c}{\psi_0, \psi_1, \Phi; c} \qquad \frac{\left[\psi_0 \wedge \psi_1\right]^i, \Phi; c}{\psi_0, \left[\psi_1\right]^i, \Phi; c} \qquad \frac{\varphi, \Phi; c}{\left[\varphi\right]^c, \Phi; c+1}\ \text{if } \varphi \in \mathtt{UNT}(\varphi_0)$$

$$\frac{\left[\varphi \mathtt{U} \psi\right]^i, \Phi; c}{\left[\psi \vee (\varphi \wedge \mathtt{X}(\varphi \mathtt{U} \psi))\right]^i, \Phi; c} \qquad \frac{\varphi \mathtt{R} \psi, \Phi; c}{\psi \wedge (\varphi \vee \mathtt{X}(\varphi \mathtt{R} \psi)), \Phi; c}$$

$$(\mathtt{X})\ \frac{\left[\mathtt{X}\psi_1\right]^{i_1}, \ldots, \left[\mathtt{X}\psi_m\right]^{i_m}, \mathtt{X}\varphi_1, \ldots, \mathtt{X}\varphi_n, q_1, \ldots, q_k; c}{\left[\psi_1\right]^{i_1}, \ldots, \left[\psi_m\right]^{i_m}, \varphi_1, \ldots, \varphi_n; c} \qquad \frac{\left[\varphi\right]^i, \left[\varphi\right]^j, \Phi; c}{\left[\varphi\right]^{\min(i,j)}, \Phi; c}$$

Fig. 2. The rules of the LTL foci games.

We will also speak of a play rather than a branch or a run in order not to lose the connection to focus games. Note that a play won by player $\exists$ is an unsuccessful branch in a tableau for unsatisfiability.

Configurations of the game $\mathcal{G}(\varphi)$ are subsets of formulas, in which several formulas are highlighted with individual foci; plus an additional counter $c \in \mathbb{N}$. Formally, a configuration is an element of $2^{Sub^*(\varphi)}$, where $Sub^*(\varphi)$ is defined as $Sub(\varphi)$ except for the case

$$\begin{aligned} Sub^*(\varphi \mathtt{U} \psi) \ :=\ & \{\varphi \mathtt{U} \psi, [\varphi \mathtt{U} \psi]^i, [\mathtt{X}(\varphi \mathtt{U} \psi)]^i, [\varphi \wedge \mathtt{X}(\varphi \mathtt{U} \psi)]^i, [\psi \vee (\varphi \wedge \mathtt{X}(\varphi \mathtt{U} \psi))]^i\} \\ & \cup Sub^*(\varphi) \cup Sub^*(\psi), \quad \text{where } i \in \mathbb{N}. \end{aligned}$$

A configuration is written

$$\left[\varphi_1\right]^{i_1}, \ldots, \left[\varphi_m\right]^{i_m}, \Phi; c$$

which means that formula φ_j is currently highlighted with focus i_j. The counter value is c.

For two configurations C_i and C_j in the game $\mathcal{G}(\varphi_0)$ we write $C_i \approx C_j$ if (1) they are equal as sets of formulas disregarding the foci and the counters, and (2) for every $\chi \in \mathtt{UNT}(\varphi_0)$: χ has a focus in C_i and in C_j.

Every play of $\mathcal{G}(\varphi_0)$ starts with the configuration $\varphi_0; 0$. The rules are presented in Figure 2. Player $\exists$ chooses disjuncts. Conjunctions are flattened and fixpoint constructs are unfolded. An occurring $\chi \in \mathtt{UNT}(\varphi_0)$ is highlighted with a new focus. Note that foci on conjunctions and disjunctions are always

passed to the right assuming that boolean junctions are ordered pairs. The reason for this is: a $\varphi \mathtt{U} \psi$ occurs rightmost in the unfolding of itself. Rule (X) strips off trailing X operators and satisfiable sets of propositions. Finally, the same formula must not occur twice with different foci. In this case, the older one replaces the newer one. We will assume that the rules introducing new and collapsing two existing foci will always be played immediately if possible.

