Mathematical Structures in Computer Science (2021), **31**, pp. 1135–1144
doi:10.1017/S0960129521000475

CAMBRIDGE
UNIVERSITY PRESS

PAPER

Branching-time logics and fairness, revisited

Markus Latte

Institute of Informatics, Ludwig Maximilian University of Munich, Munich, Germany

(Received 31 July 2020; revised 14 October 2021; accepted 19 November 2021)

Abstract

Emerson and Halpern (1986, *Journal of the Association for Computing Machinery* **33**, 151–178) prove that the Computation Tree Logic (CTL) cannot express the existence of a path on which a proposition holds infinitely often (fairness for short).

The scope is widened from CTL to a general branching-time logic. A path quantifier is followed by a language with temporal descriptions. In this extended setting, the said inexpressiveness is strengthened in two aspects. First, universal path quantifiers are unrestricted. In this way, they are relieved of any temporal quantifiers such as of those in AU and AR from CTL. Second, existential path quantifiers are allowed with any countable language. Instances are the temporal quantifiers in EU and ER from CTL. By contrast, the fairness statement is an existential path quantifier with an uncountable language. Both aspects indicate that this inexpressiveness is optimal with respect to the polarity of path quantifiers and to the cardinality of their languages.

Keywords: Computation tree logic; propositional dynamic logic; branching-time logic; fairness; diagonalisation; expressivity

1. Introduction

Propositional Dynamic Logic (PDL), cf. Fischer and Ladner (1979) or Harel et al. (2000, Chap. 5), and Computation Tree Logic (CTL), cf. Clarke and Emerson (1982) or Emerson and Halpern (1985), are proposed as specification formalisms. However, these logics were shown to be unable to express fairness, cf. Harel and Sherman (1982) for PDL and Emerson and Halpern (1986, Thm. 7) or Clarke and Draghicescu (1989) for CTL. Here, fairness means the existence of a path on that a certain proposition holds infinitely often, cf. Lamport (1980). By contrast, some extensions of PDL and CTL can express fairness, cf. Streett (1981) for PDL and Emerson and Halpern (1986) for CTL.

In essence, the semantics of CTL comprises a path quantifier and a temporal quantifier along the path. Axelsson et al. (2010*a*,*b*) incorporate ideas from PDL into CTL. Basically, the domain of the temporal quantifier is refined from the set of all times to a syntax-given set of times. In this way, the obtained logic (XCTL) can express, for instance, that a path exists on which a certain formula holds at all even times. Because the quantifier pattern remains unchanged, the logic is expected to not express fairness either. However, the proof in Axelsson et al. (2010*b*, Claim 2 in the proof to Lem. 4.3) is faulty.[1]

This article provides a unified proof that neither PDL, CTL, nor XCTL captures fairness. A second aspect concerns the universal path quantifiers.

Fairness depends on a single path. Accordingly, universal path quantifiers should not assist any formula in expressing fairness. The inexpressiveness results by Emerson and Halpern (1986, Thm. 7), Harel and Sherman (1982), and by Clarke and Draghicescu (1989, Thm. 3) are not stated for arbitrary universal path quantifiers because the results regard negation-closed logics. As an example, we consider the two formulae

$$\vartheta_1 \ := \ \mathtt{E}\,\mathtt{tt}\,\mathtt{U}\,p \qquad \text{and} \qquad \vartheta_2 \ := \ \mathtt{E}\,\mathtt{tt}\,\mathtt{U}\,(p \wedge \neg\mathtt{EGF}\,p) \tag{1}$$

where p is some proposition, $\mathtt{E}\,\mathtt{tt}\,\mathtt{U}\,\psi$ states that a vertex is reachable at which ψ holds, and $\mathtt{EGF}\,p$ stands for p-fairness. Neither ϑ_1 nor ϑ_2 is equivalent to p-fairness. More precisely, the linear loop-free structure on that p holds only at the root is a model of both formulae but is p-unfair. The formula ϑ_1 belongs to CTL and is expressible in PDL. The subformula $\mathtt{EGF}\,p$ places ϑ_2 outside of CTL and PDL. Hence, the three cited proofs can handle ϑ_1 but not ϑ_2. Nevertheless, the last two proofs can be adjusted. Basically, it is sufficient to decompose equivalences into implications according to the polarity of the subformulae, similar to the pair (18) and (19) in Lemma 19. Such an adjustment cannot be taken for granted, cf. Bojańczyk (2008) for a peculiar case. For instance, Demri et al. (2016, Thm. 10.3.7) simplify the proof by Emerson and Halpern (1986, Thm. 7) but the younger proof applies its induction hypothesis bidirectionally. Hence, a polarity-based decomposition cannot adjust this proof.

Section 3 defines a plain branching-time logic with the said logics as its sublogics. Theorem 20 in Section 4 is the main result. As a consequence, the mentioned sublogics are uniformly proven unable to express fairness. Section 6 addresses an extension of Section 4.

2. Notations

Ordinal numbers are understood in the sense of von Neumann. In particular, the set of natural numbers is written as ω, and each number is the set of all smaller numbers. For example, $\omega + 1 = \{0, 1, \ldots, \omega\}$ and $c \setminus a = \{b \mid a \leq b < c\}$.

