Automata-Theoretic Characterisations of Branching-Time Temporal Logics

Massimo Benerecetti
Università di Napoli Federico II, Italy

Laura Bozzelli
Università di Napoli Federico II, Italy

Fabio Mogavero
Università di Napoli Federico II, Italy

Adriano Peron
Università di Trieste, Italy

Abstract

Characterisations theorems serve as important tools in model theory and can be used to assess and compare the expressive power of temporal languages used for the specification and verification of properties in formal methods. While complete connections have been established for the linear-time case between temporal logics, predicate logics, algebraic models, and automata, the situation in the branching-time case remains considerably more fragmented. In this work, we provide an *automata-theoretic characterisation* of some important branching-time temporal logics, namely CTL* and ECTL* interpreted on arbitrary-branching trees, by identifying two variants of *Hesitant Tree Automata* that are proved equivalent to those logics. The characterisations also apply to *Monadic Path Logic* and the bisimulation-invariant fragment of *Monadic Chain Logic*, again interpreted over trees. These results widen the characterisation landscape of the branching-time case and solve a forty-year-old open question.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Modal and temporal logics; Theory of computation → Tree languages

Keywords and phrases Branching-Time Temporal Logics, Monadic Second-Order Logics, Tree Automata

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.128

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: https://arxiv.org/abs/2404.17421

Funding M. Benerecetti, F. Mogavero, and A. Peron are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). This work has been partially supported by the GNCS 2024 project "Certificazione, monitoraggio, ed interpretabilità in sistemi di intelligenza artificiale".

1 Introduction

Temporal logics [49] play a pivotal role in the *formal verification* of complex systems [50]. Serving as *specification languages*, they provide a framework to express and reason about time-dependent properties, capturing the intricate behaviours and interactions of system components over time. Commonly, these languages are classified into two categories: *linear-time logics*, which emphasise properties spanning the entirety of a computation, and *branching-time logics*, specifically tailored to address the non-deterministic and concurrent nature of behaviours. Well-established representatives of the former include *Linear-Time Temporal Logic* (LTL) [61, 62], its full ω-regular extension ELTL [84], and the finite-horizon variant

LTL_f [32]. Important members of the second category, instead, belong to the families of *Dynamic Logics* [30, 35] and *Computation Tree Logics*, including CTL [21, 16, 24, 17, 22], CTL* [23, 25], ECTL* [80], $CTL*_f$ [79], and $ECTL*_f$ [74]. Additionally, more expressive but lower-level languages, like μ-Calculus [42], have been considered, which suitably extend classic modal logic with monadic fix-point operators, contributing to the rich tapestry of specification languages in the field of formal verification and synthesis.

The semantics of these temporal logics are typically formalised, at the meta-level, through various flavour of *predicate logic*, frequently *First-Order Logic* (FO) or *Second-Order Logic* (SO), interpreted over either *linearly-ordered structures*, such as finite and infinite words [60], or *branching structures*, like Kripke structures [43], labelled transition systems [40], and their tree unwindings. In tandem with this, the rich body of literature on automata-theoretic techniques [75] for words and trees, originated from [41, 56, 57, 66], has proven invaluable to provide effective technical tools for the solution of related *model-checking* [18, 2, 81, 47, 27], *satisfiability* [81, 26, 78, 6, 47], and *synthesis* [15, 63, 69] decision problems. Predicate logics and automata theory offer, in addition, a rich and coherent arsenal of tools to evaluate and compare the expressive power, as well as the computational properties, of temporal languages, as witnessed by numerous *characterisation theorems*. These results provide a dual perspective on the topic, which enhance our ability to navigate the intricate landscape of language fragments and allow us to assess their pros (*e.g.*, elementary complexity of decision problems) and cons (*e.g.*, limitations on the expressive power).

The initial seminal result in this context is Kamp's theorem [39, 31, 67, 68], which establishes the equivalence of LTL and FO over infinite words. The result also extends to LTL_f and FO on finite words [19]. A direct link has been drawn between FO-*definability* and *recognition* by *counter-free finite-state automata*, in both the finite [52] and infinite [48, 71, 72, 59] cases, by means of the notions of *star-free language*, *aperiodic language*, and *aperiodic syntactic monoid* (see [70], for finite words, and [58], for the infinite ones). Together these results provide a complete characterisation of the expressive power of LTL and LTL_f in terms of predicate logics and automata. A parallel correspondence exists between ELTL and $ELTL_f$, the *Monadic Second-Order Logic* (MSO) and its *weak (finite-quantification) fragment* (WMSO), and regular automata on infinite and finite words. Notably, the equivalence between WMSO and regular automata [8, 20, 76], followed by the equivalence between MSO and ω-regular automata [9, 10, 51, 14], stands among the first results connecting the two fields of model theory and automata theory.

The landscape for branching-time temporal logics is considerably more intricate, due to the complex topology of the models and additional factors, such as *bisimulation invariance* [77] and *counting quantifiers* [29], and it is not as clear and complete as the linear-time counterpart. A significant milestone in this setting is the full correspondence between μ-Calculus, the *bisimulation-invariant* fragment of MSO interpreted over trees, and *(Symmetric) Alternating Parity Tree Automata* [38]. This result generalises the already known connection between the latter two formalisms [64]. Another noteworthy connection has been shown to exist between the *alternation-free* fragment of μ-Calculus (AFμ-Calculus), the bisimulation-invariant fragments of WMSO over bounded-branching trees, and *(Symmetric) Alternating Weak Tree Automata* [1, 37] (see [28, 11, 12, 13], for the unbounded-branching case), which extends previous partial results [45, 65]. The above equivalences lift also to the general case, by incorporating counting quantifiers into the temporal logics [37, 36]. The scenario in other cases appears significantly more fragmented. In recent developments, the equivalence between CTL and *(Symmetric) Hesitant Linear Tree Automata* [7] was proved. Nonetheless, as of today, no corresponding fragment of MSO has been identified. By contrast, several variants of CTL* have been linked to the *path* and *chain* fragments of MSO since the eighties, although no automata characterisation has been provided thus far. For instance, it was shown in [34] that, on binary trees, CTL* is equivalent to *Monadic Path Logic* (MPL) [33]. Similar correspondences have been established in [74] for CTL*_{f}, ECTL*, and ECTL*_{f}, which equate, respectively, to FO, *Monadic Chain Logic* (MCL), and its weak fragment (WMCL). The result concerning CTL* was later extended to arbitrary-branching trees, addressing both bisimulation-invariance [54] and counting quantifiers [55]. As far as we know, no similar results are available for the other three logics. Finally, the recently introduced *Monadic Tree Logic* (MTL) [3] together with its variants have yet to find a correspondence either with temporal logics or with automata.

The objective of this work is to provide an *automata-theoretic characterisation* of CTL* and ECTL*, by identifying two specific classes of alternating tree automata that are expressively equivalent to those logics (the used technique extends seamlessly to the finite-horizon variants). A first result is the proof of the equivalence of the *symmetric variant* of classic ranked *Hesitant Tree Automata* (HTA) [47] with both ECTL* and the bisimulation-invariant fragment of MCL. To this end, for technical convenience, we employ two intermediate formalisms. On the one hand, to prove the equivalence between HTA and ECTL*, we use a *syntactic variant* of ECTL*, called *Computation Dynamic Logic* (CDL), alongside its counting version (CCDL). In ECTL* temporal operators are specified by means of right-linear grammars, while CDL uses finite automata on finite words for the same purpose incorporated into the dynamic modalities. Moreover, while the path subformulae in ECTL* are part of the alphabet of the grammar, in CDL they are specified by means of a testing function over the set of states of the automaton. It is straightforward to move from one formalism to the other by means of a linear-time translation. This logic essentially lifts to the branching-time realm the *Linear Dynamic Logic* (LDL) proposed in [32, 83]. On the other hand, we consider a *first-order extension* [82] of HTAs (HFTA) and show them equivalent to MCL by proving a closure property under *chain projections*. The final result, then, follows from the equivalence between HTAs and the bisimulation-invariant fragment of HFTA. As a second result, we first identify the *graded extension* of HTAs (HGTA), together with its counter-free restriction (HGTA_{cf}), and then prove their equivalence with CCDL and CCTL*, respectively. While for the definition of HGTA the standard notion of counting modalities smoothly applies, introducing HGTA_{cf} proves quite more intricate. We show, indeed, that a naive application of counter-freeness in the context of tree-automata leads to a class of languages that are not CTL* definable. To overcome this problem, we identify the crucial *mutual-exclusion* property of a HGTA that constrains the automaton branching-behaviours. This property, together with counter-freeness of the automaton linear behaviours, provides an apt definition of HGTA_{cf}, something that was previously only hypothesised in [54, 55]. The above characterisations holds also under bisimulation-invariance assumptions. Specifically, HTA_{cf} is equivalent to both CTL* and the bisimulation-invariant fragment of MPL. All these results, coupled with the algebraic characterisation of tree languages provided in [74], brings the expressiveness landscape for branching-time temporal logics to the same level as their linear-time counterpart, thus closing a forty-year-old open question posed in [73, 74].

2 Preliminaries

Let $\mathbb{N}$ be the set of natural numbers. For $i, j \in \mathbb{N}$ with $i \leq j$, $[i, j]$ denotes the set of natural numbers k such that $i \leq k \leq j$. For a finite or infinite word ρ over some alphabet, $|\rho|$ is the length of ρ ($|\rho| = \omega$ if ρ is infinite) and for all $0 \leq i < |\rho|$, $\rho(i)$ is the $(i + 1)$-th letter of ρ.

