Model Checking Games for CTL*

Martin Lange and Colin Stirling

LFCS, Division of Informatics
The University of Edinburgh

email: {martin,cps}@dcs.ed.ac.uk

August 2000

Abstract

We define model checking games for the temporal logic **CTL*** and prove their correctness. They provide a technique for using model checking interactively in a verification/specification process. Their main feature is to construct paths in a transition system stepwise. That enables them to be the basis for a local model checking algorithm with a natural notion of justification. However, this requires configurations of a game to contain sets of formulas. Moreover, an additional structure on these sets, called focus, has to be used to guarantee correctness.

1 Introduction

Model checking is a useful and broadly accepted technique for verifying parallel processes. The system to be examined is abstracted into a mathematical interpretation for a logical formula which formalises a property the system is expected to have or to lack. A model checking algorithm decides whether the system's abstraction fulfils the formula and thus whether the system meets its specification given by the formula, provided that the abstraction is correct. We will not discuss the finding of good abstractions at all, instead we are interested in checking properties of the abstraction only. Hence, in the following, the term "system" will denote the abstraction as well.
However, verification of concurrent systems is often combined with specification in the framework of developing them. For such a process a simple yes/no answer to the question whether a system is correct w.r.t. a certain property is not sufficient. Moreover, techniques that show why or where the property is violated are required.
Model checking games being played by two players on the system and the formula provide such features. Answering the question about the property being fulfilled turns out to be equivalent to finding a winning strategy for one of the players. Once such a strategy is found, i.e. computed by a verification tool for example, it can be used to enable an interactive play between the tool and the developer.
There are various classes of interpretations which are suitable for modelling a temporal behaviour. We will deal with transition systems only.
Furthermore, there also are various logics that allow the formalisation of temporal properties over transition systems. **CTL*** (cf. [4]) is not just one of them but probably the most appropriate one for expressing temporal properties. The linear time logic **LTL** (cf. [7]) and the branching time logic **CTL** (cf. [1]) for example can be found as genuine syntactic fragments of **CTL***. A lot of interesting properties, like "something holds infinitely often", cannot be expressed in **CTL** but in **CTL***. **LTL** is capable of doing this, but cannot formalise the existence of a certain sequence of states in the system.

On the other hand, **CTL*** can be translated into the modal μ-calculus $\mathcal{L}_\mu$ (cf. [5]), for which such model checking games already exist (cf. [8, 3]). However, the alternation depth of the resulting μ-calculus formulas is bounded by two (cf. [2]). Since model checking for **CTL*** is PSPACE-complete (cf. [6]), whereas there exists a polynomial time algorithm solving the model checking problem for that fragment of $\mathcal{L}_\mu$, the translation procedure must enlarge the formulas or the transition systems exponentially, unless P=PSPACE. Although this is not necessarily bad since formulas can be assumed to be small, it is undesirable for the mentioned specification and verification process because it lacks the subformula property for an interactive play: All formulas occurring in the game should be subformulas of the formalised property. This enables the user of a verification tool best to understand the diagnosis of the underlying system that is provided by the interactive play.
In the remaining sections we will recall the syntax and semantics of **CTL***, define the model checking games and prove their correctness. There is a fairly simple way of defining games for **CTL*** which follows exactly the semantics. I.e. whenever a path formula is reached the corresponding player names a whole path on which the formula is examined. However, it is easy to give examples in which the length of a shortest path to be chosen is at least as great as the size of the transition system. Moreover, since an algorithm might have to examine all the possible choices the players can take, this feature would contradict the idea of a local algorithm. Therefore we require paths in the transition system to be constructed stepwise throughout the game.

2 Syntax and Semantics

Let *Prop* be a set of propositional constants including *true* and *false*, which is closed under complementary propositions, i.e. $Prop = \{\mathtt{tt}, \mathtt{ff}, Q_1, \overline{Q_1}, \dots\}$. A *transition system* $\mathcal{T}$ is a triple (S, T, L) with (S, T) being a directed graph. $L : S \to 2^{Prop}$ labels the states, s.t. for all $s \in S$: $\mathtt{tt} \in L(s), \mathtt{ff} \notin L(s)$ and $Q \in L(s)$ iff $\overline{Q} \notin L(s)$. We write $s \to t$ for $s, t \in S$ and $(s, t) \in T$, and assume that every state in the graph has at least one successor.

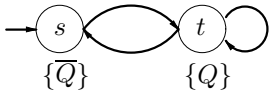

Figure 1: A transition system.