Let $\langle\, ,\, \rangle$ be a bijection from $\omega \times \omega$ to ω such that $a \leq \langle a, b\rangle$ for all $a, b \in \omega$. For instance, Cantor suggests $\langle a, b\rangle := (a + b)(a + b + 1)/2 + a$.

A function and its graph are considered as interchangeable. The image of a function f is written as $\mathsf{img}(f)$.

3. A Unified Branching-Time Logic

3.1 Syntax

Let $\mathbb{P}$ denote a non-empty set of *propositions*. The set $\mathbb{F}$ of *formulae* is the least fixed point (Phillips 1992, Subsec. 3.1) for the following rules.

$$\begin{array}{rcl} \mathbb{F} \ni \psi & ::= & p \mid \neg\psi \mid \psi \vee \psi \mid \mathtt{E}L \\ p & \in & \mathbb{P} \\ L & \subseteq & \omega \to \Psi \qquad (2) \\ \Psi & \text{is} & \text{a finite set of objects named } \psi \qquad (3) \end{array}$$

The variables φ, α, and ε refer to formulae. An entity L generated by (2) is a *language* of infinite *words*. Its *alphabet* $\Sigma(L) := \bigcup\{\mathsf{img}(w) \mid w \in L\}$ is finite due to (3). In formulae, parentheses may be inserted to improve readability.

3.2 Semantics

A *structure* is a quadruple (V, E, ℓ, r) that consists of a set V of *vertices*, a set E of *edges* as a subset of $V \times V$ such that each $v \in V$ has a $w \in V$ with $(v, w) \in E$, a *labelling function* ℓ in $V \to 2^{\mathbb{P}}$, and a *root* r in V. For such a structure $\mathscr{S}$, the root is changed to s by $\mathscr{S} @ s$. A *path* π is a function

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

in $\omega \to V$ such that $\pi(0) = r$ and $(\pi(i), \pi(i+1)) \in E$ for all $i \in \omega$. Whether a structure $\mathscr{S}$ is a *model* of a formula φ, written as $\mathscr{S} \models \varphi$, is defined as follows.

$$\begin{array}{lll} \mathscr{S} \models p & \text{iff} & p \in \ell(r) \text{ for the root } r \text{ of } \mathscr{S} \\ \mathscr{S} \models \neg\psi & \text{iff} & \mathscr{S} \not\models \psi \\ \mathscr{S} \models \psi_0 \vee \psi_1 & \text{iff} & \mathscr{S} \models \psi_j \text{ for some } j \in 2 \\ \mathscr{S} \models \mathtt{E}L & \text{iff} & \text{there is a pair of a path } \pi \text{ for } \mathscr{S} \text{ and of a word } z \text{ in } L \\ & & \text{such that } \mathscr{S} \mathbin{@} \pi(i) \models z(i) \text{ for all } i \in \omega \end{array}$$

For a chosen pair in the last clause, its components are called *witnessing path* and *witnessing word*.

Two structures $\mathscr{S}_1$ and $\mathscr{S}_2$ are 0-*bisimilar*, written as $\mathscr{S}_1 \approx_0 \mathscr{S}_2$, iff each proposition p fulfils $\mathscr{S}_1 \models p$ iff $\mathscr{S}_2 \models p$. Two formulae are *equivalent* iff they have the same models.

3.3 Abbreviations

Boolean values and operations are available by the usual abbreviations $\mathtt{tt} := p \vee \neg p$ for some $p \in \mathbb{P}$ and $\psi_0 \wedge \psi_1 := \neg(\neg\psi_0 \vee \neg\psi_1)$. A disjunction $\psi_0 \vee \psi_1$ could have been alternatively defined as $\mathtt{E}\{z_0, z_1\}$ where, for both $j \in 2$, the word z_j is constantly true, e.g. $\neg\mathtt{E}\emptyset$, but is ψ_j at 0. For concise definitions in Subsection 3.4, disjunctions are first-class citizens, though.

The EU-, EG-, EX-, ER-constructors from CTL can be imitated and generalised. For $\alpha, \varepsilon \in \mathbb{F}$ and for $A, E \subseteq \omega$, the languages

$$\alpha\ {}^{A}\mathtt{U}^{E}\ \varepsilon \;:=\; \Big\{ z \;\Big|\; \begin{array}{l} \text{some } n \in E \text{ fulfils } z(n) = \varepsilon \\ \text{and } z(k) = \alpha \ \text{ for all } k \in A \cap n \\ \text{and } z(k) = \mathtt{tt} \text{ for all } k \in \omega \setminus (n+1) \end{array} \Big\} \tag{4}$$

and

$$\mathtt{G}^{A}\alpha \;:=\; \big\{ z \;\big|\; \text{each } n \in \omega \text{ fulfils } z(n) = \alpha \text{ if } n \in A \text{ and } z(n) = \mathtt{tt} \text{ otherwise} \big\} \tag{5}$$

are defined as subsets of $\omega \to \{\mathtt{tt}, \alpha, \varepsilon\}$ and of $\omega \to \{\mathtt{tt}, \alpha\}$, respectively. For convenience,

$$\mathtt{X}\varepsilon \;:=\; \mathtt{tt}\ {}^{\emptyset}\mathtt{U}^{\{1\}}\ \varepsilon \tag{6}$$

and

$$\varepsilon\ {}^{E}\mathtt{R}^{A}\ \alpha \;:=\; \mathtt{G}^{A}\alpha \;\cup\; \alpha\ {}^{A}\mathtt{U}^{E\cap A}\ (\varepsilon \wedge \alpha) \;\cup\; \alpha\ {}^{A}\mathtt{U}^{E\setminus A}\ \varepsilon\,. \tag{7}$$

