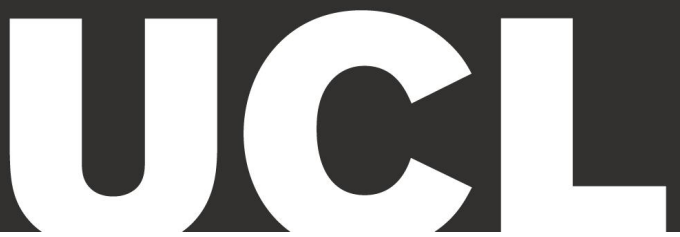
UCL DEPARTMENT OF
COMPUTER SCIENCE



Research Note

RN/14/11

Fairness for Infinite-State Systems

October 20, 2014

Byron Cook Heidy Khlaaf Nir Piterman

Abstract.

In this paper we introduce the first known tool for symbolically proving *fair*-CTL properties of (infinite-state) integer programs. Our solution is based on a reduction to existing techniques for fairness-free CTL model checking via the use of infinite non-deterministic branching to symbolically partition fair from unfair executions. We show the viability of our approach in practice using examples drawn from device drivers and algorithms utilizing shared resources.

1 Introduction

In model checking, fairness allows us to bridge between linear-time (*a.k.a.* trace-based) and branching-time (*a.k.a.* state-based) reasoning. Fairness is crucial, for example, to Vardi & Wolper's automata-theoretic technique for LTL verification [23]. Furthermore, when proving state-based CTL properties, we must often use fairness to model trace-based assumptions about the environment both in a sequential setting, and when reasoning about concurrent environments, where fairness is used to abstract away the scheduler.

In this paper we introduce the first-known fair-CTL model checking technique for (infinite-state) integer programs. Our solution reduces fair CTL to fairness-free CTL using prophecy variables to encode a partition of fair from unfair paths. Cognoscenti may at first find this result surprising. It is well known that fair termination of Turing machines cannot be reduced to termination of Turing machines. The former is Σ_1^1-complete and the latter is RE-complete [17].[1] For similar reasons fair-CTL model checking of Turing machines cannot be reduced to CTL model checking of Turing machines. The key to our reduction is the use of infinite non-deterministic branching when model checking fairness-free CTL. As a consequence, in the context of infinite branching, fair and fairness-free CTL are equally difficult (and similarly for termination).

Motivation. Current techniques for model checking CTL properties provide no support for verification of fair-CTL, thus excluding a large set of branching-time liveness properties necessitating fairness. These properties are often imperative to verifying the liveness of systems such as Windows kernel APIs that acquire resources and APIs that release resources. Below are properties which can be expressed in fair-CTL, but not CTL nor LTL. We write these properties in CTL*, a superset of both CTL and LTL[2]. For brevity, we substitute the property $\mathsf{GF}p{\rightarrow}\mathsf{GF}q$ with Ω. A state property is indicated by φ (i.e., a combination of assertions on the states of the program) and p and q are subsets of program states, constituting our fairness requirement (infinitely often p implies infinitely often q).

$$\mathsf{E}\big[\Omega \wedge \mathsf{G}\varphi\big] \tag{1}$$

$$\mathsf{A}\Big[\Omega \rightarrow \mathsf{G}\big[\varphi_1 \rightarrow \mathsf{A}\big(\Omega \rightarrow \mathsf{F}\varphi_2\big)\big]\Big] \tag{2}$$

$$\mathsf{A}\Big[\Omega \rightarrow \mathsf{G}\big[\varphi_1 \rightarrow \mathsf{A}\big(\Omega \rightarrow \mathsf{FE}(\Omega \wedge \mathsf{G}\varphi_2)\big)\big]\Big] \tag{3}$$

Property (1) generalizes fair non-termination, that is, there exists an infinite fair computation all of whose states satisfy the property φ. Property (2) indicates that on every fair path, every φ_1 state is later followed by a φ_2 state. We will later verify property (2) for a Windows device driver, indicating that a lock will

[1] Sometimes termination refers to *universal termination*, which entails termination for *all* possible inputs. This is a harder problem and is co-$\mathrm{RE}^{\mathrm{RE}}$-complete.

[2] These properties expressed in terms of the fair path quantifiers E_f and A_f are $\mathsf{E}_f\,\mathsf{G}\varphi$, $\mathsf{A}_f\,\mathsf{G}(\varphi_1 \rightarrow \mathsf{A}_f\,\mathsf{F}\varphi_2)$, and $\mathsf{A}_f\,\mathsf{G}(\varphi_1 \rightarrow \mathsf{A}_f\,\mathsf{F}\,\mathsf{E}_f\,\mathsf{G}\varphi_2)$, respectively.

always eventually be released in the case that a call to a lock occurs, provided that whenever we continue to call a Windows API repeatedly, it will eventually return a desired value (fairness). Similarly, (3) dictates that on every fair path whenever a φ_1 state is reached, on all possible futures there is a state which is a possible fair future and φ_2 is always satisfied. For example, one may wish to verify that there will be a possible active fair continuation of a server, and that it will continue to effectively serve if sockets are successfully opened.[3]

Furthermore, fair-CTL model checking is rudimental to the well known technique of verifying LTL in the finite-state setting [23]. A fair-CTL model checker for infinite-state systems would thus enable us to implement the automata-theoretic approach to linear-time model checking by reducing it to fair-CTL model checking as is done in the finite-state setting.

Fairness is also crucial to the verification of concurrent programs, as well-established techniques such as [6] reduce concurrent liveness verification to a sequential verification task. Thread-modular reductions of concurrent to sequential programs often require a concept of fairness when the resulting sequential proof obligation is a progress property such as wait-freedom, lock-freedom, or obstruction-freedom. Moreover, obstruction freedom cannot be expressed in LTL without additional assumptions. With our technique we can build practical tools for automatically proving these sequential reductions using fair-CTL model checking.

Related Work. Support for fairness in finite and other decidable settings has been well studied. Tools for these settings (*e.g.* NuSMV for finite state systems [4, 5], Moped and PuMoc for pushdown automata [22, 21], Prism for probabilistic timed automata [18], and Uppaal for timed automata [14]) provide support for fairness constraints.

Contrarily, we seek to verify the undecidable general class of (infinite-state) integer programs supporting both control-sensitive and numerical properties. Additionally, some of these tools do not fully support CTL model checking, as they do not reliably support mixtures of nested universal/existential path quantifiers, etc. The tools which consider full CTL and the general class of integer programs as we do are [2, 9, 11]. However, these tools provide no support for verifying fair-CTL.

When we consider the general class of integer programs, the use of infinite nondeterminism to encode fairness policies has been previously utilized by Olderog *et al.* [1]. However, they do not rely on nondeterminism alone but require refinement of the introduced nondeterminism to derive concrete schedulers which enforce a given fairness policy. Thus, their technique relies on the ability to force the occurrence of fair events whenever needed by the reduction. We support general fairness constraints, rather than just fair scheduling. The ability to force the occurrence of fair events is too strong for our needs. Indeed, in

[3] Notice that our definition of fair CTL considers finite paths. Thus, all path quantifications above range over finite paths as well.

the context of we rely on the program continuing a normal execution until the "natural" fulfillment of the fairness constraint.

An analysis of fair discrete systems which separates reasoning pertaining to fairness and well-foundedness through the use of inductive transition invariants was introduced in [19]. Their strategy is the basis of the support for fairness added to TERMINATOR [7]. However, this approach relies on the computation of transition invariants [20], whereas our approach does not. We have recently shown that, in practice, state-based techniques that circumvent the computation of transition invariants perform significantly better [13]. Additionally, a technique utilized to reduce LTL model checking to fairness-free CTL model checking introduced by [10] is largely incomplete, as it does not sufficiently determinize all possible branching traces. Note that these methodologies are used to verify fairness and liveness constraints expressible within linear temporal logic, and are thus not applicable to verify fair branching-time logic or branching-time logic. Indeed, this was part of our motivation for studying alternative approaches to model checking with fairness.

2 Preliminaries

Transition systems. A transition system is $M = (S, S_0, R, L)$, where S is a countable set of states, $S_0 \subseteq S$ a set of initial states, $R \subseteq S \times S$ a transition relation, and $L : S \to 2^{AP}$ a labeling function associating a set of propositions with every state $s \in S$. A *trace* or a *path* of a transition system is either a finite or infinite sequence of states. The set of infinite traces starting at $s \in S$, denoted by $\Pi_\infty(s)$, is the set of sequences $(s_0, s_1, \ldots)$ such that $s_0 = s$ and $\forall i \geq 0.\ (s_i, s_{i+1}) \in R$. The set of finite traces starting at $s \in S$, denoted by $\Pi_f(s)$, is the set of sequences $(s_0, s_1, \ldots, s_j)$ such that $s_0 = s$, $j \geq 0$, $\forall i < j.\ (s_i, s_{i+1}) \in R$, and $\forall s \in S.\ (s_j, s) \notin R$. Finally, the set of maximal traces starting at s, denoted by $\Pi_m(s)$, is the set $\Pi_\infty(s) \cup \Pi_f(s)$. For a path π, we denote the length of said path by $|\pi|$, which is ω in case that π is infinite.

Computation tree logic (CTL). We are interested in verifying state-based properties in computation tree logic (CTL). Our definition of CTL differs slightly from previous work, as it takes into account finite (maximal) paths. This semantics allows us to specify properties such as termination without requiring special atomic propositions to hold at program exit points, as proposed by Cook *et al.* in [12], and to reason about a transformation that introduces many finite paths.

A CTL formula is of the form:

$$\varphi ::= \alpha \mid \neg\alpha \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathsf{AX}\varphi \mid \mathsf{AF}\varphi \mid \mathsf{A}[\varphi \mathsf{W} \varphi] \mid \mathsf{EX}\varphi \mid \mathsf{EG}\varphi \mid \mathsf{E}[\varphi \mathsf{U} \varphi],$$

where $\alpha \in AP$ is an atomic proposition. We assume that formulae are written in negation normal form, in which negation only occurs next to atomic propositions. We introduce $\mathsf{AG}, \mathsf{AU}, \mathsf{EF}$, and EW as syntactic sugar as usual. A formula is in ACTL if it uses only universal operators, i.e., AX, AW, AF, AU, or AG.

Fig. 1 defines when a CTL property φ holds in a state $s \in S$ of a transition system M. We say that φ holds in M, denoted $M \models_m \varphi$, if $\forall s \in S_0.\ M, s \models_m \varphi$.