Definition 2.1 Let $Q \in Prop$. The logic **CTL*** is defined by

$$\varphi \quad ::= \quad Q \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X\varphi \mid \varphi U \varphi \mid \varphi R \varphi \mid E\varphi \mid A\varphi$$

The set of subformulas $Sub(\varphi)$ for a given φ is defined as follows:

$$\begin{array}{rcl}
Sub(Q) & := & \{Q\} \\
Sub(\varphi \wedge \psi) & := & \{\varphi \wedge \psi\} \cup Sub(\varphi) \cup Sub(\psi) \\
Sub(\varphi \vee \psi) & := & \{\varphi \vee \psi\} \cup Sub(\varphi) \cup Sub(\psi) \\
Sub(E\varphi) & := & \{E\varphi\} \cup Sub(\varphi) \\
Sub(A\varphi) & := & \{A\varphi\} \cup Sub(\varphi) \\
Sub(X\varphi) & := & \{X\varphi\} \cup Sub(\varphi) \\
Sub(\varphi U \psi) & := & \{\varphi U \psi, X(\varphi U \psi), \varphi \wedge X(\varphi U \psi), \psi \vee (\varphi \wedge X(\varphi U \psi))\} \cup Sub(\varphi) \cup Sub(\psi) \\
Sub(\varphi R \psi) & := & \{\varphi R \psi, X(\varphi R \psi), \varphi \vee X(\varphi R \psi), \psi \wedge (\varphi \vee X(\varphi R \psi))\} \cup Sub(\varphi) \cup Sub(\psi)
\end{array}$$

Not surprisingly, we set $Sub(\Phi) := \bigcup_{\varphi\in\Phi} Sub(\varphi)$.

Definition 2.2 The semantics of a **CTL*** formula is explained using paths $\pi = s_0 s_1 \ldots s_n \ldots$ of a transition system. With $\pi^{(i)}$ we denote the suffix of π beginning with the state s_i.

- $\mathcal{T}, \pi \models Q$ iff $Q \in L(s_0)$
- $\mathcal{T}, \pi \models \varphi \wedge \psi$ iff $\mathcal{T}, \pi \models \varphi$ and $\mathcal{T}, \pi \models \psi$
- $\mathcal{T}, \pi \models \varphi \vee \psi$ iff $\mathcal{T}, \pi \models \varphi$ or $\mathcal{T}, \pi \models \psi$
- $\mathcal{T}, \pi \models A\varphi$ iff for all paths $\sigma = s_0\sigma'$: $\mathcal{T}, \sigma \models \varphi$
- $\mathcal{T}, \pi \models E\varphi$ iff there exists a path $\sigma = s_0\sigma'$ and $\mathcal{T}, \sigma \models \varphi$
- $\mathcal{T}, \pi \models X\varphi$ iff $\mathcal{T}, \pi^{(1)} \models \varphi$
- $\mathcal{T}, \pi \models \varphi U \psi$ iff there exists $i \in \mathbb{N}$ s.t. $\mathcal{T}, \pi^{(i)} \models \psi$ and for all $j < i : \mathcal{T}, \pi^{(j)} \models \varphi$
- $\mathcal{T}, \pi \models \varphi R \psi$ iff for all $i \in \mathbb{N} : \mathcal{T}, \pi^{(i)} \models \psi$ or there exists a $j \leq i$ s.t. $\mathcal{T}, \pi^{(j)} \models \varphi$

φ and ψ are *logically equivalent*, $\varphi \equiv \psi$, if for all transition systems and paths $\mathcal{T}, \pi \models \varphi$ iff $\mathcal{T}, \pi \models \psi$ is true. A **CTL*** formula φ is called a *state formula* if $\varphi \equiv A\varphi$ holds. Clearly, every formula of the form $A\varphi$ or $E\varphi$ is a state formula, and every state formula can be brought into one of these forms, too. In the following we will regard state formulas only. Hence we may write $\mathcal{T}, s_0 \models \varphi$ if we mean $\mathcal{T}, \pi \models \varphi$. Formulas not being state formulas are called *path formulas*. They will still occur as subformulas of state formulas. For the rest of the paper we fix a transition system $\mathcal{T} = (S, T, L)$ and take the freedom to write $s \models \varphi$ and $\pi \models \varphi$ for $\mathcal{T}, s \models \varphi$ and $\mathcal{T}, \pi \models \varphi$.

3 Model Checking Games

In order to introduce games we need two players, namely *player* I and *player* II.[1] If p is one of them then $\overline{p}$ denotes the other one. It is player II's task to show that a formula is satisfied whereas player I tries to show the converse.
The set of *configurations* for a transition system $\mathcal{T}$ and a formula φ is $Conf(\mathcal{T}, \varphi) \subseteq \{\mathrm{I}, \mathrm{II}\} \times S \times Sub(\varphi) \times 2^{Sub(\varphi)}$. A configuration is written $p, s \vdash [\psi], \Phi$ where p is a player called the *path player*, $s \in S$, $\psi \in Sub(\varphi)$ and $\Phi \subseteq Sub(\varphi)$.[2] In this case ψ is said to be *in focus*. The main idea is to build a path stepwise from transitions, and the path player is the one who will take the choice of the next transition. Her opponent is in control of the focus and thus also called the *focus player*.
A *play* between player I and player II is a sequence of configurations. There are nineteen rules of the form

$$\frac{p, s \vdash [\varphi], \Phi}{p', s' \vdash [\varphi'], \Phi'} \; p''$$

for transforming configurations. They are to be read as: *If the actual configuration is* $p, s \vdash [\varphi], \Phi$ *then player* p'' *has to perform a choice and the next configuration is* $p', s' \vdash [\varphi'], \Phi'$.