The sets A and E restrict the temporal quantifiers. For instance, $\mathtt{EG}^{\{2t \mid t \in \omega\}}\varphi$ asks for a path on that the formula φ holds *at all even times* while odd times are unspecified. The standard constructors appear when $A = E = \omega$. The setting (7) rests on the corresponding decomposition of ER-formulae for CTL, cf. Emerson and Halpern (1985, proof of Thm. 8.4) or Demri et al. (2016, dual of Lem. 7.1.2).

Example 1. Let φ be a formula. The formula

$$\mathtt{AGF}\,\varphi \;:=\; \neg\mathtt{E}\mathtt{tt}\ {}^{\omega}\mathtt{U}^{\omega}\ \big(\mathtt{E}\neg\mathtt{tt}\ {}^{\omega}\mathtt{R}^{\omega}\ \neg\varphi\big)$$

asserts that φ holds infinitely often on *every* path. With the language

$$\mathtt{GF}\,\varphi \;:=\; \big\{ z\colon \omega \to \{\mathtt{tt}, \varphi\} \;\big|\; z(i) = \varphi \text{ for infinitely many } i \in \omega \big\} \tag{8}$$

the dual formula $\mathtt{EGF}\,\varphi$ asserts that φ holds infinitely often on *some* path.

Definition 2 (Fairness)**.** *Let φ be a formula. A structure is φ-*fair *iff it is a model of* $\mathtt{EGF}\,\varphi$.

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

Lemma 3. *Every language of the shape* (4), (5), (6), *or* (7) *is countable.*

Proof. Words in (4) are eventually constant. The word in (5) is unique. □

Remark 4. Each proof that is cited in Section 1 shows that a pair of two structures cannot be distinguished by some PDL- or CTL-formulae. However, $\mathtt{EG}^{\{2t \mid t\in\omega\}}p$ distinguishes the pairs by Emerson and Halpern (1986, Thm. 7) and $\mathtt{EG}^{\{2t \mid t\in\omega\}}\neg p$ the pair by Clarke and Draghicescu (1989, Sec. 4). Both formulae are handled by Theorem 20.

3.4 Sublogics

Definition 5. *The sets of negative and positive* subformulae *of a formula* φ *are the smallest sets* $\mathsf{subf}_{-1}(\varphi)$ *and* $\mathsf{subf}_1(\varphi)$ *that fulfil for both signs* $s \in \{-1, 1\}$

- $\varphi \in \mathsf{subf}_1(\varphi)$,
- $\neg\psi \in \mathsf{subf}_s(\varphi)$ *implies* $\psi \in \mathsf{subf}_{-s}(\varphi)$,
- $\psi_0 \vee \psi_1 \in \mathsf{subf}_s(\varphi)$ *implies* $\{\psi_0, \psi_1\} \subseteq \mathsf{subf}_s(\varphi)$, *and*
- $\mathtt{E}L \in \mathsf{subf}_s(\varphi)$ *implies* $\Sigma(L) \subseteq \mathsf{subf}_s(\varphi)$.

Definition 6. *Let* $S \subseteq \{-1, 1\}$ *be a set of signs. Its set of* languages in a formula *is*

$$\mathsf{lngs}_S(\varphi) \quad := \quad \{L \mid \mathtt{E}L \in \mathsf{subf}_s(\varphi) \text{ for some } s \in S\}\,.$$

Restrictions on $\mathbb{F}$ through lngs reveal various branching-time logics. Prominently, CTL is just

$$\left\{\varphi \in \mathbb{F} \;\middle|\; \text{each } L \in \mathsf{lngs}_{\{-1,1\}}(\varphi) \text{ has the shape of (6) or of (4) or (7) for } A = E = \omega \right\}. \tag{9}$$

Example 7. In continuation of Example 1, the formula $\mathtt{AGF}\,\varphi$ belongs to CTL but $\mathtt{EGF}\,\varphi$ does not.

Because any restriction of negative E-constructors is irrelevant for Theorem 20, we drop those constraints and obtain the larger logic

$$\left\{\varphi \in \mathbb{F} \;\middle|\; \text{each } L \in \mathsf{lngs}_{\{1\}}(\varphi) \text{ has the shape of (6) or of (4) or (7) for } A = E = \omega \right\}. \tag{10}$$

In other words, negative E-constructors are entirely unrestricted and may quantify along a path freely. A variant is

$$\left\{\varphi \in \mathbb{F} \;\middle|\; \text{each } L \in \mathsf{lngs}_{\{1\}}(\varphi) \text{ has the shape of (4) or (7)} \right\} \tag{11}$$

which allows to customise the range of the temporal quantifiers on both sides of an U and a R. The logic subsumes limited variants (Axelsson et al. 2010*a*,*b*) that allow the customisation on the right side of U and R only. Similarly,

$$\left\{\varphi \in \mathbb{F} \;\middle|\; \text{each } z \in \bigcup \mathsf{lngs}_{\{1\}}(\varphi) \text{ fulfils } z(i) = \mathtt{tt} \text{ for almost all } i \in \omega \right\} \tag{12}$$

captures reachability and thus subsumes PDL and its nonregular extensions (Harel et al. 2000, Chap. 5 and 9). All stated subsumptions ignore edge labels or understand them as encoded by propositions, cf. De Nicola and Vaandrager (1990). In particular, Theorem 20 disregards edge labels.