Fair CTL. For a transition system M, a fairness condition is $\Omega = (p, q)$, where $p, q \subseteq S$. When fairness is part of the transition system we denote it as $M =$

$$\frac{\alpha(s)}{M,s \models_m \alpha} \qquad \frac{\neg\alpha(s)}{M,s \models_m \neg\alpha}$$

$$\frac{M,s \models \varphi_1 \quad M,s \models \varphi_2}{M,s \models_m \varphi_1 \wedge \varphi_2} \qquad \frac{M,s \models \varphi_1 \ \vee \ M,s \models \varphi_2}{M,s \models_m \varphi_1 \vee \varphi_2}$$

$$\frac{\forall \pi = (s_0, s_1, ...) \in \Pi_m(s).\ M, s_1 \models \varphi}{M,s \models_m \mathsf{AX}\varphi} \qquad \frac{\exists \pi = (s_0, s_1, ...) \in \Pi_m(s).\ M, s_1 \models \varphi}{M,s \models_m \mathsf{EX}\varphi}$$

$$\frac{\begin{array}{c}\forall \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\forall i \in [0, |\pi|).\ M, s_i \models \varphi_1) \vee \\ (\exists j \in [0, |\pi|).\ M, s_j \models \varphi_2 \wedge \forall i \in [0, j).\ M, s_i \models \varphi_1)\end{array}}{M,s \models_m \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]} \qquad \frac{\begin{array}{c}\forall \pi = (s_0, s_1, ...) \in \Pi_m(s). \\ (\exists j \in [0, |\pi|).\ M, s_j \models \varphi)\end{array}}{M,s \models_m \mathsf{AF}\varphi}$$

$$\frac{\begin{array}{c}\exists \pi = (s_0, s_1, ...) \in \Pi_m(s). \\ (\exists j \in [0, |\pi|).\ M, s_j \models \varphi_2 \wedge \forall i \in [0, j).\ M, s_i \models \varphi_1)\end{array}}{M,s \models_m \mathsf{E}[\varphi_1 \mathsf{U} \varphi_2]} \qquad \frac{\begin{array}{c}\exists \pi = (s_0, s_1, ...) \in \Pi_m(s). \\ (\forall i \in [0, |\pi|).\ M, s_i \models \varphi)\end{array}}{M,s \models_m \mathsf{EG}\varphi}$$

Fig. 1. Semantics of CTL: $\models_m$

(S, S_0, R, L, Ω). We still include Ω as a separate component in transformations and algorithms for emphasis. We freely confuse between assertions over program variables and sets of states that satisfy them. An infinite path π is unfair under Ω if states from p occur infinitely often along π but states from q occur finitely often. Otherwise, π is fair. The condition Ω denotes a strong fairness constraint. Weak fairness constraints can be trivially expressed by $\Omega = (\mathsf{true}, q)$, that is, states from q must occur infinitely often. Equivalently, π is fair if it satisfies the LTL formula $\pi \models (\mathsf{GF}p \rightarrow \mathsf{GF}q)$. For a transition system $M = (S, S_0, R, L, \Omega)$, an infinite path π, we denote $M, \pi \models \Omega$ if π is fair [16]. We consider strong fairness with one pair of sets of states. Extending our results to strong fairness over multiple pairs is simple and omitted for clarity of exposition.

For a transition system M and a CTL property φ, Fig. 2 defines when the property holds in a state $s \in S$. We say that φ holds in M, denoted $M \models_{\Omega_+} \varphi$ if $\forall s \in S_0.R, s \models_{\Omega_+} \varphi$. When clear from the context, we sometimes omit the transition system and write $s \models_{\Omega_+} \varphi$ or $s \models_m \varphi$ instead.

3 Fair-CTL Model Checking

In this section we present a procedure for reducing fair-CTL model checking to CTL model checking. The procedure builds on a transformation of infinite-state programs by adding a prophecy variable that truncates unfair paths. We start by presenting the transformation, followed by a program's adaptation for using said transformation, and subsequently the model-checking procedure.

In Fig. 3, we propose a reduction $\textsc{Fair}(M, \Omega)$ that encodes an instantiation of the fairness constraint within a transition system. When given a transition system (S, S_0, R, L, Ω), where $\Omega = (p, q)$ is a strong-fairness constraint, $\textsc{Fair}(M, \Omega)$ returns a new transition system (without fairness) that, through the use of a prophecy variable n, infers all possible paths that satisfy the fairness constraint, while avoiding all paths violating the fairness policy. Intuitively, n is decreased whenever a transition imposing $p \wedge n' < n$ is taken. Since $n \in \mathbb{N}$, n cannot decrease infinitely often, thus enforcing the eventual invalidation of the

$$\frac{\alpha(s)}{M,s \models_{\Omega_+} \alpha} \qquad \frac{\neg\alpha(s)}{M,s \models_{\Omega_+} \neg\alpha}$$

$$\frac{M,s \models_{\Omega_+} \varphi_1 \quad M,s \models_{\Omega_+} \varphi_2}{M,s \models_{\Omega_+} \varphi_1 \wedge \varphi_2} \qquad \frac{M,s \models_{\Omega_+} \varphi_1 \ \vee \ M,s \models \varphi_2}{M,s \models_{\Omega_+} \varphi_1 \vee \varphi_2}$$

$$\frac{\forall \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\pi \in \Pi_f \vee M,\pi \models \Omega) \ \Rightarrow \ M, s_1 \models \varphi}{M,s \models_{\Omega_+} \mathsf{AX}\varphi} \qquad \frac{\exists \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\pi \in \Pi_f \vee M,\pi \models \Omega) \ \wedge \ M, s_1 \models \varphi}{M,s \models_{\Omega_+} \mathsf{EX}\varphi}$$

$$\frac{\begin{array}{c}\forall \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\pi \in \Pi_f \vee \\ M,\pi \models \Omega) \ \Rightarrow \ (\forall i \in [0, |\pi|).\ M, s_i \models \varphi_1) \vee \\ (\exists j \in [0, |\pi|).\ M, s_j \models \varphi_2 \wedge \forall i \in [0, j).\ M, s_i \models \varphi_1)\end{array}}{M,s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]}$$

$$\frac{\begin{array}{c}\forall \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\pi \in \Pi_f \vee \\ M,\pi \models \Omega) \ \Rightarrow \ (\exists j \in [0, |\pi|).\ M, s_j \models \varphi_2\end{array}}{M,s \models_{\Omega_+} \mathsf{AF}\varphi}$$

$$\frac{\begin{array}{c}\exists \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\pi \in \Pi_f \vee \\ M,\pi \models \Omega) \ \wedge \ (\exists j \in [0, |\pi|).\ M, s_j \models \varphi_2 \wedge \\ \forall i \in [0, j).\ M, s_i \models \varphi_1)\end{array}}{M,s \models_{\Omega_+} \mathsf{E}[\varphi_1 \mathsf{U} \varphi_2]}$$

$$\frac{\begin{array}{c}\exists \pi = (s_0, s_1, ...) \in \Pi_m(s).\ (\pi \in \Pi_f \vee \\ M,\pi \models \Omega) \ \wedge \ (\forall i \in [0, |\pi|).\ M, s_i \models \varphi)\end{array}}{M,s \models_{\Omega_+} \mathsf{EG}\varphi}$$

Fig. 2. Semantics of Fair CTL with finite and infinite paths: $\models_{\Omega_+}$

$$\textsc{Fair}((S, S_0, R, L), (p, q)) \triangleq (S_\Omega, S^0_\Omega, R_\Omega, L_\Omega) \text{ where}$$

$$\begin{array}{rcl} S_\Omega & = & S \times \mathbb{N} \\ R_\Omega & = & \{((s,n),(s',n')) \mid (s,s') \in R\} \wedge \begin{pmatrix} (\neg p \wedge n' \leq n) \vee \\ (p \wedge n' < n) \vee \\ q \end{pmatrix} \\ S^0_\Omega & = & S^0 \times \mathbb{N} \\ L_\Omega(s,n) & = & L(s) \end{array}$$

Fig. 3. FAIR takes a system (S, S_0, R, L, Ω), where $\Omega = (p, q)$ is a fairness constraint and $p, q \subseteq S$, system $(S_\Omega, S^0_\Omega, R_\Omega, L_\Omega)$. Note that $n \geq 0$ is implicit, as $n \in \mathbb{N}$.

transition $p \wedge n' < n$. Therefore, R_Ω would only allow a transition to proceed if q holds or $\neg p \wedge n' \leq n$ holds. That is, either q occurs infinitely often or p will occur finitely often. Note that a q-transition imposes no constraints on n', which effectively resets n' to an arbitrary value.

The conversion of M with fairness constraint Ω to $\textsc{Fair}(M, \Omega)$ involves the truncation of paths due to the wrong estimation of the number of p-s until q. This means that $\textsc{Fair}(M, \Omega)$ can include (maximal) finite paths that are prefixes of unfair infinite paths. It follows that when model checking CTL we have to ensure that these paths do not interfere with the validity of our model checking procedure. Hence, we have to distinguish between maximal (finite) paths that occur in M and those introduced by our reduction. This is done through adding a proposition t to mark all original "valid" termination states prior to the reduction in Fig. 3, followed by adjusting the CTL specification through a transformation,

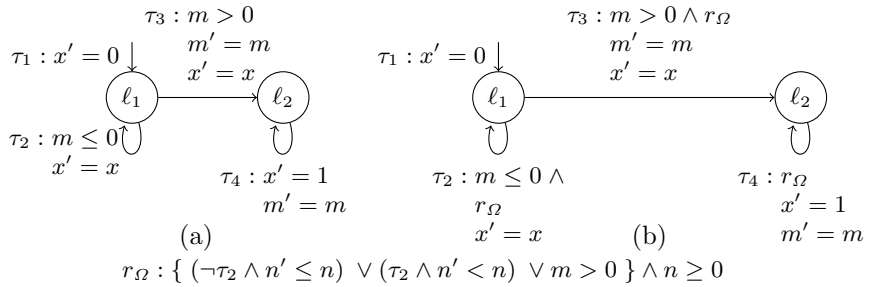


Fig. 4. Reducing a transition system with the fair CTL property $\mathsf{AG}(x = 0 \rightarrow \mathsf{AF}(x = 1))$ and the fairness constraint $\mathsf{GF}\ \tau_2 \rightarrow \mathsf{GF}\ m > 0$. The original transition system is represented in (a), followed by the application of our fairness reduction in (b).

all presented in Section 3.3. We first provide high-level understanding of our approach through an example.

3.1 Illustrative Example

Consider the example in Fig. 4 for the fair CTL property $\mathsf{AG}(x = 0 \rightarrow \mathsf{AF}(x = 1))$ and the fairness constraint $\mathsf{GF}\ \tau_2 \rightarrow \mathsf{GF}\ m > 0$ for the initial transition system introduced in (a). We demonstrate the resulting transformation for this infinite-state program, which allows us to reduce fair model checking to model checking. By applying $\textsc{Fair}(M, \Omega)$ from Fig. 3, we obtain (b) where each original transition, τ_2, τ_3, and τ_4, are adjoined with restrictions such that $\{(\neg\tau_2 \wedge n' \leq n) \vee (\tau_2 \wedge n' < n) \vee m > 0 \} \wedge n \geq 0$ holds. That is, we wish to restrict our transition relations such that if τ_2 is visited infinitely often, then the variable m must be positive infinitely often. In τ_2, the unconstrained variable m indicates that the variable m is being assigned to a nondeterministic value, thus with every iteration of the loop, m acquires a new value. In the original transition system, τ_2 can be taken infinitely often given said non-determinism, however in (b), such a case is not possible. The transition τ_2 in (b) now requires that n be decreased on every iteration. Since $n \in \mathbb{N}$, n cannot be decreased infinitely often, causing the eventual restriction to the transition τ_2. Such an incidence is categorized as a finite path that is a prefix of some unfair infinite paths. As previously mentioned, we will later discuss how such paths are disregarded. This leaves only paths where the prophecy variable "guessed" correctly. That is, it prophesized a value such that τ_3 is reached, thus allowing our property to hold.

