An Automaton-based Characterisation of First Order Logic over Infinite Trees (short paper)

Massimo Benerecetti[1,*,†], Dario Della Monica[2,*,†], Angelo Matteo[2,*,†], Fabio Mogavero[1,*,†] and Gabriele Puppis[2,*,†]

[1] Università degli Studi di Napoli "Federico II", Italy
[2] Università degli Studi di Udine, Italy

Abstract

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.

Keywords

First Order Logic, Infinite trees, Tree automata, Branching time temporal logics

1. Introduction

Characterisation theorems [1] 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* [2] (*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 [3, 4]. 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 [5, 6] and ELTL [7], with *star-free* [8, 9] and *ω-regular* [10] languages, and with automata-theoretic models, including *finite* [11, 12] and *Büchi* [13, 10, 14] automata. These connections provide deep insights into the structure of definable properties and lead to optimal decision procedures across different representations.

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 [15] 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

ICTCS '25: Italian Conference on Theoretical Computer Science, September 10-12, 2025, Pescara, Italy
*Corresponding author.
†These authors contributed equally.
massimo.benerecetti@unina.it (M. Benerecetti); dario.dellamonica@uniud.it (D. Della Monica); matteo.angelo@spes.uniud.it (A. Matteo); fabio.mogavero@unina.it (F. Mogavero); gabriele.puppis@uniud.it (G. Puppis)

only *child*, only *ancestor*, or both relations). Aside from the positive result for FO over finite trees with the child relation [16], the problem remains open in all other settings. Efforts to resolve this question have mainly followed *algebraic approaches* [17], inspired by their success in the word case (most notably Schützenberger theorem on star-free languages [8]). 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 [18, 19, 20, 21] or topologically simple infinite trees [22, 23]. 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 [24] between *Monadic Second-Order Logic* (MSO) [25], *Parity Tree Automata* [26, 27], and the *Modal* μ-Calculus [28]. More recently [29], the landscape has expanded to include the expressive equivalence of *Monadic Chain/Path Logics* (MCL/MPL) [15, 30, 31], their temporal *Computation Tree Logic* counterparts (ECTL*/CTL*) [32, 33, 34], and variants of *Hesitant Tree Automata* (HTA) [35].

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 $\mathrm{cCTL}^{p}_{\pm}$, and the *Full Computation Tree Logic with counting and finite path quantification*, denoted $\mathrm{cCTL}^{\star}_{f}$. In [36, 29] 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 [37] that relate fragments of the modal μ-Calculus, variants of *Propositional Dynamic Logic* (PDL) [38], and *Weak Monadic Chain Logic* (WMCL).

Other related work. In earlier work, Bojańczyk [19] showed that, over finite binary trees, FO with child and ancestor relations is equivalent to a *cascade product* of so-called *aperiodically wordsum automata*. While related in spirit, this result targets a different logic and a different class of structures. More recently, Ford [39] 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.

2. Temporal logics equivalent to FO and tree automata

Temporal logics. We consider FO over unranked and unordered finitely branching infinite trees. In the following we will call them just infinite trees. The syntax and semantics for FO we consider is entirely standard (we refer the reader to [31] for details). Here, we present two branching time temporal logics provably equivalent to FO over infinite trees: they are, respectively, a fragment of CTL *with past and counting*, that we call *polarized* and denote by $\mathrm{cCTL}^{p}_{\pm}$, and $\mathrm{CTL}^\star$ *with counting and finite path quantification*, denoted by $\mathrm{cCTL}^{\star}_{f}$.

The former logic $\mathrm{cCTL}^{p}_{\pm}$ has been introduced in [36] and shown equivalent to FO over infinite trees. It is a restriction of cCTL^{p}, i.e., CTL with past modalities (CTL^{p}) [40], further enhanced with counting. The following grammar (where p ranges over a set of atomic propositions AP) defines the syntax of $\mathrm{cCTL}^{p}_{\pm}$:

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

The semantics is standard, except for the counting operator [41] $\mathtt{D}^n\varphi$, which is satisfied if the node of the tree at which it is evaluated has at least n successors satisfying φ. Compared to cCTL^p, the fragment $\mathrm{cCTL}^p_\pm$ disallows formulas of the kind $\mathtt{E}(\varphi\,\mathtt{R}\,\psi)$, that cannot be restored through the use of negation. In particular, $\mathtt{U}$ must be paired with $\mathtt{E}$, while $\mathtt{R}$ (the dual of $\mathtt{U}$) must be paired with $\mathtt{A}$ (the dual of $\mathtt{E}$). This syntactic restriction is reflected in the semantics: existential quantification can predicate only about *co-safety* properties, while, dually, universal quantification one can only express *safety* properties.