[1] In the following "he" will stand for player I whereas "she" will be a synonym for either player II or both.

[2] We may write $p, s \vdash \Phi$ if we speak about a configuration without explicitly referring to the focus formula.

The *side formulas*, i.e. those that are not in focus, can be seen as an insurance for the path player's opponent to redo a move that she has done before. This is necessary because the path player chooses the path stepwise along which a formula is examined.
At each configuration the set of side formulas together with the formula in focus can be understood as a disjunction (resp. conjunction) of formulas in case the path player is player I (resp. II).
A play for a transition system $\mathcal{T}$ with starting state s and a formula φ begins with the configuration $\mathrm{I}, s \vdash [\varphi]$.[3] From then on, the play proceeds according to the following rules. Once the focus is on a quantified formula a new path has to be chosen. Thus, all current side formulas do not matter anymore. Furthermore, the next path player is determined.

$$(1)\ \frac{p, s \vdash [A\varphi], \Phi}{\mathrm{I}, s \vdash [\varphi]} \qquad (2)\ \frac{p, s \vdash [E\varphi], \Phi}{\mathrm{II}, s \vdash [\varphi]}$$

An explicit state formula can also be discarded in case the focus player does not want to prove (resp. refute) it in the actual state.

$$(3)\ \frac{p, s \vdash [\varphi], A\psi, \Phi}{p, s \vdash [\varphi], \Phi}\ \overline{p} \qquad (4)\ \frac{p, s \vdash [\varphi], E\psi, \Phi}{p, s \vdash [\varphi], \Phi}\ \overline{p} \qquad (5)\ \frac{p, s \vdash [\varphi], Q, \Phi}{p, s \vdash [\varphi], \Phi}\ \overline{p}$$

The four rules for a boolean connective in focus are almost straightforward. Note that it is not necessary to keep both disjuncts for example if the path player is player II because apparently she knows which path she is going to choose.

$$(6)\ \frac{\mathrm{I}, s \vdash [\varphi_0 \wedge \varphi_1], \Phi}{\mathrm{I}, s \vdash [\varphi_i], \Phi}\ \mathrm{I} \qquad (7)\ \frac{\mathrm{I}, s \vdash [\varphi_0 \vee \varphi_1], \Phi}{\mathrm{I}, s \vdash [\varphi_i], \varphi_{1-i}, \Phi}\ \mathrm{II}$$

$$(8)\ \frac{\mathrm{II}, s \vdash [\varphi_0 \vee \varphi_1], \Phi}{\mathrm{II}, s \vdash [\varphi_i], \Phi}\ \mathrm{II} \qquad (9)\ \frac{\mathrm{II}, s \vdash [\varphi_0 \wedge \varphi_1], \Phi}{\mathrm{II}, s \vdash [\varphi_i], \varphi_{1-i}, \Phi}\ \mathrm{I}$$

The temporal operators U and R simply are unfolded.

$$(10)\ \frac{p, s \vdash [\varphi U \psi], \Phi}{p, s \vdash [\psi \vee (\varphi \wedge X(\varphi U \psi))], \Phi} \qquad (11)\ \frac{p, s \vdash [\varphi R \psi], \Phi}{p, s \vdash [\psi \wedge (\varphi \vee X(\varphi R \psi))], \Phi}$$

Now, applying those rules might generate an $X\psi$ formula in focus. Before a play can proceed with that all side formulas have to be brought into this form, too. The rules for this are very similar to the ones above.

$$(12)\ \frac{\mathrm{I}, s \vdash [X\psi], \varphi_0 \wedge \varphi_1, \Phi}{\mathrm{I}, s \vdash [X\psi], \varphi_i, \Phi}\ \mathrm{I} \qquad (13)\ \frac{\mathrm{I}, s \vdash [X\psi], \varphi_0 \vee \varphi_1, \Phi}{\mathrm{I}, s \vdash [X\psi], \varphi_0, \varphi_1, \Phi}$$

$$(14)\ \frac{\mathrm{II}, s \vdash [X\psi], \varphi_0 \vee \varphi_1, \Phi}{\mathrm{II}, s \vdash [X\psi], \varphi_i, \Phi}\ \mathrm{II} \qquad (15)\ \frac{\mathrm{II}, s \vdash [X\psi], \varphi_0 \wedge \varphi_1, \Phi}{\mathrm{II}, s \vdash [X\psi], \varphi_0, \varphi_1, \Phi}$$

$$(16)\ \frac{p, s \vdash [X\chi], \varphi U \psi, \Phi}{p, s \vdash [X\chi], \psi \vee (\varphi \wedge X(\varphi U \psi)), \Phi} \qquad (17)\ \frac{p, s \vdash [X\chi], \varphi R \psi, \Phi}{p, s \vdash [X\chi], \psi \wedge (\varphi \vee X(\varphi R \psi)), \Phi}$$

Once a configuration is reached in which every formula begins with an X, it is possible to go over to the next state on the path currently being examined.

[3] Note that φ is a state formula, i.e., either atomic in which case the play is finished immediately, or it can be assumed to begin with a path quantifier A or E in which case the right path player will be determined in the next step.