3.2 Prefixes of Infinite Paths

We now elaborate on the transformations utilized to distinguish between the maximal (finite) paths that occur in M, and those that are prefixes of unfair infinite paths introduced by our reduction. Consider a transition system $M = (S, S_0, R, L, \Omega)$, where $\Omega = (p, q)$, and let φ be a CTL formula. Let t be an atomic proposition not appearing in L or φ. We define the transformation to

$$
\begin{aligned}
\textsc{Term}(\alpha, t) &::= \alpha \\
\textsc{Term}(\varphi_1 \wedge \varphi_2, t) &::= \textsc{Term}(\varphi_1, t) \wedge \textsc{Term}(\varphi_2, t) \\
\textsc{Term}(\varphi_1 \vee \varphi_2, t) &::= \textsc{Term}(\varphi_1, t) \vee \textsc{Term}(\varphi_2, t) \\
\textsc{Term}(\mathsf{AX}\varphi, t) &::= t \vee \mathsf{AX}(\textsc{Term}(\varphi, t)) \\
\textsc{Term}(\mathsf{AF}\varphi, t) &::= \mathsf{AF}\textsc{Term}(\varphi, t) \\
\textsc{Term}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2], t) &::= \mathsf{A}[\textsc{Term}(\varphi_1, t)\ \mathsf{W}\ \textsc{Term}(\varphi_2, t)] \\
\textsc{Term}(\mathsf{EX}\varphi, t) &::= \neg t \wedge \mathsf{EX}(\textsc{Term}(\varphi, t)) \\
\textsc{Term}(\mathsf{EG}\varphi, t) &::= \mathsf{EG}\textsc{Term}(\varphi, t) \\
\textsc{Term}(\mathsf{E}[\varphi_1 \mathsf{U} \varphi_2], t) &::= \mathsf{E}[\textsc{Term}(\varphi_1, t)\ \mathsf{U}\ \textsc{Term}(\varphi_2, t)]
\end{aligned}
$$

Fig. 5. Transformation $\textsc{Term}(\varphi, t)$.

mark "valid" termination states as $\textsc{Term}(M, t) = (S, S_0, R', L', \Omega')$, where R', L', and Ω' are as follows:

$$
\begin{array}{l}
R' = R \cup \{(s, s) \mid \forall s'.(s, s') \notin R\} \\
\Omega' = (p, q \vee t)
\end{array}
\qquad
L'(s) = \begin{cases} L(s) \cup \{t\}, & \text{if } \forall s'.(s, s') \notin R \\ L(s), & \text{otherwise} \end{cases}
$$

That is, we eliminate all finite paths in $\textsc{Term}(M, t)$ by instrumenting self loops and adding the proposition t on all terminal states. The fairness constraint is adjusted to include paths that end in such states. We now adjust the CTL formula φ that we wish to verify on M. Recall that t does not appear in φ. Now let $\textsc{Term}(\varphi, t)$ denote the CTL formula transformation in Fig. 5.

The combination of the two transformations maintains the validity of a CTL formula in a given system.

Theorem 1. $M \models_{\Omega_+} \varphi \Leftrightarrow \textsc{Term}(M, t) \models_{\Omega_+} \textsc{Term}(\varphi, t)$

Proof Sketch (full proof in Appendix A). We show that every fair path of $\textsc{Term}(M, t)$ corresponds to a maximal path in M and vice versa. The proof then proceeds by induction on the structure of the formula. For existential formulas, witnesses are translated between the models. For universal formulas, we consider arbitrary paths and translate them between the models.

After having marked the "valid" termination points in M by using the transformation $\textsc{Term}(M, t)$, we must ensure that our fair-CTL model-checking procedure ignores "invalid" finite paths in $\textsc{Fair}(M, \Omega)$. The finite paths that need to be removed from consideration are those that arise by wrong prediction of the prophecy variable n. The formula $\mathsf{term} = \mathsf{AFAX}\,\mathsf{false}$ holds in a state s iff all paths from s are finite. We denote its negation $\mathsf{EGEX}\,\mathsf{true}$ by $\neg\mathsf{term}$. Intuitively, when considering a state (s, n) of $\textsc{Fair}(M, \Omega)$, if (s, n) satisfies term, then (s, n) is part of a wrong prediction. If (s, n) satisfies $\neg\mathsf{term}$, then (s, n) is part of a correct prediction. Further on, we will set up our model checking technique such that universal path formulas ignore violations that occur on terminating paths (which correspond to wrong predictions) and existential path formulas use only non-terminating paths (which correspond to correct predictions).

3.3 Fair-CTL Model Checking

We use $\textsc{Fair}(M, \Omega)$ to handle fair-CTL model checking. Our procedure employs an existing CTL model checking algorithm for infinite-state systems. We assume

```
1  let FairCTL(M, Ω, φ) : assertion =
2
3      match(φ) with
4      | Q φ1 OP φ2
5      | φ1 bool_OP φ2 →
6          a_φ1 = FairCTL(M, Ω, φ1);
7          a_φ2 = FairCTL(M, Ω, φ2)
8      | Q OP φ1 →
9          a_φ1 = FairCTL(M, Ω, φ1)
10     | α →
11         a_φ1 = α
12
13     match(φ) with
14     | E φ1 U φ2 →
15         φ' = E[a_φ1 U (a_φ2 ∧ ¬term)]
16     | E Gφ1 →
17         φ' = EG(a_φ1 ∧ ¬term)
18     | E Xφ1 →
19         φ' = EX(a_φ1 ∧ ¬term)
20     | A φ1 W φ2 →
21         φ' = A[a_φ1 W (a_φ2 ∨ term)]
22
23     | A Fφ1 →
24         φ' = AF(a_φ1 ∨ term)
25     | A Xφ1 →
26         φ' = AX(a_φ1 ∨ term)
27     | φ1 bool_OP φ2 →
28         φ' = a_φ1 bool_OP a_φ2
29     | α →
30         φ' = a_φ1
31
32     M' = Fair(M, Ω)
33     a = CTL(M', φ')
34
35     match(φ) with
36     | E φ' →
37         return ∃n ≥ 0 . a
38     | A φ' →
39         return ∀n ≥ 0 . a
40     | _ →
41         return a
```

Fig. 6. Our procedure FairCTL(M, Ω, φ) which employs both an existing CTL model checker and the reduction Fair(M, Ω). An assertion characterizing the states in which φ holds under the fairness constraint Ω is returned.

```
1  let Verify(M, Ω, φ) : bool =
2
3      a = FairCTL(Term(M, t), Ω, Term(φ, t))
4      return S_0 ⇒ a
```

Fig. 7. CTL model checking procedure Verify, which utilizes the subroutine in Fig. 6 to verify if a CTL property φ holds under the fairness constraints Ω for a transition system M.

that the CTL model checking algorithm returns an assertion characterizing all the states in which a CTL formula holds. Tools such as Beyene *et al.* [2] and Cook *et al.* [9] support this functionality. We denote such CTL verification tools by CTL(M, φ), where M is a transition system and φ a CTL formula.

Our procedure adapting Fair(M, Ω) is presented in Fig. 6. Given a transition system M, a fairness constraint Ω, and a CTL formula φ, FairCTL returns an assertion characterizing the states in which φ fairly holds. Initially, our procedure is called by Verify in Fig. 7 where M and φ are initially transformed by Term(M, t) and Term(φ, t) discussed in Section 3.2. That is, Term(M, t) marks all "valid" termination states in M to distinguish between maximal (finite) paths that occur in M and those introduced by our reduction. Term(φ, t) allows us to disregard all aforementioned finite paths, as we only consider infinite paths, which correspond to a fair path in the original system.

Our procedure then begins by recursively enumerating over each CTL subproperty, wherein we attain an assertion characterizing all the states in which

the sub-property holds under the fairness constraint Ω. These assertions will subsequently replace their corresponding CTL sub-properties as shown on lines 15,17,19, and so on. A new CTL formula φ' is then acquired by adding an appropriate termination or non-termination clause (lines 13-30). This clause allows us to ignore finite paths that are prefixes of unfair infinite paths. Recall that other finite paths were turned infinite and marked by the proposition t in $\textsc{Term}(M, t)$.

Ultimately, our reduction $\textsc{Fair}(M, \Omega)$ is utilized on line 32, where we transform the input transition system M according to Fig. 3. With our modified CTL formula φ' and transition system M', we call upon the existing CTL model checking algorithm to return an assertion characterizing all the states in which the formula holds. The returned assertion is then examined on lines 35-39 to determine whether or not φ' holds under the fairness constraint Ω. If the property is existential, then it is sufficient that there exists at least one value of the prophecy variable such that the property holds. If the property is universal, then the property must hold for all possible values of the prophecy variable.

We state the correctness and completeness of our model checking procedure.

Theorem 2. *For every CTL formula φ and every transition system M with no terminating states we have $M \models_{\Omega_+} \varphi \Leftrightarrow S_0 \rightarrow \textsc{FairCTL}\ (M, \Omega, \varphi)$.*

Proof Sketch (full proof in Appendix A). First, we demonstrate that every infinite path in $\textsc{Fair}(M, \Omega)$ starting in (s, n) for some $n \in \mathbb{N}$ corresponds to an infinite path in M starting in s satisfying Ω, and vice versa. From this correspondence of fair paths in M and infinite paths in $\textsc{Fair}(M, \Omega)$, we can safely disregard all the newly introduced finite paths given a transition system with no finite paths (i.e., $\textsc{Term}(M, t)$).

We then proceed to show by induction on the structure of the formula that the assertion returned by $\textsc{FairCTL}(M, \Omega, \varphi)$ characterizes the set of states of M that satisfy φ. For a universal property, we show that if it holds from s in M then it(s modified form) holds from (s, n) for every n in $\textsc{Fair}(M, \Omega)$ and vice versa. For an existential property, we show that if it holds from s in M then its modified form holds from (s, n) for some n in $\textsc{Fair}(M, \Omega)$ and vice versa.

Corollary 1. *For every CTL formula φ and every transition system M we have $M \models_{\Omega_+} \varphi \Leftrightarrow \textsc{Verify}(M, \Omega, \varphi)$ returns true.*

Proof. Verify calls FairCTL on $\textsc{Term}(M, t)$ and $\textsc{Term}(\varphi, t)$. It follows that $\textsc{Term}(M, t)$ has no terminating states and hence Theorem 2 applies to it. By Theorem 1, the mutual transformation of M and φ to $\textsc{Term}(M, t)$ and $\textsc{Term}(\varphi, t)$ preserves whether or not $M \models_{\Omega_+}$. The corollary follows.

4 Fair-ACTL Model Checking

In this section we show that in the case that we are only interested in universal path properties, i.e., formulas in ACTL, there is a simpler approach to fair-CTL model checking. In this simpler case, we can solely use the transformation $\textsc{Fair}(M, \Omega)$. Just like in FairCTL, we still must ignore truncated paths that

$$
\begin{aligned}
\textsc{NTerm}(\alpha) &::= \alpha \\
\textsc{NTerm}(\varphi_1 \wedge \varphi_2) &::= \textsc{NTerm}(\varphi_1) \wedge \textsc{NTerm}(\varphi_2) \\
\textsc{NTerm}(\varphi_1 \vee \varphi_2) &::= \textsc{NTerm}(\varphi_1) \vee \textsc{NTerm}(\varphi_2) \\
\textsc{NTerm}(\mathsf{AX}\varphi) &::= \mathsf{AX}(\textsc{NTerm}(\varphi) \vee \mathsf{term}) \\
\textsc{NTerm}(\mathsf{AF}\varphi) &::= \mathsf{AF}(\textsc{NTerm}(\varphi) \vee \mathsf{term}) \\
\textsc{NTerm}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]) &::= \mathsf{A}[\textsc{NTerm}(\varphi_1)\ \mathsf{W}\ (\textsc{NTerm}(\varphi_2) \vee \mathsf{term})]
\end{aligned}
$$

Fig. 8. Transformation NTerm( ).