The latter logic $\mathrm{cCTL}^\star_f$ can be shown equivalent to FO by adapting the model-theoretic argument of [41]: this was noticed for the first time in [29]. The syntax is the same as the one of $\mathrm{cCTL}^\star$ [41]:

$$\varphi ::= \mathtt{D}^n\varphi \;\big|\; p \;\big|\; \neg\varphi \;\big|\; \varphi\vee\varphi \;\big|\; \mathtt{E}\varphi \;\big|\; \mathtt{X}\,\varphi \;\big|\; \varphi\,\mathtt{U}\,\varphi$$

The difference is in the semantics: $\mathrm{cCTL}^\star$ features infinite path quantification, while $\mathrm{cCTL}^\star_f$ predicates on finite nonempty paths. Unlike $\mathrm{cCTL}^p_\pm$, it is possible in $\mathrm{cCTL}^\star_f$ to pair $\mathtt{R}$ with $\mathtt{E}$, as in $\mathtt{E}(\varphi\,\mathtt{R}\,\psi)$. However, this syntactic ability does not reflect in a real (semantic) gain. As a matter of fact, the semantic constraint of $\mathrm{cCTL}^\star_f$ causes the validity of formulas such as $\varphi\,\mathtt{R}\,\psi \leftrightarrow \psi\,\mathtt{U}\,((\tilde{X}\bot\vee\varphi)\wedge\psi)$, making it possible to equivalently rewrite the above formula $\mathtt{E}(\varphi\,\mathtt{R}\,\psi)$ as $\mathtt{E}(\psi\,\mathtt{U}\,((\tilde{X}\bot\vee\varphi)\wedge\psi))$. Therefore, the aforementioned polarized behavior of $\mathrm{cCTL}^p_\pm$ is observed in $\mathrm{cCTL}^\star_f$ as well: somehow, the syntactic restriction in the former logic and the semantic constraint in the latter one are equivalent.

In this work, we provide automaton-based characterisations for both logics, allowing the identification of normal forms.

Tree automata. A *Graded Alternating Büchi Tree Automaton* (*GTA*) is a tuple $\mathcal{A} = \langle Q,\Sigma,\delta,q_I,F\rangle$, where Q is a set of states, Σ is a finite alphabet, $\delta : Q\times\Sigma \to \mathcal{B}^+(\{\Diamond_k,\Box_k \mid k>0\}\times Q)$ is the transition function, q_I is the starting state and $F\subseteq Q$ is the set of accepting states. Given a set X, $\mathcal{B}^+(X)$ denotes as usual the set of positive boolean formulas over X. We skip the details on how a run of a GTA is structured (the reader can find them in, e.g., [42, 43]), limiting ourselves to stating that it is itself a tree and is accepting if every branch visits a state in F infinitely often.

A *Weak* GTA (*WGT*) is a GTA $\mathcal{A} = \langle Q,\Sigma,\delta,q_I,F\rangle$ such that there is a partition of Q into non-empty disjoint sets $\{Q_1,\ldots,Q_n\}$, called *components*, 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. Moreover, every component Q_i is either entirely composed of states in F or entirely of states outside F. This is the notion of weak automaton as introduced by [44].

An *Hesitant* WGT (*HWGT*) is a WGT $\mathcal{A} = \langle Q,\Sigma,\delta,q_I,F\rangle$ such that every component Q_i 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 a pair of the form $(\Diamond_1,q')$, and only disjunctively related to other pairs 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 a pair of the form $(\Box_1,q')$, and only conjunctively related to other pairs 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 pair in $\delta(q,\sigma)$.

Given this structural restriction on components, every path of a run of a HWGT gets eventually stuck in an existential or in a universal component: imposing that every state in a universal component is accepting and every state in an existential one is not, one gets that HWGT are also weak. The hesitant constraint was first introduced in [35].

In what follows, we present our result: we identify two restrictions of HWGTs that are respectively equivalent to $\mathrm{cCTL}^p_\pm$ and $\mathrm{cCTL}^\star_f$.