Kripke Trees and Tree Languages. Given a non-empty set of directions D, a tree T (with set of directions in D) is a non-empty subset of D^* which is prefix closed (i.e., for each $w \cdot d \in T$ with $d \in D$, $w \in T$). Elements of T are called nodes and the empty word ε is the root of T. For $w \in T$, a *child* of w in T is a node in T of the form $w \cdot d$ for some $d \in D$. For $w \in T$, the *subtree of* T *rooted at node* w is the tree consisting of the nodes of the form w' such that $w \cdot w' \in T$. A *subtree of* T is a tree T′ such that for some $w \in T$, T′ is a subset of the subtree of T rooted at w. A *path* of T is a subtree π of T which is totally ordered by the child-relation (i.e., each node of π has at most one child in π). In the following, a path π of T is also seen as a word over T in accordance to the total ordering in π induced by the child relation. A *chain* of T is a subset of a path of T. A tree is *non-blocking* if each node has some child. A non-blocking tree T is infinite, and maximal paths in T are infinite as well.

For an alphabet Σ, a Σ-labelled tree is a pair (T, Lab) consisting of a tree and a labelling $Lab : T \mapsto \Sigma$ assigning to each node in T a symbol in Σ. A *tree-language* over Σ is a set of Σ-labeled trees. In this paper, we consider formalisms whose specifications denote *tree-languages* over a given alphabet Σ. For the easy of presentation, we assume that the labeled trees of a tree-language are non-blocking. All the results of this paper can be easily adapted to the general case, where the non-blocking assumption is relaxed. For a finite set AP of atomic propositions, a *Kripke tree* over AP is a non-blocking 2^{AP}-labelled tree.

Automata over Infinite and Finite Words. We first recall the class of parity nondeterministic automata on infinite words (parity NWA for short) which are tuples $\mathcal{A} = \langle \Sigma, Q, \delta, q_I, \Omega \rangle$, where Σ is a finite input alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \mapsto 2^Q$ is the transition function, $q_I \in Q$ is an initial state, and $\Omega : Q \mapsto \mathbb{N}$ is a parity acceptance condition over Q assigning to each state a natural number (color). Given a word ρ over Σ, a *path* of $\mathcal{A}$ over ρ is a word π over Q of length $|\rho| + 1$ ($|\rho| + 1$ is ω if ρ is infinite) such that $\pi(i+1) \in \delta(\pi(i), \rho(i))$ for all $0 \leq i < |\rho|$. A *run* over ρ is a path over ρ starting at the initial state. The NWA $\mathcal{A}$ is *counter-free* if for all $n > 0$, states $q \in Q$ and finite words ρ over Σ, the following holds: if there is a path from q to q over ρ^n, then there is also a path from q to q over ρ.

A run π of $\mathcal{A}$ over an infinite word ρ is *accepting* if the highest color of the states appearing infinitely often along π is even. The ω-language $\mathsf{L}(\mathcal{A})$ accepted by $\mathcal{A}$ is the set of infinite words ρ over Σ such that there is an accepting run π of $\mathcal{A}$ over ρ.

A parity acceptance condition $\Omega : Q \mapsto \mathbb{N}$ is a *Büchi* (resp., *coBüchi*) condition if there is an even (resp., odd) color $n \in \mathbb{N}$ such that $\Omega(Q) \subseteq \{n-1, n\}$. A *Büchi* (resp., *coBüchi*) NWA is a parity NWA whose acceptance condition is Büchi (resp., coBüchi).

We also consider NWA over finite words (NWA_f for short) which are defined as parity NWA but the parity condition Ω is replaced with a set $F \subseteq Q$ of accepting states. A run π over a finite word is *accepting* if its last state is accepting.

3 Branching-Time Temporal Logics

In this section, we recall syntax and semantics of Counting-CTL* (CCTL* for short) [55], which extends the classic branching-time temporal logic CTL* [25] with counting operators. Moreover, we introduce a novel branching-time temporal logic more expressive than CCTL*, called *Counting Computation Dynamic Logic* (CCDL for short). CCDL can be viewed as a branching-time extension of Linear Dynamic Logic (LDL) [32]. However, unlike LDL, we consider NWA_f over finite words, instead of regular expressions, as the building blocks of formulae. This approach is similar to the one adopted in [83] for Visibly Linear Dynamic Logic, a context-free extension of LTL.

The Logic CCTL*. The syntax of CCTL* is given by specifying inductively the set of *state formulae* φ and the set of *path formulae* ψ over a given finite set AP of atomic propositions:

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

where $p \in \mathrm{AP}$, $\mathtt{X}$ and $\mathtt{U}$ are the standard "next" and "until" temporal modalities, $\mathtt{E}$ is the existential path quantifier, and $\mathtt{D}^n$ is the counting operator with $n \in \mathbb{N} \setminus \{0\}$. The language of CCTL* consists of the state formulae of CCTL*. Standard CTL* is the fragment of CCTL* where the counting operators $\mathtt{D}^n$ with $n > 1$ are disallowed, and standard LTL [61] corresponds to the set of path formulae of CTL* where the path quantifiers are disallowed.

Given a Kripke tree $\mathcal{T} = (\mathrm{T}, \mathit{Lab})$ (over AP), a node w of T, an infinite path π of T, and $0 \leq i < |\pi|$, the satisfaction relations $(\mathcal{T}, w) \models \varphi$, for a state formula φ, (meaning that φ holds at node w of $\mathcal{T}$), and $(\mathcal{T}, \pi, i) \models \psi$, for a path formula ψ, (meaning that ψ holds at position i of the path π in $\mathcal{T}$) are defined as usual:

$(\mathcal{T}, w) \models p \quad \Leftrightarrow p \in \mathit{Lab}(w)$;
$(\mathcal{T}, w) \models \mathtt{E}\psi \quad \Leftrightarrow (\mathcal{T}, \pi, 0) \models \psi$ for some infinite path π of T starting at node w;
$(\mathcal{T}, w) \models \mathtt{D}^n\varphi \quad \Leftrightarrow$ there are at least n distinct children w' of w in T s.t. $(\mathcal{T}, w') \models \varphi$;
$(\mathcal{T}, \pi, i) \models \varphi \quad \Leftrightarrow (\mathcal{T}, \pi(i)) \models \varphi$;
$(\mathcal{T}, \pi, i) \models \mathtt{X}\,\psi \quad \Leftrightarrow (\mathcal{T}, \pi, i+1) \models \psi$;
$(\mathcal{T}, \pi, i) \models \psi_1\,\mathtt{U}\,\psi_2 \quad \Leftrightarrow$ for some $j \geq i$: $(\mathcal{T}, \pi, j) \models \psi_2$ and $(\mathcal{T}, \pi, k) \models \psi_1$ for all $i \leq k < j$.

Note that $\mathtt{D}^1\varphi$ corresponds to $\mathtt{EX}\,\varphi$. A Kripke tree $\mathcal{T}$ satisfies (or is a model of) a state formula φ, written $\mathcal{T} \models \varphi$, if $\mathcal{T}, \varepsilon \models \varphi$. The tree-language $\mathtt{L}(\varphi)$ of φ is the set of models of φ. For an LTL formula ψ and an infinite word ρ over 2^{AP}, ρ satisfies ψ, written $\rho \models \psi$, if $\mathcal{T}_\rho \models \mathtt{E}\psi$, where $\mathcal{T}_\rho$ is a trivial tree-encoding of ρ. For an LTL formula ψ, $\mathtt{L}(\psi)$ denotes the set of infinite words over 2^{AP} satisfying ψ.

The New Logic CCDL. Like CCTL*, the syntax of CCDL is composed of *state formulae* φ and *path formulae* ψ over a given finite set AP of atomic propositions, defined as follows:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathtt{E}\psi \mid \mathtt{D}^n\varphi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \langle\mathcal{A}\rangle\psi$$

where $p \in \mathrm{AP}$ and $\langle\mathcal{A}\rangle$ is the *existential sequencing* modality applied to a *testing* $\mathrm{NWA_f}$ $\mathcal{A}$. We define a *testing* $\mathrm{NWA_f}$ $\mathcal{A} = \langle 2^{\mathrm{AP}}, \mathrm{Q}, \delta, q_I, \mathrm{F}, \tau\rangle$ as consisting of an $\mathrm{NWA_f}$ $\langle 2^{\mathrm{AP}}, \mathrm{Q}, \delta, q_I, \mathrm{F}\rangle$ over finite words over 2^{AP} and a test function τ mapping states in Q to CCDL path formulae. Intuitively, along an infinite path π of a Kripke tree, the testing automaton accepts the labeling of a (possibly empty) infix $\pi(i) \ldots \pi(j-1)$ of π if the embedded $\mathrm{NWA_f}$ has an accepting run $q_i \ldots q_j$ over the labeling of such an infix so that, for each position $k \in [i, j]$, the formula $\tau(q_k)$ holds at position k along π. A test function τ is *trivial* if it maps each state to $\top$. We also use the shorthand $[\mathcal{A}]\psi \triangleq \neg\langle\mathcal{A}\rangle\neg\psi$ (*universal sequencing* modality). The language of CCDL consists of the state formulae of CCDL. We also consider the *bisimulation-invariant* fragment CDL of CCDL where the counting operators $\mathtt{D}^n$ with $n > 1$ are disallowed. Given a Kripke tree $\mathcal{T} = (\mathrm{T}, \mathit{Lab})$, an infinite path π of T, and $0 \leq i < |\pi|$, the semantics of modality $\langle\mathcal{A}\rangle$ is defined as follows, where $\mathcal{A} = \langle 2^{\mathrm{AP}}, \mathrm{Q}, \delta, q_I, \mathrm{F}, \tau\rangle$:

$(\mathcal{T}, \pi, i) \models \langle\mathcal{A}\rangle\psi \quad \Leftrightarrow$ for some $j \geq i$, $(i, j) \in \mathrm{R}_{\mathcal{A}}(\mathcal{T}, \pi)$ and $(\mathcal{T}, \pi, j) \models \psi$

where $\mathrm{R}_{\mathcal{A}}(\mathcal{T}, \pi)$ is the set of pairs (i, j) with $j \geq i$ such that there is an accepting run $q_i \ldots q_j$ of the $\mathrm{NWA_f}$ embedded in $\mathcal{A}$ over $\mathit{Lab}(\pi(i)) \ldots \mathit{Lab}(\pi(j-1))$ and, for all $k \in [i, j]$, it holds that $(\mathcal{T}, \pi, k) \models \tau(q_k)$. The notions of a model and tree-language of a CCDL formula are defined as for CCTL*.