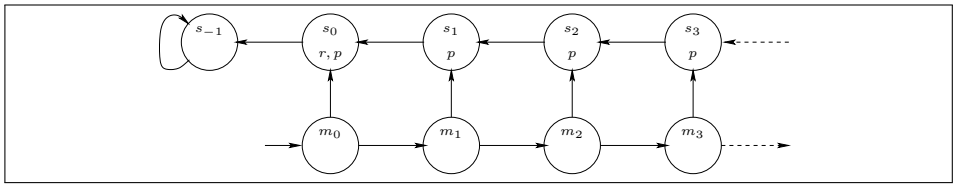


Fig. 9. A system showing that ECTL model checking is more complicated.

correspond to wrong predictions. However, in this case, this can be done by a formula transformation.

Let $\textsc{NTerm}(\varphi)$ denote the transformation in Figure 8. The transformation ensures that universal path quantification ignores states that lie on finite paths that are due to wrong estimations of the number of p-s until q. Using this transformation, it is possible to reduce fair-ACTL model checking to (A)CTL model checking over $\textsc{Fair}(M, \Omega)$. Formally, this is stated in the following theorem.

Theorem 3. *For every ACTL formula φ and every transition system M with no terminating states, we have $M \models_{\Omega_+} \varphi \Leftrightarrow \textsc{Fair}(M, \Omega) \models \textsc{NTerm}(\varphi) \vee \mathit{term}$.*

Proof Sketch (full proof in Appendix A). As before, the proof proceeds by induction on the structure of the formula. We show that if the property holds from s in M then for every n the (modified) property holds from (s, n) in $\textsc{Fair}(M, \Omega)$ and vice versa.
Note that the initial states of $\textsc{Fair}(M, \Omega)$ are all the initial states of M annotated by all possible options of $n \in \mathbb{N}$. It follows that the combination of all transformations reduce fair ACTL model checking to ACTL model checking.

Corollary 2. *For every ACTL formula φ we have*

$$M \models_{\Omega_+} \varphi \Leftrightarrow \textsc{Fair}(\textsc{Term}(M, t), \Omega) \models \textsc{NTerm}(\textsc{Term}(\varphi, t)) \vee \mathit{term}$$

Proof. As $\textsc{Term}(M, t)$ produces a transition system with no terminating states and $\textsc{Term}(\varphi, t)$ converts an ACTL formula to an ACTL formula, the proof then follows from Theorem 1 and Theorem 3.

The direct reduction presented in Theorem 3 works well for ACTL but does not work for existential properties. We now demonstrate why Fig. 3 is not sufficient to handle existential properties alone. Consider the transition system M in Figure 9, the fairness constraint $\Omega = \{(p, q)\}$, and the property $\mathsf{EG}(\neg p \wedge \mathsf{EF} r)$. One can see that $M, m_0 \models_{\Omega_+} \mathsf{EG}(\neg p \wedge \mathsf{EF} r)$. Indeed, from each

state s_i there is a unique path that eventually reaches s_0, where it satisfies r, and then continues to s_{-1}, where p does not hold. As the path visits finitely many p states it is clearly fair. So, every state m_i satisfies $\mathsf{EF}r$ by considering the path $m_i, s_i, s_{i-1}, \ldots, s_0, s_{-1}, \ldots$. Then the fair path $m_0, m_1, \ldots$ satisfies $\mathsf{EG}(\neg p \wedge \mathsf{EF}r)$. On the other hand, it is clear that no other path satisfies $\mathsf{EG}(\neg p \wedge \mathsf{EF}r)$.

Now consider the transformation $\textsc{Fair}(M, \Omega)$ and consider model checking of $\mathsf{EG}(\neg p \wedge \mathsf{EF}r)$. In $\textsc{Fair}(M, \Omega)$ there is no path that satisfies this property. To see this, consider the transition system $\textsc{Fair}(M, \Omega)$ and a value $n \in \mathbb{N}$. For every value of n the path $(m_0, n), (m_1, n), (m_2, n), \ldots$ is an infinite path in $\textsc{Fair}(M, \Omega)$ as it never visits p. This path does not satisfy $\mathsf{EG}(\neg p \wedge \mathsf{EF}r)$. Consider some state (m_j, n_j) reachable from (m_0, n) for $j > 2n$. The only infinite paths starting from (m_j, n_j) are paths that never visit the states s_i. Indeed, paths that visit s_i are terminated as they visit too many p states. Thus, for every $n \in \mathbb{N}$ we have $(m_0, n) \not\models \mathsf{EG}(\neg p \wedge \mathsf{EF}r)$. Finite paths in $\textsc{Fair}(M, \Omega)$ are those of the form $(m_0, n_0), \ldots, (m_i, n_i), (s_i, n_{i+1}), \ldots$. Such paths clearly cannot satisfy the property $\mathsf{EG}(\neg p \wedge \mathsf{EF}r)$ as the states s_i do satisfy p. Allowing existential paths to ignore fairness is clearly unsound. We note also that in $\textsc{Fair}(M, \Omega)$ we have $(m_0, n) \models \textsc{NTerm}(\mathsf{AF}(p \vee \mathsf{AG}\neg r))$.

Reducing Fair Termination to Termination.

Given the importance of termination as a system property, we emphasize the reduction of fair termination to termination. Note that termination can be expressed in ACTL as $\mathsf{AFAX}\ \mathsf{false}$, thus the results in Corollary 2 allow us to reduce fair termination to model checking (without fairness). Intuitively, a state that satisfies $\mathsf{AX}\ \mathsf{false}$ is a state with no successors. Hence, every path that reaches a state with no successors is a finite path. Here, we demonstrate that for infinite-state infinite-branching systems, fair termination can be reduced to termination.

A transition system M terminates if for every initial state $s \in S_0$ we have $\Pi_\infty(s) = \emptyset$. System M fair-terminates under fairness Ω if for every initial state $s \in S_0$ and every $\pi \in \Pi_\infty(s)$ we have $\pi \not\models \Omega$, i.e., all infinite paths are unfair.

The following corollary follows from the proof of Theorem 3, where we establish a correspondence between fair paths of M and infinite paths of $\textsc{Fair}(M, \Omega)$.

Corollary 3. *M fair terminates iff $\textsc{Fair}(M, \Omega)$ terminates.*

Recall that the reduction relies on transition systems having an infinite branching degree. For transition systems with finite-branching degree, we cannot reduce fair termination of finite-branching programs to termination of finite-branching programs, as the former is Σ^1_1-complete and the latter is RE-complete [17].

5 Example

Consider the example in Fig. 10. We will demonstrate the resulting transformations which will disprove the CTL property $\mathsf{EG}\ x \leq 0$ under the weak fairness constraint $\mathsf{GF}\ \mathsf{true} \rightarrow \mathsf{GF}\ \mathsf{y} \geq 1$ for the initial transition system introduced in (a). We begin by executing Verify in Fig. 7. In Verify the transition system in (a) is transformed according to $\textsc{Term}(M, t)$ and the CTL formula $\mathsf{EG}\ x \leq 0$

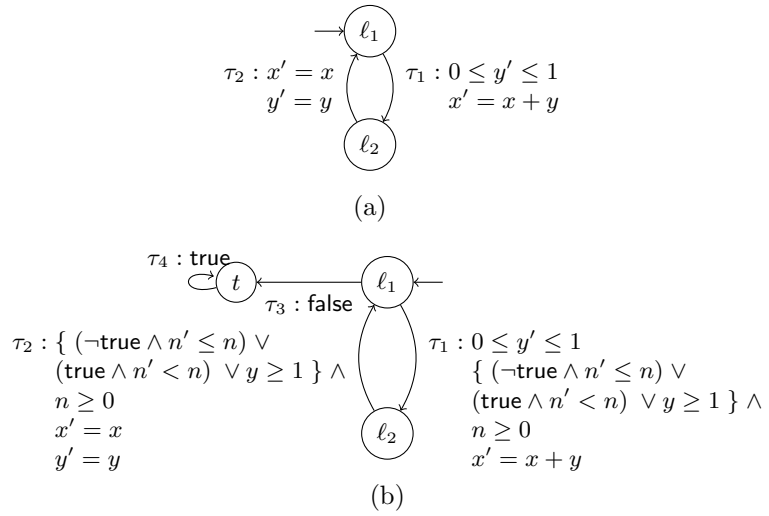


Fig. 10. Verifying a transition system with the CTL property $\mathsf{EG}\ x \leq 0$ and the weak fairness constraint $\mathsf{GF}\ \mathsf{true} \rightarrow \mathsf{GF}\ y \geq 1$. The original transition system is represented in (a), followed by the application of our fairness reduction in (b).

is transformed according to $\textsc{Term}(M, t)$, as discussed in 3.2. Our main procedure FairCTL in Fig. 6 is then called. First, we recursively enumerate over the most inner sub-property $x \leq 0$, wherein $x \leq 0$ is returned as it is our base case. In lines 13-30, a new CTL formula φ' is then acquired by adding an appropriate termination or non-termination clause. This clause allows us to ignore finite paths that are prefixes of some unfair infinite paths, that is, those that have not been marked by $\textsc{Term}(M, t)$. We then obtain (b) in Fig. 10 by applying $\textsc{Fair}(M, \Omega)$ from Fig. 3 on line 32. Thus, we must restrict each transition such that $\{ (\neg\mathsf{true} \wedge n' \leq n) \vee (\mathsf{true} \wedge n' < n) \vee y \geq 1 \} \wedge n \geq 0$ holds. This can be seen in transitions τ_1 and τ_2.

Recall that $\textsc{Fair}(M, \Omega)$ can include (maximal) finite paths that are prefixes of unfair infinite paths. We thus have to ensure that these paths do not interfere with the validity of our model checking procedure. We have shown how to distinguish between maximal (finite) paths that occur in M and those introduced by our transformation in Theorem 1. This is demonstrated by τ_3 and τ_4 in (b): in τ_3 we simply take the negation of the loop invariant (in this case it is false), as it would indicate a terminating path given that no other transitions follow the loop termination. In τ_4 we instrument a self loop and add the proposition t to eliminate all terminal states. Additionally, utilizing $\textsc{Term}(\varphi, t)$ on $\mathsf{EG}\ x \leq 0$ allows us to disregard all aforementioned marked finite paths, as we only consider infinite paths which correspond to a fair path in the original system.