Let $\mathtt{foc}(\Phi) := \{\ i \mid \text{there is a } \psi \text{ with } \left[\psi\right]^i \in \Phi\ \}$ be the set of all focus indices occurring in Φ.

Player $\exists$ wins the play $C_1, \ldots, C_n$ iff

(i) $C_n = q_1, \ldots, q_k$ and for all $i = 1, \ldots, k$: $\overline{q_i} \notin C_n$ and $q_i \neq \mathtt{ff}$,

(ii) there is an $i < n$, s.t. $C_i \approx C_n$ and $\mathtt{foc}(C_i) \cap \mathtt{foc}(C_n) = \emptyset$.

Player $\forall$ wins the play $C_1, \ldots, C_n$ iff

(iii) $\mathtt{ff} \in C_n$ or there is a $q \in \mathcal{P}$ s.t. $q \in C_n$ and $\overline{q} \in C_n$, or

(iv) there is an $i < n$, s.t. $C_i \approx C_n$ and $\mathtt{foc}(C_i) \cap \mathtt{foc}(C_n) \neq \emptyset$.

Note that the definition of $\approx$ ensures that only configurations are compared in which all U formulas and their unfoldings have a focus.

Lemma 4.1 *Every play of $\mathcal{G}(\varphi)$ has length at most $2^{|\varphi|+1}$.*

Proof. Note that there are only $2^{|\varphi|+1} = 2 \cdot 2^{|\varphi|}$ many different configurations modulo foci indices since there are only $2^{|\varphi|}$ many different sets of subformulas of φ. Moreover, introducing foci can at most double the length of a play. □

Lemma 4.2 *Every play of $\mathcal{G}(\varphi)$ has a unique winner.*

Proof. According to Lemma 4.1, every play has a winner. Furthermore, winning conditions (i) and (iii), as well as (ii) and (iv) are mutually exclusive. Besides, if a play is won with condition (i) or (iii), then the last configuration C_n cannot have a companion C_i as in winning condition (ii) or (iv). Otherwise the play would have been finished at C_i already. □

Definition 4.3 Let $P = C_0, C_1, \ldots, C_n$ be a play of a game $\mathcal{G}(\varphi_0)$. Take a $\varphi \mathtt{U} \psi \in \mathit{Sub}(\varphi_0)$. It is called *persisting* for P iff there is an i with $0 \leq i \leq n$, s.t. for all $j < i$: $C_j \cap \mathtt{unf}(\varphi \mathtt{U} \psi) = \emptyset$ and for all $j \geq i$: $C_j \cap \mathtt{unf}(\varphi \mathtt{U} \psi) \neq \emptyset$.

Lemma 4.4 *A play P of $\mathcal{G}(\varphi_0)$ is won by player $\forall$ with his winning condition* (iv) *iff there is a $\varphi \mathtt{U} \psi \in \mathit{Sub}(\varphi_0)$ which is persisting for P.*

Proof. Suppose that there is a persisting U in P. Then the moment it becomes top-level in a configuration it will receive a focus i. Furthermore, since it persists, it retains its focus. Note that another unfolding of it can obtain a different focus index j later on. But then $j > i$. Moreover, before rule (X) can

be applied, both unfoldings will have been unified and index j will be replaced by i. Therefore, whenever a repeat on a configuration occurs, focus i will have survived and, hence, player $\forall$'s winning condition (iv) will apply.

