An Automaton-based Characterisation of First-Order Logic over Infinite Trees

Massimo Benerecetti
Università degli Studi di Napoli Federico II
massimo.benerecetti@unina.it

Dario Della Monica
Università degli Studi di Udine
dario.dellamonica@uniud.it

Angelo Matteo
Università degli Studi di Udine
matteo.angelo@spes.uniud.it

Fabio Mogavero
Università degli Studi di Napoli Federico II
fabio.mogavero@unina.it

Gabriele Puppis
Università degli Studi di Udine
gabriele.puppis@uniud.it

In this paper, we study First Order Logic (FO) over (unordered) infinite trees and its connection with branching-time temporal logics. More specifically, we provide an automata-theoretic characterisation of FO interpreted over infinite trees. To this end, two different classes of hesitant tree automata are introduced and proved to capture precisely the expressive power of two branching time temporal logics, denoted $\mathrm{cCTL}^{p}_{\pm}$ and $\mathrm{cCTL}^{\star}_{f}$, which are, respectively, a restricted version of counting CTL with past and counting $\mathrm{CTL}^{\star}$ over finite paths, both of which have been previously shown equivalent to FO over infinite trees. The two automata characterisations naturally lead to normal forms for the two temporal logics, and highlight the fact that FO can only express properties of the tree branches which are either safety or co-safety in nature.

1 Introduction

Characterisation theorems [25] are powerful model-theoretic tools that offer a principled approach to understanding the intrinsic features of formal systems. They allow us to mark the *expressive boundaries* of specification languages, compare these formalisms *w.r.t.* their *descriptive power* on specific classes of models, and design new languages starting from a given set of requirements, in the spirit of *Lindström-style theorems* [34] (*e.g.*, based on maximality principles). They also play a central role in *definability theory*, guiding the identification of expressive fragments and meaningful extensions of known logics, thus supporting the selection of suitable languages for the specification of the correct behaviour of systems in verification and synthesis tasks.

A foundational distinction exists between *linear-time* and *branching-time* languages [36, 37]. The former capture properties of computations viewed as totally-ordered sets of events, while the latter account for the branching structure inherent in concurrent and nondeterministic system behaviours.

The linear-time case, where models are isomorphic to (finite or infinite) *words*, is by now well understood. A rich and intertwined network of equivalences connects *predicate logics* over linear orders with *temporal logics*, such as LTL [44, 45] and ELTL [54], with *star-free* [50, 43] and ω*-regular* [12] languages, and with automata-theoretic models, including *finite* [42, 48] and *Büchi* [19, 12, 13] automata. These connections provide deep insights into the structure of definable properties and lead to optimal decision procedures across different representations.

Submitted to:
Sixteenth International Symposium on Games, Automata, Logics, and Formal Verification (GandALF 2025)

By contrast, the branching-time setting remains more fragmented. Even for *First-Order Logic* (FO) interpreted over (finite or infinite) trees many fundamental definability questions remain unsettled. A longstanding open problem posed by Thomas in the 1980s [51] asks whether it is decidable if a given regular-tree language is definable in FO. This question has been studied under various combinations of tree types (*ranked/unranked*, *ordered/unordered*) and interpreted vocabularies (*e.g.*, including only *child*, only *ancestor*, or both relations). Aside from the positive result for FO over finite trees with the child relation [1], the problem remains open in all other settings. Efforts to resolve this question have mainly followed *algebraic approaches* [6], inspired by their success in the word case (most notably the Schützenberger theorem on star-free languages [50]). These approaches rely on the compositionality and structural insights provided by syntactic algebras. Despite significant progress, they have provided only partial results, mostly for classes of finite trees [46, 4, 23, 9] or topologically simple infinite trees [7, 8]. An alternative and often complementary line of work seeks direct characterisations of FO-definable tree languages via automata. This route, highly successful in the linear-time case, has also led to fruitful results in the branching-time setting, including a correspondence [29] between *Monadic Second-Order Logic* (MSO) [47], *Parity Tree Automata* [40, 22], and the *Modal* μ-CALCULUS [30]. More recently [2], the landscape has expanded to include the expressive equivalence of *Monadic Chain/Path Logics* (MCL/MPL) [51, 52, 28], their temporal *Computation Tree Logic* counterparts (ECTL*/CTL*) [53, 20, 21], and variants of *Hesitant Tree Automata* (HTA) [32].

In this work, we continue this line of development, by providing the first, to the best of our knowledge, complete automata-theoretic characterisation of first-order logic with the descendant relation of unranked unordered infinite trees. Our approach builds on previous results for two branching-time temporal logics, namely a *fragment of Computation Tree Logic with past and counting*, denoted cCTL$^p_\pm$, and the *Full Computation Tree Logic with counting and finite path quantification*, denoted cCTL$^\star_f$. In [49, 2] these logics were shown to be expressively equivalent to FO when interpreted on unordered infinite trees. For these two logics, we introduce corresponding variants of hesitant graded automata, called *Two-Way Hesitant Linear Tree Automata* (2HLGT) and *counter-free Hesitant Weak Tree Automata* (HWGT$_{cf}$), and prove that they capture precisely the expressive power of the considered logics, and therefore of FO as well. This establishes a full mutual equivalence between logics and automata. These characterisations also uncover a *polarised normal form* for both temporal logics, revealing a noteworthy semantic feature of FO over infinite trees: formulas that quantify existentially on branches can only express *co-safety* properties, while those quantifying universally correspond exclusively to *safety* properties. This observation aligns with earlier findings [15] that relate fragments of the modal μ-CALCULUS, variants of *Propositional Dynamic Logic* (PDL) [24], and *Weak Monadic Chain Logic* (WMCL).

Other related work. In earlier work, Bojańczyk [4] showed that, over finite binary trees, FO with child and ancestor relations is equivalent to a *cascade product* of so-called *aperiodic wordsum automata*. While related in spirit, this result targets a different logic and a different class of structures. More recently, Ford [26] focused on the same tree structures that are considered here, and introduced the class of *antisymmetric path parity automata*, which are shown to be no more expressive than FO. However, that work does not provide a translation from FO to automata, leaving the equivalence question open.

Organization. The paper is organised as follows. In Section 2 we give the necessary preliminaries on words, trees, first order and temporal logics. In Section 3 we recall the two branching-time temporal logics equivalent to FO and investigate their most interesting properties. Section 4 is devoted to the introduction of the class of *graded tree automata* and of its weak and hesitant restrictions. Sections 5 and 6 are the main sections, in which we prove the equivalence of two classes of automata with the two temporal logics discussed in Section 3, while in Section 7 we present the normal forms obtained for them. Finally, Section 8 discusses the results.

2 Preliminaries

Words and trees. Given a finite alphabet Σ, a *finite* (resp., *infinite*) word over Σ is a finite (resp. infinite) sequence of letters in Σ. A *word language* over Σ is a set of words over Σ.

We consider unranked and unordered infinite trees with arbitrary finite branching. A *tree* is a connected acyclic graph $T = (V_T, E_T)$, where V_T is its set of *nodes* and $E_T \subseteq V_T \times V_T$ is its *transition* relation; E_T is total, that is, for every $n \in V_T$, $(n, n') \in E_T$ for some n'. We denote the reflexive and transitive closure of E_T by E_T^*. We denote the root of a tree T by ε_T. Given two nodes v, w in T, we say that v is the *parent* of w if $(v, w) \in E_T$ and that v is a *child* or *successor* of w if $(w, v) \in E_T$. A tree is *unranked* if there is no restriction (apart from finiteness) on the number of children a node might have. If two nodes have the same parent we say they are *siblings*. A tree is *unordered* if the order of the siblings is irrelevant. Moreover, we say that v is an *ancestor* of w if $(v, w) \in E_T^*$, and that v is a *descendant* of w if $(w, v) \in E_T^*$.

The *subtree* of T rooted at a node w is the tree consisting of all the descendants of w. A *path* π of a tree T is a finite or infinite sequence of nodes of T, whose first element is the root of T and where every element but the first one is a child of its predecessor in the sequence. Given a path $\pi = n_0 n_1 \ldots$, we write $\pi(i)$ to refer to n_i.

Given a finite alphabet Σ, a Σ-labelled tree (Σ-tree for short) is a pair $\mathcal{T} = (T, \tau)$ such that T is a tree and $\tau : V_T \to \Sigma$ is a labelling function assigning to each node in T a letter of Σ. A *tree language* over Σ is a set of Σ-trees. We denote by $\mathcal{T}_\Sigma$ the language of all Σ-trees.

First-Order logic. We introduce *monadic First-Order logic on trees with the ancestor relation* (FO for short). Let AP be a set of atomic propositions. Formulas of FO are generated by the following grammar, where p ranges over AP and $x, x_0, x_1, \ldots$ are first-order variables from a set Var:

$$\varphi ::= (x_0 = x_1) \mid (x_0 \leq x_1) \mid p(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi$$

The usual abbreviations $\top, \bot, \wedge, \to, \leftrightarrow, \forall$ are allowed. A variable is *free* if it is not bound to any quantifier. A formula without free variables is called a *sentence*.

An FO formula φ is interpreted over a structure $\mathcal{M} = (\mathcal{T}, \zeta)$, where $\mathcal{T} = (T, \tau)$ is a Σ-tree,[1] with $\Sigma = 2^{\mathrm{AP}}$, and $\zeta : \mathrm{Var} \to V_T$ is a function associating a node of T with each variable. The relational symbols $=$ and $\leq$ are interpreted, respectively, by the identity on the nodes and by the reflexive and transitive closure E_T^* of the transition relation of T; the rest of the semantics is standard. Note that we can equally interpret FO formulas on words, seen as trees whose nodes have at most one successor.