4. Inexpressiveness of Fairness

To reveal fairness as inexpressible, a fair structure and a bunch of unfair structures are proven as indistinguishable, cf. Theorem 20 with its elaboration in Section 4.1. Finally, Subsection 4.2 applies Theorem 20 to the logics from Subsection 3.4.

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

4.1 Indistinguishable structures

Assumption 8. *Until the end of this subsection, we let p be a proposition and let ϑ be an arbitrary formula such that $\bigcup \mathsf{lngs}_{\{1\}}(\vartheta)$ is a countable set of words.*

In Definition 15, we shall construct a set $\{\mathscr{T}_k \mid k \in \omega + 1\}$ of structures. The structure $\mathscr{T}_\omega$ is p-fair while $\mathscr{T}_k$ is not for any $k \in \omega$, cf. Lemma 16.

Lemma 19 demonstrates that both kinds of structures are indistinguishable. The crucial cases are the $\mathtt{E}$-subformulae of ϑ. Positive $\mathtt{E}$-subformulae shall transfer from the p-fair structure $\mathscr{T}_\omega$ to a p-unfair structure $\mathscr{T}_k$ for $k \in \omega$. The direction of negative $\mathtt{E}$-subformulae is reverse.

In a first approximation, each structure is linear. Along the sole path in the structure $\mathscr{T}_\omega$, the proposition p is placed infinitely often but sparsely. The places are enumerated by the function T, as determined by Definition 11 with Lemma 10. The vertices for p are so seldom that, for each possible witnessing word of each positive $\mathtt{E}$-subformulae, each asked formula is also asked at a vertex for $\neg p$. Hence, some suffix of the witnessing word can cope with only $\neg p$. Along the corresponding suffix of the path, the proposition p can be erased stealthily, cf. the proof of Lemma 18. The modification for the k-th suffix yields the p-unfair structure $\mathscr{T}_k$. So far, the backbone of $\mathscr{T}_k$ for $k \in \omega + 1$ is described.

The other direction has to convey a negative $\mathtt{E}$-subformula from $\mathscr{T}_k$ for $k \in \omega$ to $\mathscr{T}_\omega$. In such a formula $\mathtt{E}L$, the language L is not captured by Assumption 8. As a compensation, we attach for each $\mathtt{E}$-subformula a p-unfair model to $\mathscr{T}_\omega$ to capture any witnessing path from the p-unfair structure $\mathscr{T}_k$, cf. Lemma 17. Back to the first direction, witnessing paths in $\mathscr{T}_\omega$ can enter the now attached structures. Thus, these structures are also attached to $\mathscr{T}_k$ for each $k \in \omega$. All in all, we collected all ingredients for Definition 15.

The just drafted diagonalisation against positive $\mathtt{E}$-subformulae requires that the set of all possible witnessing words is enumerable, cf. Assumption 8 and Definition 9.

Definition 9. *We fix an arbitrary enumeration $Z\colon \omega \to (\omega \to \mathbb{F})$ of all words in $\bigcup \mathsf{lngs}_{\{1\}}(\vartheta)$.*

Lemma 10. *There exists a function $T\colon \omega \to \omega$ such that all $n \in \omega$ meet the condition*

$$\left.\begin{array}{l} \text{each } t_2 \in \omega \text{ with } T(n+1) < t_2 \\ \text{has a } t_1 \in \omega \text{ with } T(n) < t_1 < T(n+1) \\ \text{such that } z(t_1 - t_0) = z(t_2 - t_0) \\ \qquad \text{where } t_0 \text{ and } i \text{ fulfil } n = \langle t_0, i\rangle \\ \qquad \text{and } z \text{ abbreviates } Z(i). \end{array}\right\} \tag{13}$$

Proof. An induction on n constructs $T(n)$ and entails that $n \leq T(n)$ as a support. Given n, the values t_0 and i exist. Because $t_0 \leq \langle t_0, i\rangle = n \leq T(n)$, the set $X := \{\, z(t - t_0) \mid T(n) < t \in \omega \,\}$ is well defined and not empty. As z is a word, the set is finite. Hence, the conditions

$$\min\{\, t \mid z(t - t_0) = x \text{ and } T(n) < t \in \omega \,\} \quad < \quad T(n+1) \qquad \text{for each } x \in X$$

characterise $T(n+1)$ completely. As a by-product, $n \leq T(n) < T(n+1)$. □

Definition 11. *Henceforth, an arbitrary function provided by Lemma 10 is designated as T.*

Definition 12 (Default Unfair Models). *For each formula, φ let $\mathscr{U}_\varphi = (V_\varphi, E_\varphi, \ell_\varphi, r_\varphi)$ be some model of $\varphi \wedge \neg \mathtt{EGF}\, p$ if existing and an arbitrary model of $\neg\mathtt{EGF}\, p$ otherwise. We consider their sets of vertices as pairwise disjoint and disjoint from ω.*

Attaching a structure to another structure can change the modelled formulae at the shared root. As Lemma 17 bridges from an attached structure to the entire structure, Definition 15 filters each $\mathtt{E}$-subformula of ϑ by the formulae that can be asked at a root.