Conversely, if it applies then there is a focus index which has survived. But this can only be if once it was given to an $\mathtt{U}$ formula or one of its unfoldings and then remained on it. Note that it is always the right path in a syntax tree of an $\mathtt{U}$'s unfolding which leads back to it. Furthermore, the focus is always passed on to the right successor in its syntax tree. Thus, this $\mathtt{U}$ is persisting in the play at hand. □

Theorem 4.5 (Soundness) *If player $\exists$ has a winning strategy for $\mathcal{G}(\varphi_0)$ then φ_0 is satisfiable.*

Proof. If player $\exists$ has a winning strategy then there is at least one play won by her. A model π for φ_0 can easily be extracted from this play. Its states are sets of configurations separated by applications of rule ($\mathtt{X}$). A simple induction on the structure of formulas shows that every formula in a configuration is fulfilled by the suffix of π beginning with its corresponding state. In particular, $\pi \models \varphi_0$. Lemma 4.4 shows that every occurring $\mathtt{U}$ formula becomes fulfilled eventually. By assumption, no $\mathtt{U}$ is persisting for this play. Hence, for every $\varphi \mathtt{U} \psi$, player $\exists$ must have chosen ψ at some point. By hypothesis, it is fulfilled on the corresponding suffix. □

Theorem 4.6 (Completeness) *If φ_0 is satisfiable then player $\exists$ has a winning strategy for $\mathcal{G}(\varphi_0)$.*

Proof. Suppose π is a model for φ. Then player $\exists$ can follow the game $\mathcal{G}(\varphi_0)$ in π. I.e. starting with $\pi^{(0)}$ she annotates configurations with a state. Each time rule ($\mathtt{X}$) is played, she moves to the next state in π. If a disjunction comes up she chooses the disjunct which is fulfilled by the suffix of π beginning with the actual state. If both are fulfilled she chooses the smaller one.

The following invariant holds: in a configuration Φ annotated with $\pi^{(i)}$ we have $\pi^i \models \varphi$ for all $\varphi \in \Phi$. Suppose now that the resulting play is won by player $\forall$. This cannot be with winning condition (iii) since no suffix of π can satisfy both q and $\overline{q}$ for some $q \in \mathcal{P}$. Suppose he wins it with winning condition (iv). According to Lemma 4.4, the play must have a persisting formula of the form $\varphi \mathtt{U} \psi$. While it persisted, player $\exists$ has never chosen ψ in its unfolding. By the definition of her strategy, ψ was never satisfied by any suffix of π so far. Note that this play of $\mathcal{G}(\varphi_0)$ could be continued ad infinitum showing that ψ is never fulfilled on any suffix of π. Thus, $\varphi \mathtt{U} \psi$ cannot have been satisfied contradicting the invariant. □

Corollary 4.7 ***(Small model property)*** *If $\varphi \in LTL$ is satisfiable then it*

has a model of size $O(2^{|\varphi|})$.

Proof. Directly from Theorem 4.5 and Lemma 4.1. □

Theorem 4.8 ***(Complexity)*** *Deciding the winner of* $\mathcal{G}(\varphi)$ *is in PSPACE.*

Proof. A nondeterministic algorithm only needs to store two configurations – the actual one that gets overwritten with each application of a game rule, and a C_i s.t. player $\exists$ believes that a C_j will occur later on with $C_i \approx C_j$. Furthermore, the algorithm needs to store a counter in order to abort the play in case it has not found a repeat. The size of a configuration is polynomial in the size of the input – note that Lemma 4.1 bounds the value of the foci index component by $2^{|\varphi|+1}$ – and so is the size of the counter. Applying Savitch's Theorem [11] yields the claim. □

Example 4.9 We will define a family $\varphi_{n,k}$ of unsatisfiable LTL formulas for every $n, k \geq 1$ exhibiting the foci games' advantage over focus games. The length of a longest play in the game tree for $\varphi_{n,k}$ will be longer by a factor $O(n)$ for the focus games compared to the foci games. Thus, n measures the gap between the two sorts of games; k acts as a an amplifier.