On line 33, a CTL model checker is then employed with the transition system in (b) and the CTL formula φ'. We then apply tools such as Beyene *et al.* [2] and Cook *et al.* [9]to the transformation introduced to verify CTL for infinite-state systems. An assertion characterizing the states in which φ' holds is returned and then further examined on lines 36 and 37, where it is discovered that this

Program	LOC	Property	FC	Time(s)	Result
WDD1	20	AG(BlockInits() ⇒ AF UnblockInits())	Yes	14.4	✓
WDD1	20	AG(BlockInits() ⇒ AF UnblockInits())	No	2.1	χ
WDD2	374	AG(AcqSpinLock() ⇒ AF RelSpinLock())	Yes	18.8	✓
WDD2	374	AG(AcqSpinLock() ⇒ AF RelSpinLock())	No	14.1	χ
WDD3	58	AF(EnCritRegion() ⇒ EG ExCritRegion())	Yes	12.5	χ
WDD3	58	AF(EnCritRegion() ⇒ EG ExCritRegion())	No	9.6	✓
WDD4	302	AG(added_socket > 0 ⇒ AFEG STATUS_OK)	Yes	30.2	✓
WDD4	302	AG(added_socket > 0 ⇒ AFEG STATUS_OK)	No	72.4	χ
Bakery	37	AG(Noncritical ⇒ AF Critical)	Yes	2.9	✓
Bakery	37	AG(Noncritical ⇒ AF Critical)	No	16.4	χ
Prod-Cons	30	AG(p_i > 0 ⇒ AF q_i <= 0)	Yes	18.5	✓
Prod-Cons	30	AG(p_i > 0 ⇒ AF q_i <= 0)	No	5.5	χ
Chain	48	AG(x ≥ 8 ⇒ AF x = 0)	Yes	1.8	✓
Chain	48	AG(x ≥ 8 ⇒ AF x = 0)	No	4.7	χ

Fig. 11. Experimental evaluations of infinite-state programs such as Windows device drivers (WDD) and concurrent systems, which were reduced to non-deterministic sequential programs via [6]. Each program is tested for both the success of a branching-time liveness property with a fairness constraint and its failure due to a lack of fairness. A ✓ represents the existence of a validity proof, while χ represents the existence of a counterexample. We denote the lines of code in our program by LOC and the fairness constraint by FC. There exist no competing tools available for comparison.

property does not hold due to the restrictive fairness constraint applied to the existential CTL property. The weak fairness constraint requires that infinitely often $y \geq 1$ holds, which interferes with the existential property that $\mathsf{EG}\ x \leq 0$. This shows that for the existential fragment of CTL, fairness constraints restrict the transition relations required to prove an existential property. This can be beneficial when attempting to disprove systems and their negations.

6 Experiments

In this section we demonstrate the results of preliminary experiments with a prototype implementation. We applied our tool to several small programs: a classical mutual exclusion algorithm as well as code fragments drawn from device drivers. Our implementation is based on an extension to T2 [3, 13, 9].[4] As discussed in the related work, there are currently no known tools supporting fair-CTL for infinite-state systems, thus we are not able to make experimental comparisons.

Fig. 11 shows experimental evaluations of sequential Windows device drivers (WDD) and various concurrent systems [5]. WDD1 uses the fairness constraint $\mathsf{GF}(\texttt{IoCreateDevice.exit}\{1\}) \Rightarrow \mathsf{GF}(\texttt{status} = \texttt{SUCCESS})$, while WDD2 and 3 utilize the same fairness constraint in relation to checking the acquisition and release of spin locks and the entrance and exit of critical regions, respectively. WDD4 requires a weak fairness constraint indicating that `STATUS_OK` will hold a value of `true` infinitely often, that is, whenever sockets are successfully opened, the server will eventually return a successful status infinitely often.

[4] T2 can be acquired at `http://research.microsoft.com/en-us/projects/t2/`

[5] Benchmarks can be found at `http://heidyk.com/experiments.html`

Note that the initially concurrent programs are reduced to sequential programs via [6], which uses rely-guarantee reasoning to reduce multi-threaded verification to liveness. We verify the traditional Bakery algorithm, requiring that any thread requesting access to the critical region will eventually be granted the right to do so. The producer-consumer algorithm requires that any amount of input data produced, must be eventually consumed. The Chain benchmark consists of a chain of threads, where each thread decreases its own counter, but the next thread in the chain can counteract, and increase the counter of the previous thread, thus only the last thread in the chain can be be decremented unconditionally. These algorithms are verified on 2, 4, and 8 threads, respectively.

For the the existential fragment of CTL, fairness constraints can often restrict the transition relations required to prove an existential property, as demonstrated by WDD3. For universal CTL properties, fairness policies can assist in enforcing properties to hold that previously did not. Thus, our tool allows us to both prove and disprove the negation of each of the properties.

7 Discussion

We have described the first-known fair-CTL model checking technique for integer based infinite-state programs. Our approach is based on a reduction to existing techniques for fairness-free CTL model checking. The reduction relies on utilizing prophecy variables to introduce additional information into the state-space of the program under consideration. This allows fairness-free CTL proving techniques to reason only about fair executions. Our implementation seamlessly builds upon existing CTL proving techniques, resulting in experiments which demonstrate the practical viability of our approach.

Furthermore, our technique allows us to bridge between linear-time (LTL) and branching-time (CTL) reasoning. Not only so, but a seamless integration between LTL and CTL reasoning may make way for further extensions supporting CTL* verification of infinite-state programs [15]. We hope to further examine both the viability and practicality of such an extension.

We include the definition of fair-CTL considering only infinite paths and show how to change transition systems to use either definition in our technical report which can be acquired at [8]. Additionally, we show how to modify the proof system to incorporate an alternative approach to CTL verification advocated by Cook & Koskinen [11].

References

1. K. Apt and E. Olderog. Fairness in parallel programs: The transformational approach. *ACM Transactions on Programming Languages and Systems*, 10, 1988.
2. T. Beyene, C. Popeea, and A. Rybalchenko. Solving existentially quantified horn clauses. In *CAV '13*, LNCS, 2013.
3. M. Brockschmidt, B. Cook, and C. Fuhs. Better termination proving through cooperation. In *CAV '13*, LNCS, 2013.
4. A. Cimatti, E.M. Clarke, G. Enrico, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *CAV '02*, LNCS, 2002.

5. E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2), April 1986.
6. B. Cook, A. Gotsman, M. Parkinson, and V. Vafeiadis. Proving that non-blocking algorithms don't block. In *POPL '09*. ACM, 2009.
7. B. Cook, A. Gotsman, A. Podelski, A. Rybalchenko, and M. Vardi. Proving that programs eventually do something good. In *POPL '07*. ACM, 2007.
8. B. Cook, Khlaaf H, and N. Piterman. Fairness for infinite-state systems. Technical Report RN/14/11, University College London, 2014.
9. B. Cook, H. Khlaaf, and N. Piterman. Faster temporal reasoning for infinite-state programs. In *FMCAD'14*, To Appear.
10. B. Cook and E. Koskinen. Making prophecies with decision predicates. In *POPL '11*. ACM, 2011.
11. B. Cook and E. Koskinen. Reasoning about nondeterminism in programs. In *PLDI '13*. ACM, 2013.
12. B. Cook, E. Koskinen, and M. Vardi. Temporal property verification as a program analysis task. In *CAV '11*, LNCS, 2011.
13. B. Cook, A. See, and F. Zuleger. Ramsey vs. lexicographic termination proving. In *TACAS'13*, LNCS, 2013.
14. A. David, J. Håkansson, K.G. Larsen, and P. Pettersson. Model checking timed automata with priorities using dbm subtraction. In *FORMATS'06*, 2006.
15. E.A. Emerson and J.Y Halpern. "sometimes" and "not never" revisited: On branching versus linear time temporal logic. *J. ACM*, 33(1), January 1986.
16. E.A. Emerson and C.-L. Lei. Temporal reasoning under generalized fairness constraints. In *3rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 210 of *LNCS*, pages 21–36. Springer, 1986.
17. D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes and fairness. *Journal of the ACM*, 33:224–248, 1986.
18. M. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In *CAV'11*, LNCS, 2011.
19. A. Pnueli, A. Podelski, and A. Rybalchenko. Separating fairness and well-foundedness for the analysis of fair discrete systems. In *TACAS'05*, LNCS, 2005.
20. A. Podelski and A. Rybalchenko. Transition invariants. In *LICS '04*, 2004.
21. S. Schwoon. Moped - A Model-Checker for Pushdown Systems. `http://www7.in.tum.de/~schwoon/moped`, 2002.
22. F. Song and T. Touili. Pushdown model checking for malware detection. In *ESEC/FSE 2013*, 2013.
23. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Inf. Comput.*, 115(1):1–37, 1994.

A Appendix: Proofs

We prove Theorem 1:

Proof.

$\Leftarrow$ $\textsc{Term}(M,t), s \models_{\Omega_+} \textsc{Term}(\varphi,t) \Rightarrow M, s \models_{\Omega_+} \varphi$.

Base case: Consider an atomic proposition $\alpha \neq t$, assume $\textsc{Term}(M,t), s \models_{\Omega_+} \textsc{Term}(\alpha,t)$. Given that $\textsc{Term}(\alpha,t) = \alpha$, we now have $\textsc{Term}(M,t), s \models_{\Omega_+} \alpha$ and $M, s \models_{\Omega_+} \alpha$. These statements are equivalent.

Induction hypothesis: For every state s and subformula φ (of lower temporal height) assume $\textsc{Term}(M,t), s \models_{\Omega_+} \textsc{Term}(\varphi,t) \Rightarrow M, s \models_{\Omega_+} \varphi$.

Proof by structural induction:

1. $\textsc{Term}(M,t), s \models_{\Omega_+} \textsc{Term}(\mathsf{AX}\varphi,t) \Rightarrow M, s \models_{\Omega_+} \mathsf{AX}\varphi$.
   Given that $\textsc{Term}(\mathsf{AX}\varphi_1,t) = t \vee \mathsf{AX}(\textsc{Term}(\varphi,t))$ we now have $\textsc{Term}(M,t), s \models_{\Omega_+} t \vee \mathsf{AX}(\textsc{Term}(\varphi,t))$. That is either $\textsc{Term}(M,t), s \models_{\Omega_+} t$ or $\textsc{Term}(M,t), s \models_{\Omega_+} \mathsf{AX}(\textsc{Term}(\varphi,t))$. Consider the case $\textsc{Term}(M,t)$, $s \models_{\Omega_+} t$, denoting that s is a terminal state in M since self loops are added to terminating states with the proposition t being set to true in them. Under the semantics of $\models_{\Omega_+}$ a terminating state would have no successors, the property $\mathsf{AX}\varphi$ then holds. Now consider that $\textsc{Term}(M,t), s \not\models t$, meaning $\textsc{Term}(M,t), s \models_{\Omega_+} \mathsf{AX}(\textsc{Term}(\varphi,t))$. We have $\forall \pi = (s_0, s_1, \ldots)$. $\textsc{Term}(M,t), s_1 \models_{\Omega_+} \textsc{Term}(\varphi,t)$. As $\textsc{Term}(M,t)$, $s \not\models t$ this implies $\forall \pi = (s_0, s_1, \ldots)$. $M, s_1 \models_{\Omega_+} \varphi$.
2. $\textsc{Term}(M,t), s \models_{\Omega_+} \textsc{Term}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2],t) \Rightarrow M, s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$.
   Recall that $\textsc{Term}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2],t) = \mathsf{A}[\textsc{Term}(\varphi_1,t) \mathsf{W} \textsc{Term}(\varphi_2,t)]$. Consider a maximal path $\pi = (s_0, s_1, \ldots)$ starting at s in M. If $\pi \in \Pi_\infty$ and π is fair, then π is in $\textsc{Term}(M,t)$ as well and by $\textsc{Term}(M,t), s \models_{\Omega_+} \mathsf{A}[\textsc{Term}(\varphi_1,t) \mathsf{W} \textsc{Term}(\varphi_2,t)]$ we know either $\exists j \geq 0$. $\textsc{Term}(M,t), s_j \models_{\Omega_+} \textsc{Term}(\varphi_2,t) \wedge \forall i \in [0,j)$. $\textsc{Term}(M,t), s_i \models_{\Omega_+} \textsc{Term}(\varphi_1,t)$ or $\forall i \geq 0$ $\textsc{Term}(M,t), s_j \models_{\Omega_+} \textsc{Term}(\varphi_1,t)$. Using our induction hypothesis we then have either $M, s_j \models_{\Omega_+} \varphi_2 \wedge \forall i \in [0,j)$. $M, s_i \models_{\Omega_+} \varphi_1$ or $\forall i \geq 0$. $M, s_i \models_{\Omega_+} \varphi_2$. If $\pi \in \Pi_f$ then $\pi = (s_0, s_1, \ldots, s_n)$ and $\pi' = (s_0, s_1, \ldots, s_n, s_{n+1}, s_{n+2}, \ldots)$, where for every $l > 0$ we have $s_{n+l} = s_n$, is a path in $\textsc{Term}(M,t)$. As $\textsc{Term}(M,t), s \models_{\Omega_+} \mathsf{A}[\textsc{Term}(\varphi_1,t) \mathsf{W} \textsc{Term}(\varphi_2,t)]$ then either there is a j such that $\textsc{Term}(M,t), s_j \models_{\Omega_+} \textsc{Term}(\varphi_2,t)$ and $\forall i \in [0,j)$. $\textsc{Term}(M,t), s_i \models_{\Omega_+} \textsc{Term}(\varphi_1,t)$ or $\forall i \in [0,|\pi|)$. $\textsc{Term}(M,t), s_i \models_{\Omega_+} \textsc{Term}(\varphi_2,t)$. In the first case, if $j < n$, then clearly π satisfies $\varphi_1 \mathsf{W} \varphi_2$. If $j \geq n$ then $s_j = s_n$ and then $\textsc{Term}(M,t), s_n \models_{\Omega_+} \textsc{Term}(\varphi_2,t) \wedge \forall i \in [0,n)$. $\textsc{Term}(M,t)$, $s_i \models_{\Omega_+} \textsc{Term}(\varphi_1,t)$ implying $M, s_n \models_{\Omega_+} \varphi_2 \wedge \forall i \in [0,n)$. $M, s_i \models_{\Omega_+} \varphi_1$. In the second case then forall $i \in [0,|\pi|)$. $M, s_i \models_{\Omega_+} \varphi_1$. It follows that $M, s \models_{\Omega_+} \mathsf{A}\varphi_1 \mathsf{W} \varphi_2$.
3. $\textsc{Term}(M,t), s \models_{\Omega_+} \textsc{Term}(\mathsf{EX}\varphi,t) \Rightarrow M, s \models_{\Omega_+} \mathsf{EX}\varphi$.
   Given that $\textsc{Term}(\mathsf{EX}\varphi,t) = \neg t \wedge \mathsf{EX}(\textsc{Term}(\varphi,t))$ we now have $\textsc{Term}(M,t), s \models_{\Omega_+} \neg t \wedge \mathsf{EX}(\textsc{Term}(\varphi,t))$. First, consider $\textsc{Term}(M,t), s \models_{\Omega_+}$

$\mathsf{EX}(\textsc{Term}(\varphi, t))$, which implies $\exists \pi = (s_0, s_1, \ldots)$. $\textsc{Term}(M, t), s_1 \models_{\Omega_+} \textsc{Term}(\varphi, t)$. Now consider $\textsc{Term}(M, t), s \models_{\Omega_+} \neg t$, that is, the proposition $\neg t$ indicates that (s, s_1) was not an instrumented transition. Given that we only instrument transitions in terminal states, $\neg t$ guarantees that s_1 is a successor of s as s is a non-terminal state in M. Recall our assumption that $\forall s'.\ \textsc{Term}(M, t), s' \models_{\Omega_+} \textsc{Term}(\varphi, t) \Rightarrow M, s' \models_{\Omega_+} \varphi$ and $(s, s_1) \in R$ thus $M, s_1 \models_{\Omega_+} \varphi$.

4. A similar proof to AW follows for AF, EU, and EG.

$\Rightarrow$ $M, s \models_{\Omega_+} \varphi \Rightarrow \textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\varphi, t)$.

Base case: For an atomic proposition α, $\forall s.\ M, s \models_{\Omega_+} \alpha \Rightarrow \textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\alpha, t)$. Given that $\textsc{Term}(\alpha, t) = \alpha$, we now have $\textsc{Term}(M, t), s \models_{\Omega_+} \alpha$ and $M, s \models_{\Omega_+} \alpha$. Under the semantics of both $\models_{\Omega_+}$ and $\models_{\Omega_+}$, these statements are equivalent.

Induction hypothesis: For every state s and every formula φ (of lower temporal height) assume $M, s \models_{\Omega_+} \varphi \Rightarrow \textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\varphi, t)$.

Proof by structural induction:

1. $M, s \models_{\Omega_+} \mathsf{AX}\varphi \Rightarrow \textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\mathsf{AX}\varphi, t)$.
   We expand our formula to $M \models_{\Omega_+} \mathsf{AX}\varphi \Rightarrow \textsc{Term}(M, t) \models_{\Omega_+} t \vee \mathsf{AX}\textsc{Term}(\varphi, t)$. Consider a state s in M that is a non-terminal state, clearly it is also a non-terminating state in $\textsc{Term}(M, t)$. By $M \models_{\Omega_+} \mathsf{AX}\varphi$ then for every s_1 such that $(s, s_1) \in R$ we have $M, s_1 \models_{\Omega_+} \varphi$. As s is not terminal for every s_1 such that $(s, s_1) \in R'$ we have $(s, s_1) \in R$ and hence $\textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\mathsf{AX}\varphi, t)$. In the case that s is a terminal state in M, self loops were added to terminating states with the proposition t being set to true in $\textsc{Term}(M, t)$ in s. Then t holds in s in $\textsc{Term}(M, t)$, allowing $\textsc{Term}(M, t), s \models_{\Omega_+} t$ implying $\textsc{Term}(M, t) \models_{\Omega_+} t \vee \mathsf{AX}\textsc{Term}(\varphi, t)$.
2. $M, s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2] \Rightarrow \textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2], t)$.
   Recall that $\textsc{Term}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2], t) = \mathsf{A}[\textsc{Term}(\varphi_1, t) \mathsf{W} \textsc{Term}(\varphi_2, t)]$. Consider a path π' starting at s in $\textsc{Term}(M, t)$, if π' is of the form $\pi' = (s_0, s_1, \ldots, s_n, s_{n+1}, s_{n+2}, \ldots)$ such that for every $l > 1$ we have $s_{n+l} = s_n$ and $s_n \models_{\Omega_+} t$, then $\pi = (s_0, s_1, \ldots, s_n)$ is a maximal path in M since terminating paths are instrumented as self-loops in $\textsc{Term}(M, t)$ and marked with the proposition t. If $M, s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$ then either there is a $j \leq n$ such that $M, s_j \models_{\Omega_+} \varphi_2$ and $\forall i \in [0, j).\ M, s_i \models_{\Omega_+} \varphi_1$ or $\forall i \in [0, n].\ M, s_i \models_{\Omega_+} \varphi_1$. By our induction hypothesis in the first case we have $\textsc{Term}(M, t), s_j \models_{\Omega_+} \textsc{Term}(\varphi_2, t) \wedge \forall i \in [0, j).\ \textsc{Term}(M, t), s_i \models_{\Omega_+} \textsc{Term}(\varphi_1, t)$ and in the second case we have $\forall i \in [0, n].\ \textsc{Term}(M, t), s_i \models \textsc{Term}(\varphi_1, t)$. If π' is of the form $\pi' = (s_0, s_1, \ldots)$ such that the proposition t does not hold anywhere along the path, then $\pi = (s_0, s_1, \ldots)$ is an infinite path in M and by $M, s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$ we know that either $\exists j \geq 0.\ M, s_j \models_{\Omega_+} \varphi_2 \wedge \forall i \in [0, j).\ M, s_i \models_{\Omega_+} \varphi_1$ or $\forall i \geq 0.\ M, s_i \models_{\Omega_+} \varphi_1$. Using our induction hypothesis we then have that either $\exists j \geq 0.\ \textsc{Term}(M, t), s_j \models_{\Omega_+} \textsc{Term}(\varphi_2, t) \wedge \forall i \in [0, j).\ \textsc{Term}(M, t)$,

$s_i \models_{\Omega_+} \textsc{Term}(\varphi_1, t)$ or $\forall i \geq 0$. $\textsc{Term}(M, t) \models \textsc{Term}(\varphi_1, t)$. It follows that $\textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\mathsf{A}[\varphi_1 \mathsf{W} \varphi_2], t)$.

3. $M, s \models_{\Omega_+} \mathsf{EX}\varphi \Rightarrow \textsc{Term}(M, t), s \models_{\Omega_+} \textsc{Term}(\mathsf{EX}\varphi, t)$.
   We expand our formula to $M \models_{\Omega_+} \mathsf{EX}\varphi \Rightarrow \textsc{Term}(M, t) \models_{\Omega_+} \neg t \wedge \mathsf{EX}\textsc{Term}(\varphi, t)$. By $M, s \models_{\Omega_+} \mathsf{EX}\varphi$ we know that there is s_1 such that $(s, s_1) \in R$ and $M, s_1 \models_{\Omega_+} \varphi$, hence s is not terminal in M and $\textsc{Term}(M, t)$, $s \models_{\Omega_+} \neg t$. Through our induction hypothesis it then follows that $\textsc{Term}(M, t), s_1 \models_{\Omega_+} \textsc{Term}(\varphi, t)$. Both arguments of the conjunction are thus satisfied hence $\textsc{Term}(M, t), s \models_{\Omega_+} \neg t \wedge \mathsf{EX}\textsc{Term}(\varphi, t)$.
4. A similar proof to AW follows for AF, EU, and EG.

We prove Theorem 2:

Proof. First, we demonstrate that every infinite path in $\textsc{Fair}(M, \Omega)$ starting in (s, n) for some $n \in \boldsymbol{N}$ corresponds to an infinite path in M starting in s satisfying Ω. Consider an infinite path $\pi = (s_0, n_0), (s_1, n_1), \ldots$ in $\textsc{Fair}(M, \Omega)$. Let $\pi' = s_0, s_1, \ldots$ be it's projection in M. Suppose that π' is unfair. That is, states of the form (s, n) where $s \in p$ occur infinitely often along π but states of the form (s, n) where $s \in q$ occur only finitely often. This implies that for infinitely many i we have $n_i > n_{i+1}$. However, $\forall i \geq 0$ we have $n \geq 0$. As the only conjunct in R_Ω that allows n to increase requires q to hold it follows that this is a contradiction.

We demonstrate that every fair path in M according to Ω starting in s corresponds to an infinite path in $\textsc{Fair}(M, \Omega)$ starting in (s, n) for some $n \in \boldsymbol{N}$. Consider an infinite path $\pi = s_0, s_1, \ldots$ in M such that $\pi \models \Omega$. If there are finitely many occurrences of states from p in π then n_0 be the number of such occurrences plus 1. Then, the path $\pi' = (s_0, n_0), (s_1, n_1), \ldots$ where n_{i+1} is $n_i - 1$ if $s_i \in p$ and $n_{i+1} = n_i$ otherwise is a path in $\textsc{Fair}(M, \Omega)$. Indeed, every transition such that $s_i \in p$ satisfies $p \wedge n' < n$ and every other transition satisfies $n' \leq n$.