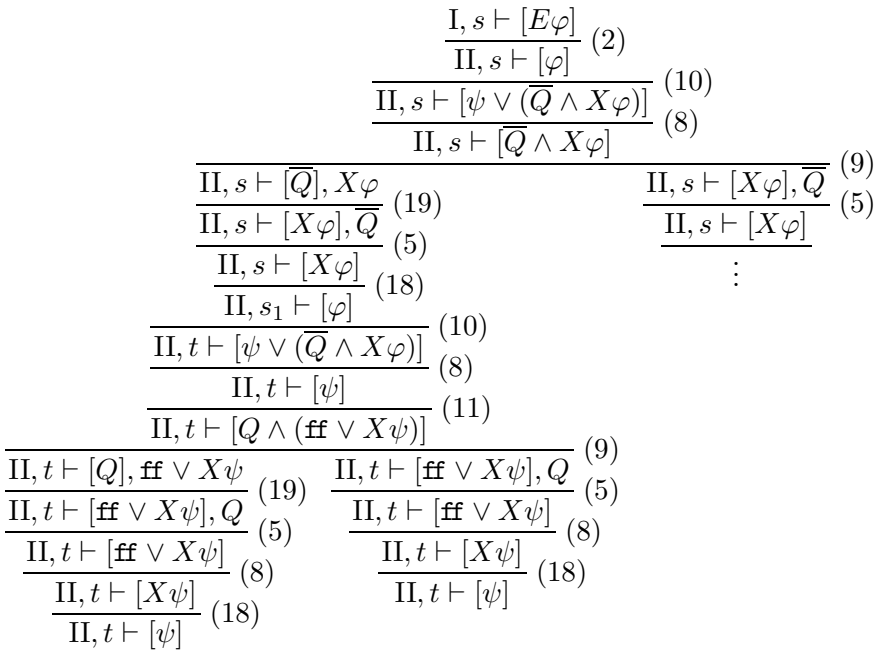

Figure 2: The game tree of example 3.1.

$$(18)\ \frac{p, s \vdash [X\varphi_0], X\varphi_1, \dots, X\varphi_k}{p, t \vdash [\varphi_0], \varphi_1, \dots, \varphi_k}\ p,\ \ s \to t$$

Finally, there is a special rule that enables the focus player to react appropriately to the path player's moves.

$$(19)\ \frac{p, s \vdash [\varphi], \psi, \Phi}{p, s \vdash [\psi], \varphi, \Phi}\ \overline{p}$$

A *move* in a play consists of two steps. First, the path player and the focus determine which of the rules (1) – (18) applies,[4] and hence which player takes the next choice.[5] After that the path player's opponent has the chance to reset the focus using rule (19).
A play is finished after a full move if it has reached a configuration

1. $p, s \vdash [Q], \Phi$, or else
2. $C = \mathrm{II}, s \vdash [\varphi U \psi], \Phi$ (resp. $C = \mathrm{I}, s \vdash [\varphi R \psi], \Phi$) after the play already went through C and player $\overline{p}$ never applied rule (19) in between, or else
3. $p, s \vdash [\varphi], \Phi$ for the second time possibly using rule (19) in between.

In the first case player II wins if $Q \in L(s)$, otherwise player I wins. In the second case player I wins if the formula in focus is $\varphi U \psi$, and player II if it is $\varphi R \psi$. In the third case p wins.

[4] A situation in which two different rules are applicable is possible. However, the order in which they are used does not effect the outcome of the game.

[5] Remember that first the formula in focus has to be brought into the form $X\psi$ before the players work on the side formulas.

Example 3.1 To illustrate the game rules we give an example. Let $\mathcal{T}$ be the transition system of figure 1. The formula to be examined is $E(\overline{Q}U(\mathtt{ff}RQ))$.[6] Obviously, $\mathcal{T}$ with starting state s satisfies it. The tree showing a winning strategy for player II with the rule numbers annotated is given in figure 2. The dots indicate a branch of the gametree that occurs twice. We use the abbreviations $\psi := \mathtt{ff}RQ$ and $\varphi := \overline{Q}U\psi$. Player II wins the play of the leftmost branch because of winning condition three, and the one right beside it because of condition two.

4 Correctness

We will show that player II has a winning strategy for a game if and only if the transition system and the starting state is a model for the formula. In order to do this we need a few technical lemmas.

Lemma 4.1 Let $C_0, \ldots, C, C_1, \ldots, C_k, C$ be a play with $C = p, s \vdash \Phi$. Then all intermediate configurations $C_1, \ldots, C_k$ are also of the form $p, s_i \vdash \Phi_i$ for $i = 1, \ldots, k$.

Proof: For simplicity reasons we assume $p = \mathrm{I}$. Suppose there is an $i \in \{1, \ldots, k\}$ with $C_i = \mathrm{II}, s_i \vdash \Phi_i$. Take the least such i. All formulas in Φ_i must have been subformulas of formulas in Φ. One of them must have been of the form $E\varphi$ which caused the pathplayer to become player II with rule (2). From this follows $\Phi_i = \{\varphi\}$. As C is also a configuration following C_i all formulas in Φ must have been generated by φ only, in particular $E\varphi$. This would cause $E\varphi$ to be a proper subformula of itself. The $p = \mathrm{II}$ case is dual. □

Proposition 4.2 Every play has a uniquely determined winner.