Let $\mathcal{P} = \{p_0, \ldots, p_n\}$ and define abbreviations $P!_n := \bigvee_{i=0}^{n-1}(p_i \wedge \bigwedge_{j \neq i} \neg p_j)$ and $\bar{P}_n := \bigwedge_{i=0}^{n} \neg p_i$ saying that exactly one proposition other than p_n, resp. none holds. Let

$$\psi^1_{n,k} \;:=\; \Big(\big(P!_n \wedge \bigwedge_{i=1}^{k-1} \mathtt{X}^i \bar{P}_n\big) \rightarrow \mathtt{X}^k P!_n \Big) \;\wedge\; \bigwedge_{i=1}^{k-1} \Big(\mathtt{X}^i P!_n \wedge \bigwedge_{j \neq i} \mathtt{X}^j \bar{P}_n \;\rightarrow\; \mathtt{X}^k \bar{P}_n \Big)$$

say: if among k successive states there is only one in which one of the propositions $\mathcal{P} \setminus \{p_n\}$ holds, then this pattern gets repeated ad infinitum, i.e. only every k-th state satisfies one of these propositions. Finally, let

$$\psi^0_{n,k} \;:=\; \bigwedge_{i=0}^{k-1} \mathtt{X}^i \bar{P}_n \;\wedge\; \mathtt{X}^k P!_n$$

say that the first $k-1$ states of a linear time structure are labelled with no proposition and the k-th state is labelled with one of $\mathcal{P} \setminus \{p_n\}$. Note that $\psi_0 \wedge G\psi_1$ expresses that one of these propositions holds exactly in every k-th state. Now we are able to define

$$\varphi_{n,k} \;:=\; \psi^0_{n,k} \wedge \mathtt{G}(\psi^1_{n,k} \wedge \bigwedge_{i=0}^{n} \mathtt{F} p_i)$$

Note that $\varphi_{n,k}$ is unsatisfiable for every $n, k \in \mathbb{N}$ because $\psi_{n,k,0} \wedge G\psi_{n,k,1}$ requires that p_n holds nowhere, i.e. $\mathtt{F}p_n$ can never become fulfilled.

The game trees for both $\Gamma(\varphi_{n,k})$ and $\mathcal{G}(\varphi_{n,k})$ are too big to be depicted here – even for small n and k. However, note that the maximal nesting depth of $\mathtt{X}$ operators in $\varphi_{n,k}$ is k. This plus the fact that the outermost G regenerates the pattern defining $\psi_{n,k,1}$ and all the formulas $\mathtt{F}p_i$ after each application of rule ($\mathtt{X}$) ensures that after $O(k)$ applications of rule ($\mathtt{X}$), a repeat on a configuration is found if the position of the focus is ignored.

Since the foci game $\mathcal{G}(\varphi_{n,k})$ focuses on all $\mathtt{U}$ formulas – i.e. all $\mathtt{F}$ formulas in this case – in parallel, it is able to show that $\mathtt{F}p_n$ does not get fulfilled. Thus player $\forall$ wins each play after at most k applications of rule ($\mathtt{X}$).

Now consider the focus game $\Gamma(\varphi_{n,k})$. W.l.o.g. assume that at the beginning player $\forall$ has fixed his priority list as $[\mathtt{F}p_0, \mathtt{F}p_1, \ldots, \mathtt{F}p_n]$. Then he will focus on each of them in this order. The game tree for player $\forall$ will contain all of player $\exists$'s possible choices, in particular those she takes after every k applications of rule ($\mathtt{X}$) in order to choose a proposition out of $\mathcal{P} \setminus \{p_n\}$. Thus, there is a play in which she chooses p_0 after k applications of rule ($\mathtt{X}$), p_1 after $2k$ applications, etc. until p_{n-1} after $n \cdot k$ applications. But then player $\forall$ will have focused on $\mathtt{F}p_0$ for k steps, then on $\mathtt{F}p_1$ for another k steps, etc. Eventually, after $n \cdot k$ applications of rule ($\mathtt{X}$) he will focus on $\mathtt{F}p_n$ which does not get fulfilled and win the play after k steps.