3. Restrictions of HWGT and equivalence results

Two-way linear HWGT. The first restriction of HWGTs we introduce is designed to obtain the equivalence with $\mathrm{cCTL}^p_\pm$. First, we actually enhance HWGTs, giving them the chance of going up the input tree as $\mathrm{cCTL}^p_\pm$ does using past operators. This is achieved through the following.

A *Two-Way* HWGT (*2HWGT*) is a HWGT $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, Q_\forall \rangle$ such that $\delta : Q \times \Sigma \to \mathcal{B}^+(\{\Diamond_k, \Box_k, -1\} \times Q)$. Basically, the automaton can send also states in the parent of the current node, traversing the input tree *upwards*. This suggests an extension of the hesitant types. Given a 2HWGT, a component Q_i can be existential, universal, transient but also *upward*:

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

This is not enough for our purposes. Currently, components are not restricted enough to get FO's expressive power. So, we introduce the following.

A *Linear* HWGT (*HLGT*) is a HWGT $\mathcal{A} = \langle Q, \Sigma, \delta, q_I, Q_\forall \rangle$ in which every component is a singleton.

Combining the two restrictions, one gets *Two-Way Linear* HWGT (*2HLGT*) and this class of automata is provably equivalent to $\mathrm{cCTL}^p_\pm$ formula.

Theorem 1. *2HLGT and* $\mathrm{cCTL}^p_\pm$ *are expressively equivalent formalisms.*

This characterisation is effective, meaning that given a $\mathrm{cCTL}^p_\pm$ formula one can obtain an equivalent 2HLGT, and vice versa given a 2HLGT one can translate it in an equivalent $\mathrm{cCTL}^p_\pm$.

Counter-free HWGT. The automaton-based characterisation of $\mathrm{cCTL}^\star_f$ turns out to be more involved. The Two-Way head movement here is not needed, since the logic only employs future temporal operators. Moreover, the singleton restriction would be too strong, making it impossible to translate formulas of the form $\mathtt{E}((\varphi\,\mathtt{U}\,\psi)\,\mathtt{U}\,\gamma)$. Thus, we have to find a suitable and reasonable restriction of the non-transient components, wihout making them too weak. This is achieved by looking at components as word automata, in a similar fashion to what has been done in [42, Definition 5.2]. The crucial point of this construction is the acceptance condition: since we are dealing with a *weak* automaton, universal components are entirely accepting, while existential components are entirely rejecting. To retain this property when we see components as word automata, we let every universal (resp., existential) component be a *Universal* (resp. *Non-deterministic) Büchi* (resp., *Co-Büchi) word automaton* with every state in F, but a sink state. For detailed definitions of Universal, Nondeterministic, Büchi and Co-Büchi word automata we refer the reader to [45]. In this way we clearly have limited the expressive power to *co-safety* and *safety* properties: a word automaton for an existential component Q_i accepts only exiting Q_i, while dually a word automaton for a universal component Q_i accepts only staying inside Q_i. However, not every safety and co-safety property can be expressed by FO. To get a further restriction, we recall that Linear-time temporal logic, usually denoted by LTL, is equivalent to FO over words. LTL is generated according to the following grammar: $\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{X}\,\varphi \mid \varphi\,\mathtt{U}\,\varphi$. It is well known [46] that if one only allows LTL formulas employing atomic propositions, boolean operations (except negation), and temporal operators $\mathtt{X}$ and $\mathtt{U}$, one gets coSafeLTL, while replacing $\mathtt{U}$ with $\mathtt{R}$, one gets SafeLTL: these are the co-safety and safety fragments of LTL, respectively, and consequently also of FO. With a restriction on the structure of word automata (namely *counter-freeness*, as defined in [47, Definition 11.1]), we get the following.

Lemma 2. SafeLTL *(resp.,* coSafeLTL*) and counter-free universal Büchi (resp. non-deterministic Co-Büchi) automata are expressively equivalent formalisms.*

The counter-freeness of a word automaton basically implies that the automaton cannot count modulo $n \geq 2$ while reading an input. Thus, to limit the expressiveness of the components of HWGT, we require them to be counter-free. We also impose another restriction on the word automata called *mutual exclusion*, but it is of rather technical nature and so we do not recall it here. It can be found in [42, Definition 5.5]. We are finally done. An HWGT in which every non-transient component, seen as a word automaton, is counter-free and satisfies mutual exclusion is denoted $HWGT_{cf}$.