Definition 13. *The* filtering *of a language L by a formula ψ is $L \downarrow \psi := \{z \in L \mid z(0) = \psi\}$.*

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

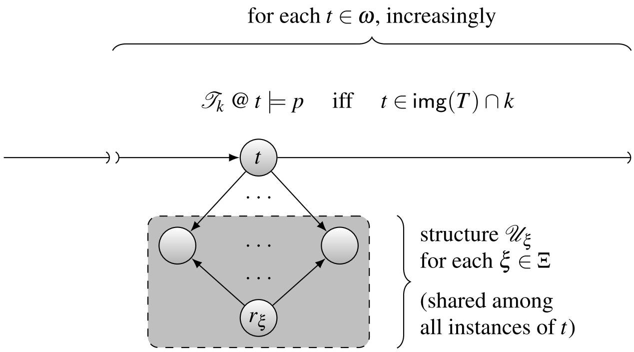

Figure 1. The structure $\mathcal{T}_k$ where $k \in \omega + 1$.

Definition 14. E*-subformulae of* ϑ *are split by their first letter through*

$$\Xi \quad := \quad \left\{ \mathtt{E}(L \downarrow \psi) \;\middle|\; \mathtt{E}L \in \mathsf{subf}_{-1}(\vartheta) \cup \mathsf{subf}_1(\vartheta) \textit{ and } \psi \in \Sigma(L) \right\}.$$

Definition 15 (Sample Structures)**.** *Let* $k \in \omega + 1$*. The structure* $\mathcal{T}_k$ *is the quadruple* $(V, E, \ell, 0)$ *where the set* V *of vertices is*

$$\bigcup_{\xi \in \Xi} V_\xi \quad \cup \quad \omega\,,$$

the set E *of edges is*

$$\bigcup_{\xi \in \Xi} E_\xi \quad \cup \quad \bigcup_{\xi \in \Xi} \left\{ (t, s) \;\middle|\; t \in \omega \textit{ and } (r_\xi, s) \in E_\xi \right\} \quad \cup \quad \{ (t, t+1) \mid t \in \omega \}\,,$$

and the labelling function ℓ *is*

$$\bigcup_{\xi \in \Xi} \ell_\xi \quad \cup \quad \left\{ \left(t, \{p \mid t \in \mathsf{img}(T) \cap k\}\right) \;\middle|\; t \in \omega \right\}.$$

Figure 1 sketches the structure $\mathcal{T}_k$. Basically, $\mathcal{T}_k$ is linear where at each vertex t the structure $\mathcal{U}_\xi$ is attached for all $\xi \in \Xi$. Along the linear basis, the places for the proposition p are determined by $\mathsf{img}(T)$ until k. For $k \in \omega + 1$ and $t \in \omega$, the *backbone* of $\mathcal{T}_k @ t$ is the path $\{(i, t+i) \mid i \in \omega\}$, that is, the t-th suffix of the underlying linear structure.

Lemma 16. *Let* $k \in \omega + 1$ *and* $t \in \omega$*. Then,* $\mathcal{T}_k @ t \models \mathtt{EGF}\, p$ *iff* $k = \omega$*.*

Proof. The function T is unbounded. On the backbone of $\mathcal{T}_k @ t$, the set of p-labelled vertices is $\mathsf{img}(T) \cap k \setminus t$. Its cardinality is infinite iff $k = \omega$. Because each attached structure $\mathcal{U}_\xi$ for $\xi \in \Xi$ is a model of $\neg\mathtt{EGF}\, p$, any witnessing path for $\mathcal{T}_k @ t \models \mathtt{EGF}\, p$ sits on the backbone. □

As tools for the proof of Lemma 19 later, Lemma 17 basically exchanges a p-unfair model of an E-subformula for a p-fair model, and Lemma 18 shows that each witnessing word from Z can be misled by some p-unfair structure.

Lemma 17. *Let* $k \in \omega + 1$*, let* $t \in \omega$*, and let* $\xi \in \Xi$ *of the shape* $\mathtt{E}(L \downarrow \psi)$*. If* $\xi \wedge \neg\mathtt{EGF}\, p$ *has a model and* $\mathcal{T}_k @ t \models \psi$ *then* $\mathcal{T}_k @ t \models \xi$*.*

Proof. A witness for $\mathcal{U}_\xi \models \xi$ is adjusted to a witness for $\mathcal{T}_k @ t \models \xi$. The first vertex of the witnessing path is replaced with t while the witnessing word remains. The adjustment is sound. First, $\mathcal{U}_\xi$ is a model of ψ because any word in $L \downarrow \psi$ begins with ψ. Second, $\mathcal{T}_k @ t$ is also a model

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

of ψ by assumption. Third, $\mathscr{U}_\xi$ is a substructure of $\mathscr{T}_k @ t$ and no edge in $\mathscr{T}_k @ t$ leaves this substructure. □

In Lemma 18, the line (15) switches a p-fair model of an $\mathtt{E}$-formula to a p-unfair model if the witnessing word is captured by the enumeration Z. The premise (14) is provided by the induction hypothesis when (15) is applied in the proof of Lemma 19.