Embedding of CCTL* into CCDL. The logic CCTL* can be easily embedded into CCDL as follows. Let $\mathcal{A}$ be the testing $\mathrm{NWA_f}$ having trivial tests and accepting all and only the words of length 1, and for CCDL path formulae ψ_1, ψ_2, let $\mathcal{A}_{\psi_1,\psi_2} = \langle 2^{\mathrm{AP}}, \{q_1, q_2\}, \delta, q_1, \{q_2\}, \tau\rangle$ be the testing $\mathrm{NWA_f}$ where, for all $a \in 2^{\mathrm{AP}}$, $\delta(q_1, a) = \{q_1, q_2\}$, $\delta(q_2, a) = \emptyset$, $\tau(q_1) = \psi_1$, and $\tau(q_2) = \psi_2$. Then, the next and until formulae $\mathtt{X}\,\psi_1$ and $\psi_1\,\mathtt{U}\,\psi_2$ can be expressed as follows: $\mathtt{X}\,\psi_1 \equiv \langle\mathcal{A}\rangle\psi_1$ and $\psi_1\,\mathtt{U}\,\psi_2 \equiv \psi_2 \vee \langle\mathcal{A}_{\psi_1,\psi_2}\rangle\top$.

4 Alternating Tree Automata

In this section, we recall the class of parity *alternating tree automata with first-order constraints* (FTA for short), introduced in [82] to provide an automata-theoretic characterization of MSO interpreted on arbitrary labeled trees. Moreover, we also recall the class of *graded alternating tree automata* (GTA for short), a subclass of FTA, which was introduced in [44] and allows for expressing counting modal requirements on the child relation of an input tree. The transition relation of both FTA and GTA is based on constraints on the set of states Q written as formulae in a suitable language, called *one-step logic*. The *one-step interpretations* of such formulae over Q are pairs (S, I), where S is an arbitrary (possibly infinite) non-empty set and I is a mapping $\mathrm{I} : \mathrm{S} \mapsto 2^{\mathrm{Q}}$, assigning to each element of S a subset of Q. Intuitively, the pair (S, I) describes the local behaviour of the automaton on reading a node w of the input tree. The set S corresponds to the set of children of the current input node w and, for each $w' \in \mathrm{S}$, $\mathrm{I}(w')$ is the set of states associated with the copies of the automaton which are sent to the child w' of w.

One-Step Logic for GTA. The one-step relation of GTA is specified by means of formulae θ of one-step positive graded modal logic over Q, we call *graded* Q*-constraints*, defined as:

$\theta ::= \top \mid \bot \mid \theta \vee \theta \mid \theta \wedge \theta \mid \Diamond_k\alpha \mid \Box_k\alpha$

where $k \in \mathbb{N} \setminus \{0\}$ and α is a *positive* Boolean formula over Q. The atomic formulae $\Diamond_k\alpha$ and $\Box_k\alpha$ are called Q*-atoms*. The atom $\Diamond_1\alpha$ (resp., $\Box_1\alpha$) is also denoted by $\Diamond\alpha$ (resp., $\Box\alpha$). A formula θ is *symmetric* if the atoms occurring in θ are of the form $\Diamond\alpha$ or $\Box\alpha$.

The satisfaction relation $(\mathrm{S}, \mathrm{I}) \models \theta$ for a one-step interpretation (S, I) over Q is inductively defined as follows (we omit the clauses for positive Boolean connectives which are standard):

- $(\mathrm{S}, \mathrm{I}) \models \Diamond_k\alpha$ if $|\{s \in \mathrm{S} \mid \mathrm{I}(s) \models \alpha\}| \geq k$;
- $(\mathrm{S}, \mathrm{I}) \models \Box_k\alpha$ if $|\{s \in \mathrm{S} \mid \mathrm{I}(s) \not\models \alpha\}| < k$.

If $(\mathrm{S}, \mathrm{I}) \models \theta$, we say that (S, I) is a model of θ. Intuitively, for an alternating automaton $\mathcal{A}$ with set of states Q, the atom $\Diamond_k\alpha$ requires that at the current input node w, there are at least k children of w and, for each of such nodes w', (**) there is a subset $\mathrm{Q}' \subseteq \mathrm{Q}$ satisfying α such that a copy of $\mathcal{A}$ is sent to node w' in state q, for each $q \in \mathrm{Q}'$. For an atom $\Box_k\alpha$, the previous condition (**) is required to hold for all but at most $k - 1$ children w' of w.

One-Step Logic for FTA. The one-step language $\mathrm{FOE}_1^+(\mathrm{Q})$ of positive first-order formulae with equality and monadic predicates over Q and first-order variables in Vr_1 is given by the sentences (formulae without free variables) generated by the following grammar:

$\theta ::= \top \mid \bot \mid q(x) \mid x = y \mid x \neq y \mid \theta \vee \theta \mid \theta \wedge \theta \mid \exists x. \theta \mid \forall x. \theta$

where $q \in \mathrm{Q}$ and $x, y \in \mathrm{Vr}_1$. An $\mathrm{FOE}_1^+(\mathrm{Q})$-sentence θ is called *first-order* Q*-constraint*; θ is *symmetric* if it does not contain equality and inequality atomic formulae. In FTA, these constraints allow formulae that refer to the children of a node of a tree by means of explicit first-order variables.

Given a one-step interpretation (S, I) over Q and an assignment $\mathsf{V} : \mathrm{Vr}_1 \to \mathrm{S}$ of the first-order variables, the satisfaction relation $(\mathrm{S}, \mathrm{I}), \mathsf{V} \models \theta$ is defined in a standard way. For sentences θ, this relation is independent of V, and we simply write $(\mathrm{S}, \mathrm{I}) \models \theta$. Note that graded Q-constraints can be trivially expressed in $\mathrm{FOE}_1^+(\mathrm{Q})$, and first-order Q-constraints θ are *monotonic*, i.e., for all one-step interpretations (S, I) and $(\mathrm{S}, \mathrm{I}')$ such that $\mathrm{I}(s) \subseteq \mathrm{I}'(s)$ for each $s \in \mathrm{S}$, it holds that $(\mathrm{S}, \mathrm{I}) \models \theta$ entails $(\mathrm{S}, \mathrm{I}') \models \theta$. A *minimal model* of θ is a model (S, I) of θ such that there is no model $(\mathrm{S}, \mathrm{I}')$ of θ with $\mathrm{I}' \neq \mathrm{I}$ and $\mathrm{I}'(s) \subseteq \mathrm{I}(s)$ for each $s \in \mathrm{S}$.

Parity GTA and Parity FTA. A *parity* GTA $\mathcal{A}$ is a tuple $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \Omega \rangle$, where Σ, Q, q_I, and Ω are defined as for parity NWA, while the transition function δ is a mapping from $\mathrm{Q} \times \Sigma$ to the set of graded Q-constraints. The set $\mathsf{Atoms}(\mathcal{A})$ is the set of Q-atoms occurring in the transition function of $\mathcal{A}$. Parity FTA $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \Omega \rangle$ are defined similarly but the transition function δ is of the form $\delta : \mathrm{Q} \times \Sigma \mapsto \mathrm{FOE}_1^+(\mathrm{Q})$. A GTA (resp., FTA) $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \Omega \rangle$ is *symmetric* if for all $(q, a) \in \mathrm{Q} \times \Sigma$, the constraint $\delta(q, a)$ is symmetric. GTA (resp., FTA) $\mathcal{A}$ operate over non-blocking Σ-labeled trees $(\mathrm{T}, \mathit{Lab})$. A run of $\mathcal{A}$ over $(\mathrm{T}, \mathit{Lab})$ is a $(\mathrm{Q} \times \mathrm{T})$-labeled tree $\mathsf{r} = (\mathrm{T}_\mathsf{r}, \mathit{Lab}_\mathsf{r})$, where each node of T_r labelled by (q, w) describes a copy of $\mathcal{A}$ that is in state q reading the node w of T. Moreover, we require that:

- $\mathit{Lab}_\mathsf{r}(\varepsilon) = (q_I, \varepsilon)$ (initially, the automaton is in state q_I reading the root of the input T);
- for each node $y \in \mathrm{T}_\mathsf{r}$ with $\mathit{Lab}_\mathsf{r}(y) = (q, w)$ and denoted by S_w the set of children of node w in the input T, there is a one-step interpretation $(\mathrm{S}_w, \mathrm{I})$ over Q satisfying $\delta(q, \mathit{Lab}(w))$ such that the set of labels associated with the children of y in T_r consists of the pairs (q', w') with $w' \in \mathrm{S}_w$ and $q' \in \mathrm{I}(w')$.