Temporal logics. We now introduce variants of the branching-time temporal logics CTL and CTL*. In doing so, we adopt a suggestive notation that annotates a base logic, e.g. CTL, with superscripts and subscripts, denoting enhancements and restrictions, respectively. The most expressive logical formalism in this setting is denoted cCTL^{*,p}, and can be seen as the extension of CTL* with counting modalities (e.g., for counting the number of successors satisfying a certain property) and past operators (for navigating the tree along ancestors). Formulas of cCTL^{*,p} are generated by the following grammar, where n ranges over the natural numbers and p ranges over a set AP of atomic propositions:

$$\varphi ::= \underbrace{\mathtt{D}^n\varphi}_{\text{counting operators}} \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{E}\varphi \mid \mathtt{X}\varphi \mid \varphi\,\mathtt{U}\,\varphi \mid \underbrace{\mathtt{Y}\varphi \mid \varphi\,\mathtt{S}\,\varphi}_{\text{past operators}}$$

Some abbreviations are $\mathtt{F}\varphi = \top\,\mathtt{U}\,\varphi$, $\mathtt{G}\varphi = \neg\mathtt{F}\neg\varphi$, $\varphi\,\mathtt{R}\,\varphi' = \neg(\neg\varphi\,\mathtt{U}\,\neg\varphi')$, $\varphi\,\mathtt{W}\,\varphi' = \mathtt{G}\varphi \vee (\varphi\,\mathtt{U}\,\varphi')$, $\mathtt{A}\varphi = \neg\mathtt{E}\neg\varphi$.

[1] It is sometimes convenient to identify Σ with AP. Since here we are not concerned with succinctness of formulas or complexity of satisfiability problems, this detail is immaterial.

Formulas of cCTL^{*,p} are evaluated with respect to a Σ-tree $\mathcal{T} = (T, \tau)$, where $\Sigma = 2^{\mathrm{AP}}$, an infinite path π of T, and a position $i \in \mathbb{N}$ along this path. The satisfaction relation $\vDash$ is defined by induction over cCTL^{*,p} formulas as follows (we omit the obvious Boolean cases):

- $\mathcal{T}, \pi, i \vDash \mathtt{D}^n \psi$ *iff* there are n distinct infinite paths $\pi_1, \ldots, \pi_n$ such that (i) they coincide with π up to position i, (ii) they all differ at position $i+1$, and (iii) they satisfy $\mathcal{T}, \pi_j, i+1 \vDash \varphi$ for all $1 \leq j \leq n$,
- $\mathcal{T}, \pi, i \vDash p$ *iff* $p \in \tau(\pi(i))$,
- $\mathcal{T}, \pi, i \vDash \mathtt{E}\psi$ *iff* there is an infinite path π' that coincides with π up to position i and satisfies $\mathcal{T}, \pi', i \vDash \psi$,
- $\mathcal{T}, \pi, i \vDash \mathtt{X}\,\psi$ *iff* $\mathcal{T}, \pi, i+1 \vDash \psi$,
- $\mathcal{T}, \pi, i \vDash \psi\,\mathtt{U}\,\psi'$ *iff* there is $j \geq i$ such that $\mathcal{T}, \pi, j \vDash \psi'$ and $\mathcal{T}, \pi, k \vDash \psi$ for all $i \leq k < j$,
- $\mathcal{T}, \pi, i \vDash \mathtt{Y}\,\psi$ *iff* $i > 0$ and $\mathcal{T}, \pi, i-1 \vDash \psi$,
- $\mathcal{T}, \pi, i \vDash \psi\,\mathtt{S}\,\psi'$ *iff* there is $j \leq i$ such that $\mathcal{T}, \pi, j \vDash \psi'$ and $\mathcal{T}, \pi, k \vDash \psi$ for all $i \geq k > j$.

We say that two formulas ψ and ψ' are *equivalent*[2] if for every Σ-tree $\mathcal{T} = (T, \tau)$ and every infinite path π of T, we have that $\mathcal{T}, \pi, 0 \vDash \psi$ iff $\mathcal{T}, \pi, 0 \vDash \psi'$. We say that a Σ-tree $\mathcal{T}$ is a *model* of a formula φ, denoted $\mathcal{T} \vDash \psi$, if $\mathcal{T}, \pi, 0 \vDash \psi$ for every infinite path π. A formula is *valid* if every Σ-tree is a model of it.

We denote by $\mathcal{L}(\psi)$ the language of models of a given formula ψ, and we shall consider classes of languages defined by formulas in certain fragments of cCTL^{*,p}. In particular, we say that two such fragments are *expressively equivalent* if they define the same class of languages, and that one fragment is *strictly less expressive* than another one when the class of languages defined by the former is strictly contained in the class of languages defined by the latter.

Let us now discuss the main fragments of cCTL^{*,p}. A first fragment is CTL^{*,p}, which is cCTL^{*,p} devoid of the counting operators $\mathtt{D}^n$. In its turn, CTL^{*,p} without the past operators corresponds to the classical CTL* logic. Fragments CTL^p and CTL are obtained by applying the same restrictions (analogous removals of counting and past operators) to cCTL^p, the fragment of cCTL^{*,p} where path quantifiers and future temporal operators must be paired together, as indicated by the following grammar:

$$\overbrace{\varphi ::= \mathtt{D}^n\varphi \mid \overbrace{\underbrace{p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{EX}\,\varphi \mid \mathtt{E}(\varphi\,\mathtt{U}\,\varphi) \mid \mathtt{E}(\varphi\,\mathtt{R}\,\varphi)}_{\text{CTL}} \mid \mathtt{Y}\varphi \mid \varphi\,\mathtt{S}\,\varphi}^{\text{CTL}^p}}^{\text{cCTL}^p}$$

Finally, cCTL* and cCTL are, respectively, cCTL^{*,p} and cCTL^p without past operators. It is well known that CTL is strictly less expressive than CTL^p [33, Theorem 4.1, 4.2], and consequently cCTL is strictly less expressive than cCTL^p.

Finally, we have the Linear-time Temporal Logic with and without past operators, abbreviated, respectively, LTL^p and LTL for short. These logics can be seen as the fragments of CTL^{*,p} and CTL* that are evaluated over unary trees, that is words (notice that in this setting the path quantifier $\mathtt{E}$ becomes pointless). It is worth recalling that LTL and LTL^p are expressively equivalent [27, Theorem 2.2], and they correspond to FO when the latter is evaluated on words as well:

Proposition 1. *[18, Theorem 1.1]* LTL *and* LTL^p *are equivalent to* FO *over finite and infinite words.*

[2] This is a standard notion called *initial equivalence*, sometimes contrasted with the stronger equivalence that evaluates formulas at arbitrary nodes. In this paper, we are only concerned with initial equivalence, and therefore we opt for saying just "equivalent" without further specification.

Word automata. A *Nondeterministic Parity word Automaton* (*NPA*) is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to 2^Q$ is a transition relation (represented as a function towards the powerset of Q), $q_I \in Q$ is an initial state, and $\Omega : Q \to \mathbb{N}$ is a *priority function* associating a number with each state. An NPA is *deterministic* if $\delta : Q \times \Sigma \to Q$.

A *path* of $\mathcal{A}$ on a finite (resp., infinite) word $w = a_0 a_1 \dots$ is a finite (resp., infinite) sequence of states $q_0 q_1 \dots$ such that $q_{i+1} \in \delta(q_i, a_i)$ for all positions i in w. Such a path is called a *run* of $\mathcal{A}$ on w if it is infinite and, moreover, its first state q_0 is the initial state q_I of the automaton. We denote by $\mathsf{inf}(r)$ the set of states visited infinitely often along a run r, and say that r is *successful* if the highest priority of the states in $\mathsf{inf}(r)$ is even. An infinite word w is *accepted* by $\mathcal{A}$ if there is a successful run of $\mathcal{A}$ on w. The language $\mathcal{L}(\mathcal{A})$ *recognized* by $\mathcal{A}$ is the set of words accepted by $\mathcal{A}$. Two automata are *equivalent* if they recognize the same language.

Variants of NPAs are obtained by changing the acceptance conditions. One way is to simply turn existential non-determinism into universal non-determinism, that is, to declare that the automaton accepts a given word whenever *every* run on it is successful; we call these models of automata *universal* (e.g., we have *Universal Parity word Automata*, abbreviated UPA). Another way is to constrain the co-domain of the priority function to be of cardinality 2 and to contain both an odd and an even number. An NPA with this restriction is a *Büchi* (resp., *coBüchi*) automaton if the largest number in the co-domain $cod(\Omega)$ of Ω is even (resp., odd). Note that such an acceptance condition can be equivalently specified by the set $F = \{q \in Q \mid \Omega(q) = \mathsf{max}(cod(\Omega))\}$, whose states are usually called *accepting* or *rejecting* depending on whether we deal with Büchi or coBüchi conditions. Accordingly, a run r is successful for a Büchi (resp., coBüchi) condition if $\mathsf{inf}(r) \cap F \neq \emptyset$ (resp., $\mathsf{inf}(r) \cap F = \emptyset$).

An automaton $\mathcal{A}$ is *counter-free* if the following holds for all states q, all finite words ι, and all numbers n: if $\mathcal{A}$ admits a path on ι^n that starts and ends at q, then it admits a path also on ι that starts and ends at q. By [18, Theorem 1.1], counter-free Büchi nondeterministic automata and FO capture the same class of word languages.

3 Temporal Logics and FO

In this section, we introduce two temporal logics provably equivalent to FO over unranked and unordered finitely branching infinite trees.