Lemma 18. *Let $t_0, i \in \omega$. Set $c := T\big(\langle t_0, i\rangle + 1\big) + 1$ and $z := Z(i)$. If*

$$\mathscr{T}_\omega @ t_1 \;\models\; \psi \quad \textit{implies} \quad \mathscr{T}_c @ t_2 \;\models\; \psi \tag{14}$$

for all $\psi \in \Sigma(\{z\})$ and for all pairs $t_1, t_2 \in \omega$ such that $\mathscr{T}_\omega @ t_1 \approx_0 \mathscr{T}_c @ t_2$, then

$$\mathscr{T}_\omega @ t_0 \;\models\; \mathtt{E}\{z\} \quad \textit{implies} \quad \mathscr{T}_c @ t_0 \;\models\; \mathtt{E}\{z\}\,. \tag{15}$$

Proof. Let π be a witnessing path for $\mathscr{T}_\omega @ t_0 \models \mathtt{E}\{z\}$. We take this path for $\mathscr{T}_c @ t_0 \models \mathtt{E}\{z\}$. Because the structures that are attached to the backbone are shared among $\mathscr{T}_\omega @ t_0$ and $\mathscr{T}_c @ t_0$, it remains to consider the backbone. So, let t_2 be a vertex of π on the backbone. As the backbone is linear,

$$\mathscr{T}_\omega @ t_2 \;\models\; \psi \;:=\; z(t_2 - t_0)\,.$$

Next, we show that

$$\mathscr{T}_\omega @ t_1 \;\models\; \psi \quad \text{and} \quad \mathscr{T}_\omega @ t_1 \;\approx_0\; \mathscr{T}_c @ t_2 \quad \text{for some } t_1 \in \omega \tag{16}$$

because this property together with (14) entails that $\mathscr{T}_c @ t_2 \models \psi$ and complete the proof.

Case: $t_2 < c$. We take t_2 as t_1. In particular, Definition 15 ensures the second part of (16) because $t_1 \in \omega$ on the backbone and because $t_2 \in c$ in this case.

Case: $c \leq t_2$. The condition (13) for $n := \langle t_0, i\rangle$ yields a t_1 such that

$$T(n) \;<\; t_1 \;<\; T(n+1) \quad \text{and} \quad z(t_1 - t_0) \;=\; \psi\,. \tag{17}$$

Because the first conjunct implies $t_1 \leq t_2$ in this case, t_1 also refers to the backbone. Hence, the second conjunct in (17) shows that $\mathscr{T}_\omega @ t_1 \models \psi$. The second part of (16) is a consequence of Definition 15 for the following reason. First, $\mathscr{T}_\omega @ t_1 \models \neg p$ due to the first conjunct in (17). Second, $\mathscr{T}_c @ t_2 \models \neg p$ because $t_2 \notin c$ in this case. □

Lemma 19. *Each formula φ fulfils*

$$\varphi \in \mathsf{subf}_1(\vartheta) \;\; \textit{and} \;\; \mathscr{T}_\omega @ t_\omega \models \varphi \quad \textit{imply} \quad \mathscr{T}_k @ t_k \models \varphi \tag{18}$$

and

$$\varphi \in \mathsf{subf}_{-1}(\vartheta) \;\; \textit{and} \;\; \mathscr{T}_k @ t_k \models \varphi \quad \textit{imply} \quad \mathscr{T}_\omega @ t_\omega \models \varphi \tag{19}$$

for all $k \in \omega$, and all pairs $t_k, t_\omega \in \omega$ such that $\mathscr{T}_\omega @ t_\omega \approx_0 \mathscr{T}_k @ t_k$.

Proof. Induction on φ. A proposition is handled by the definition of $\approx_0$. The induction hypothesis yields negation and disjunction where a negation swaps between (18) and (19). So, let φ be $\mathtt{E}L$ for some language L.

Case: the premises of (19) hold. Let z be a witnessing word for $\mathscr{T}_k @ t_k \models \varphi$. Hence,

$$\mathscr{T}_k @ t_k \;\models\; z(0) \quad \text{and} \quad \mathscr{T}_k @ t_k \;\models\; \mathtt{E}\big(L \downarrow z(0)\big) \;=:\; \xi\,. \tag{20}$$

The induction hypothesis for the first conjunct and Lemma 16 for the second conjunct entail that

$$\mathscr{T}_\omega @ t_\omega \;\models\; z(0) \quad \text{and} \quad \mathscr{T}_k @ t_k \;\models\; \xi \wedge \neg\mathtt{EGF}\, p\,.$$

Because $\xi \in \Xi$, Lemma 17 combines both parts to $\mathscr{T}_\omega @ t_\omega \models \xi$. Finally, as the semantics of $\mathtt{E}$ is monotone in the language, $\mathscr{T}_\omega @ t_\omega \models \varphi$.