Proof: Every play is finite because the number of states of the transition system is finite, and so is the number of subformulas of a given φ. Therefore, the number of configurations is finite and every play will eventually reach a configuration that has been visited before and the third winning condition will apply. The first or the second could apply beforehand. If the play ended with an atomic proposition in focus then the winner is uniquely determined because $\{Q, \overline{Q}\} \subseteq L(s)$ is by definition excluded for each state s. If a configuration is visited twice then the path player, who is unique according to Lemma 4.1, wins. It may happen that a configuration $p, s \vdash \Phi$ with $\varphi U\psi, \varphi' R\psi' \in \Phi$ occurs twice, but only one of these formulas can stay in focus permanently. Hence, the winner is unique in this case, too. □

The *game* $\Gamma_{\mathcal{T}}(s, \varphi)$ for a transition system $\mathcal{T}$ with starting state s and a formula φ consists of all possible plays for $\mathcal{T}, s$ and φ. Since the number of configurations for a game is finite, the game can be viewed as a finite directed graph. This structure will be called the *game graph*. A path in a game graph can have a loop and thus be of infinite length. The play that is represented by that path ends when a configuration is visited for the second time, such that winning condition two or three applies.
We say p *wins* $\Gamma_{\mathcal{T}}(s, \varphi)$ or has a *winning strategy* for $\Gamma_{\mathcal{T}}(s, \varphi)$ if she can force every play into a configuration that makes her win the play.

Lemma 4.3 a) For every game one of the players has a winning strategy.
b) Player II wins the game $\Gamma_{\mathcal{T}}(s, \varphi)$ iff player I does not win $\Gamma_{\mathcal{T}}(s, \varphi)$.

[6] The expressed property is "There exists a path with a finite prefix and an infinite suffix. On the prefix Q never holds, on the suffix it always does."

Proof: a) Consider the tree of all possible plays in a given game. At each leaf one of the players has a winning strategy by doing nothing. Let C be a configuration with successors $C_1, \ldots, C_k$. By induction hypothesis, there is a winning strategy for either of the players in each game beginning with C_i. The branching in C can only be caused by one of the rules that require a choice to be made by, say, p. Now p also has a winning strategy for the game beginning in C if there exists an i such that p has a winning strategy in C_i, because she may choose to play on with C_i. If there is no such one, $\overline{p}$ has a winning strategy in C, because he will win no matter which C_i she chooses.
b) The "only if" part is obvious. The "if" part follows from part a). □

Corollary 4.4 The game graph for a game $\Gamma_{\mathcal{T}}(s, \varphi)$ can be partitioned into blocks. These blocks can be ordered, such that every play
a) never leaves a block i into a block j with $j < i$, and
b) finally stays in one block.

Proof: This follows from Lemma 4.1 if one also considers changes from a path player p to p herself by using game rule (1) or (2). The order on the blocks can be found in a breadth-first-search that labels the reachable configurations with natural numbers, beginning with 1. A new number is assigned to a configuration whenever game rule (1) or (2) is applied. Part b) follows from this and the finiteness of the game graph. □

The *game tree* of a game $\Gamma_{\mathcal{T}}(s, \varphi)$ is specified in the following way. Every path in the tree is a play of the game. Furthermore, if p is the winner of $\Gamma_{\mathcal{T}}(s, \varphi)$, then at every configuration C that gives p the choice all but one successor C' are eliminated, such that p still wins if she chooses C'. If $\overline{p}$ has the choice in C then all successors from the game graph are preserved in the game tree. Abusing notation $\Gamma_{\mathcal{T}}(s, \varphi)$ should stand for both the game and the game tree.[7]
A game tree representing a winning strategy for player p will also be called a *successful game tree for player p*.

Lemma 4.5 Let $\Gamma_1 = C_0 \ldots$ and $\Gamma_2 = C_0' \ldots$ be two games with $C_0 = p, s \vdash \Phi$ and $C_0' = p, s \vdash \Psi$. Assume that they both stay in one block only according to Lemma 4.4. Consider the game $\Gamma_3 = C_0'' \ldots$ with $C_0'' = p, s \vdash \Phi \cup \Psi$.
a) If $p =$ I and player II wins Γ_1 or Γ_2 then she also wins Γ_3.
b) If $p =$ II and she wins Γ_1 and Γ_2 then she also wins Γ_3.