From this correspondence of fair paths in M and infinite paths in $\textsc{Fair}(M, \Omega)$, we can safely disregard all the newly introduced finite paths given a transition system with no finite paths (i.e., $\textsc{Term}(M, t)$).

We now turn to the main theorem. We prove it by induction on the structure of the formula. Namely, we show that the assertion returned by $\textsc{Fair}\mathrm{CTL}(M, \Omega, \varphi)$ characterizes the set of states of M that satisfy φ. We note that for atomic propositions and Boolean operators the proof is immediate.

$\Rightarrow$ *Induction hypothesis:* Let a_i be the assertion characterizing the set of states satisfying φ_i. For every state s and every formula φ_i (of lower temporal height) assume $M, s \models \varphi_i \Rightarrow s \rightarrow a_i$ for $i \in \{1, 2\}$.

1. Consider the case that $\varphi = \mathsf{AX}\varphi_1$. Suppose that $M, s \models_{\Omega_+} \mathsf{AX}\varphi_1$. We have to show that for every $n \geq 0$ we have $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{AX}(a_1 \vee \mathsf{term})$.
   Consider a state (s, n) such that (s, n) has no successors. Then clearly, $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{AX}(a_1 \vee \mathsf{term})$.
   Consider a state (s, n) such that (s, n) has some successors. Let (s_1, n_1) be some successor of (s, n). If $(s_1, n_1) \models \mathsf{term}$ then we are done. Otherwise, there is an infinite path $(s_1, n_1), (s_2, n_2), \ldots$ in $\textsc{Fair}(M, \Omega)$. By

the correspondence established above it must be the case that $s_1, s_2, \ldots$ is a fair path in M. Thus, as s_1 is a successor of s in M and $s, s_1, \ldots$ is a fair path in M it must be the case that $s_1 \models \varphi_1$. Hence, by induction, $s_1 \models a_1$ and we are done.
As n was arbitrary it follows that this holds for every possible value of n.

2. Consider the case that $\varphi = \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$. Suppose that $M, s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$. We have to show that for every $n \geq 0$ we have $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{A}[a_1 \mathsf{W}(a_2 \vee \mathsf{term})]$.
Consider a state (s, n) such that (s, n) has no infinite paths starting from it. Then clearly, $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{term}$ and we are done.
Consider a state (s, n) such that (s, n) has some infinite path starting from it. Let $\pi = (s, n), (s_1, n_1), (s_2, n_2), \ldots$ be a maximal path in $\textsc{Fair}(M, \Omega)$.
If π is infinite then by the correspondence established above it must be the case that $\pi' = s, s_1, s_2, \ldots$ is a fair path in M. As $M, s \models_{\Omega_+} \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$ it must be the case that π' satisfies $\varphi_1 \mathsf{W} \varphi_2$. Then, it must be the case that π satisfies $a_1 \mathsf{W} a_2$.
If π is finite then let (s_i, n_i) be the last state on π such that from (s_i, n_i) there is some infinite path. Let $(s, n), (s_1, n_1), \ldots, (s_i, n_i), (s'_{i+1}, n'_{i+1}), \ldots$ be this infinite path. Then $s, s_1, \ldots, s_i, s'_{i+1}, \ldots$ is a fair path in M and it must satisfy $\varphi_1 \mathsf{W} \varphi_2$. If φ_2 is satisfied on or before i then π satisfies $a_1 \mathsf{W} a_2$. Otherwise, all states $s, s_1, \ldots, s_i$ satisfy φ_1. It follows that state (s_{i+1}, n_{i+1}) satisfies term in $\textsc{Fair}(M, \Omega)$ and hence π satisfies $a_1 \mathsf{W}(a_2 \vee \mathsf{term})$.
As n was arbitrary it follows that this holds for every possible value of n.

3. The case of $\varphi = \mathsf{AF}\varphi_1$ is similar to the above.

4. Consider the case that $\varphi = \mathsf{EX}\varphi_1$. Suppose that $M, s \models_{\Omega_+} \mathsf{EX}\varphi_1$. We have to find some $n \geq 0$ such that $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{EX}(a_1 \wedge \neg \mathsf{term})$. By assumption, there is a fair path $s, s_1, s_2, \ldots$ in M such that $s_1 \models \varphi_1$. Recall the correspondence between infinite paths in $\textsc{Fair}(M, \Omega)$ and fair paths in M. Then, there is an infinite path $(s, n), (s_1, n_1), (s_2, n_2), \ldots$ in $\textsc{Fair}(M, \Omega)$. Thus, state (s_1, n_1) satisfies $\neg \mathsf{term}$. As $s_1 \to a_1$, it must be the case that $(s, n) \models \mathsf{EX}(a_1 \wedge \neg \mathsf{term})$.

5. Consider the case that $\varphi = \mathsf{E}[\varphi_1 \mathsf{U} \varphi_2]$. Suppose that $M, s \models_{\Omega_+} \mathsf{E}[\varphi_1 \mathsf{U} \varphi_2]$. Let a_1 be the assertion characterizing the set of states satisfying φ_1 and a_2 the assertion characterizing the set of states satisfying φ_2. By induction for every state s we have $M, s \models \varphi_i \Leftrightarrow s \to a_i$, for $i \in \{1, 2\}$.
We have to find some $n \geq 0$ such that $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{E}[a_1 \mathsf{U}(a_2 \wedge \neg \mathsf{term})]$.
By assumption, there is a fair path $\pi' = s, s_1, s_2, \ldots$ in M such that $\pi' \models \varphi_1 \mathsf{U} \varphi_2$. Recall the correspondence between infinite paths in $\textsc{Fair}(M, \Omega)$ and fair paths in M. Then, there is an infinite path $\pi = (s, n), (s_1, n_1), (s_2, n_2), \ldots$ in $\textsc{Fair}(M, \Omega)$. Clearly, due to π we have that $(s, n) \models \neg \mathsf{term}$ and sim-

ilarly for every state on this path. As $s_i \to a_j$ iff $M, s_i \models \varphi_j$ the claim follows.

6. The case of $\varphi = \mathsf{EG}\varphi_1]$ is similar.

$\Leftarrow$ *Induction hypothesis:* Let a_i be the assertion characterizing the set of states satisfying φ_i. For every state s and every formula φ_i (of lower temporal height) assume $M, s \models \varphi_i \Leftarrow s \to a_i$ for $i \in \{1, 2\}$.

1. Consider the case that $\varphi = \mathsf{AX}\varphi_1$. Suppose that in $\textsc{Fair}(M, \Omega)$ we have that for every $n \geq 0$ we have $(s, n) \models \mathsf{AX}(a_1 \vee \mathsf{term})$.
   We have to show that $s \models \mathsf{AX}\varphi_1$.
   If there are no fair paths starting in s then this clearly holds. Consider a fair path $\pi' = s, s_1, s_2, \ldots$ in M. Recall the correspondence established above between fair paths in M and infinite paths in $\textsc{Fair}(M, \Omega)$. We conclude that $\pi = (s, n), (s_1, n_1), (s_2, n_2), \ldots$ is an infinite path in $\textsc{Fair}(M, \Omega)$. Hence, (s_1, n_1) cannot satisfy term.
   It must be the case that s_1 satisfies a_1 and the claim follows.
2. Consider the case that $\varphi = \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$. Suppose that for every value of n we have $(s, n) \models \mathsf{A}[a_1 \mathsf{W} (a_2 \vee \mathsf{term})]$.
   We have to show that $s \models \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$.
   If there are no fair paths starting in s then this clearly holds. Consider a fair path $\pi' = s, s_1, s_2, \ldots$ in M. Recall the correspondence established above between fair paths in M and infinite paths in $\textsc{Fair}(M, \Omega)$. We conclude that $\pi = (s, n), (s_1, n_1), (s_2, n_2), \ldots$ is an infinite path in $\textsc{Fair}(M, \Omega)$. Hence, every state on π cannot satisfy term and we conclude that π satisfies $a_1 \mathsf{W} a_2$.
   It must be the case that as π' satisfies $\varphi_1 \mathsf{W} \varphi_2$ as required.
3. The the case of $\varphi = \mathsf{AF}\varphi_1$ is similar.
4. Consider the case that $\varphi = \mathsf{EX}\varphi_1$. Suppose that there is some n such that $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{EX}(a_1 \wedge \neg\mathsf{term})$.
   We have to show that $M, s \models_{\Omega_+} \mathsf{EX}\varphi_1$.
   As $(s, n) \models \mathsf{EX}(a_1 \wedge \neg\mathsf{term})$ it follows that there is an infinite path $\pi = (s, n), (s_1, n_1), \ldots$ such that $(s_1, n_1) \models a_1$.
   Recall the correspondence between fair paths in M and infinite paths in $\textsc{Fair}(M, \Omega)$. We conclude that $\pi' = s, s_1, \ldots$ is a fair path in M and $s_1 \models \varphi_1$.
   Thus, $s \models \mathsf{EX}\varphi_1$.
5. Consider the case that $\varphi = \mathsf{E}[\varphi_1 \mathsf{U} \varphi_2]$. Suppose that for some $n \geq 0$ we have $\textsc{Fair}(M, \Omega), (s, n) \models \mathsf{E}[a_1 \mathsf{U} (a_2 \wedge \neg\mathsf{term})]$.
   We have to show that $M, s \models_{\Omega_+} \mathsf{E}[\varphi_1 \mathsf{U} \varphi_2]$.
   There is a path $(s, n), (s_1, n_1), \ldots, (s_i, n_i)$ that satisfies $a_1 \mathsf{U} (a_2 \wedge \neg\mathsf{term})$ such that $s, s_1, \ldots, s_{i-1}$ satisfy a_1 and s_i satisfies $a_2 \wedge \neg\mathsf{term}$. From (s_i, n_i) satisfying $\neg\mathsf{term}$ there is an infinite path $(s, n), (s_1, n_1), \ldots, (s_i, n_i), (s_{i+1}, n_{i+1}), \ldots$ in $\textsc{Fair}(M, \Omega)$.
   Recall the correspondence between fair paths in M and infinite paths in $\textsc{Fair}(M, \Omega)$. We conclude that $\pi' = s, s_1, \ldots, s_i, s_{i+1}, \ldots$ is a fair path in M such that $s, s_1, \ldots, s_{i-1}$ satsify φ_1 and s_i satisfies φ_2.

6. The case of $\varphi = \mathsf{EG}\varphi_1$ is similar.

We prove Theorem 3:

Proof. We start with an auxiliary claim.

Claim. For every ACTL formula φ, every transition system with no terminating states M, and every state s of M if for infinitely many $n \in \boldsymbol{N}$ we have $\textsc{Fair}(M, \Omega), (s, n) \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$ then $\forall n \in \boldsymbol{N}$ we have $\textsc{Fair}(M, \Omega), (s, n) \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$.

We prove this claim by induction on the structure of the formula. It holds trivially for propositions and Boolean combinations of formulas.