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

Case: the premises of (18) *hold.* Let z be a witnessing word for $\mathscr{T}_\omega @ t_\omega \models \varphi$. As $z \in L$ and $L \in \mathsf{lngs}_{\{1\}}(\vartheta)$, Definition 9 yields an $i \in \omega$ such that $Z(i) = z$. Thus,

$$\mathscr{T}_\omega @ t_\omega \models z(0) \quad \text{and} \quad \mathscr{T}_\omega @ t_\omega \models \mathtt{E}\{z\}\,.$$

Because the induction hypothesis supplies the premise (14) of Lemma 18, the implication (15) affects the second conjunct. As the semantics of E is monotone in the language,

$$\mathscr{T}_\omega @ t_\omega \models z(0) \quad \text{and} \quad \mathscr{T}_c @ t_\omega \models \mathtt{E}\big(L \downarrow z(0)\big) =: \xi$$

for some $c \in \omega$. This situation reminds of (20). The reasoning there results in $\mathscr{T}_k @ t_k \models \varphi$ here. □

The scope of Assumption 8 ends here.

4.2 Consequences

Theorem 20. *Let p be a proposition, and let ϑ be a formula. If $\bigcup \mathsf{lngs}_{\{1\}}(\vartheta)$ is a countable set of words then ϑ is not equivalent to* EGF p.

Proof. Assumption 8 is fulfilled. We consider Lemma 19 for $\varphi := \vartheta$, $k := 1$, and $t_k := 0 =: t_\omega$. In particular, $\mathscr{T}_\omega @ t_\omega \approx_0 \mathscr{T}_k @ t_k$ due to Definition 15. The implication (18) and Lemma 16 reject an equivalence of ϑ and EGF p. □

The language GF p is uncountable. Consequently, Theorem 20 also shows that EL is not equivalent to EGF p when L is only a countable subset of GF p.

Corollary 21. *Let p be a proposition. The logics in* (10), (11), *and* (12) *do not contain any formula that is equivalent to* EGF p.

Proof. By Theorem 20 with Lemma 3. In particular, the number of subformulae is finite. □

For example, Corollary 21 handles both formulae from (1). They belong to (10) although the second formula hosts EGF p as a negative subformula.

Given the subsumptions between logics in Subsection 3.4, Corollary 21 remedies the faulty argument in Axelsson et al. (2010*b*, Lem. 4.3) and reproves the results by Emerson and Halpern (1986, Thm. 7) on CTL and by Harel and Sherman (1982) on PDL.

Corollary 22. *Let p be a proposition. Neither* CTL, XCTL, PDL *nor its nonregular extensions contain any formula that is equivalent to* EGF p.

5. Remark on Unfair Models

The proofs by Harel and Sherman (1982) and by Emerson and Halpern (1986, Thm. 7) as well as the proof here attach some unfair structures to a fair structure. The cited proofs construct each attached unfair structure explicitly as an *open box*. By contrast, Definition 12 picks unfair structures and attach them as *closed boxes*.

The closed-box principle can simplify the definition of structures and shorten argumentations. An example is the proof by Emerson and Halpern that CTL cannot express p-fairness. Figure 2 depicts a pair of a p-fair structure $\mathscr{S}_\vartheta^+$ and a p-unfair structure $\mathscr{S}_\vartheta^-$ for a CTL-formula ϑ. However, this formula cannot distinguish between both structures. In fact, an induction yields

$$\mathscr{S}_\vartheta^+ @ v \models \varphi \quad \text{iff} \quad \mathscr{S}_\vartheta^- @ v \models \varphi \tag{21}$$

for both $v \in 2$ and each subformula φ of ϑ. The transfer of a witnessing path from $\mathscr{S}_\vartheta^+$ to $\mathscr{S}_\vartheta^-$ basically tightens the cycle $0 \to 1 \to 0$ to the loop $0 \to 0$ and possibly redirects the edge $0 \to 1$ to $0 \to r_\varepsilon$ if a formula E α U ε is considered and $\mathscr{S}_\vartheta^- @ 1 \models \varepsilon$.

https://doi.org/10.1017/S0960129521000475 Published online by Cambridge University Press

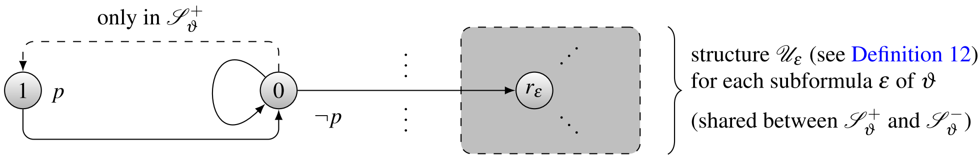

Figure 2. The structures $\mathscr{S}_\vartheta^+$ and $\mathscr{S}_\vartheta^-$ for a CTL-formula ϑ. Both differ in the edge $0 \rightarrow 1$.

Looking back to the introduction, the previous argument can be expanded to any formula ϑ from (10). For this purpose, the equivalence (21) is split by the polarity of φ similar to (18) and (19).

6. Extension

Theorem 20 requires any language L to be countable if $\mathtt{E}L$ is a positive subformula. This constraint cannot be relaxed from "countable" to "uncountable" as the opponent language $\mathtt{GF}\,p$ is already uncountable. Hence, the cardinality is a border between unfairness and fairness.