The run r is accepting if, for all infinite paths π starting from the root, the infinite sequence of states in $\mathit{Lab}_\mathsf{r}(\pi(0))\mathit{Lab}_\mathsf{r}(\pi(1))\ldots$ satisfies the parity acceptance condition Ω. The language $\mathsf{L}(\mathcal{A})$ accepted by $\mathcal{A}$ is the tree-language over Σ consisting of the non-blocking Σ-labeled trees $(\mathrm{T}, \mathit{Lab})$ such that there is an accepting run of $\mathcal{A}$ over $(\mathrm{T}, \mathit{Lab})$.

Dualization. For a graded Q-constraint θ, the *dual* $\widetilde{\theta}$ of θ is obtained from θ by exchanging $\vee$ with $\wedge$, $\top$ with $\bot$, and Q-atoms $\Diamond_k \alpha$ with $\Box_k \widetilde{\alpha}$, and vice versa, where $\widetilde{\alpha}$ is obtained from α by exchanging $\vee$ with $\wedge$. For example, the dual of $\Diamond_{k_1}(q_0 \vee q_1) \wedge \Box_{k_2} q_2$ is $\Box_{k_1}(q_0 \wedge q_1) \vee \Diamond_{k_2} q_2$. Similarly, the dual $\widetilde{\theta}$ of a first-order Q-constraint θ is obtained from θ by exchanging $\vee$ with $\wedge$, $\top$ with $\bot$, $x = y$ with $x \neq y$, and existential quantification $\exists x$ with universal quantification $\forall x$. For a parity GTA (resp., parity FTA) $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \Omega \rangle$, the *dual automaton* of $\mathcal{A}$ is the parity GTA (resp., parity FTA) $\widetilde{\mathcal{A}} = \left\langle \Sigma, \mathrm{Q}, \widetilde{\delta}, q_I, \widetilde{\Omega} \right\rangle$, where for all $(q, a) \in \mathrm{Q} \times \Sigma$, $\widetilde{\Omega}(q) = \Omega(q) + 1$ and $\widetilde{\delta}(q, a)$ is the dual of $\delta(q, a)$. By [82, 12], the following holds.

▶ **Proposition 4.1** ([82, 12]). *Let $\mathcal{A}$ be a parity* GTA *(resp., parity* FTA*). Then, the dual automaton of $\mathcal{A}$ is a parity* GTA *(resp., parity* FTA*) accepting the complement of* $\mathsf{L}(\mathcal{A})$.

5 Automata Characterisations of CDL and CCTL*

In this section, we provide effective automata-theoretic characterisations of the logics CCDL and CCTL*. We first consider the graded version of the class of *hesitant alternating tree automata* (HTA, for short), the latter being a well-known formalism introduced in [47] as an optimal automata-theoretic framework for model checking and synthesis of CTL*. We show that the graded version of HTA (HGTA for short) characterises the logic CCDL. In order to capture the logic CCTL*, we consider a subclass of HGTA obtained by enforcing a counter-freeness requirement on the linear-time behaviour of the automaton along an existential component together with an additional condition (we call *mutual-exclusion property*) on the alphabet of the linearization word automaton.

In the following, for a GTA $\mathcal{A}$ and a set $\mathrm{A} \subseteq \mathsf{Atoms}(\mathcal{A})$, we denote by $\mathsf{Con}(\mathrm{A})$ (resp., $\mathsf{Dis}(\mathrm{A})$) the conjunction (resp., disjunction) of the atoms occurring in A. As usual, the empty conjunction is $\top$ and the empty disjunction is $\bot$.

Hesitant GTA. An *hesitant* GTA (HGTA for short) is a tuple $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$, where $\langle \Sigma, \mathrm{Q}, \delta, q_I, \Omega \rangle$ is a parity GTA, $\mathsf{H} = \langle \mathrm{Q}_1, \ldots, \mathrm{Q}_n \rangle$ is an *ordered* tuple of non-empty pairwise disjunct subsets Q_i of Q (called *components* of $\mathcal{A}$) which form a partition of Q, and $\mathsf{H}_\exists$ is a subset of the components in H (the so called *existential components*). Thus, like HTA [47], there is an ordered partition of Q into disjoint sets $\mathrm{Q}_1, \ldots, \mathrm{Q}_n$. Moreover, each component Q_i is classified as *transient*, *existential*, or *universal*, and the following holds:

- *transient requirement*: for each transient component Q_i and $(q, a) \in \mathrm{Q}_i \times \Sigma$, $\delta(q, a)$ only refers to states in components Q_j such that $j < i$;
- *existential requirement*: for each existential component Q_i and $(q, a) \in \mathrm{Q}_i \times \Sigma$, $\delta(q, a)$ can be rewritten as a disjunction of conjunctions of the form $\Diamond q' \wedge \mathsf{Con}(\mathrm{A})$, where $q' \in \mathrm{Q}_i$ and the atoms in A only refer to states in components Q_j such that $j < i$;
- *universal requirement*: for each universal component Q_i and $(q, a) \in \mathrm{Q}_i \times \Sigma$, $\delta(q, a)$ can be rewritten as a conjunction of disjunctions of the form $\Box q' \vee \mathsf{Dis}(\mathrm{A})$, where $q' \in \mathrm{Q}_i$ and the atoms in A only refer to states in components Q_j such that $j < i$;
- *hesitant acceptance requirement*: for each existential (resp., universal) component Q_i, the restriction Ω_{Q_i} of Ω to the set Q_i is a Büchi condition (resp., coBüchi condition).

The first three requirements ensure that every infinite path of a run of $\mathcal{A}$ gets trapped within some existential or universal component Q_i. The existential requirement establishes that from each existential state $q \in \mathrm{Q}_i$, exactly one copy is sent to a child of the current input node in component Q_i (all the other copies move to states with order lower than i). The universal requirement corresponds to the dual of the existential requirement. Finally, the hesitant acceptance requirement ensures that for each infinite path π of a run that gets trapped into an existential (resp., universal) component, π is accepting iff π visits infinitely many times states with even color (resp., π visits finitely many times states with odd color).

▶ **Example 5.1.** Let $\mathrm{AP} = \{p\}$ and φ_p be the CTL* formula $\mathtt{EX}\, p$ asserting that the root of the given Kripke tree has a child where p holds. We consider the tree-language L_2 consisting of the Kripke trees $\mathcal{T}$ such that there is an infinite path π from the root so that p never holds along π and at the even positions $2i$, φ_p holds at node $\pi(2i)$. L_2 requires counting modulo 2 and cannot be expressed in CCTL*. The language L_2 is recognised by the HGTA $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \langle \mathrm{Q}_1, \mathrm{Q}_2 \rangle, \{\mathrm{Q}_2\}, \Omega \rangle$ consisting of three states having colour 0: the existential states q_I and q having the same and highest order ($\mathrm{Q}_2 = \{q_I, q\}$) and the transient state q_p ($\mathrm{Q}_1 = \{q_p\}$). Moreover, (i) $\delta(q_p, \{p\}) = \top$ and $\delta(q_p, \emptyset) = \bot$, (ii) $\delta(q_I, \emptyset) = \Diamond q \wedge \Diamond q_p$ and $\delta(q_I, \{p\}) = \bot$, and (iii) $\delta(q, \emptyset) = \Diamond q_I$ and $\delta(q, \{p\}) = \bot$.

Linearization. Fix an HGTA $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$. Given a component Q_i of $\mathcal{A}$ and $\mathrm{A} \subseteq \mathsf{Atoms}(\mathcal{A})$, the set A *is lower than* Q_i if the atoms in A only refer to states with order $j < i$. For each existential (resp., universal) component Q_i and $q \in \mathrm{Q}_i$, we introduce a Büchi (resp., coBüchi) NWA $\mathcal{A}_{\mathrm{Q}_i,q}$ over the alphabet $\Sigma \times \mathsf{Atoms}(\mathcal{A})$. Intuitively, $\mathcal{A}_{\mathrm{Q}_i,q}$ encodes the *modular* behaviour of $\mathcal{A}$ starting at state q, which is composed of the behaviour along Q_i (which is linear-time when Q_i is existential), plus additional moves that lead to states with order lower than i: the input alphabet $\Sigma \times \mathsf{Atoms}(\mathcal{A})$ keeps track of these additional moves. When Q_i is universal, then $\mathcal{A}_{\mathrm{Q}_i,q}$ can be viewed as a universal tree automaton.