Retrieving winning strategies for focus games. We will explain how to obtain winning strategies for $\Gamma(\varphi)$ from those of $\mathcal{G}(\varphi)$. This is easy for player $\exists$. She simply makes the same choices in both games.

The situation is different for player $\forall$. He cannot simply leave the focus on a formula and change it to the $\mathtt{U}$ that makes him win a play in the moment it becomes top-level because it is player $\exists$ who chooses the play subsequently. Thus, he cannot know which play is going to be played.

Definition 4.10 A formula of the form $\varphi\mathtt{U}\psi$ is called winning for a play P if P is won by player $\forall$ with his winning condition (iv), and $\varphi\mathtt{U}\psi$ is persisting in P and has the least focus index among the persisting ones.

Take the full tree of all plays in $\mathcal{G}(\varphi_0)$ for a $\varphi_0 \in$ LTL. Every configuration C can be annotated with the set of $\mathtt{U}$ formulas that are winning for a play containing C. In $\Gamma(\varphi_0)$ player $\forall$ can now set the focus to the $\varphi\mathtt{U}\psi$ in C that is contained in C's annotations and has least focus index there.

Note that a configuration can have several $\mathtt{U}$ formulas in its annotation and the one chosen by player $\forall$ might be winning for a different play than the one that player $\exists$ reveals later on. In this case, player $\forall$ has to change focus in this moment. This can prolong the play P at hand in $\Gamma(\varphi_0)$ compared to

the one in $\mathcal{G}(\varphi_0)$. Thus even if P repeats on C in $\mathcal{G}(\varphi_0)$ it might not in $\Gamma(\varphi_0)$. However, continuing to play according to the rules will ultimately result in a win for player $\forall$. Each time the play in $\mathcal{G}(\varphi)$ needs to be prolonged in this way, the set of annotated $\mathtt{U}$ formulas gets smaller which eventually means that player $\forall$ does not have to change focus in $\Gamma(\varphi)$ anymore.

5 Foci Games for LTL Model Checking

The model checking problem for LTL is usually defined via transition systems rather than linear time structures. A (labelled) total transition system $\mathcal{T}$ over a set $\mathcal{P}$ of atomic propositions is a tuple $(\mathcal{S}, \rightarrow, \lambda)$ where $(\mathcal{S}, \rightarrow)$ is a graph s.t. for every $s \in \mathcal{S}$ there is at least one $t \in \mathcal{S}$ with $s \rightarrow t$.

$\lambda : \mathcal{S} \rightarrow 2^{\mathcal{P}}$ labels the vertices (also called states) in a maximally consistent manner. Note that every infinite path through $\mathcal{T}$ induces a linear time structure π. Then the model checking problem for LTL and transition systems is: given an LTL formula φ and a $\mathcal{T}$ with state s, does $\pi \models \varphi$ hold for every linear time structure π from $\mathcal{T}$ starting in s? We will write this succinctly as $\mathcal{T}, s \models \varphi$ or even $s \models \varphi$ if the underlying transition system can be derived from the context.

Remember that LTL satisfiability and its model checking are closely related: both are complete for PSPACE, both can be reduced to the emptiness problem for nondeterministic Büchi automata and both have similar game-theoretic characterisations in terms of focus games. However, they are dual to each other: satisfiability focus games naturally lead to a nondeterministic [6], model checking games to a universal procedure [7]. From a complexity-theoretic point of view, this is of course the same [11].

From a game-theoretic point of view, the roles of the players are swapped. It is player $\forall$ who becomes active while player $\exists$ is left with nothing to choose.