Proof: a) Say she wins Γ_1. She will win Γ_3 by setting the focus as she would have done in Γ_1. Thus, she will also do the same moves. Since the set of side formulas in Γ_3 is larger than in Γ_1, rules (3) – (5) or (12) – (17) might have to be invoked. However, the set is still finite such that only a finite number of new moves in Γ_3 can occur between two moves from Γ_1. If she wins Γ_1 with winning condition one then she obviously does so in Γ_3, too. Assume she wins with condition two. The finiteness of the number of new side formulas from Γ_2 ensures that every play in Γ_3 performs a loop as well. Since formulas from Γ_2 do not occur in focus in this play the winner is the same as the one in Γ_1.
It is possible to create new branchings by using rule (12) for example. But the new plays only differ in the set of the side formulas which have no effect on the winner at all. Thus, every play in Γ_3 corresponds to a play in Γ_1 with the same winner.
b) Here, player I is in charge of the focus. Similar arguments as in the preceeding case hold for the use of the rules (3) – (5) or (12) – (17), as well as for the loops in Γ_3. Player I can ignore side formulas, but he will lose because the plays correspond to similar plays in Γ_1 or Γ_2 where he would lose, too. Thus, his only chance is to reset the focus from a

[7] Thus, the game tree is a minimal finite unfolding of the game graph.

formula of, say, Γ_1 to a formula of Γ_2 before he loses like he would in Γ_1. Again, he will lose there as he would in Γ_2, or he resets the focus back to a formula from Γ_1 again. Since $|Sub(\Phi \cup \Psi)| < \infty$ he will eventually create a loop such that he used rule (19) on this loop. Thus, player II also wins every possible play in Γ_3. □

Theorem 4.6 Player II wins $\Gamma_{\mathcal{T}}(s, \varphi_0)$ iff $\mathcal{T}, s \models \varphi_0$.

Proof: Because of Lemma 4.3 it is enough to show one direction only. We will do the "if"-part by constructing a (possibly infinite) game tree for player II out of the model of φ_0. This will later become a successful game tree by cutting infinite paths at appropriate positions. Furthermore, because of Corollary 4.4 it suffices to consider games on formulas with one path quantifier only. Nested $A\varphi$ or $E\varphi$ formulas can be seen as entirely new games and, hence, can be considered as atomic propositions. An induction on the number of the blocks finally proves the theorem for arbitrary **CTL*** formulas. Therefore, we may assume the path player to stay the same throughout a whole game.
There are two distinguishable cases depending on the path quantifier of φ_0. First, let $\varphi_0 = A\varphi$ and $\Pi = \{\pi \mid \pi = s\pi', \pi \text{ is a path in } \mathcal{T}\}$. Assuming that the path formula φ holds on all paths in Π, we will construct a successful game tree for player II by induction on the syntactical structure of φ. This construction may add new side formulas to existing subtrees. Lemma 4.5 a) shows that the resulting subtrees remain successful. The path set Π may differ depending on the case of the induction, but it can always be decomposed into "smaller" sets for the induction hypothesis.
Remember the following facts for this case: Every configuration is of the form $\mathrm{I}, t \vdash \Phi$, thus it is always player II who is allowed to reset the focus. The third winning condition cannot apply.
Case $\varphi = Q$**:** $\pi \models Q$ iff $Q \in L(t)$ for t being the first state in π and thus the first state of all $\pi' \in \Pi$. The subtree of $\Gamma_{\mathcal{T}}(s, \varphi_0)$ is the leaf $\mathrm{I}, t \vdash [Q], \Phi$ for any $\pi \in \Pi$ and any Φ which can be ignored by player II.
Case $\varphi = \psi \wedge \chi$**:** Here, $\pi \models \psi$ and $\pi \models \chi$ for any $\pi \in \Pi$. Thus, there are two subtrees for the plays on ψ and χ, and the resulting subtree is

$$\dfrac{\mathrm{I}, t \vdash [\psi \wedge \chi], \Phi}{\dfrac{\mathrm{I}, t \vdash [\psi], \Phi}{\vdots} \quad \dfrac{\mathrm{I}, t \vdash [\chi], \Phi}{\vdots}}$$

Case $\varphi = X\psi$**:** For all $\pi \in \Pi$ we have $\pi \models \varphi$ iff $\pi^{(1)} \models \psi$. No play can proceed with φ at the current stage before potentially present side formulas are stripped to the form $X\chi$, too. This is always feasible possibly dropping state formulas, but may create several branches.

$$\dfrac{\mathrm{I}, s_0 \vdash [X\psi], \Phi}{\vdots \qquad \vdots \qquad \vdots}$$

$$\dfrac{\dfrac{\dfrac{\mathrm{I}, s_0 \vdash [X\psi], X\chi_1^1, \ldots, X\chi_{k_1}^1}{\mathrm{I}, s_1 \vdash [\psi], \chi_1^1, \ldots, \chi_{k_1}^1} \quad \dfrac{\cdots}{\vdots}}{\vdots}}{} \qquad \dfrac{\dfrac{\dfrac{\mathrm{I}, s_0 \vdash [X\psi], X\chi_1^m, \ldots, X\chi_{k_m}^m}{\mathrm{I}, s_1' \vdash [\psi], \chi_1^m, \ldots, \chi_{k_m}^m} \quad \dfrac{\cdots}{\vdots}}{\vdots}}{}$$