The polarized fragment of counting CTL with past operators. The first temporal logic we discuss is a syntactic restriction on cCTL^p, introduced and studied by Schlingloff in [49]. More precisely, in [49, Theorem 4.5] it is shown that there is a fragment of cCTL^p that is expressively equivalent to FO over infinite finitely branching trees. We call this fragment *Polarized* cCTL^p, abbreviated cCTL$^p_\pm$. Its formulas are generated by the following grammar:

$$\varphi ::= \mathtt{D}^n \varphi \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{EX}\,\varphi \mid \mathtt{E}(\varphi\,\mathtt{U}\,\varphi) \mid \mathtt{Y}\,\varphi \mid \varphi\,\mathtt{S}\,\varphi$$

Accordingly, we denote by cCTL$_\pm$ the sub-fragment obtained from cCTL$^p_\pm$ by disallowing past operators.

The only difference w.r.t. cCTL^p and cCTL is that $\mathtt{E}(\varphi\,\mathtt{R}\,\varphi)$ is not included as primitive in the syntax (and it cannot be restored as an abbreviation). Indeed, it is possible to define the following abbreviations: $\mathtt{AX}\,\varphi = \neg\mathtt{EX}\neg\varphi$, $\mathtt{EF}\,\varphi = \mathtt{E}(\top\,\mathtt{U}\,\varphi)$, $\mathtt{AG}\,\varphi = \neg\mathtt{EF}\neg\varphi$, $\mathtt{A}(\varphi\,\mathtt{R}\,\varphi') = \neg\mathtt{E}(\neg\varphi\,\mathtt{U}\neg\varphi')$; however, formulas of the form $\mathtt{EG}\,\varphi$, $\mathtt{E}(\varphi\,\mathtt{R}\,\varphi')$, $\mathtt{AF}\,\varphi$, $\mathtt{A}(\varphi\,\mathtt{U}\,\varphi')$ are not derivable. The semantic intuition behind this syntax is that formulas that existentially quantify over branches can only express co-safety properties; dually, universal quantifications can only be paired with safety properties. This will become transparent in the following of

the paper, but note already that this is similar to what happens in a fragment of the modal μ-CALCULUS isolated in [15], that the authors call *completely additive*, and that allows only the interplay of least fixpoint operators and existential modalities, and, dually, of greatest fixpoints and universal modalities.

Before proceeding, we address two natural issues: whether $\mathrm{cCTL}^p_\pm$ is strictly less expressive than cCTL^p and whether $\mathrm{cCTL}_\pm$ is strictly less expressive than $\mathrm{cCTL}^p_\pm$. It turns out that the answer is positive in both cases, for the following two propositions.

Proposition 2. *[2, Theorem 3]] No* FO *formula can express the* CTL *formula* $\mathtt{AF}\,p$.

The result follows from the equivalence of FO and $\mathrm{cCTL}^p_\pm$.

Proposition 3. *[33, Lemma A.2] No* $\mathrm{cCTL}_\pm$ *formula can express the* $\mathrm{CTL}^\star$ *formula* $\mathtt{E}((p \vee q\,\mathtt{U}\,r)\,\mathtt{U}\,s)$.

By investigating the proof of [33, Lemma A.1], it turns out that there is indeed a formula of $\mathrm{cCTL}^p_\pm$ equivalent to $\mathtt{E}((p \vee q\,\mathtt{U}\,r)\,\mathtt{U}\,s)$ (it is quite long so we omit it here). Note how such a formula predicates about a *finite prefix* of the selected path. Finally, to sum up: $\mathrm{cCTL}_\pm$ is strictly less expressive than $\mathrm{cCTL}^p_\pm$, which is in turn less expressive than cCTL^p.

Counting CTL* over finite paths. The second temporal logic equivalent to FO that we investigate is $\mathrm{cCTL}^\star_f$. The equivalence of this logic with FO can be established easily by adapting the model-theoretic argument of [39] and it was first noticed in [2, Proposition 3].

$\mathrm{cCTL}^\star_f$ is cCTL* with path quantification ranging over *finite paths*. Given a 2^{AP}-tree $\mathcal{T} = (T, \tau)$, a finite non-empty path π of T, and a position i on the path, the satisfaction relation $\vDash$ of a $\mathrm{cCTL}^\star_f$ formula is defined as follows (once again we omit the obvious Boolean cases):

- $\mathcal{T}, \pi, i \vDash \mathtt{D}^n\psi$ *iff* path π does not end at $\pi(i)$ and there are n distinct finite paths $\pi_1, \ldots, \pi_n$ such that (i) they coincide with π up to position i, (ii) they all differ at position $i+1$, and (iii) they satisfy $\mathcal{T}, \pi_j, i+1 \vDash \varphi$ for all $1 \le j \le n$,
- $\mathcal{T}, \pi, i \vDash p$ *iff* $p \in \tau(\pi(i))$,
- $\mathcal{T}, \pi, i \vDash \mathtt{E}\psi$ *iff* there is a finite path π' coinciding with π up to i, such that $\mathcal{T}, \pi', i \vDash \psi$,
- $\mathcal{T}, \pi, i \vDash \mathtt{X}\,\psi$ *iff* path π does not end at $\pi(i)$ and $\mathcal{T}, \pi, i+1 \vDash \psi$.
- $\mathcal{T}, \pi, i \vDash \psi\,\mathtt{U}\,\psi'$ *iff* there is $j \ge i$ such that $\mathcal{T}, \pi, j \vDash \psi'$ and $\mathcal{T}, \pi, k \vDash \psi$ for all $i \le k < j$,

While the semantics for atomic propositions and the temporal operator until $\mathtt{U}$ is unchanged compared to cCTL*, the other clauses must be slightly adapted to deal with finite path quantification. The notions of formula equivalence and model must be adapted accordingly: two formulas ψ and ψ' are *equivalent* if for every Σ-tree $\mathcal{T} = (T, \tau)$ and every finite non-empty path π of T, we have that $\mathcal{T}, \pi, 0 \vDash \psi$ iff $\mathcal{T}, \pi, 0 \vDash \psi'$; a Σ-tree $\mathcal{T}$ is a *model* of a formula φ, denoted $\mathcal{T} \vDash \psi$, if $\mathcal{T}, \pi, 0 \vDash \psi$ for every finite non-empty path π.

In $\mathrm{cCTL}^\star_f$, it makes sense to define an abbreviation for the dual of $\mathtt{X}$, defined as $\tilde{\mathtt{X}}\varphi = \neg\mathtt{X}\neg\varphi$, thus corresponding to the semantic clause:

- $\mathcal{T}, \pi, i \vDash \tilde{\mathtt{X}}\,\psi$ *iff* the path π ends at $\pi(i)$ or $\mathcal{T}, \pi, i+1 \vDash \psi$.

Notice that, in cCTL*, the semantics of $\mathtt{X}$ and $\tilde{X}$ coincide ($\mathtt{X}$ is the dual of itself). The introduction of $\tilde{\mathtt{X}}$ changes a few things. Recall that $\mathtt{F}\varphi \leftrightarrow \varphi \vee \mathtt{X}\mathtt{F}\varphi$ and $\varphi\,\mathtt{U}\,\psi \leftrightarrow \psi \vee (\varphi \wedge \mathtt{X}(\varphi\,\mathtt{U}\,\psi))$. Now, by putting $\mathtt{G}\varphi = \neg\mathtt{F}\neg\varphi$ and $\varphi\,\mathtt{R}\,\psi = \neg(\neg\varphi\,\mathtt{U}\neg\psi)$, one obtains $\mathtt{G}\varphi \leftrightarrow \varphi \wedge \tilde{\mathtt{X}}\mathtt{G}\varphi$ and $\varphi\,\mathtt{R}\,\psi \leftrightarrow \psi \wedge (\varphi \vee \tilde{\mathtt{X}}(\varphi\,\mathtt{R}\,\psi))$. Moreover, $\neg\mathtt{E}\mathtt{X}\neg\psi = \mathtt{A}\tilde{\mathtt{X}}\psi$.

The equivalence of $\mathrm{cCTL}^\star_f$ with $\mathrm{cCTL}^p_\pm$ could come as a surprise, since the former does not share the syntactic restriction of the latter. In the following proposition we show how finite path quantification actually enforces semantically the same behaviour of $\mathrm{cCTL}^p_\pm$.

Proposition 4. *Let φ, ψ, and γ be* $\mathrm{cCTL}^\star_f$ *formulas. Then, the following are valid* $\mathrm{cCTL}^\star_f$ *equivalences:*

1. $\mathtt{E}\tilde{\mathtt{X}}\varphi \leftrightarrow \top$ *and* $\mathtt{A}\mathtt{X}\varphi \leftrightarrow \bot$,
2. $\varphi\,\mathtt{R}\,\psi \leftrightarrow \alpha^{\mathtt{R}}_{\varphi,\psi}$ *and* $\varphi\,\mathtt{U}\,\psi \leftrightarrow \alpha^{\mathtt{U}}_{\varphi,\psi}$, *where* $\alpha^{\mathtt{R}}_{\varphi,\psi} = \psi\,\mathtt{U}\,((\tilde{X}\bot \vee \varphi) \wedge \psi)$ *and* $\alpha^{\mathtt{U}}_{\varphi,\psi} = \psi\,\mathtt{R}\,((\mathtt{X}\top \wedge \varphi) \vee \psi)$