The intuition behind this is as follows. Note that in the satisfiability foci games conjunctions are preserved and disjunctions are chosen because the latter is satisfiable iff one of its disjuncts is satisfiable. The former however is satisfiable if both conjuncts do not contradict each other. In the model checking game player $\forall$ wants to refute $s \models \varphi$. Hence, he has to name a path π in the underlying transition system $\mathcal{T}$ s.t. $\pi \not\models \varphi$. Since the number of paths in a transition system can easily become exponential in the number of its states we want player $\forall$ to select this path pointwise, i.e. reveal a new state only when the formula requires to do so. But then player $\exists$ cannot choose disjuncts anymore. The choice of which disjunct is fulfilled depends on which path player $\forall$ chooses. However, he would only do this after player $\exists$ has committed to a particular disjunct. Thus, the games would not be sound

$$\frac{s \vdash \left[\psi_0 \wedge \psi_1\right]^k, \Phi; c}{s \vdash \psi_0, \Phi; c \qquad s \vdash \left[\psi_1\right]^k, \Phi; c}\ \forall \qquad \frac{s \vdash \psi_0 \wedge \psi_1, \Phi; c}{\psi_0, \Phi; c \qquad \psi_1, \Phi; c}\ \forall$$

$$\frac{s \vdash \left[\psi_0 \vee \psi_1\right]^k, \Phi; c}{s \vdash \psi_0, \left[\psi_1\right]^k, \Phi; c} \qquad \frac{s \vdash \psi_0 \vee \psi_1, \Phi; c}{s \vdash \psi_0, \psi_1, \Phi; c}$$

$$\frac{s \vdash \varphi, \Phi; c}{s \vdash \left[\varphi\right]^c, \Phi; c+1}\ \text{if } \varphi \in \mathtt{REL}(\varphi_0) \qquad \frac{s \vdash \left[\varphi\right]^i, \left[\varphi\right]^k, \Phi; c}{s \vdash \left[\varphi\right]^{\min\{i,k\}}, \Phi; c}$$

$$\frac{s \vdash \left[\varphi \mathtt{R} \psi\right]^k, \Phi; c}{s \vdash \left[\psi \wedge (\varphi \vee \mathtt{X}(\varphi \mathtt{R} \psi))\right]^k, \Phi; c} \qquad \frac{s \vdash \varphi \mathtt{U} \psi, \Phi; c}{s \vdash \psi \vee (\varphi \wedge \mathtt{X}(\varphi \mathtt{U} \psi)), \Phi; c}$$

$$(\mathtt{X})\ \frac{s \vdash \left[\mathtt{X}\psi_1\right]^{i_1}, \ldots, \left[\mathtt{X}\psi_m\right]^{i_m}, \mathtt{X}\varphi_1, \ldots, \mathtt{X}\varphi_n, q_1, \ldots, q_k; c}{t \vdash \left[\psi_1\right]^{i_1}, \ldots, \left[\psi_m\right]^{i_m}, \varphi_1, \ldots, \varphi_n; c}\ \forall s \to t$$

Fig. 3. The rules for the LTL model checking foci games.

anymore. For a detailed example cf. [7].

Given a transition system $\mathcal{T} = (\mathcal{S}, \to, \lambda)$, a distinguished state s_0 and an LTL formula φ_0, the LTL model checking foci game $\mathcal{G}_{\mathcal{T}}(s_0, \varphi_0)$ is played by players $\forall$ and $\exists$ in order to determine whether or not $s_0 \models \varphi_0$ holds. Configurations are written as $t \vdash \Phi; c$ where $t \in \mathcal{S}$, $\Phi \subseteq \mathit{Sub}^*(\varphi_0)$ possibly equipped with foci and $c \in \mathbb{N}$. Φ is interpreted disjunctively. The starting configuration is $s_0 \vdash \varphi_0; 0$, and the rules are presented in Figure 3.

We redefine the relation $\approx$ on configurations as: $C_i \approx C_j$ if (1) they are equal as sets of formulas disregarding the states, the foci and the counters, and (2) for every $\chi \in \mathtt{REL}(\varphi_0)$: χ has a focus in C_i and in C_j. The following is not hard to see.