However, all subtrees at these branches are successful by induction hypothesis because they still all contain ψ in focus.
Case $\varphi = \chi R\psi$**:** The constructed game tree may follow several paths $\pi \in \Pi$ because of the $\vee$ that is implicit in a $\chi R\psi$ formula.
There are two ways for $\chi R\psi$ to be fulfilled on a particular path. Either ψ holds on all states of the path. Then by induction hypothesis all occurring subtrees on left sides are

successful. Also, there must be $k, m \in \mathbb{N}$, such that $k \neq m$ but $\pi^{(k)} = \pi^{(m)}$, and thus the right branch of the tree is won by player II, too.

$$\dfrac{\dfrac{\dfrac{\mathrm{I}, s_0 \vdash [\chi R \psi], \Phi}{\mathrm{I}, s_0 \vdash [\psi \wedge (\chi \vee X(\chi R \psi))], \Phi}}{\dfrac{\mathrm{I}, s_0 \vdash [\psi], \Phi}{\vdots} \qquad \dfrac{\mathrm{I}, s_0 \vdash [\chi \vee X(\chi R \psi)], \Phi}{\vdots}} \qquad \dfrac{\dfrac{\mathrm{I}, s_1 \vdash [\chi R \psi], \Phi'}{\dfrac{\vdots}{\mathrm{I}, s_1 \vdash [\psi], \Phi'}\vdots} \qquad \dfrac{\dfrac{\vdots}{\vdots}}{\dfrac{\mathrm{I}, s_k \vdash [\chi R \psi], \Psi}{\dfrac{\vdots}{\mathrm{I}, s_m \vdash [\chi R \psi], \Psi}\vdots}}}$$

The other possibility is for ψ to hold until finally ψ and χ are fulfilled in a $\pi^{(k)}$. Then the right branch of this tree can be substituted by

$$\dfrac{\mathrm{I}, s_k \vdash [\chi], \Psi}{\vdots}$$

Again, by induction hypothesis all occurring subtrees are successful and so is the whole tree.

Case $\varphi = \chi U \psi$: This case is almost the same as the second part of the preceeding one. The only difference lies in the boolean connectives.

Case $\varphi = \psi \vee \chi$: By induction hypothesis we have a successful game tree for either the play on ψ or the play on χ depending on what path player I chooses. As the path is constructed stepwise he may do the choices according to player II's choices of the formula in focus. Say, player II chooses ψ, as the game rules force her to do the first move at this stage. The rules (12) – (17) ensure that both players have to play simultaneously on χ, too, if player I wants to proceed on a certain path.

There are two distinguishable cases. Either one of the formulas can be proved with an atomic proposition Q in a state s_k on a given path $\pi \in \Pi$. Then player II sets the focus to Q once the play has reached that state.

$$\dfrac{\dfrac{\dfrac{\mathrm{I}, s_0 \vdash [\psi \vee \chi], \Phi}{\mathrm{I}, s_0 \vdash [\psi], \chi, \Phi}}{\vdots}}{\dfrac{\mathrm{I}, s_k \vdash [\psi'], Q, \Phi'}{\mathrm{I}, s_k \vdash [Q], \psi', \Phi'}}$$

In the remaining case player II has to win with condition two. That means she must win with a $\gamma R \psi$ formula in focus whose first argument is irrelevant. It suffices to consider formulas of the form $\chi U(\gamma R \psi) \in \Phi'$,[8] where Φ' is the set of all formulas in the actual configuration.

[8]Note that $\alpha R \beta \equiv \mathtt{ff} U(\alpha R \beta)$ and $\alpha U(\beta U(\gamma R \delta) \equiv (\alpha U(\beta U \delta)) U(\gamma R \delta)$. However, this is an argument on the meta-level that reduces the effort to prove this case. It does not necessarily break the subformula property that has been pointed out in the introduction.

Let $\Phi' \supseteq \Phi = \{\varphi_1, \dots, \varphi_k\}$ be all those formulas, i.e. $\varphi_i = \chi_i U(\gamma_i R \psi_i)$ for $i = 1, \dots, k$. Consider the set $\Pi(s)$ of all paths starting with state s as an infinite tree. As Φ is to be understood disjunctively, there must be a $j \in \{1, \dots, k\}$ for every $\pi \in \Pi(s)$, s.t. $\pi \models \varphi_j$. $\Pi(s)$ can be partitioned by

$$\Pi'_i = \{\pi \in \Pi \mid \pi \models \varphi_i, \text{ and } \forall j < i : \pi \not\models \varphi_j\}, \text{ for } i = 1, \dots, k$$

Now extend this partition with finite prefixes:

$$\Pi_i := \Pi'_i \cup \{\alpha \mid i \text{ is the least } j, \text{ s.t. } \exists \pi \in \Pi'_j, \pi = \alpha\sigma\}, \text{ for } i = 1, \dots, k$$

This is a partition on the set of all infinite and finite paths beginning with state s.
Now let $\alpha \in \Pi_l$ be non-empty and finite. Then for all paths $\pi = \alpha\sigma$, $\pi \in \Pi_m$ implies $m \geq l$. $\alpha \in \Pi_l$ only if there is an infinite path $\pi = \alpha\sigma \in \Pi_l$. Suppose there also is a $\pi' = \alpha\sigma' \in \Pi_m$, but $m < l$. Then $\alpha \notin \Pi_l$ because l would not be the least index anymore. This simple result gives a strategy for player II in the play with the set Φ. She begins to set the focus to the formula with the least index l, s.t. $\Pi_l \neq \emptyset$. As player I chooses a path π stepwise he will, at any stage in the play, have selected a finite prefix α of π. Once such one $\alpha \notin \Pi_l$ she will reset the focus to the corresponding formula φ_m, s.t. $\alpha \in \Pi_m$. Since $m > l$ there are only $|\Phi| - 1$ possibilities to reset the focus until player I cannot choose a path anymore that does not fulfill the formula in focus. By induction hypothesis she will win the game from then on.