3. $\mathtt{E}(\varphi\,\mathtt{R}\,\psi) \leftrightarrow \mathtt{E}(\alpha^{\mathtt{R}}_{\varphi,\psi})$ *and* $\mathtt{A}(\varphi\,\mathtt{U}\,\psi) \leftrightarrow \mathtt{A}(\alpha^{\mathtt{U}}_{\varphi,\psi})$,
4. $\mathtt{EX}(\varphi\,\mathtt{R}\,\psi) \leftrightarrow \mathtt{EX}(\alpha^{\mathtt{R}}_{\varphi,\psi})$ *and* $\mathtt{A\tilde{X}}(\varphi\,\mathtt{U}\,\psi) \leftrightarrow \mathtt{A\tilde{X}}(\alpha^{\mathtt{U}}_{\varphi,\psi})$
5. $\mathtt{E}(\varphi\,\mathtt{U}(\psi\,\mathtt{R}\,\gamma)) \leftrightarrow \mathtt{E}(\varphi\,\mathtt{U}\,\alpha^{\mathtt{R}}_{\psi,\gamma})$ *and* $\mathtt{A}(\varphi\,\mathtt{R}(\psi\,\mathtt{U}\,\gamma)) \leftrightarrow \mathtt{A}(\varphi\,\mathtt{R}\,\alpha^{\mathtt{U}}_{\psi,\gamma})$,
6. $\mathtt{E}((\varphi\,\mathtt{R}\,\psi)\,\mathtt{U}\,\gamma) \leftrightarrow \mathtt{E}(\alpha^{\mathtt{R}}_{\varphi,\psi}\,\mathtt{U}\,\gamma)$ *and* $\mathtt{A}((\varphi\,\mathtt{U}\,\psi)\,\mathtt{R}\,\gamma) \leftrightarrow \mathtt{A}(\alpha^{\mathtt{U}}_{\varphi,\psi}\,\mathtt{R}\,\gamma)$.

Equivalences with F and G can be obtained from the equivalences with U and R. Thus, even if it seems that cCTL$^{\star}_{f}$ is able to syntactically specify properties impossible to express with cCTL$^{p}_{\pm}$, the finite path quantification indeed trivializes many of these properties. Interestingly, both logics have an ability the other one lacks, but as we have already mentioned they share the same expressive power: cCTL$^{p}_{\pm}$ enjoys past operators and thus the ability to reason backwards over the input tree, while cCTL$^{\star}_{f}$ is able to nest U and R operators without a path quantifiers in between. The equivalence with FO shows that these abilities yield the same expressiveness. Indeed, notice that the cCTL$^{p}_{\pm}$ formula equivalent to the above cited CTL* formula $\mathtt{E}((p \vee q\,\mathtt{U}\,r)\,\mathtt{U}\,s)$, that makes cCTL$^{p}_{\pm}$ strictly more expressive than cCTL$_{\pm}$ (see Proposition 3 above), preserves its meaning also in cCTL$^{\star}_{f}$. Basically, the syntactic restrictions of cCTL$^{p}_{\pm}$ are semantically retained in cCTL$^{\star}_{f}$. Once again, we mention the completely additive fragment of the modal μ-CALCULUS to emphasize its relationship with these two logics and to better understand their behaviour. In [15, Theorem 3.6], this fragment is shown equivalent to PDL, while in [14, Theorem 1] it is shown that PDL is equivalent to a fragment of Weak Monadic Chain Logic (WMCL). Since FO is itself a fragment of WMCL, it makes sense that the temporal logics equivalent to it enjoys such "polarized" properties, meaning that the existential and universal modalities are bounded to express dual properties, such as co-safety and safety ones.

4 Hesitant and Weak Graded Tree Automata

Here we study a general automaton model inspired by the class of *Graded Tree Automata*, originally introduced in [31, Section 3]. We progressively restrict the recognising power of this model, until we obtain two classes of automata provably equivalent to cCTL$^{p}_{\pm}$ and cCTL$^{\star}_{f}$, and consequently to FO.

Graded tree automata. A set of *positive Boolean formula* over a set X is denoted by $\mathcal{B}^{+}(X)$, and is composed of formulas of the form $\top, \bot, x, \theta \vee \theta, \theta \wedge \theta$, where $x \in X$.

A *Graded Alternating Parity Tree Automaton* (GTA) is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$, where Q is a finite set of states, Σ is a finite alphabet, $q_I \in Q$ is an initial state, $\Omega : Q \to \mathbb{N}$ is a priority function, and $\delta : Q \times \Sigma \to \mathcal{B}^{+}(\{\Diamond_k, \Box_k \mid k > 0\} \times Q)$ is an alternating transition function. We call *atom* any pair $(\heartsuit_k, q)$, where $\heartsuit$ is either $\Diamond$ or $\Box$. When $k = 1$, we simply write $\Diamond$ instead of $\Diamond_k$ and $\Box$ instead of $\Box_k$.

A *run* of a GTA $\mathcal{A}$ on a Σ-labeled tree $\mathcal{T} = (T, \tau)$, is a $V_T \times Q$-labelled tree $\mathcal{R} = (R, \rho)$, where $R = (V_R, E_R)$ is a tree (possibly with some leaves) and $\rho : V_R \to V_T \times Q$ is a labeling function that associates every mnode of R to a node of the input tree and a state of the automaton. Before describing the additional conditions that must be satisfied by a run, we briefly explain how to evaluate a positive Boolean formula that may appear in the transition function of the automaton. We start by considering the case of atoms. Given a candidate run $\mathcal{R} = (R, \rho)$ on $\mathcal{T} = (T, \tau)$ and a node $s \in V_R$, with $\rho(s) = (n, q)$, we let:

- $s \vDash (\Diamond_k, q')$ if n has at least k successors in T, say, $n_1, \ldots, n_k$, and, for each $i = 1, \ldots, k$, s has at least one successor in R labeled by (n_i, q');
- $s \vDash (\Box_k, q')$ if for all but $k-1$ successors n' of n in T, there is a successor of s in R that is labeled by (n', q').

The satisfaction relation $\vDash$ is then extended to the constants $\top$ and $\bot$ and to the Boolean connectives $\wedge$ and $\vee$ in the obvious way, e.g. by letting $s \vDash \top$ when s is a leaf. Now, the additional conditions that must be satisfied by a run $\mathcal{R} = (R, \rho)$ of $\mathcal{A}$ on $\mathcal{T} = (T, \tau)$ are as follows: (a) $\rho(\varepsilon_R) = (\varepsilon_T, q_I)$, (b) for every node

$s \in R$, with $\rho(s) = (n, q)$, $s \vDash \delta(q, \tau(n))$. Given such a run $\mathcal{R} = (R, \rho)$ and a maximal path π in it (which can be finite or infinite), we denote by $\mathsf{inf}(\pi)$ the set of states visited infinitely often along π, and we say that π is *accepting* if either it terminates in a leaf (implying that there is a transition reaching $\top$), or it is infinite and the highest priority assigned by Ω to the states in $\mathsf{inf}(\pi)$ is even. Finally, a run $\mathcal{R}$ is *successful* if every maximal path in it is accepting.

Weak and Hesitant tree automata. We are interested in a proper subclass of GTAs, characterized by a *hesitant* partition on the state set and a *weak* acceptance condition. We define the weak acceptance condition first, following [41].

A *Weak Graded Alternating Parity Tree Automaton* $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$ (WGT) is a GTA such that there is a partition of Q into disjoint non-empty sets (from now on, *components*) $\{Q_1, ..., Q_k\}$ and a partial order $\leq$ such that the transitions from a state in Q_i can only lead to states in Q_i or to states in a component with lower order. The acceptance condition is weak because we enforce that every component either contains only states marked even by the priority function or contains only states marked odd.

A *Hesitant Weak Graded Alternating Tree Automaton* (HWGT) $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$ is a WGT such that each component is of one of the following three types:

- Q_i is *existential*, if for all $\sigma \in \Sigma$ and for all $q, q' \in Q_i$, q' can appear in the disjunctive normal form of $\delta(q, \sigma)$ only in an atom $(\Diamond, q')$, and only disjunctively related to other atoms with states in Q_i;
- Q_i is *universal*, if for all $\sigma \in \Sigma$ and for all $q, q' \in Q_i$, q' can appear in the conjunctive normal form of $\delta(q, \sigma)$ only in an atom $(\Box, q')$, and only conjunctively related to other atoms with states in Q_i;
- Q_i is *transient*, if for all $\sigma \in \Sigma$ and for all $q, q' \in Q_i$, q' does not appear in any atom in $\delta(q, \sigma)$.

Note that in the existential and universal components $\Diamond_k$ and $\Box_k$ must have $k = 1$, if they are paired with a state of the component. This is the *hesitant* constraint on the partition of the state set, introduced in [32, Section 5.1]. Clearly, every path of a given run will eventually get stuck in an existential or a universal component. We let every state in a universal (resp., existential) component be marked even (resp., odd) by the priority function. Let Q_i be the component in which a path π of a given run $\mathcal{R}$ gets stuck: if Q_i is universal, π satisfies the accepting condition if it visits infinitely often a state in Q_i; if Q_i is existential, π satisfies the accepting condition if it visits finitely often every state in Q_i. Notice that this accepting condition is weak, since it basically states that the universal components are entirely accepting and the other components are entirely rejecting. It is possible to have a HWGT $\mathcal{A}$ with a positive boolean formula θ as initial condition instead of a state. It is then possible to convert it to an automaton with an initial state by having as initial an extra state q_θ, such that $\delta(q_\theta, \sigma) = \theta$ for every $\sigma \in \Sigma$. Such an automaton will be denoted by $\mathcal{A}^\theta$. Given a HWGT $\mathcal{A}$ and a state $q \in Q$, we denote by $\mathcal{A}^q$ the automaton $\mathcal{A}$ where q is the starting state.