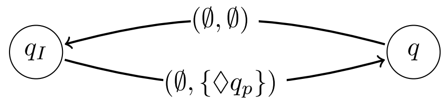
▶ **Definition 5.2** (Linearization word automata). *For each non-transient component* Q_i *of* $\mathcal{A}$ *and* $q \in \mathrm{Q}_i$*, we denote by* $\mathcal{A}_{\mathrm{Q}_i,q}$ *the parity* NWA $\mathcal{A}_{\mathrm{Q}_i,q} = \langle \Sigma \times 2^{\mathsf{Atoms}(\mathcal{A})}, \mathrm{Q}_i, \delta_{\mathrm{Q}_i}, q, \Omega_{\mathrm{Q}_i} \rangle$ *where for all* $q' \in \mathrm{Q}_i$*,* $a \in \Sigma$*, and* $\mathrm{A} \subseteq \mathsf{Atoms}(\mathcal{A})$*,* $\delta_{\mathrm{Q}_i}(q', (a, \mathrm{A}))$ *is defined as follow:*

- *Case* Q_i *is existential:* $q'' \in \delta_{\mathrm{Q}_i}(q', (a, \mathrm{A}))$ *if there is conjunction* ξ *in the disjunctive normal form of* $\delta(q', a)$ *such that* $\xi = \Diamond q'' \wedge \mathsf{Con}(\mathrm{A})$ *(note that* A *is lower than* Q_i*).*
- *Case* Q_i *is universal:* $q'' \in \delta_{\mathrm{Q}_i}(q', (a, \mathrm{A}))$ *if there is disjunction* ξ *in the conjunctive normal form of* $\delta(q', a)$ *such that* $\xi = \Box q'' \vee \mathsf{Dis}(\mathrm{A})$ *(note that* A *is lower than* Q_i*).*

Let Υ_{Q_i} *be the set of elements* $\mathrm{A} \subseteq \mathsf{Atoms}(\mathcal{A})$ *s.t.* $\delta_{\mathrm{Q}_i}(q', (a, \mathrm{A})) \neq \emptyset$ *for some* $(q', a) \in \mathrm{Q}_i \times \Sigma$*.*

▶ Remark 5.3. Note that the transition function of $\mathcal{A}_{\mathrm{Q}_i,q}$ is independent of q, and $\mathcal{A}_{\mathrm{Q}_i,q}$ is a Büchi (resp., coBüchi) NWA if Q_i is existential (resp., universal). We can equate the parity NWA $\mathcal{A}_{\mathrm{Q}_i,q}$ to the parity NWA over the alphabet $\Sigma \times \Upsilon_{\mathrm{Q}_i}$ which is obtained from $\mathcal{A}_{\mathrm{Q}_i,q}$ by restricting the transition function to the alphabet $\Sigma \times \Upsilon_{\mathrm{Q}_i}$. In the following, we write $\mathcal{A}_{\mathrm{Q}_i,q}$ to denote this automaton. Observe that each set of atoms $\mathrm{A} \in \Upsilon_{\mathrm{Q}_i}$ is lower than Q_i.

If we consider the HGTA $\mathcal{A}$ of Example 5.1, the Büchi NWA $\mathcal{A}_{\mathrm{Q}_2,q_I}$ associated with the existential component Q_2 is illustrated below. Note that $\Upsilon_{\mathrm{Q}_2} = \{\emptyset, \{\Diamond q_p\}\}$.



Let us fix an HGTA $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$ with $\mathsf{H} = \langle \mathrm{Q}_1, \ldots, \mathrm{Q}_n \rangle$. For each graded Q-constraint θ, we denote by $\mathcal{A}^\theta$ the HGTA $\langle \Sigma, \mathrm{Q} \cup \{\theta\}, \delta_\theta, \theta, \mathsf{H}_\theta, \mathsf{H}_\exists, \Omega \cup (\theta \to 0) \rangle$ where for the states in Q, δ_θ agrees with δ, for the initial state θ, $\delta_\theta(\theta, a) = \theta$ for all $a \in \Sigma$, and $\mathsf{H}_\theta = \langle \mathrm{Q}_1, \ldots, \mathrm{Q}_n, \{\theta\} \rangle$. Note that $\{\theta\}$ is a transient component with highest order. Thus, from the root of the input tree, $\mathcal{A}^\theta$ send copies of $\mathcal{A}$ to the children of the root according to the constraint θ. By construction, for each existential state q of an HGTA $\mathcal{A}$, we obtain the following characterisation of the language $\mathsf{L}(\mathcal{A}^q)$, where $\mathcal{A}^q$ is the HGTA obtained from $\mathcal{A}$ by setting q as initial state instead of q_I, in terms of the linearization of $\mathcal{A}$.

▶ **Proposition 5.4.** *Let* $\mathcal{A}$ *be an* HGTA*,* Q_i *be an existential component of* $\mathcal{A}$*, and* $q \in \mathrm{Q}_i$*. Then, for each input* $\mathcal{T} = (\mathrm{T}, Lab)$*,* $\mathcal{T} \in \mathsf{L}(\mathcal{A}^q)$ if and only if *there is an infinite path* π *of* $\mathcal{T}$ *starting at the root and an infinite word* $\rho \in \mathsf{L}(\mathcal{A}_{\mathrm{Q}_i,q})$ *such that* ρ *is of the form* $\rho = (Lab(\pi(0)), \mathrm{A}_0)(Lab(\pi(1)), \mathrm{A}_1) \ldots$ *and for each* $i \geq 0$*,* $\mathcal{T}_{\pi(i)} \in \mathsf{L}(\mathcal{A}^{\mathsf{Con}(\mathrm{A}(i))})$*, where* $\mathcal{T}_{\pi(i)}$ *is the labelled subtree of* $\mathcal{T}$ *rooted at node* $\pi(i)$*.*

Counter-free HGTA. In order to capture CCTL*, we introduce a subclass of HGTA obtained by enforcing additional conditions. By Proposition 5.4 and the equivalence of LTL and Büchi *counter-free* NWA [19], a natural condition consists in requiring that for each non-transient component Q_i of the HGTA and state $q \in \mathrm{Q}_i$, the NWA $\mathcal{A}_{\mathrm{Q}_i,q}$ is counter-free (*counter-freeness requirement*).[1] However, this condition is not sufficient for characterising the logic CCTL*. A counterexample is the HGTA $\mathcal{A}$ of Example 5.1 which clearly satisfies the counter-freeness requirement but recognises a tree-language which is not expressible in CCTL*. We introduce an additional condition (*mutual-exclusion property*) on the alphabets of the linearization automata (see Definition 5.5 below). A *Counter-free* HGTA ($\mathrm{HGTA}_{\mathrm{cf}}$ for short) is an HGTA satisfying both the counter-free requirement and the mutual-exclusion condition.

[1] Note that the property of an NWA to be counter-free is independent of the initial state.

▶ **Definition 5.5.** *An* HGTA $\mathcal{A}$ *satisfies the* mutual-exclusion property *if for each non-transient component* Q_i *and for all* $A, A' \in \Upsilon_{Q_i}$ *such that* $A \neq A'$*, it holds that there exists an atom* $\mathsf{atom} \in A$ *and an atom* $\mathsf{atom}' \in A'$ *such that* $\mathsf{L}(\mathcal{A}^{\mathsf{atom}})$ *is the complement of* $\mathsf{L}(\mathcal{A}^{\mathsf{atom}'})$*. Note that if* Υ_{Q_i} *is a singleton, then the previous property is fulfilled.*

Evidently, if $\mathcal{A}$ satisfies the mutual-exclusion condition, then for each non-transient component Q_i and for all $A, A' \in \Upsilon_{Q_i}$ such that $A \neq A'$, it holds that $\mathsf{L}(\mathcal{A}^{\mathsf{Con}(A)}) \cap \mathsf{L}(\mathcal{A}^{\mathsf{Con}(A')}) = \emptyset$. Intuitively, the mutual-exclusion condition requires that along a non-transient component Q_i, the distinct moves $A \in \Upsilon_{Q_i}$ (these moves lead to components with order lower than i) are mutually exclusive. Let us consider again the HGTA $\mathcal{A}$ of Example 5.1. Since $\Upsilon_{Q_2} = \{\emptyset, \{\Diamond q_p\}\}$, by Definition 5.5, $\mathcal{A}$ does not satisfy the mutual-exclusion condition. Note that $\mathsf{Con}(\emptyset) = \top$ and $\mathsf{Con}(\{\Diamond q_p\}) = \Diamond q_p$. Hence, $\mathsf{L}(\mathcal{A}^{\top}) \cap \mathsf{L}(\mathcal{A}^{\mathsf{Con}(\{\Diamond q_p\})}) = \mathsf{L}(\mathcal{A}^{\mathsf{Con}(\{\Diamond q_p\})}) = \mathsf{L}(\mathtt{EX}\, p) \neq \emptyset$.

The dual $\widetilde{\mathcal{A}}$ of an HGTA $\mathcal{A} = \langle \Sigma, Q, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$ is the tuple $\left\langle \Sigma, Q, \widetilde{\delta}, q_I, \mathsf{H}, \widetilde{\mathsf{H}_\exists}, \widetilde{\Omega} \right\rangle$, where $\widetilde{\delta}$ and $\widetilde{\Omega}$ are defined as for the dual of an arbitrary parity GTA and $\widetilde{\mathsf{H}_\exists}$ consists of the universal components of $\mathcal{A}$. By construction and Proposition 4.1, the considered subclasses of GTA are closed under Boolean language operations.

▶ **Proposition 5.6.** HGTA *(resp.,* HGTA_{cf}*) and* HGTA *satisfying the mutual-exclusion property are effectively closed under Boolean language operations.*

Enforcing the Mutual-exclusion Property. By exploiting dualization, an HGTA $\mathcal{A}$ can be converted into an equivalent HGTA $\mathcal{A}_s$ satisfying the mutual-exclusion condition. Intuitively, $\mathcal{A}_s$ is obtained by merging in a syntactical and *modular* way $\mathcal{A}$ with a renaming of the dual HGTA $\widetilde{\mathcal{A}}$.