Harel and Sherman (1982) prove that a certain extension of PDL cannot express fairness. Basically, this extension allows languages of finite-state automata that regard each infinite run as accepting. An instance is the *uncountable* language $\{\psi_0, \psi_1\}^\omega$ for two different formulae ψ_0 and ψ_1. Nevertheless, this class of languages is tame through its pumping property (Harel and Sherman 1982, case "$loop(\alpha)$" in the proof of Lemma 7): each suffix s of each word in such a language admits two finite words x and y such that the length of xy is language-specifically bounded, xy is a prefix of s, and s can be replaced with xy^ω. Such languages can be incorporated into Subsection 4.1 with ease. First, the enumeration Z has to list those languages as well. Second, (13) is expanded such that if n refers to such a language then $T(n+1) - T(n)$ exceeds the language-specific bound. Third, Lemma 18 has to be adjusted. The line (15) was formulated with a singleton language just for simplicity. Now, the pumping property requires to mention the entire language because the witnessing word will be replaced.

7. Conclusion

Theorem 20 draws the line between unfairness and fairness at the level of cardinality for positive $\mathtt{E}$-subformulae. In particular, the rigidity of the temporal constructors $\mathtt{EU}$, $\mathtt{EG}$, ... alone cannot be hold responsible for unfairness. In truth, their implicit countability *is* responsible. On the other hand, Theorem 20 additionally reveals that negative $\mathtt{E}$-subformulae have no impact on fairness.

Acknowledgements

The article in hand is dedicated to Martin Hofmann. I am deeply grateful for many years of delightful discussions, his keen perception, scientific inspiration, permissive supervisions, and much more.

We met at his seminar about "proofs from THE BOOK" for the first time. So, I like to continue the tradition of short ~~stories~~ proofs. When we were discussing my PhD project on CTL years later, he objected that CTL is rather weak and advertised more goal-driven formal systems. Along these lines, this contribution has two ingredients: shortness and expressiveness.

As words fail me, I just want to say: Martin, thank you! In the same year of the seminar, you also gave a lecture on "efficient algorithms" and offered some challenging exercises. The bottle of "corona extra" beer that you rewarded is still adorning my home and is a precious reminder for me.

Note

1 The formula $\mathtt{EG}^{\Sigma(\Sigma^{k+2})^*}q$ does distinguish the considered pair of structures at the respective $(n, k{+}1)$-indexed vertex.

References

Axelsson, R., Hague, M., Kreutzer, S., Lange, M., and Latte, M. (2010a). Extended computation tree logic. In: Fermüller, C. and Voronkov, A. (eds.), *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, volume 6397 of Lecture Notes in Computer Science*, pp. 67–81. Springer, Heidelberg, Germany.

Axelsson, R., Hague, M., Kreutzer, S., Lange, M. and Latte, M. (2010b). Extended Computation Tree Logic. arXiv.org. Available at https://arxiv.org/abs/1006.3709v1.

Bojańczyk, M. (2008). The common fragment of ACTL and LTL. In Amadio, R. (ed.), *Foundations of Software Science and Computational Structures, volume 4962 of Lecture Notes in Computer Science*, pp. 172–185. Springer, Heidelberg, Germany.

Clarke, E. and Draghicescu, I. (1989). Expressibility results for linear-time and branching-time logics. In: de Bakker, J., de Roever, W. and Rozenberg, G. (eds.), *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, volume 354 of Lecture Notes in Computer Science*, pp. 428–437. Springer, Heidelberg, Germany.

Clarke, E. M. and Emerson, E. A. (1982). Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.), *Logics of Programs: Workshop, Yorktown Heights, New York, May 1981, volume 131 of Lecture Notes in Computer Science*, pp. 52–71. Springer, Heidelberg, Germany.

De Nicola, R. and Vaandrager, F. (1990). Action versus state based logics for transition systems. In: Guessarian, I. (ed.), *Semantics of Systems of Concurrent Processes, volume 469 of Lecture Notes in Computer Science*, pp. 407–419. Springer, Heidelberg, Germany.

Demri, S., Goranko, V. and Lange, M. (2016). *Temporal Logics in Computer Science: Finite-State Systems*, Cambridge, UK, Cambridge University Press.

Emerson, E. A. and Halpern, J. Y. (1985). Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences* **30** 1–24.

Emerson, E. A. and Halpern, J. Y. (1986). "Sometimes" and "Not Never" revisited: On branching versus linear time temporal logic. *Journal of the Association for Computing Machinery* **33** 151–178.

Fischer, M. J. and Ladner, R. E. (1979). Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences* **18** (2) 194–211.

Harel, D., Kozen, D. and Tiuryn, J. (2000). *Dynamic Logic*, Cambridge, MA, The MIT Press.

Harel, D. and Sherman, R. (1982). Looping vs. repeating in dynamic logic. *Information and Control* **55** (1–3) 175–192.

Lamport, L. (1980). "Sometime" is sometimes "Not Never": On the temporal logic of programs. In: *Proceedings of the 7th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 174–185. New York, USA, Association for Computing Machinery.

Phillips, I. (1992). Recursion theory. In: Abramsky, S., Gabbay, D. and Maibaum, T. (eds.) *Handbook of Logic in Computer Science*, Background: Mathematical Structures, vol. 1, Oxford, UK, Oxford University Press, 79–187.

Streett, R. S. (1981). Propositional dynamic logic of looping and converse. In: *Proceedings of the thirteenth annual ACM Symposium on Theory of Computing*, pp. 375–383. New York, USA, Association for Computing Machinery.

Cite this article: Latte M (2021). Branching-time logics and fairness, revisited. *Mathematical Structures in Computer Science* **31**, 1135–1144. https://doi.org/10.1017/S0960129521000475