Theorem 3. *$HWGT_{cf}$ and $\mathrm{cCTL}^{\star}_{f}$ are expressively equivalent formalisms.*

Again, the translation in the two directions is effective.

4. Normal forms of temporal logics

In the previous section, we have sketched the definition of two classes of automata provably equivalent to $\mathrm{cCTL}^{p}_{\pm}$ and $\mathrm{cCTL}^{\star}_{f}$. Now, thanks to the class of automaton proven equivalent to $\mathrm{cCTL}^{\star}_{f}$, it is possible to highlight the *semantic* behaviour of the latter. Namely, whenever an existential path quantification is involved, a $\mathrm{cCTL}^{\star}_{f}$ 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 4. *For any $\mathrm{cCTL}^{\star}_{f}$ formula, there is an equivalent formula generated by the grammar*

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

Note that this grammar allows to state that $\mathtt{E}$ is only followed by coSafeLTL and, by the use of negation, that $\mathtt{A}$ 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 that a *polarized* version of $\mathrm{cCTL}^{\star}$, that we will denote by $\mathrm{cCTL}^{\star}_{\pm}$, creating a symmetry with [36] and also showing that the semantic content provided by finite path quantification is useless when one restricts the syntax as above. To conclude, let LTL^{p} be LTL enhanced with past operators $\mathtt{Y}$ and $\mathtt{S}$. Then, the following is well known.

Lemma 5. *[48] 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 6. *For any $\mathrm{cCTL}^{p}_{\pm}$ formula, there is an equivalent formula generated by the grammar*

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

(as usual, using negation one can construct shorthands of the form $\mathtt{AG}\,\psi$).

5. 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 characterisations give us two corresponding characterisations of FO and also allowed us to unveil a peculiar behaviour of FO over infinite trees, namely the fact that 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. As a by-product, we also obtained two normal forms for the two temporal logics considered. Despite the advancements, several problems remain open and we plan to further investigate FO over trees, with a focus on the definability problem.

References

[1] J. Flum, Characterizing Logics, in: Model-Theoretic Logics, Springer, 1985, pp. 77–120.

[2] P. Lindström, On Extensions of Elementary Logic, Theoria 35 (1969) 1–11.

[3] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems - Specification, Springer, 1992.

[4] Z. Manna, A. Pnueli, Temporal Verification of Reactive Systems - Safety, Springer, 1995.

[5] A. Pnueli, The Temporal Logic of Programs, in: FOCS'77, IEEE Computer Society, 1977, pp. 46–57.

[6] A. Pnueli, The Temporal Semantics of Concurrent Programs, TCS 13 (1981) 45–60.

[7] P. Wolper, Temporal Logic Can Be More Expressive, IC 56 (1983) 72–99.

[8] M. Schützenberger, On Finite Monoids Having Only Trivial Subgroups, IC 8 (1965) 190–194.

[9] D. Perrin, J. Pin, First-Order Logic and Star-Free Sets, JCSS 32 (1986) 393–406.

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

[11] A. Nerode, Linear Automaton Transformations, PAMS 9 (1958) 541–544.

[12] M. Rabin, D. Scott, Finite Automata and their Decision Problems, IBMJRD 3 (1959) 115–125.

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

[14] J. Büchi, On a Decision Method in Restricted Second Order Arithmetic, in: Studies in Logic and the Foundations of Mathematics, volume 44, Elsevier, 1966, pp. 1–11.

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

[16] M. Benedikt, L. Segoufin, Regular Tree Languages Definable in FO and in FO mod, TOCL 11 (2009) 1–32.

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

[18] A. Potthoff, First-Order Logic on Finite Trees, in: TAPSOFT'95, Springer, 1995, pp. 123–139.

[19] M. Bojańczyk, Decidable Properties of Tree Languages, Ph.D. thesis, Warsaw University, Warsaw, Poland, 2004.

[20] Z. Esik, P. Weil, Algebraic Characterization of Logically Defined Tree Languages, IJCM 20 (2010) 195–239.

[21] M. Bojańczyk, H. Straubing, I. Walukiewicz, Wreath Products of Forest Algebras, with Applications to Tree Logics, LMCS 8 (2012).