Now, let $\varphi_0 = E\varphi$. Here, all winning conditions are possible. Unlike the first part, a single path $\pi = s_0 s_1 \dots$ satisfying φ is sufficient as opposed to a set of paths.
Case $\varphi = Q$: The subtree is $\text{II}, s \vdash [Q], \Phi$. Either $\Phi = \emptyset$, in which case player II wins. Or player I drops Q and reduces the set of sideformulas. The game rules require at least one formula to be present, and as $\{Q\} \cup \Phi$ can be regarded as a conjunction all formulas are satisfied by some suffix $\pi^{(n)}$. Thus, player II will eventually win.
Case $\varphi = \psi \vee \chi$: Here, player II simply chooses the formula that is fulfilled along the path she will choose later on.
Case $\varphi = \psi \wedge \chi$: By induction hypothesis, there are gametrees for both ψ and χ. According to Lemma 4.5 b) the resulting subtree

$$\dfrac{\text{II}, t \vdash [\psi \wedge \chi], \Phi}{\dfrac{\text{II}, t \vdash [\psi], \chi, \Phi}{\vdots} \quad \dfrac{\text{II}, t \vdash [\chi], \psi, \Phi}{\vdots}}$$

is successful, too. If player I chooses to reset the focus in one of the subtrees the play will continue in the other subtree. Either player II wins there by induction hypothesis or she wins because of winning condition three as player I created a loop by resetting the focus. Furthermore, as this is the only possibility for a new sideformula to be added we can assume for the other cases that all sideformulas are fulfilled.
Case $\varphi = X\psi$: Again, potentially present sideformulas must be stripped to the form $X\chi$, too. Lemma 4.5 b) says that all subtrees at these branches are successful. We need not take the resetting of the focus in the shown part into account because it has the same effect as being done on the top level of this tree or after the stripping of the formulas.

$$\dfrac{\text{II}, s_j \vdash [X\psi], \Phi}{\dfrac{\vdots}{\dfrac{\text{II}, s_j \vdash [X\psi], X\chi_1^1, \dots, X\chi_{k_1}^1}{\dfrac{\text{II}, s_{j+1} \vdash [\psi], \chi_1^1, \dots, \chi_{k_1}^1}{\vdots}}} \quad \dfrac{\vdots}{\dfrac{\text{II}, s_j \vdash [X\psi], X\chi_1^m, \dots, X\chi_{k_m}^m}{\dfrac{\text{II}, s_{j+1} \vdash [\psi], \chi_1^m, \dots, \chi_{k_m}^m}{\vdots}}}}$$

Case $\varphi = \chi R \psi$ or $\varphi = \chi U \psi$: These case are very similar to the ones in the previous part of the proof. The only difference is that there is just one successor configuration whenever rule (18) is applied.

To finish the proof of this part of the theorem every infinite branch of the constructed gametree must be cut after the first position in which a configuration occured for the second time. The resulting finite tree is exactly a successful gametree for player II.
The "only if"-part of the theorem follows immediately from Lemma 4.3 and the duality of the games. □

References

[1] E. M. Clarke and E. A. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In *Logics of Programs: Workshop*, volume 131 of *LNCS*, Yorktown Heights, New York, May 1981. Springer.

[2] M. Dam. CTL* and ECTL* as fragments of the modal μ-calculus. *TCS*, 126(1):77–96, April 1994.

[3] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In IEEE, editor, *Proc. 32nd Annual Symp. on Foundations of Computer Science*, pages 368–377, San Juan, Porto Rico, October 1991. IEEE Computer Society Press.

[4] E. A. Emerson and A. P. Sistla. Deciding full branching time logic. *Information and Control*, 61(3):175–201, June 1984.

[5] D. Kozen. Results on the propositional mu-calculus. *TCS*, 27:333–354, December 1983.

[6] F. Moller and G. M. Birtwistle. *Logics for concurrency: structure versus automata*, volume 1043 of *LNCS*. Springer, New York, NY, USA, 1996.

[7] A. Pnueli. The temporal logic of programs. In *Proc. 18th IEEE Symp. on the Foundations of Computer Science, FOCS-77*, pages 46–57, Providence, Rhode Island, October 31–November 2 1977. IEEE, IEEE Computer Society Press.

[8] C. Stirling. Local model checking games. In I. Lee and S. A. Smolka, editors, *Proc. 6th Int. Conf. on Concurrency Theory, CONCUR'95*, volume 962 of *LNCS*, pages 1–11, Berlin, GER, August 1995. Springer.