▶ **Proposition 5.7.** *Given an* HGTA $\mathcal{A}$*, one can construct an* HGTA $\mathcal{A}_s$ *such that* $\mathcal{A}_s$ *satisfies the mutual-exclusion condition and* $\mathsf{L}(\mathcal{A}_s) = \mathsf{L}(\mathcal{A})$*.*

Note that the translation in Proposition 5.7 changes the second component Υ_{Q_i} of the alphabets of the linearization automata. Since counter-free NWA are not closed under inverse projection, the construction does not preserve the counter-freeness property. For example, for the HGTA of Example 5.1, the translation replaces the edge from q to q_I with label $(\emptyset, \emptyset)$ of the NWA $\mathcal{A}_{Q_2, q_I}$ with two edges from q to q_I: one with label $(\emptyset, \{\Diamond q_p\})$ and the other one with label $(\emptyset, \{\Box q'_p\})$ where $\mathsf{L}(\mathcal{A}^{\mathsf{Con}(\{\Box q'_p\})}) = \mathsf{L}(\neg \mathtt{EX}\, p)$. The resulting NWA is not counter-free.

5.1 From Automata to Logics and Back

In this section, we show that the class of HGTA and the logic CCDL are effectively equivalent, and the class of HGTA_{cf} effectively characterizes CCTL*. We start with the translations from automata to logics.

▶ **Theorem 5.8.** *Let* $\mathcal{A}$ *be an* HGTA *(resp., an* HGTA_{cf}*) over* 2^{AP}*. Then, one can construct a* CCDL *(resp.,* CCTL**) formula* $\varphi_{\mathcal{A}}$ *such that* $\mathsf{L}(\varphi_{\mathcal{A}}) = \mathsf{L}(\mathcal{A})$*. Moreover,* $\varphi_{\mathcal{A}}$ *is a* CDL *(resp. a* CTL**) formula if* $\mathcal{A}$ *is symmetric.*

Proof. We focus on the translation from HGTA_{cf} $\mathcal{A} = \langle \Sigma, Q, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$ to CCTL*. For each $q \in Q$, we construct a CCTL* formula φ_q such that $\mathsf{L}(\varphi_q) = \mathsf{L}(\mathcal{A}^q)$ and φ_q is a CTL* formula if $\mathcal{A}$ is symmetric. Thus, by setting $\varphi_{\mathcal{A}} \triangleq \varphi_{q_I}$, Theorem 5.8 directly follows. The proof is by induction on the order ℓ of the component Q_ℓ such that $q \in Q_\ell$. We distinguish the cases where q is transient, existential, or universal. The transient case is easy and the universal case follows from the existential case by a dualization argument. Now, assume that q is existential. Let us consider the Büchi NWA $\mathcal{A}_{\mathrm{Q}_\ell,q}$ over $2^{\mathrm{AP}} \times \Upsilon_{\mathrm{Q}_\ell}$ as defined in Definition 5.2. Recall that $\mathcal{A}_{\mathrm{Q}_\ell,q}$ is counter-free. Moreover, $\Upsilon_{\mathrm{Q}_\ell} \subseteq 2^{\mathsf{Atoms}(\mathcal{A})}$ contains only elements A such that states occurring in the atoms of A have order j lower than ℓ. Thus, by the induction hypothesis, for each $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$, one can build a CCTL* formula φ_{A} such that $\mathsf{L}(\mathcal{A}^{\mathsf{Con}(\mathrm{A})}) = \mathsf{L}(\varphi_{\mathrm{A}})$. Hence, since $\mathcal{A}$ satisfies the mutual-exclusion condition, the following holds:

Claim 1. For all $\mathrm{A}, \mathrm{A}' \in \Upsilon_{\mathrm{Q}_\ell}$ such that $\mathrm{A} \neq \mathrm{A}'$, $\mathsf{L}(\varphi_{\mathrm{A}}) \cap \mathsf{L}(\varphi_{\mathrm{A}'}) = \emptyset$.

For each $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$, let p_{A} be a fresh atomic proposition. We denote by $\mathrm{AP}_{\mathsf{ex}}$ the extension of AP with these fresh propositions. Moreover, let $\mathcal{A}_{\mathsf{ex},\mathrm{Q}_\ell,q}$ be the Büchi NWA over $2^{\mathrm{AP}_{\mathsf{ex}}}$ having the same set of states, initial state, acceptance condition as $\mathcal{A}_{\mathrm{Q}_\ell,q}$ and whose transition function $\delta_{\mathsf{ex},\mathrm{Q}_\ell}$ is obtained from the transition function δ_{Q_ℓ} of $\mathcal{A}_{\mathrm{Q}_\ell,q}$ as follows: for all $q' \in \mathrm{Q}_\ell$ and $a_{\mathsf{ex}} \in 2^{\mathrm{AP}_{\mathsf{ex}}}$, if a_{ex} is of the form $a \cup \{p_{\mathrm{A}}\}$, for some $a \in 2^{\mathrm{AP}}$ and $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$, (i.e., a_{ex} contains a unique proposition in $\mathrm{AP}_{\mathsf{ex}} \setminus \mathrm{AP}$), then $\delta_{\mathsf{ex},\mathrm{Q}_\ell}(q', a_{\mathsf{ex}}) = \delta_{\mathrm{Q}_\ell}(q', (a, \mathrm{A}))$; otherwise, $\delta_{\mathsf{ex},\mathrm{Q}_\ell}(q', a_{\mathsf{ex}}) = \emptyset$. Being $\mathcal{A}_{\mathrm{Q}_\ell,q}$ counter-free, $\mathcal{A}_{\mathsf{ex},\mathrm{Q}_\ell,q}$ is clearly counter-free as well. Thus, by [19], one can construct an LTL formula ψ over $\mathrm{AP}_{\mathsf{ex}}$ such that $\mathsf{L}(\psi) = \mathsf{L}(\mathcal{A}_{\mathsf{ex},\mathrm{Q}_\ell,q})$. Since $\mathsf{L}(\mathcal{A}^{\mathsf{Con}(\mathrm{A})}) = \mathsf{L}(\varphi_{\mathrm{A}})$ for all $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$, by construction and Proposition 5.4, we obtain the following characterization of the tree-language $\mathsf{L}(\mathcal{A}^q)$.

Claim 2. For each Kripke tree $\mathcal{T} = (\mathrm{T}, \mathit{Lab})$, $\mathcal{T} \in \mathsf{L}(\mathcal{A}^q)$ iff there is an infinite path π of $\mathcal{T}$ from the root and an infinite word ρ over $2^{\mathrm{AP}_{\mathsf{ex}}}$ such that $\rho \models \psi$ and, for all $j \geq 0$, (i) $\rho(j) \cap \mathrm{AP} = \mathit{Lab}(\pi(j))$, (ii) for all $p_{\mathrm{A}} \in \rho(j)$, $(\mathcal{T}, \pi(j)) \models \varphi_{\mathrm{A}}$, and (iii) there is a unique $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$ such that $p_{\mathrm{A}} \in \rho(j)$.

Note that since $\mathsf{L}(\psi) = \mathsf{L}(\mathcal{A}_{\mathsf{ex},\mathrm{Q}_\ell,q})$, by construction, point (iii) in Claim 2 follows from the fact that $\rho \models \psi$. By exploiting the always modality $\mathtt{G}$ ($\mathtt{G}\,\xi$ is a shorthand of $\neg(\top\,\mathtt{U}\,\neg\xi)$) and both conjunction and disjunction, w.l.o.g. we can assume that the LTL formula ψ is in negation normal form, i.e., negation is applied only to atomic propositions. Now, let $f(\psi)$ be the CCTL* path formula over AP obtained from ψ by replacing each literal of the form p_{A} (resp., $\neg p_{\mathrm{A}}$), where $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$, with the CCTL* state formula φ_{A} (resp., $\bigvee_{\mathrm{A}' \in \Upsilon_{\mathrm{Q}_\ell} \setminus \{\mathrm{A}\}} \varphi_{\mathrm{A}'}$). Finally, let us consider the CCTL* state formula φ_q defined as follows:

$\varphi_q \triangleq \mathtt{E}(f(\psi) \wedge \mathtt{G} \bigvee_{\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}} \varphi_{\mathrm{A}})$.

Note that the second conjunct in the state formula φ_q ensures that, for the infinite path π selected by the path quantifier $\mathtt{E}$ and for each $j \geq 0$, the state formula φ_{A} holds at node $\pi(j)$ for some $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$. We show that a Kripke tree $\mathcal{T} = (\mathrm{T}, \mathit{Lab})$ satisfies φ_q iff the characterization of $\mathsf{L}(\mathcal{A}^q)$ in Claim 2 holds. Hence, the result follows.

We shall now focus on the left-right implication of the equivalence (the right-left implication is similar). Thus, assume that $\mathcal{T} \models \varphi_q$. Hence, there exists an infinite path π of $\mathcal{T}$ from the root and an infinite sequence $\nu = \mathrm{A}_0, \mathrm{A}_1, \ldots$ over $\Upsilon_{\mathrm{Q}_\ell}$ such that $(\mathcal{T}, \pi, 0) \models f(\psi)$ and for each $j \geq 0$, $(\mathcal{T}, \pi(j)) \models \varphi_{\mathrm{A}_j}$. Let $\mathit{Lab}(\pi) \otimes \nu$ be the infinite word over $2^{\mathrm{AP}_{\mathsf{ex}}}$ defined as follows for all $j \geq 0$: $(\mathit{Lab}(\pi) \otimes \nu)(j) = \mathit{Lab}(\pi(j)) \cup \{p_{\mathrm{A}_j}\}$. By Claim 2, it suffices to show that $\mathit{Lab}(\pi) \otimes \nu \models \psi$. To this purpose, we show by structural induction that for each $j \geq 0$ and subformula θ of ψ if $(\mathcal{T}, \pi, j) \models f(\theta)$, then $(\mathit{Lab}(\pi) \otimes \nu, j) \models \theta$. Since the formula ψ is in negation normal form, by the induction hypothesis, the unique non-trivial case is when θ is either of the form p_{A} or of the form $\neg p_{\mathrm{A}}$ for some $\mathrm{A} \in \Upsilon_{\mathrm{Q}_\ell}$.