Summing up, we will work with a HWGT $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$, with an hesitant partition on the state set and a weak acceptance condition, stating that every state in the existential components are only visited finitely often or some state in the universal components is visited infinitely often. However, this class of automata is still not *weak* enough to be equivalent to FO. Indeed, note the striking similarity of HWGTs with the *Additive Weak Parity Automata* (*AWA*) introduced in [14, Section IV] and proven equivalent to WMCL on trees. This suggests that HWGTs as above defined are equivalent to this logic, too. In the following, we will further restrict their recognising power and yield equivalence results with the two temporal logics above introduced.

5 Automaton-based Characterization of $\mathrm{cCTL}^p_\pm$

What currently prevents us from saying that HWGTs and $\mathrm{cCTL}^p_\pm$ are equivalent are the following two observations: an HWGT cannot reason upward along the input tree and its components have no restriction other than being purely accepting or purely rejecting. In this section we will solve both these problems in a straightforward way, and prove that the obtained class of automata is equivalent to $\mathrm{cCTL}^p_\pm$.

Two-Way Linear HGTA. A *Two-Way HWGT* (*2HWGT*) is defined exactly like a HWGT, namely, as a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$, but where $\delta : Q \times (\Sigma \times \{0,1\}) \to \mathcal{B}^+(\{\Diamond_k, \Box_k, -1\} \times Q)$. The idea is that the automaton can send states towards the successors of the currently visited node, as well as towards its parent, if it exists. These transitions are determined, as usual, from the current state and from the label of the visited node, but they might also depend on whether the head is at the root or not — for this, we explicitly mark the nodes of the input tree with a flag that distinguishes the root from the other nodes.

Given a run $\mathcal{R} = (R, \rho)$ of a 2HWGT $\mathcal{A}$ over a Σ-labelled tree as above, and a node s in V_R such that $\rho(s) = (n', q')$, the semantics of an atom $(-1, q) \in \mathcal{B}^+(\{\Diamond_k, \Box_k, -1\} \times Q)$, for some $q \in Q$, is as follows:

- $s \vDash (-1, q)$ *iff* n' has a parent n'' and s has a parent labelled by (n'', q) – note in particular that when the head is at the root this atom is always false.

Moreover, we enlarge the hesitant types adding to the above defined transient, existential and universal the following. Given a 2HWGT $\mathcal{A}$ and a component Q_i of $\mathcal{A}$:

- Q_i is *upward*, if for all $\sigma \in \Sigma$ and for all $q, q' \in Q_i, q'$ can appear in $\delta(q, \sigma)$ only in the form $(-1, q)$.

Finally, a *linear* HWGT, or HLGT for short, is a HWGT $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$ in which the partitioning of the state set is composed entirely of *singletons*. This latter condition is reminiscent of a restriction on Additive Weak Parity Automata (AWA) (already cited above at the end of section 4) introduced in [26, Definition 3.3.1], and giving rise to the so-called *Antisymmetric* AWA. In that paper, the author states the expressive equivalence between AWA and WMCL, and conjectures an equivalence between Antisymmetric AWA and FO (he was only able to prove that FO is at least as expressive as Antisymmetric AWA). This conjectured correspondence will be discussed again at the end of the section.

Finally, merging the two restrictions together, we get the class of 2HLGT, i.e., two-way automata in which every state is either transient, existential, universal or upward. We will now prove that 2HLGT are equivalent to $\mathrm{cCTL}^p_\pm$.

5.1 Equivalence of 2HLGT and $\mathrm{cCTL}^p_\pm$

Here we prove that given a 2HLGT, it is always possible to obtain a $\mathrm{cCTL}^p_\pm$ formula equivalent to it. The approach is a combination of the translation procedure of [35, Theorem 6] and [11, Theorem 3.1].

Theorem 5. *Given a 2HLGT $\mathcal{A}$ over $\Sigma = 2^{AP}$, for a set of atomic propositions AP, it is always possible to obtain a $\mathrm{cCTL}^p_\pm$ formula $\varphi_\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi_\mathcal{A})$.*

Proof. Fix a 2HLGT $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, \Omega \rangle$. We will show how to translate any given $q \in Q$ by induction over the order of the singletons in the partition. This will yield the result, by the translation of q_I. For a given state q or a positive boolean formula θ, we will use in the following $\chi(q)$ and $\chi(\theta)$ for the $\mathrm{cCTL}^p_\pm$ formulas equivalent to $\mathcal{A}^q$ and $\mathcal{A}^\theta$, respectively.

First of all, we translate any $\sigma \in \Sigma$ by the $\mathrm{cCTL}^p_\pm$ formula $\psi_\sigma = \bigwedge_{p \in \sigma} p \wedge \bigwedge_{p \notin \sigma} \neg p$. By construction, for every $q \in Q$ and every $\sigma \in \Sigma$, if $q' \in \delta(q, \sigma)$ and $q' \neq q$, we know that q' is of lower order than q with respect to the partition of the state set. By inductive hypothesis, we could assume to have a $\mathrm{cCTL}^p_\pm$ formula $\chi(q')$ for each one of them, if it wasn't for atoms. Indeed, since $\mathrm{cCTL}^p_\pm$ is closed under $\vee$ and $\wedge$, and $\top$ and $\bot$ are already allowed formulas of the logic, we can assume to translate this kind of formulas without

any problem. With regards to atoms, however, the translation is also easy: $(\Diamond_k, q)$ is translated as $\mathtt{D}^k\chi(q)$, $(\Box_k, q)$ is translated as $\neg\mathtt{D}^k\neg\chi(q)$ and $(-1, q) = \mathtt{Y}\chi(q)$. We just have to show how to obtain $\chi(q)$. This is done in the following way. Having fixed a $q \in Q$ and a $\sigma \in \Sigma$, $\delta(q,\sigma)$ is a positive boolean formula θ. Moreover, if we ignore a (possible) atom in wich q itself occurs, it is a positive boolean formula composed entirely of states of order lower than q, denoted by $\chi(\theta_{q,\sigma})$. The subscript says that it is the formula obtained by $\delta(q,\sigma)$. Now, we define the recursion to obtain χ. This is done by induction on the order on the components of the partition, that is well-founded.

Given $q \in Q$ and $\sigma \in \Sigma$, we know that q can only be of a specific type because the automaton is hesitant. For any one of these types we can present $\delta(q,\sigma)$ in a precise way, as follows:

- if q is transient, $\delta(q,\sigma)$ is of the form θ',
- if q is existential, $\delta(q,\sigma)$ is of the form $((\Diamond, q) \wedge \theta) \vee \theta')$,
- if q is universal, $\delta(q,\sigma)$ is of the form $((\Box, q) \wedge \theta) \vee \theta')$,
- if q is upward, $\delta(q,\sigma)$ is of the form $((-1, q) \wedge \theta) \vee \theta')$.

We impose that q does not appear neither in θ nor θ' and this is always possible because if there are more instances of q in $\delta(q,\sigma)$ we can always rewrite it with just one. The reasoning behind this form is that θ is the positive boolean formula that appears conjunctively related with q itself, while θ' is only disjunctively related to q. This means that whenever θ is true, the automaton has to keep staying in state q, while whenever θ' is true, it must leave q.

Recall that by induction we can assume to have $\chi(\theta_{q,\sigma})$ and $\chi(\theta'_{q,\sigma})$. Thanks to this, we can obtain two formulas related to q, that tells if the automaton stays in the state or leaves it. Namely:

$$\gamma_q = \bigvee_{\sigma \in \Sigma} \psi_\sigma \wedge \chi(\theta_{q,\sigma}) \qquad \gamma'_q = \bigvee_{\sigma \in \Sigma} \psi_\sigma \wedge \chi(\theta'_{q,\sigma})$$

What is missing is how to translate a single state:

- if q is transient, $\varphi_q = \gamma'_q$,
- if q is existential, $\varphi_q = \mathtt{E}(\gamma_q \mathtt{U} \gamma'_q)$,
- if q is universal, $\varphi_q = \mathtt{A}(\gamma_q \mathtt{W} \gamma'_q)$,
- if q is upward, $\varphi_q = \gamma_q \mathtt{S} \gamma'_q$.

□

Proposition 6. *For any $q \in Q$, $\mathcal{L}(\mathcal{A}^q) = \mathcal{L}(\chi(q))$.*

The other direction of the translation is classic and can be obtained by adapting in a straightforward way the proof of [17, Section 14.7.2].

Theorem 7. *Given a* cCTL$^p_\pm$ *formula φ one can construct a 2HLGT $\mathcal{A}_\varphi$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$.*

Clearly, the equivalence between 2HLGT and cCTL$^p_\pm$ is allowed by backward transitions. If one removes the possibility of going "up" the input tree, one also loses the expressive power given by the past operators of the logic. By inspecting the translations, it is clear that the two-way head movements and the past temporal operators are intertwined, and that if one removes them both from the two formalisms there is still a possibility of translations between the formalisms thus obtained, yielding the following lemma.

Lemma 8. *HLGT and* cCTL$_\pm$ *are equivalent formalisms.*

By combining this with Propostion 3, one obtains that HLGT are strictly less expressive than FO. The following conjecture implies that the class of Antisymmetric AWA introduced in [26] and above mentioned is strictly less expressive than FO over trees.

Conjecture. *HLGT and Antisymmetric Additive-Weak Parity Automata are equivalent formalisms.*

6 Automaton-based Characterization of cCTL$^\star_f$

In this section, we complete the picture of logics-vs-automata correspondences by showing a class of automata, in fact, restrictions of HWGTs, that are expressively equivalent to cCTL$^\star_f$. We will follow the approach from [3], with minor cosmetic changes. We shall mostly focus on the translation from the restriction of HWGTs (to be defined soon) to equivalent cCTL$^\star_f$ formulas, which is the interesting part of the proof.