1. Consider the case that $\varphi = \mathsf{AX}\varphi_1$. We recall that $\textsc{NTerm}(\varphi)$ is $\mathsf{AX}\varphi_1 \vee \mathsf{term}$. Suppose that for infinitely many n we have $\textsc{Fair}(M, \Omega) \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. Consider a successor s' of s and a value $n' \in \boldsymbol{N}$. There is $n'' > n$ such that $\textsc{Fair}(M, \Omega) \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. If $\textsc{Fair}(M, \Omega), (s, n'') \models \mathsf{term}$ then clearly $\textsc{Fair}(M, \Omega), (s', n') \models \mathsf{term}$ as well. Otherwise, $\textsc{Fair}(M, \Omega), (s', n') \models \varphi_1 \vee \mathsf{term}$. Thus, for infinitely many n' we have $\textsc{Fair}(M, \Omega), (s', n') \models \varphi_1 \vee \mathsf{term}$ and by assumption it follows that this holds for all values of n.
   As s' was arbitrarily chosen it follows that the same holds for every successor of s'. Thus, for every $n' \in \boldsymbol{N}$ we have $\textsc{Fair}(M, \Omega), (s, n') \models \varphi \vee \mathsf{term}$.
2. Consider the case that $\varphi = \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$. We recall that $\textsc{NTerm}(\varphi)$ is $\mathsf{A}[\textsc{NTerm}(\varphi_1) \mathsf{W} (\textsc{NTerm}(\varphi_2) \vee \mathsf{term})]$. Suppose that for infinitely many n we have $\textsc{Fair}(M, \Omega), (s, n) \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. Consider a value n' and assume by contradiction that $\textsc{Fair}(M, \Omega), (s, n') \not\models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. It follows that (s, n') is not terminating. Hence, there is an infinite path π starting in (s, n) that does not satisfy $\textsc{NTerm}(\varphi)$. By assumption there is an $n'' > n$ such that $\textsc{Fair}(M, \Omega), (s, n'') \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. The path π' that is identical to π except that it starts in (s, n'') instead of in (s, n') is also an infinite path from (s, n''). It follows that (s, n'') is not terminating. And the path π' does satisfy $\textsc{NTerm}(\varphi)$. The only difference between π and π' is the initial state (and in it the value of n). It follows that either $(s, n'') \models \textsc{NTerm}(\varphi_1)$ or $(s, n'') \models \textsc{NTerm}(\varphi_2)$. In either case, we can find infinitely many different $n''' > n''$ such that the same thing will hold. It follows by induction that for infinitely many value n''' we have $(s, n''') \models \textsc{NTerm}(\varphi_1)$ or $(s, n''') \models \textsc{NTerm}(\varphi_2)$. In either case, the same follows for all values of n and we conclude that π cannot be a counter example to $\textsc{NTerm}(\varphi_1)$.
3. Consider the case that $\varphi = \mathsf{AF}\varphi_1$. We recall that $\textsc{NTerm}(\varphi)$ is $\mathsf{AF}(\textsc{NTerm}(\varphi_1) \vee \mathsf{term})$. Suppose that for infinitely many n we have $\textsc{Fair}(M, \Omega), (s, n) \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. Consider a value n' and assume by contradiction that $\textsc{Fair}(M, \Omega), (s, n') \not\models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. It follows that (s, n') is not terminating. Hence, there is an infinite path π starting in (s, n) that does not satisfy $\textsc{NTerm}(\varphi)$. By assumption there is an $n'' > n$ such that $\textsc{Fair}(M, \Omega), (s, n'') \models \textsc{NTerm}(\varphi) \vee \mathsf{term}$. The path π' that is identical to π except that it starts in (s, n'') instead of in (s, n') is also an infinite path from (s, n''). It follows that (s, n'') is not terminating. And the path π' does satisfy $\textsc{NTerm}(\varphi)$. The

only difference between π and π' is the initial state (and in it the value of n). It follows that $(s, n'') \models \text{NTERM}(\varphi_1)$. Thus, we can find infinitely many values for which $(s, n''') \models \text{NTERM}(\varphi_1)$. By assumption it follows that for every value of n we have $(s, n) \models \varphi_1 \vee \mathsf{term}$. Thus, it must be the case that $(s, n') \models \text{NTERM}(\varphi) \vee \mathsf{term}$.

We prove a slightly stronger claim, which implies the Theorem. For every ACTL formula φ, every transition system with no terminating states M, and every state s of M we have

$$M, s \models_{\Omega_+} \varphi \Leftrightarrow \forall n \in \boldsymbol{N}.\ \text{FAIR}(M, \Omega), (s, n) \models \text{NTERM}(\varphi) \vee \mathsf{term}$$

We prove this claim by induction on the structure of the formula. Consider an atomic proposition $\alpha \neq t$. Clearly, for every s and every n we have $M, s \models_{\Omega_+} \alpha \Leftrightarrow \text{FAIR}(M, \Omega), (s, n) \models \text{NTERM}(\alpha)$. The proof for Boolean operators is immediate.

$\Rightarrow$ 1. Consider the case that $\varphi = \mathsf{AX}\varphi_1$. Suppose that $M, s \models_{\Omega_+} \mathsf{AX}\varphi_1$.
We have to show that $\text{FAIR}(M, \Omega), (s, n) \models \text{NTERM}(\mathsf{AX}\varphi_1) \vee \mathsf{term}$ for an arbitrary $n \in \boldsymbol{N}$. Recall that $\text{NTERM}(\mathsf{AX}\varphi_1) = \mathsf{AX}(\varphi_1 \vee \mathsf{term})$. If (s, n) is terminating in $\text{FAIR}(M, \Omega)$ then clearly $\text{FAIR}(M, \Omega), (s, n) \models \mathsf{term}$. Otherwise, consider an infinite path $\pi' = (s_0, n_0), (s_1, n_1), \ldots$ starting in (s, n) in $\text{FAIR}(M, \Omega)$. Let $\pi = s_0, s_1, \ldots$ be the projection of π' on the states in M. As shown above π is a fair path in M. Hence, as $M, s \models_{\Omega_+} \mathsf{AX}\varphi_1$ it follows that $M, s_1 \models_{\Omega_+} \varphi_1$ and by induction assumption $\text{FAIR}(M, \Omega), (s_1, n_1) \models \varphi_1 \vee \mathsf{term}$. A similar argument works for every successor (s_1, n_1) of (s, n).

2. Consider the case that $\varphi = \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$. Suppose that $M, s \models_{\Omega_+} \varphi$.
We have to show that $\text{FAIR}(M, \Omega), (s, n) \models \text{NTERM}(\varphi) \vee \mathsf{term}$. Recall that $\text{NTERM}(\varphi) = \mathsf{A}[\text{NTERM}(\varphi_1) \mathsf{W} (\text{NTERM}(\varphi_2) \vee \mathsf{term})]$.
If (s, n) is terminating we are done. Otherwise, consider an infinite path $\pi' = (s_0, n_0), (s_1, n_1), \ldots$ starting in (s, n) in $\text{FAIR}(M, \Omega)$. Let $\pi = s_0, s_1, \ldots$ be the projection of π' on the states in M. As shown above π is a fair path in M. Hence, as $M, s \models_{\Omega_+} \varphi$ it follows that either for every $i \geq 0$ we have $M, s_i \models_{\Omega_+} \varphi_1$ or there is some $j \geq 0$ such that $M, s_j \models_{\Omega_+} \varphi_2$ and for every $i \in [0, j)$ we have $M, s_i \models_{\Omega_+} \varphi_1$. In both cases, it follows from the induction hypothesis that the path π' satisfies $\text{NTERM}(\varphi_1) \mathsf{W} (\text{NTERM}(\varphi_2) \vee \mathsf{term})$.
Consider a finite path $\pi' = (s_0, n_0), (s_1, n_1), \ldots, (s_m, n_m)$ starting in (s, n) in $\text{FAIR}(M, \Omega)$. Let $\pi'' = (s_0, n_0), \ldots, (s_i, n_i), (s'_{i+1}, n'_{i+1}), \ldots$ be an infinite path in $\text{FAIR}(M, \Omega)$ that has a maximal joint prefix with π'. That is, (s_{i+1}, n_{i+1}) is terminating in $\text{FAIR}(M, \Omega)$. As above, the projection of π'' is going to be a fair path in M and thus, the prefix of π'' that agrees with π' either all satisfies $\text{NTERM}(\varphi_1)$ or $\text{NTERM}(\varphi_2)$ holds somewhere along it. In the first case, as (s_{i+1}, n_{i+1}) is terminating we have that π' satisfies the AW formula. In the second case, the AW formula holds already by inspecting only the prefix of π'.

3. The case of AF is similar.

$\Leftarrow$ 1. Consider the case that $\varphi = \mathsf{AX}\varphi_1$. Suppose that $\forall n \in \boldsymbol{N}$ we have $\text{FAIR}(M, \Omega), (s, n) \models \varphi \vee \mathsf{term}$.

If there are no fair paths starting from s in M, then, clearly, $M, s \models \varphi$. Otherwise, consider a fair path $\pi = s_0, s_1, \ldots$ starting in s. Let n_i be the size of $\{j \geq i \mid s_j \in q \text{ and } \forall k \in [i, j] s_k \notin p\}$. Then, $(s_0, n_0), (s_1, n_1), \ldots$ is an infinite path in $\textsc{Fair}(M, \Omega)$. This implies that $\textsc{Fair}(M, \Omega), (s_1, n_1) \models \textsc{NTerm}(\varphi_1) \vee \mathsf{term}$. As we have shown, $\textsc{Fair}(M, \Omega), (s_1, n_1) \not\models \mathsf{term}$. It follows that $\textsc{Fair}(M, \Omega), (s_1, n_1) \models \textsc{NTerm}(\varphi_1)$.

The same clearly holds for every $n_1' > n_1$. It follows by the claim above that for every $n \in \boldsymbol{N}$ we have $(s_1, n) \models \textsc{NTerm}(\varphi_1) \vee \mathsf{term}$.

It follows that for every successor s_1' of s and for every $n \in \boldsymbol{N}$ we have $(s_1', n) \models \textsc{NTerm}(\varphi_1 \vee \mathsf{term})$. By induction, $s_1' \models \varphi_1$.

2. Consider the case that $\varphi = \mathsf{A}[\varphi_1 \mathsf{W} \varphi_2]$. Suppose that $\forall n \in \boldsymbol{N}$ we have $\textsc{Fair}(M, \Omega), (s, n) \models \varphi \vee \mathsf{term}$.

   If there are no fair paths starting from s in M, then, clearly, $M, s \models \varphi$. Otherwise, consider a fair path $\pi = s_0, s_1, \ldots$ starting in s. As in the proof of AX we can find an annotation $\pi' = (s_0, n_0), (s_1, n_1), \ldots$ that is an infinite path in $\textsc{Fair}(M, \Omega)$. Either $\forall i \geq 0$ we have $(s_i, n_i) \models \textsc{NTerm}(\varphi_1)$ or there is $j \geq 0$ such that $(s_j, n_j) \models \textsc{NTerm}(\varphi_2) \vee \mathsf{term}$ and $\forall 0 \leq i < j$ we have $(s_i, n_i) \models \textsc{NTerm}(\varphi_1)$. As by assumption $\textsc{NTerm}(\varphi) \vee \mathsf{term}$ holds for (s, n') for every n' we can show that for every one of these states and for every $n_i' > n_i$ the same property (i.e., $\textsc{NTerm}(\varphi_1)$ or $\textsc{NTerm}(\varphi_2) \vee \mathsf{term}$) holds for (s_i, n_i'). Thus, there are infinitely many values for which these properties hold. Consider, without loss of generality the property φ_1. Then, by the claim above $\textsc{NTerm}(\varphi_1) \vee \mathsf{term}$ holds for all n_i' in the state (s_i', n_i'). Thus, we can conclude from the induction assumption that the same pattern of satisfaction implies that π satisfies the AW property.
3. The case of AF is similar.