Lemma 5.1 *$\approx$ is an equivalence relation.*

Player $\exists$ wins the play $C_0, C_1, \ldots$ iff

(i) there is an $n \in \mathbb{N}$ s.t. $C_n = t \vdash q, \Phi; c$ and $q \in \lambda(t)$, or

(ii) there are infinitely many $i_j \in \mathbb{N}$ s.t. for all $j, k \in \mathbb{N}$: $C_{i_j} \approx C_{i_k}$ and $\bigcap_{j \in \mathbb{N}} \mathtt{foc}(C_{i_j}) \neq \emptyset$.

Player $\forall$ wins the play $C_0, C_1, \ldots$ iff

(iii) there is an $n \in \mathbb{N}$ s.t. $C_n = t \vdash q_1, \ldots, q_k; c$ and for all $i = 1, \ldots, k$: $q_i \notin \lambda(t)$, or

(iv) there are infinitely many $i_j \in \mathbb{N}$ s.t. for all $j, k \in \mathbb{N}$: $C_{i_j} \approx C_{i_k}$ and $\bigcap_{j \in \mathbb{N}} \mathtt{foc}(C_{i_j}) = \emptyset$.

Lemma 5.2 *Every play has a unique winner.*

Proof. This is proved in the same way as Lemma 4.2 but uses Lemma 5.1 as well since the foci of infinitely many configurations are taken into account. □

Theorem 5.3 *If $\mathcal{T}, s_0 \models \varphi_0$ then player $\exists$ wins $\mathcal{G}_\mathcal{T}(s_0, \varphi_0)$.*

Proof. We call a configuration $t \vdash \Phi$ *true* if for all paths π starting with t there is a $\varphi \in \Phi$ s.t. $\pi \models \varphi$. Note that the starting configuration of $\mathcal{G}_\mathcal{T}(s_0, \varphi_0)$ is true. Furthermore, the rules preserve truth. The only exception is rule (X) which only preserves truth if the formula satisfying a given path is not an atomic proposition. However, if this is the case then rule (X) will not get played since player $\exists$'s winning condition (i) would apply beforehand. Thus, we can assume every configuration of $\mathcal{G}_\mathcal{T}(s_0, \varphi_0)$ to be true.

We call a formula φ a *witness* for a true configuration $t \vdash \Phi$ if it is syntactically smallest among those that are satisfied by a path starting in t. Note that only the rules for unfolding a R or an U do not decrease the size of a witness. However, if the witness in a configuration is an U then its size will eventually get decreased – recall the semantics of U.

Thus the witness becomes smaller and smaller until it finally is either a proposition or remains a R, resp. its unfoldings. In the first case, the play is won by player $\exists$ with her winning condition (i). In the second case the R will get a focus index and keep it, according to a variant of Lemma 4.4. Hence, the play will be won by player $\exists$ with her winning condition (ii). Altogether, player $\exists$ has a winning strategy. □

Theorem 5.4 *If $\mathcal{T}, s_0 \not\models \varphi_0$ then player $\forall$ wins $\mathcal{G}_\mathcal{T}(s_0, \varphi_0)$.*

Proof. Suppose $\mathcal{T}, s_0 \not\models \varphi_0$, i.e. there is a path π starting with s_0 s.t. $\pi \not\models \varphi_0$. Player $\forall$'s strategy consists of two parts. Each time rule (X) is played he selects the next state of π. If a conjunct needs to be chosen he picks the one that is not satisfied on the remaining suffix of π. This guarantees that the following invariant holds in all configurations of the outlined play: In a configuration $\pi^{(k)} \vdash \Phi$ we have $\pi^k \not\models \varphi$ for all $\varphi \in \Phi$.