Linearization of components. The first step consists in focusing on the non-transient components of an HWGT, and thinking of them as suitable word automata over an expanded alphabet. We remark that in this step, we move from automata that process trees to automata that process words. This abstraction is possible thanks to the restrictions imposed to an HWGT.

Definition 9. *Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, (Q_\forall, Q_\exists)\rangle$ be an HWGT and let B be the set of all atoms that occur in the images of its transition function. For each component Q_i of $\mathcal{A}$ and each state $q \in Q_i$, let q_{exit} be a fresh state and let $\mathcal{A}_{Q_i,q} = \langle Q_i \uplus \{q_{\text{exit}}\}, \Sigma \times 2^B, \delta', q, Q_i\rangle$ be the word automaton obtained from $\mathcal{A}$ by restricting the state set to Q_i and adding q_{exit}, declaring q to be the new initial state, annotating the input letters with subsets of B, and redefining the transition relation and the acceptance condition as follows:*

1. *if Q_i is an existential component, then $\mathcal{A}_{Q_i,q}$ is a non-deterministic coBüchi automaton (NCA), with Q_i as set of rejecting states (implying that the only successful runs are those that eventually exit Q_i), and with transitions defined by*
   - *$q_{\text{exit}} \in \delta'(q', (\sigma, C))$ (for $q' \in Q_i \uplus \{q_{\text{exit}}\}$, $\sigma \in \Sigma$, $C \subseteq B$) whenever $q' = q_{\text{exit}}$ or the disjunctive normal form of $\delta(q', \sigma)$ contains a clause of the form $\bigwedge C$ (note that, thanks to the restrictions imposed to an HWGT, the states appearing in C have all lower order than those in Q_i);*
   - *$q'' \in \delta_{Q_i}(q', (\sigma, C))$ (for $q', q'' \in Q_i$, $\sigma \in \Sigma$, $C \subseteq B$) whenever the disjunctive normal form of $\delta(q', \sigma)$ contains a clause of the form $(\Diamond, q'') \wedge \bigwedge C$;*
2. *dually, if Q_i is a universal component, then $\mathcal{A}_{Q_i,q}$ is a universal Büchi automaton (UBA), with Q_i as set of accepting states (implying that the successful runs are those that remain inside Q_i), and with transitions defined by*
   - *$q_{\text{exit}} \in \delta'(q', (\sigma, C))$ (for $q' \in Q_i \uplus \{q_{\text{exit}}\}$, $\sigma \in \Sigma$, $C \subseteq B$) whenever $q' = q_{\text{exit}}$ or the conjunctive normal form of $\delta(q', \sigma)$ contains a clause of the form $\bigvee C$;*
   - *$q'' \in \delta_{Q_i}(q', (\sigma, C))$ (for $q', q'' \in Q_i$, $\sigma \in \Sigma$, $C \subseteq B$) whenever the conjunctive normal form of $\delta(q', \sigma)$ contains a clause of the form $(\Box, q'') \vee \bigvee C$.*

We denote by B_{Q_i} the set of subsets C of B for which there is $q' \in Q_i$ with $\delta'(q', (\sigma, C)) \cap Q_i \neq \varnothing$ – intuitively, these are the annotations that allow the automaton $\mathcal{A}_{Q_i,q}$ to enter a lower component.

The word languages recognized by the $\mathcal{A}_{Q_i,q}$'s are quite simple. Specifically, when Q_i is an existential component, $\mathcal{A}_{Q_i,q}$ accepts by exiting Q_i, implying that it recognizes a *co-safety language*, namely, a language of the form $F\Sigma^\omega$, for some finite $F \subseteq \Sigma^*$. Dually, when Q_i is universal, $\mathcal{A}_{Q_i,q}$ accepts by staying inside Q_i, implying that it recognizes a *safety language*, i.e. the complement of a co-safety language. Summing up, with the above definition we have achieved two crucial goals. First, we moved from tree automata to word automata, thus enabling the use of characterizations of subclasses of languages. Second, we encoded the behaviour of each component as a (co-)safety language, which is essentially a property referred to *finite* prefixes. This justifies the existence of a translation from (a suitable restriction of) HWGTs to cCTL$^\star_f$, the variant of cCTL* with quantification on finite paths.

Counter-free word automata. The next step is to bound the expressive power of these word automata to get FO expressiveness, that over words is equivalent to LTL, as already remarked. The goal is then to

translate each automaton $\mathcal{A}_{Q_i,q}$ to an equivalent LTL formula and, combining together the formulas thus obtained, we will get a single cCTL$^\star_f$ formula that defines the entire language of trees recognized by the original HWGT.

Of course not all (co-)safety languages are LTL-definable. This means that, in order to enable the desired translation, we need to first identify which HWGTs produce component automata $\mathcal{A}_{Q_i,q}$ that are within the expressiveness of LTL. For this, we shall rely on the well-known characterization of LTL in terms of languages recognized by counter-free automata [18, Theorem 1.1]. In view of this, it is tempting to simply require that all non-transient components of the given HWGT induce counter-free word automata $\mathcal{A}_{Q_i,q}$: this would indeed guarantee that each $\mathcal{A}_{Q_i,q}$ translates to an equivalent LTL formula. However, a straightforward adaptation of [3, Example 5.1] shows an HWGT in which all non-transient components induce counter-free word automata, and yet the language recognized by the HWGT is not definable in FO. In particular, this means that the counter-free condition on the components will not guarantee the possibility of combining together the LTL formulas obtained in the previous step so as to obtain a single cCTL$^\star_f$ equivalent to the HWGT. However, in [3] it is also shown that one remains in FO if, in addition to the counter-free condition on the components, a mutual exclusion property is also enforced:

Definition 10. *An HWGT $\mathcal{A}$ satisfies the* mutual exclusion *property if for every non-transient component Q_i and every $C \neq C' \in B_{Q_i}$, there are atoms $\alpha \in C$ and $\beta \in C'$ such that $\mathcal{L}(\mathcal{A}^\alpha)$ is the complement of $\mathcal{L}(\mathcal{A}^\beta)$.*

We denote by HWGT$_{cf}$ the subclass of HWGTs that satisfy the mutual exclusion property and the counter-free condition on the word automata induced by non-transient components. Accordingly, from now on, we assume that $\mathcal{A}$ is an HWGT$_{cf}$.

Let us now inspect the translation from the word automata $\mathcal{A}_{Q_i,q}$'s to equivalent LTL formulas. Since the $\mathcal{A}_{Q_i,q}$'s recognize safety or co-safety languages, it is useful to first recall the *safe* and *co-safe fragments* of LTL, that capture precisely these languages within LTL. The safe (resp., co-safe) fragment of LTL, denoted SafeLTL (resp., coSafeLTL), allows the usual atomic propositions and Boolean connectives (except for negation), the next operator `X`, and the release operator `R` (resp., the until operator `U`). On the automaton side, we also recall that SafeLTL (resp., coSafeLTL) languages are recognized precisely by the so-called *looping*, counter-free deterministic Büchi (resp., deterministic coBüchi) automata [10], where looping means that all states are accepting (resp., rejecting), with the exception of a single sink state. Building upon this result, the following can be obtained with an adaptation of the argument in [38, Theorem 5.1]:

Lemma 11. SafeLTL *(resp.,* coSafeLTL*) and counter-free, looping, universal Büchi (resp., non-deterministic coBüchi) automata are expressively equivalent formalisms.*

6.1 Equivalence of HWGT$_{cf}$ and cCTL$^\star_f$

In this section, we prove the equivalence of HWGT$_{cf}$ and the logic cCTL$^\star_f$. This equivalence yields a normal form for cCTL$^\star_f$ formulas, which turns out to be a sort of *polarized* cCTL*.

Theorem 12. *Let $\mathcal{A}$ be a HWGT$_{cf}$ over the alphabet $\Sigma = 2^{AP}$. Then, one can construct a* cCTL$^\star_f$ *formula $\varphi_\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi_\mathcal{A})$.*

Proof. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, (Q_\forall, Q_\exists)\rangle$ be a HWGT$_{cf}$. For every state $q \in Q$, we can obtain a cCTL$^\star_f$ formula $\chi(q)$ such that $\mathcal{L}(\mathcal{A}^q) = \mathcal{L}(\chi(q))$ and the theorem follows by setting q to be q_I. The proof is by induction on the order of the components. Moreover, for every $\sigma \in \Sigma$, we denote by ψ_σ the cCTL$^\star_f$ formula: $\bigwedge_{p\in\sigma} p \wedge \bigwedge_{p\notin\sigma} \neg p$. So, let's fix a component Q_i, and let's say $q \in Q_i$. We consider the case in which Q_i is universal: the transient case is straightforward and the existential case can

be restored by a dualization argument. Thus, suppose Q_i is universal. Then, we can focus on the counter-free looping UBA $\mathcal{A}_{Q_i,q} = \langle Q_i \uplus \{\mathrm{q_{exit}}\}, 2^{AP} \times B_{Q_i}, \delta_{Q_i}, q, Q_i\rangle$, as given in Definition 9. Moreover, since every $C \in B_{Q_i}$ is composed of atoms whose states belong to components of lower order than i, by induction hypothesis we can assume to have a cCTL$^\star_f$ formula $\chi(C)$ for every $C \in \Gamma_{Q_i}$, such that $\mathcal{L}(\mathcal{A}^{\vee C}) = \mathcal{L}(\chi(C))$. Since $\mathcal{A}$ satisfies the mutual exclusion property, it also holds that for every distinct $C, C' \in \Gamma_{Q_i}$, $\mathcal{L}(\chi(C)) \cup \mathcal{L}(\chi(C')) = \mathcal{T}_\Sigma$.