[22] M. Bojańczyk, T. Idziaszek, Algebra for Infinite Forests with an Application to the Temporal logic EF, in: CONCUR'09, LNCS 5710, Springer, 2009, pp. 131–145.

[23] M. Bojańczyk, T. Idziaszek, M. Skrzypczak, Regular Languages of Thin Trees, in: STACS'13, LIPIcs 20, Leibniz-Zentrum fuer Informatik, 2013, pp. 562–573.

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

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

[26] A. Mostowski, Regular Expressions for Infinite Trees and a Standard Form of Automata, in: SCT'84, LNCS 208, Springer, 1984, pp. 157–168.

[27] E. Emerson, C. Jutla, Tree Automata, muCalculus, and Determinacy, in: FOCS'91, IEEE Computer Society, 1991, pp. 368–377.

[28] D. Kozen, Results on the Propositional muCalculus, TCS 27 (1983) 333–354.

[29] M. Benerecetti, L. Bozzelli, F. Mogavero, A. Peron, Quantifying over Trees in Monadic Second-Order Logic, in: LICS'23, IEEE Computer Society, 2023, pp. 1–13.

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

[31] T. Hafer, W. Thomas, Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory

of the Binary Tree, in: ICALP'87, LNCS 267, Springer, 1987, pp. 269–279.
[32] M. Vardi, P. Wolper, Yet Another Process Logic, in: LP'83, LNCS 164, Springer, 1984, pp. 501–512.
[33] E. Emerson, J. Halpern, "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time, in: POPL'83, Association for Computing Machinery, 1983, pp. 127–140.
[34] E. Emerson, J. Halpern, Decision Procedures and Expressiveness in the Temporal Logic of Branching Time, JCSS 30 (1985) 1–24.
[35] O. Kupferman, M. Vardi, P. Wolper, An Automata Theoretic Approach to Branching-Time Model Checking, JACM 47 (2000) 312–360.
[36] B.-H. Schlingloff, Expressive Completeness of Temporal Logic of Trees, JANCL 2 (1992) 157–180.
[37] F. Carreiro, Y. Venema, Pdl inside the μ-calculus: A syntactic and an automata-theoretic characterization, in: AIML'14, College Publications, 2014, pp. 74–93.
[38] M. Fischer, R. Ladner, Propositional Dynamic Logic of Regular Programs, JCSS 18 (1979) 194–211.
[39] C. Ford, Investigations into the Expressiveness of First-order Logic and Weak Path Automata on Infinite Tree, Master's thesis, University of Amsterdam, Amsterdam, Netherlands, 2019.
[40] F. Laroussinie, P. Schnoebelen, A Hierarchy of Temporal Logics with Past, TCS 148 (1995) 303–324.
[41] F. Moller, A. Rabinovich, Counting on CTL*: On the Expressive Power of Monadic Path Logic, IC 184 (2003) 147–159.
[42] M. Benerecetti, L. Bozzelli, F. Mogavero, A. Peron, Automata-Theoretic Characterisations of Branching-Time Temporal Logics, in: ICALP'24, LIPIcs 297, Leibniz-Zentrum fuer Informatik, 2024, pp. 128:1–20.
[43] U. Boker, Y. Shaulian, Automaton-Based Criteria for Membership in CTL, in: LICS'18, Association for Computing Machinery, 2018, pp. 155–164.
[44] D. Muller, A. Saoudi, P. Schupp, Alternating Automata, the Weak Monadic Theory of Trees and its Complexity, in: ICALP'86, LNCS 226, Springer, 1984, pp. 275–283.
[45] T. Wilke, ω-automata, in: Handbook of Automata Theory, Volume I - Theoretical Foundations, European Mathematical Society Publishing House, Zürich, Switzerland, 2021, pp. 189–234.
[46] A. Sistla, Safety, Liveness and Fairness in Temporal Logic, FAC 6 (1994) 495–511.
[47] V. Diekert, P. Gastin, First-Order Definable Languages, in: Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas], volume 2 of *Texts in Logic and Games*, Amsterdam University Press, 2008, pp. 261–306.
[48] E. Chang, Z. Manna, A. Pnueli, Characterization of Temporal Property Classes, in: ICALP'92, LNCS 623, Springer, 1992, pp. 474–486.