- $\theta = p_{\mathrm{A}}$: hence, $f(\theta) = \varphi_{\mathrm{A}}$. Since $(\mathcal{T}, \pi, j) \models f(\theta)$ and $(\mathcal{T}, \pi(j)) \models \varphi_{\mathrm{A}_j}$, by Claim 1, it follows that $\mathrm{A} = \mathrm{A}_j$, i.e. $\theta = p_{\mathrm{A}_j}$. Hence, $(\mathit{Lab}(\pi) \otimes \nu, j) \models \theta$, and the result follows.
- $\theta = \neg p_{\mathrm{A}}$: hence $f(\theta) = \bigvee_{\mathrm{A}' \in \Upsilon_{\mathrm{Q}_\ell} \setminus \{\mathrm{A}\}} \varphi_{\mathrm{A}'}$. Since $(\mathcal{T}, \pi, j) \models f(\theta)$ and $(\mathcal{T}, \pi(j)) \models \varphi_{\mathrm{A}_j}$, by Claim 1, $\mathrm{A} \neq \mathrm{A}_j$. Hence, $(\mathit{Lab}(\pi) \otimes \nu, j) \models \theta$, and we are done. ◀

From Logics to Automata. As to the translation from CCTL* to HGTA_{cf}, in order to ensure the mutual-exclusion property of the resulting HGTA_{cf}, we need a restricted syntactic form of CCTL* formulae, which is still expressively complete. A CCTL* formula is in *simple form* if each occurrence of the path quantifier $\mathtt{E}$ is immediately preceded by the counter modality $\mathtt{D}^1$ (note that $\mathtt{D}^1$ corresponds to the standard $\mathtt{EX}$ modality of CTL*). Formally, the set of state formulae φ of CCTL* in simple form is defined according to the following syntax: $\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathtt{D}^1\mathtt{E}\psi \mid \mathtt{D}^n\varphi$. One can easily show that the simple form is indeed expressively complete.

▶ **Theorem 5.9.** *Given a* CCDL *(resp.,* CCTL**) formula φ, one can construct an equivalent* HGTA *(resp.,* HGTA_{cf}*) $\mathcal{A}_\varphi$ such that $\mathsf{L}(\mathcal{A}_\varphi) = \mathsf{L}(\varphi)$. Moreover, $\mathcal{A}_\varphi$ is symmetric if φ is a* CDL *(resp., a* CTL**) formula.*

Proof. We focus on the translation from CCTL* to HGTA_{cf}. Fix a CCTL* formula Φ. W.l.o.g., we can assume that Φ is in simple form. As in the case of the alternating hesitant automata for CTL* [47], we construct the automaton by induction on the structure of Φ. With each state subformula φ of Φ we associate an HGTA_{cf} $\mathcal{A}_\varphi$ over $\Sigma = 2^{\text{AP}}$ such that $\mathsf{L}(\mathcal{A}_\varphi) = \mathsf{L}(\varphi)$. The cases where φ is an atomic proposition, or the root operator of φ is the counting modality $\mathtt{D}^n$ are straightforward. The cases where the root operator of φ is a Boolean connective directly follow from Proposition 5.6. Now, assume that $\varphi = \mathtt{E}\psi$ for some path formula ψ. Let $\mathsf{max}(\psi)$ be the set of state subformulae of ψ of the form $\mathtt{E}\xi$ or $\mathtt{D}^n\xi$ which are not preceded by the modality $\mathtt{E}$ or the counting modality in the syntax tree of ψ. Since ψ is in simple form, $\mathsf{max}(\psi)$ is of the form $\{\mathtt{D}^{n_1}\varphi_1, \ldots, \mathtt{D}^{n_k}\varphi_k\}$ for some $k \geq 0$, where $\varphi_1, \ldots, \varphi_k$ are CCTL* formulae in simple form. Note that if ψ is a CTL* formula, then $n_1 = \ldots = n_k = 1$. By the induction hypothesis, for each $i \in [1, k]$, one can construct an HGTA_{cf} $\mathcal{A}_i = \langle 2^{\text{AP}}, \text{Q}_i, \delta_i, q_{I\,i}, \mathsf{H}_i, \mathsf{H}_{\exists,i}, \Omega_i \rangle$ such that $\mathsf{L}(\mathcal{A}_i) = \mathsf{L}(\varphi_i)$. For each $i \in [1, k]$, let $\widetilde{\mathcal{A}}_i = \left\langle 2^{\text{AP}}, \widetilde{\text{Q}}_i, \widetilde{\delta}_i, \widetilde{q_I}_i, \widetilde{\mathsf{H}}_i, \widetilde{\mathsf{H}}_{\exists,i}, \widetilde{\Omega}_i \right\rangle$ be a renaming of the dual automaton of $\mathcal{A}_i$.

Let AP_{ex} be an extension of AP obtained by adding for each state formula $\mathtt{D}^{n_i}\varphi_i$ a fresh proposition p_i. Then, the path formula ψ can be viewed as an LTL formula ψ_{ex} over AP_{ex}. By [19], one can build a Büchi *counter-free* NWA $\mathcal{N}_\psi = \langle 2^{\text{AP}_{\mathsf{ex}}}, \text{Q}, \delta, q_I, \Omega \rangle$ s.t. $\mathsf{L}(\mathcal{N}_\psi) = \mathsf{L}(\psi_{\mathsf{ex}})$. By construction, we easily deduce the following characterization of $\mathsf{L}(\varphi) = \mathsf{L}(\mathtt{E}\psi)$:

Claim 1: for each Kripke tree $\mathcal{T} = (\text{T}, \textit{Lab})$, $\mathcal{T} \in \mathsf{L}(\varphi)$ *iff* there exists an infinite path π of $\mathcal{T}$ from the root and an infinite word ρ over $2^{\text{AP}_{\mathsf{ex}}}$ such that $\rho \in \mathsf{L}(\mathcal{N}_\psi)$ and the following holds for each $i \geq 0$: (i) $\rho(i) \cap \text{AP} = \textit{Lab}(\pi(i))$, (ii) for each $\ell \in [1, k]$ such that $p_\ell \in \rho(i)$, $(\mathcal{T}, \pi(i)) \models \mathtt{D}^{n_\ell}\varphi_\ell$, and (iii) for each $\ell \in [1, k]$ such that $p_\ell \notin \rho(i)$, $(\mathcal{T}, \pi(i)) \models \neg\mathtt{D}^{n_\ell}\varphi_\ell$.

We define $\mathcal{A}_\varphi$ as follows: $\mathcal{A}_\varphi$ simulates the Büchi NWA $\mathcal{N}_\psi$ along a guessed infinite path of the input tree from the root and starts additional copies of the HGTA_{cf} $\mathcal{A}_1, \ldots, \mathcal{A}_k, \widetilde{\mathcal{A}}_1, \ldots, \widetilde{\mathcal{A}}_k$. According to Claim 1, these copies guarantee that whenever the NWA $\mathcal{N}_\psi$ assumes that proposition p_ℓ labels (resp., p_ℓ does not label) the current node along the guessed path, then $\mathtt{D}^{n_\ell}\varphi_\ell$ holds (resp., $\mathtt{D}^{n_\ell}\varphi_\ell$ does not hold) at this node. The components of $\mathcal{A}$ consist of the existential component Q (the set of states of the Büchi counter-free NWA $\mathcal{N}_\psi$) and the components of the HGTA_{cf} automata $\mathcal{A}_i$ and $\widetilde{\mathcal{A}}_i$ for each $i \in [1, k]$. Moreover, the existential component Q has highest order and the ordering of the components of $\mathcal{A}_i$ (resp., $\widetilde{\mathcal{A}}_i$) is preserved for each $i \in [1, k]$. For the transition function δ_φ of $\mathcal{A}_\varphi$, we have that for states in Q_i (resp., $\widetilde{\text{Q}}_i$), δ_φ agrees with the corresponding δ_i (resp., $\widetilde{\delta}_i$). For states $q \in \text{Q}$ and $a \in 2^{\text{AP}}$, $\delta_\varphi(q, a)$ is defined as follows, where for each $\text{I} \subseteq [1, k]$, $\text{I}(a)$ denotes the subset of AP_{ex} given by $a \cup \bigcup_{\ell \in \text{I}} \{p_\ell\}$:

$$\delta_\varphi(q,a) \triangleq \bigvee_{\mathrm{I}\subseteq[1,k]} \bigvee_{q'\in\delta(q,\mathrm{I}(a))} (\Diamond q' \wedge \bigwedge_{\ell\in\mathrm{I}} \Diamond_\ell q_{I_i} \wedge \bigwedge_{\ell\in[1,k]\setminus\mathrm{I}} \Box_\ell \widetilde{q}_{I_i})$$