For every $C \in B_{Q_i}$, we define a fresh atomic proposition p_C. We denote by AP_B the union of AP with all these new atomic propositions. Now, we can see the Büchi automaton $\mathcal{A}_{Q_i,q}$ as $\mathcal{A}_B = \langle Q_i \uplus \{\mathrm{q_{exit}}\}, 2^{AP_B}, \delta_B, q, Q_i\rangle$. Basically, we just hide B_{Q_i} in the alphabet. The only thing that is modified from $\mathcal{A}_{Q_i,q}$ to $\mathcal{A}_B$ is the transition function, i.e., $\delta_B(q, \sigma_C) = \delta_{Q_i}(q, (\sigma, C))$ if $\sigma_C = \sigma \cup p_C$ (where $\sigma \subseteq AP$), otherwise $\delta_B(q, \sigma_C) = \varnothing$. Clearly, these little changes do not modify any property of the starting automaton, so $\mathcal{A}_B$ is still counter-free and looping. By Lemma 11, we can assume to have a SafeLTL formula ψ_B, such that $\mathcal{L}(\mathcal{A}_B) = \mathcal{L}(\psi_B)$. Now, we have to find a way to link this formula to the tree language defined by the automaton. First, we note that the following holds by construction.

Fact. Let $\mathcal{A}$ be an HWGT$_{cf}$, Q_i be a universal component of $\mathcal{A}$ and $q \in Q_i$. Then, for each input tree $\mathcal{T} = (T, \tau)$, $\mathcal{T} \in \mathcal{L}(\mathcal{A}^q)$ *iff* for every infinite path π of $\mathcal{T}$ starting at the root, there is a word ξ of the form $\xi = (\tau(\pi(0)), C_0), (\tau(\pi(1)), C_1), (\tau(\pi(2)), C_2), \dots$, such that either $\xi \in \mathcal{L}(\mathcal{A}_{Q_i,q})$ or there is a a position $i = (\tau(\pi(i)), C_i)$ such that $\mathcal{T}_{\pi(i+1)} \in \mathcal{L}(\mathcal{A}^{\vee C_{i+1}})$, where $\mathcal{T}_{\pi(i+1)}$ is the subtree of $\mathcal{T}$ rooted at node $\pi(i+1)$.

Note this about the above claim: it basically states that a node such that its subtree is in $\mathcal{L}(\mathcal{A}^{\vee C_i})$ *releases* the path on which it is from the satisfaction of the constraint imposed by $\mathcal{A}_{Q_i,q}$. Moreover, we saw above that by induction hypothesis we have for every $C \in B_{Q_i}$ a cCTL$^\star_f$ formula $\chi(C)$ such that $\mathcal{L}(\mathcal{A}^{\vee C}) = \mathcal{L}(\chi(C))$. Combining this with the fact above, we get the following claim about the tree language defined by $\mathcal{A}^q$. This characterization can be used to prove the correctness of the translation.

Fact. For each Σ-labelled tree $\mathcal{T}$, $\mathcal{T} \in \mathcal{L}(\mathcal{A}^q)$ *iff* for every infinite path π, there is an infinite word ξ over 2^{AP_B} such that either $\xi \vDash \psi_B$, or there is $i \geq 0$ such that $\xi(i) \cap AP = \pi(i)$, for all $p_C \in \xi(i)$, $\mathcal{T}, \pi, i \vDash \chi(C)$ and there is a unique $C \in B_{Q_i}$, such that $p_C \in \xi(i)$.

Now, replace every p_C and $\neg p_C$ in ψ_Γ by the cCTL$^\star_f$ formulas $\chi(C)$ and $\bigvee_{C' \in \Gamma_{Q_i} \setminus \{C\}} \chi(C')$, respectively. The formula thus constructed is denoted $f(\psi_B)$. We can finally define the formula $\chi(q)$:

$$\chi(q) = \mathtt{A}\Big(f(\psi_B)\ \mathtt{W}\ \bigvee_{C \in B_{Q_i}} \chi(C)\Big)$$

The formula basically states that on every path the satisfaction of the formula defined by the word automaton can only be released by the satisfaction of a $\bigvee C$. □

Proposition 13. *For any $q \in Q_i$, where Q_i is a universal component, $\mathcal{L}(\mathcal{A}^q) = \mathcal{L}(\chi(q))$.*

The translation from a cCTL$^\star_f$ formula to a HWGT$_{cf}$ is basically the same provided in [3, Theorem 5.9].

Theorem 14. *Given a cCTL$^\star_f$ formula φ, one can construct a HWGT$_{cf}$ $\mathcal{A}_\varphi$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$.*

7 Normal Forms of Temporal Logics

In the previous sections, we have proved the equivalence of the two logics considered in Section 3, i.e., cCTL$^p_\pm$ and cCTL$^\star_f$, with two classes of automata. In particular, thanks to Proposition 4 and the class of automaton proven equivalent to cCTL$^\star_f$, it was possible to highlight the *semantic* behaviour of the latter. Namely, whenever an existential path quantification is involved, a cCTL$^\star_f$ formula can only express a co-safety property, while, dually, whenever a universal path quantification is involved, it can only express

a safety property. These observations give rise to the following normal form, that captures *syntactically* the *semantic* content provided by the finite path quantification.

Lemma 15. *For any* $\mathrm{cCTL}^\star_f$ *formula, there is an equivalent formula generated by the grammar*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{D}^n\varphi \mid \mathtt{E}\psi$$
$$\psi ::= \varphi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathtt{X}\psi \mid \psi \mathtt{U} \psi$$

Note that this grammar allows to state that E is only followed by coSafeLTL and, by the use of negation, that A is only followed by SafeLTL. Moreover, the difference between finite and infinite path quantification becomes redundant. Indeed, every *finite* path property can also be seen as an *infinite* path property and vice versa. This implies that the normal form of $\mathrm{cCTL}^\star_f$ is nothing else than a *polarized* version of $\mathrm{cCTL}^\star$, that we will denote by $\mathrm{cCTL}^\star_\pm$, creating a symmetry with Schlingloff's work and also showing that the semantic content provided by finite path quantification is useless when one restricts the syntax as above. To conclude, the following is well known.

Proposition 16. *[16]* SafeLTL *(resp.,* coSafeLTL*) and* LTL^p *formulas of the form* $\mathtt{G}\varphi$ *(resp.,* $\mathtt{F}\varphi$*), where* φ *is a formula using only past temporal operators, are equivalent formalisms.*

This suggests a normal form also for $\mathrm{cCTL}^p_\pm$. Since $\mathrm{cCTL}^p_\pm$ and CTL_f are equivalent formalisms, $\mathrm{cCTL}^p_\pm$ can express co-safety properties existentially and safety properties universally. Combining this with the proposition above, we get the following normal form for $\mathrm{cCTL}^p_\pm$.

Lemma 17. *For any* $\mathrm{cCTL}^p_\pm$ *formula, there is an equivalent formula generated by the grammar*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathtt{D}^n\varphi \mid \mathtt{EF}\psi$$
$$\psi ::= \varphi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathtt{Y}\psi \mid \psi \mathtt{S} \psi$$

8 Conclusions

In this work, we provided two automaton-based characterisations of the temporal logics $\mathrm{cCTL}^p_\pm$ and $\mathrm{cCTL}^\star_f$, both of which are known to be equivalent to FO over infinite trees. These gives us two corresponding characterisations of FO. The automata-theoretic perspective reveals two distinctive features of FO in this setting: (a) when expressing existential properties over paths, it can capture only co-safety properties of the node sequences along those paths, whereas universal path quantification allows it to express only safety properties; (b) every formula can be normalised into a Boolean combination of formulae where only the variable bound to the outermost quantifier is independent, while all others depend on the variable(s) quantified first. These insights were obtained by establishing corresponding normal forms for $\mathrm{cCTL}^p_\pm$ and $\mathrm{cCTL}^\star_f$, each derived from its associated class of automata.

Despite these advancements, several open problems remain. First, while [49] shows that $\mathrm{cCTL}^p_\pm$ is equivalent to FO, and [39] establishes that $\mathrm{cCTL}^\star$ corresponds to MPL, no analogous result is known for cCTL^p. As shown in Proposition 2, cCTL^p is strictly more expressive than $\mathrm{cCTL}^p_\pm$, while it is well known that it is strictly less expressive than $\mathrm{cCTL}^\star$. Determining the exact expressive power of cCTL^p remains an important open question. Second, although the two classes of automata we introduced are equivalent, current translations between them passes through FO, which leads to a non-elementary blowup. An interesting direction for future work would be to develop direct translations between the two automata classes, bypassing the intermediate FO encoding. Third, we observe a strong similarity between the normal form of $\mathrm{cCTL}^p_\pm$ and the logic studied in [5]. While $\mathrm{cCTL}^p_\pm$ supports both counting and the S operator, the logic in [5] includes only the past-time version of F. Importantly, the definability problem for that logic is decidable over finite trees. Investigating whether the same holds for $\mathrm{cCTL}^p_\pm$ in the finite-tree setting would be a valuable contribution. Finally, a more immediate objective is to prove the conjectured equivalence between HLGTs and Antisymmetric Additive-Weak Parity Automata.

References

[1] M. Benedikt & L. Segoufin (2009): *Regular Tree Languages Definable in FO and in FO mod*. *TOCL* 11(1), pp. 1–32, doi:10.1145/1614431.1614435.

[2] M. Benerecetti, L. Bozzelli, F. Mogavero & A. Peron (2023): *Quantifying over Trees in Monadic Second-Order Logic*. In: *LICS'23*, IEEE Computer Society, pp. 1–13, doi:10.1109/LICS56636.2023.10175832.