Suppose now that player $\exists$ wins this play. Winning condition (i) is impossible since it immediately contradicts this invariant. But in order for her to win with winning condition (ii) there would have to be a persisting $\varphi \mathtt{R} \psi$

according to a variant of Lemma 4.4. But then we can replace the first occurrence of $\varphi \mathtt{R} \psi$ by $\varphi \mathtt{R}^m \psi$ according to Lemma 2.1. Since the game rules follow the approximants the play would contain a configuration in which $\varphi \mathtt{R} \psi$ is interpreted as $\varphi \mathtt{R}^0 \psi$ which also contradicts the invariant. Hence, player $\forall$'s strategy is winning. □

Theorem 5.5 *For finite $\mathcal{T}$ deciding the winner of $\mathcal{G}_{\mathcal{T}}(s_0, \varphi)$ is in PSPACE.*

Proof. The proof is similar to the proof of Theorem 4.8. Note that configurations of the model checking games extend those of the satisfiability games by a state. This only needs additional logarithmic space. Furthermore, if $\mathcal{T}$ is finite then the winner of a play is already determined after $|\mathcal{T}| \cdot 2^{|\varphi|+1}$ steps since this is an upper bound on the number of different configurations possibly occurring in $\mathcal{G}_{\mathcal{T}}(s_0, \varphi)$. Then the winner can be decided by a nondeterministic algorithm using polynomial space only. With Savitch's Theorem [11], it is in PSPACE. □

References

[1] G. Bhat, R. Cleaveland, and O. Grumberg. Efficient on-the-fly model checking for CTL*. In *Proc. 10th Symp. on Logic in Computer Science, LICS'95*, pages 388–397, San Diego, CA, USA, June 1995. IEEE.

[2] A. Biere, A. Cimatti, E. M. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In R. Cleaveland, editor, *Proc. 5th Int. Conf. on Tools and Algorithms for the Analysis and Construction of Systems, TACAS'99*, volume 1579 of *LNCS*, Amsterdam, NL, March 1999.

[3] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, January 1981.

[4] E. Clarke, O. Grumberg, and K. Hamaguchi. Another look at LTL model checking. *LNCS*, 818:415–427, 1994.

[5] M. Fisher. A resolution method for temporal logic. In J. Mylopoulos and R. Reiter, editors, *Proc. 12th Joint Conf. on Artificial Intelligence*, pages 99–104, Sydney, Australia, August 1991. Morgan Kaufmann.

[6] M. Lange and C. Stirling. Focus games for satisfiability and completeness of temporal logic. In *Proc. 16th Symp. on Logic in Computer Science, LICS'01*, Boston, MA, USA, June 2001. IEEE.

[7] M. Lange and C. Stirling. Model checking games for branching time logics. *Journal of Logic and Computation*, 12(4):623–639, 2002.

[8] O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proc. 12th Symp. on Principles of Programming Languages, POPL'85*, pages 97–107, New York, January 1985. ACM.

[9] R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65(2):149–184, December 1993.

[10] A. Pnueli. The temporal logic of programs. In *Proc. 18th Symp. on Foundations of Computer Science, FOCS'77*, pages 46–57, Providence, RI, USA, October 1977. IEEE.

[11] W. J. Savitch. Deterministic simulation of nondeterministic Turing Machines. In *Symp. on Theory of Computing, STOC'69*, pages 247–248, New York, May 1969. ACM.

[12] P. H. Schmitt and J. Goubault-Larrecq. A tableau system for linear time temporal logic. In *Proc. 3rd Workshop on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'97*, volume 1217 of *LNCS*, pages 130–144. Springer, Enschede, Netherlands, April 1997.

[13] A. P. Sistla, M. Y. Vardi, and P. Wolper. Reasoning about infinite computation paths. In *Proc. 24th Symp. on Foundations of Computer Science, FOCS'83*, pages 185–194, Los Alamitos, Ca., USA, November 1983. IEEE.

[14] M. Y. Vardi. *An Automata-Theoretic Approach to Linear Temporal Logic*, volume 1043 of *LNCS*, pages 238–266. Springer, New York, NY, USA, 1996.