[3] M. Benerecetti, L. Bozzelli, F. Mogavero & A. Peron (2024): *Automata-Theoretic Characterisations of Branching-Time Temporal Logics*. In: *ICALP'24*, LIPIcs 297, Leibniz-Zentrum fuer Informatik, pp. 128:1–20, doi:10.4230/LIPIcs.ICALP.2024.128.

[4] M. Bojańczyk (2004): *Decidable Properties of Tree Languages*. Ph.D. thesis, Warsaw University, Warsaw, Poland.

[5] M. Bojańczyk (2009): *Two-Way Unary Temporal Logic over Trees*. *LMCS* 5(3), pp. 1–29, doi:10.2168/LMCS-5(3:5)2009.

[6] M. Bojańczyk (2021): *Algebra for Trees*. In: *Handbook of Automata Theory, Volume I - Theoretical Foundations*, European Mathematical Society Publishing House, Zürich, Switzerland, p. 801–838.

[7] M. Bojańczyk & T. Idziaszek (2009): *Algebra for Infinite Forests with an Application to the Temporal logic EF*. In: *CONCUR'09*, LNCS 5710, Springer, pp. 131–145, doi:10.1007/978-3-642-04081-8_10.

[8] M. Bojańczyk, T. Idziaszek & M. Skrzypczak (2013): *Regular Languages of Thin Trees*. In: *STACS'13*, LIPIcs 20, Leibniz-Zentrum fuer Informatik, pp. 562–573, doi:10.1007/s00224-014-9595-z.

[9] M. Bojańczyk, H. Straubing & I. Walukiewicz (2012): *Wreath Products of Forest Algebras, with Applications to Tree Logics*. *LMCS* 8(3), doi:10.2168/LMCS-8(3:19)2012.

[10] U. Boker, K. Lehtinen & S. Sickert (2022): *On the Translation of Automata to Linear Temporal Logic*. In: *FOSSACS'22*, LNCS 13242, Springer, pp. 140–160, doi:10.1007/978-3-030-99253-8_8.

[11] U. Boker & Y. Shaulian (2018): *Automaton-Based Criteria for Membership in CTL*. In: *LICS'18*, Association for Computing Machinery, pp. 155–164, doi:10.1145/3209108.3209143.

[12] J.R. Büchi (1962): *On a Decision Method in Restricted Second-Order Arithmetic*. In: *ICLMPS'62*, Stanford University Press, pp. 1–11.

[13] J.R. Büchi (1966): *On a Decision Method in Restricted Second Order Arithmetic*. In: *Studies in Logic and the Foundations of Mathematics*, 44, Elsevier, pp. 1–11.

[14] F. Carreiro (2015): *PDL is the Bisimulation-Invariant Fragment of Weak Chain Logic*. In: *LICS'15*, IEEE Computer Society, pp. 341–352, doi:10.1109/LICS.2015.40.

[15] F. Carreiro & Y. Venema (2014): *PDL inside the μ-Calculus: A Syntactic and an Automata-Theoretic Characterization*. In: *AIML'14*, College Publications, pp. 74–93.

[16] E. Chang, Z. Manna & A. Pnueli (1992): *Characterization of Temporal Property Classes*. In: *ICALP'92*, LNCS 623, Springer, pp. 474–486.

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

[18] V. Diekert & P. Gastin (2008): *First-Order Definable Languages*. In: *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, *Texts in Logic and Games* 2, Amsterdam University Press, pp. 261–306.

[19] C.C. Elgot (1961): *Decision Problems of Finite Automata Design and Related Arithmetics*. *TAMS* 98, pp. 21–51.

[20] E.A. Emerson & J.Y. Halpern (1983): *"Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time*. In: *POPL'83*, Association for Computing Machinery, pp. 127–140, doi:10.1145/567067.567081.

[21] E.A. Emerson & J.Y. Halpern (1985): *Decision Procedures and Expressiveness in the Temporal Logic of Branching Time*. *JCSS* 30(1), pp. 1–24, doi:10.1016/0022-0000(85)90001-7.

[22] E.A. Emerson & C.S. Jutla (1991): *Tree Automata, muCalculus, and Determinacy*. In: *FOCS'91*, IEEE Computer Society, pp. 368–377, doi:10.1109/SFCS.1991.185392.

[23] Z. Esik & P. Weil (2010): *Algebraic Characterization of Logically Defined Tree Languages*. *IJCM* 20(02), pp. 195–239, doi:10.1142/S0218196710005595.

[24] M.J. Fischer & R.E. Ladner (1979): *Propositional Dynamic Logic of Regular Programs*. *JCSS* 18(2), pp. 194–211, doi:10.1016/0022-0000(79)90046-1.

[25] J. Flum (1985): *Characterizing Logics*. In: *Model-Theoretic Logics*, Springer, pp. 77–120.

[26] C. Ford (2019): *Investigations into the Expressiveness of First-order Logic and Weak Path Automata on Infinite Tree*. Master's thesis, University of Amsterdam, Amsterdam, Netherlands.

[27] D.M. Gabbay, A. Pnueli, S. Shelah & J. Stavi (1980): *On the Temporal Analysis of Fairness*. In: *POPL'80*, Association for Computing Machinery, pp. 163–173.

[28] T. Hafer & W. Thomas (1987): *Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree*. In: *ICALP'87*, LNCS 267, Springer, pp. 269–279.

[29] D. Janin & I. Walukiewicz (1996): *On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic*. In: *CONCUR'96*, LNCS 1119, Springer, pp. 263–277.

[30] D. Kozen (1983): *Results on the Propositional muCalculus*. *TCS* 27(3), pp. 333–354, doi:10.1016/0304-3975(82)90125-6.

[31] O. Kupferman, U. Sattler & M.Y. Vardi (2002): *The Complexity of the Graded muCalculus*. In: *CADE'02*, LNCS 2392, Springer, pp. 423–437.

[32] O. Kupferman, M.Y. Vardi & P. Wolper (2000): *An Automata Theoretic Approach to Branching-Time Model Checking*. *JACM* 47(2), pp. 312–360, doi:10.1145/333979.333987.

[33] F. Laroussinie & P. Schnoebelen (1995): *A Hierarchy of Temporal Logics with Past*. *TCS* 148(2), pp. 303–324, doi:10.1016/0304-3975(95)00035-U.

[34] P. Lindström (1969): *On Extensions of Elementary Logic*. *Theoria* 35(1), pp. 1–11.

[35] C. Löding & W. Thomas (2000): *Alternating Automata and Logics over Infinite Words*. In: *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*, Springer, pp. 521–535.

[36] Z. Manna & A. Pnueli (1992): *The Temporal Logic of Reactive and Concurrent Systems - Specification*. Springer.

[37] Z. Manna & A. Pnueli (1995): *Temporal Verification of Reactive Systems - Safety*. Springer.

[38] S. Miyano & T. Hayashi (1984): *Alternating Finite Automata on ω-Words*. *TCS* 32(3), pp. 321–330.

[39] F. Moller & A.M. Rabinovich (2003): *Counting on CTL*: On the Expressive Power of Monadic Path Logic*. *IC* 184(1), pp. 147–159, doi:10.1016/S0890-5401(03)00104-4.

[40] A.W. Mostowski (1984): *Regular Expressions for Infinite Trees and a Standard Form of Automata*. In: *SCT'84*, LNCS 208, Springer, pp. 157–168.

[41] D.E. Muller, A. Saoudi & P.E. Schupp (1984): *Alternating Automata, the Weak Monadic Theory of Trees and its Complexity*. In: *ICALP'86*, LNCS 226, Springer, pp. 275–283.

[42] A. Nerode (1958): *Linear Automaton Transformations*. *PAMS* 9(4), pp. 541–544.

[43] D. Perrin & J. Pin (1986): *First-Order Logic and Star-Free Sets*. *JCSS* 32(3), pp. 393–406.

[44] A. Pnueli (1977): *The Temporal Logic of Programs*. In: *FOCS'77*, IEEE Computer Society, pp. 46–57, doi:10.1109/SFCS.1977.32.

[45] A. Pnueli (1981): *The Temporal Semantics of Concurrent Programs*. *TCS* 13, pp. 45–60, doi:10.1016/0304-3975(81)90110-9.

[46] A. Potthoff (1995): *First-Order Logic on Finite Trees*. In: *TAPSOFT'95*, Springer, pp. 123–139, doi:10.1080/11663081.1992.10510780.

[47] M.O. Rabin (1969): *Decidability of Second-Order Theories and Automata on Infinite Trees*. *TAMS* 141, pp. 1–35.

[48] M.O. Rabin & D.S. Scott (1959): *Finite Automata and their Decision Problems*. *IBMJRD* 3, pp. 115–125.

[49] B.-H. Schlingloff (1992): *Expressive Completeness of Temporal Logic of Trees*. *JANCL* 2(2), pp. 157–180.

[50] M.P. Schützenberger (1965): *On Finite Monoids Having Only Trivial Subgroups*. *IC* 8(2), pp. 190–194.

[51] W. Thomas (1984): *Logical Aspects in the Study of Tree Languages*. In: *CAAP'84*, Cambridge University Press, pp. 31–50.

[52] W. Thomas (1987): *On Chain Logic, Path Logic, and First-Order Logic over Infinite Trees*. In: *LICS'87*, IEEE Computer Society, pp. 245–256.

[53] M.Y. Vardi & P. Wolper (1984): *Yet Another Process Logic*. In: *LP'83*, LNCS 164, Springer, pp. 501–512, doi:10.1007/3-540-12896-4_383.

[54] P. Wolper (1983): *Temporal Logic Can Be More Expressive*. *IC* 56(1-2), pp. 72–99, doi:10.1016/S0019-9958(83)80051-5.