By construction, the induction hypothesis, and Claim 1, $\mathcal{A}_\varphi$ is an HGTA satisfying the mutual-exclusion property such that $\mathsf{L}(\mathcal{A}_\varphi) = \mathsf{L}(\varphi)$. It remains to show that for each $q \in \mathrm{Q}$, the Büchi NWA $\mathcal{A}_{\mathrm{Q},q}$ over the alphabet $2^{\mathrm{AP}} \times \Upsilon_{\mathrm{Q}}$ (see Definition 5.2) driven by the existential component Q of $\mathcal{A}_\varphi$ is counter-free. Let us consider the mapping g assigning to each $a_{\mathsf{ex}} \in 2^{\mathrm{AP}_{\mathsf{ex}}}$ the pair $(a, \bigcup_{\ell\in\mathrm{I}}\{\Diamond_\ell q_{I_i}\} \cup \bigcup_{\ell\in[1,k]\setminus\mathrm{I}}\{\Box_\ell \widetilde{q}_{I_i}\})$, where $a = \mathrm{AP} \cap a_{\mathsf{ex}}$ and I is the set of indexes in $j \in [1,k]$ such that $p_j \in a_{\mathsf{ex}}$. Clearly, g is a bijection between $2^{\mathrm{AP}_{\mathsf{ex}}}$ and $2^{\mathrm{AP}} \times \Upsilon_{\mathrm{Q}}$. Moreover, for the transition functions δ_{Q} and δ of $\mathcal{A}_{\mathrm{Q},q}$ and $\mathcal{N}_\psi$, respectively, it holds that, for each $(a, \mathrm{A}) \in 2^{\mathrm{AP}} \times \Upsilon_{\mathrm{Q}}$ and $q' \in \mathrm{Q}$, $\delta_{\mathrm{Q}}(q', (a, \mathrm{A})) = \delta(q', g^{-1}(a, \mathrm{A}))$, where g^{-1} is the inverse of g. Thus, since $\mathcal{N}_\psi$ is counter free, $\mathcal{A}_{\mathrm{Q},q}$ is counter free as well, and the result follows. ◀

6 Automata Characterisation of Monadic Chain Logic (MCL)

Monadic Chain Logic (MCL) is the fragment of MSO over Kripke trees where monadic second-order quantification is restricted to sets of nodes which forms chains, *i.e.* a subset of a path. In this section, we provide an automata-theoretic characterisation of MCL in terms of a subclass of parity FTA, called *Hesitant* FTA (HFTA for short), which represents the FTA counterpart of hesitant GTA. Moreover, we show that the bisimulation-invariant fragment of MCL and CDL are expressively equivalent.

The class of HFTA. An HFTA is a tuple $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$, where $\langle \Sigma, \mathrm{Q}, \delta, q_I, \Omega \rangle$ is an FTA and H and $\mathsf{H}_\exists$ are defined as for HGTA. Moreover, we require that $\mathcal{A}$ satisfies the transient requirement and the hesitant acceptance requirement of HGTA and the following variants of the existential and universal requirements of HGTA:

- for each existential component Q_i and $(q,a) \in \mathrm{Q}_i \times \Sigma$, $\delta(q,a)$ is a disjunction of formulae of the form $\exists x.\, (q'(x) \wedge \theta(x))$ where $q' \in \mathrm{Q}_i$ and $\theta(x)$ only refers to states in lower components Q_j with $j < i$ (*first-order existential requirement*);
- for each universal component Q_i and $(q,a) \in \mathrm{Q}_i \times \Sigma$, $\delta(q,a)$ is a conjunction of formulae of the form $\forall x.\, (q'(x) \vee \theta(x))$ where $q' \in \mathrm{Q}_i$ and $\theta(x)$ only refers to states in lower components Q_j with $j < i$ (*first-order universal requirement*).

HFTA can be easily translated into equivalent MCL sentences.

▶ **Theorem 6.1.** *Given an* HFTA $\mathcal{A}$ *over* 2^{AP}*, one can construct in polynomial time an* MCL *sentence* $\varphi_{\mathcal{A}}$ *over* AP *such that* $\mathsf{L}(\varphi_{\mathcal{A}}) = \mathsf{L}(\mathcal{A})$*.*

Chain Projection. Like HGTA, the tree-languages accepted by HFTA are closed under Boolean operations. Thus, in the translation of MCL formulae into equivalent parity HFTA, the only non-trivial part concerns the treatment of MCL existential quantification. For this purpose, we define an operation on tree languages that captures the semantics of MCL existential quantification. Let L be a tree language over 2^{AP} and $p \in \mathrm{AP}$. The *chain projection of* L *over* p, denoted by $\exists^{\mathsf{c}} p.\mathsf{L}$, is the language consisting of the Kripke trees $(\mathrm{T}, \mathit{Lab})$ over $\mathrm{AP} \setminus \{p\}$ such that there is an infinite path π of T from the root and a Kripke tree $(\mathrm{T}, \mathit{Lab}') \in \mathsf{L}$ so that: $\mathit{Lab}'(w) = \mathit{Lab}(w)$, for each $w \in \mathrm{T} \setminus \pi$, and $\mathit{Lab}'(w) \setminus \{p\} = \mathit{Lab}(w)$, otherwise.

We show that HFTA are effectively closed under chain projection, i.e., for each HFTA $\mathcal{A}$ over 2^{AP} and $p \in \mathrm{AP}$, one can construct an HFTA accepting $\exists^{\mathsf{C}} p.\mathsf{L}(\mathcal{A})$. The proof is divided in two steps. In the first step, we define a subclass of HFTA, called HFTA that *are nondeterministic in one path* (see Definition 6.3), whose closure under chain projection can be easily established (see Proposition 6.4). Then, in the second step, we show that an HFTA can be converted into an equivalent HFTA that is nondeterministic in one path.

We now introduce this subclass of automata. By exploiting the known notion of *basic formula* [82, 12], we first define a fragment of the one-step language $\mathrm{FOE}_1^+(\mathrm{Q})$ for a given non-empty set Q. A Q-*type* is a (possibly empty) set $\mathrm{A} \subseteq \mathrm{Q}$. It defines the first-order constraint $\mathtt{t}(\mathrm{A})(x) \triangleq \bigwedge_{q \in \mathrm{A}} q(x)$. Note that $\mathtt{t}(\mathrm{A})(x)$ is $\top$ if A is empty. Let $\mathtt{T}_\exists$ and $\mathtt{T}_\forall$ be two sets of Q-type. The *basic formula for the pair* $(\mathtt{T}_\exists, \mathtt{T}_\forall)$, denoted $\theta^{=}(\mathtt{T}_\exists, \mathtt{T}_\forall)$, is the $\mathrm{FOE}_1^+(\mathrm{Q})$ sentence defined as follows, where $\mathtt{T}_\exists = \{\mathrm{A}_1, \ldots, \mathrm{A}_n\}$ for some $n \geq 0$ and for variables $z_1, \ldots, z_k$, $\mathtt{diff}(z_1, \ldots, z_k) \triangleq \bigwedge_{i \neq j} z_i \neq z_j$:

$$\exists x_1 \ldots \exists x_n. \Big(\mathtt{diff}(x_1, \ldots, x_n) \wedge \bigwedge_{i=1}^{n} \mathtt{t}(\mathrm{A}_i)(x_i) \wedge \forall y. (\mathtt{diff}(x_1, \ldots, x_n, y) \rightarrow \bigvee_{\mathrm{A} \in \mathtt{T}_\forall} \mathtt{t}(\mathrm{A})(y))\Big).$$

Intuitively, $\theta^{=}(\mathtt{T}_\exists, \mathtt{T}_\forall)$ asserts that there are n-distinct elements $s_1, \ldots, s_n$ of the given domain S such that each s_i satisfies the Q-type A_i of the existential part $\mathtt{T}_\exists$, and every other element of the domain satisfies some Q-type in the universal part $\mathtt{T}_\forall$.

▶ **Definition 6.2.** *Let* $\mathrm{Q}' \subseteq \mathrm{Q}$ *with* $\mathrm{Q}' \neq \emptyset$. *A basic formula* $\theta^{=}(\mathtt{T}_\exists, \mathtt{T}_\forall)$ *is* Q′-functional in one direction *if there exists* $\mathrm{A} \in \mathtt{T}_\exists$ *such that* A *is a singleton consisting of an element in* Q′, *and for each* $\mathrm{B} \in (\mathtt{T}_\exists \setminus \{\mathrm{A}\}) \cup \mathtt{T}_\forall$, B *does not contain elements in* Q′. *A first-order* Q-*constraint is* Q′-functional in one direction *if it is the disjunction of basic formulae which are* Q′*-functional in one direction.*

Intuitively, when the local behaviour of an HFTA $\mathcal{A}$ at the current input node w is driven by a constraint θ that is Q′-functional in one direction, then there is a child w' of w such that exactly one copy of $\mathcal{A}$ is sent to w'. Moreover, the state of this copy is in Q′ and the states of the copies sent to the children of w distinct from w' are in $\mathrm{Q} \setminus \mathrm{Q}'$.

▶ **Definition 6.3.** *An* HFTA $\mathcal{A} = \langle \Sigma, \mathsf{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$ *is* nondeterministic in one path *if the initial state* q_I *belongs to some existential component* Q_ℓ *of* $\mathcal{A}$ *and the following hold:*

1. *for each* $q \in \mathrm{Q}_\ell$ *and* $a \in \Sigma$, $\delta(q, a)$ *is* Q_ℓ*-functional in one direction;*
2. *for each* $\mathcal{T} \in \mathsf{L}(\mathcal{A})$ *and for each infinite path* π *of* $\mathcal{T}$ *from the root, there is an accepting run* $r = (\mathrm{T}_r, Lab_r)$ *of* $\mathcal{A}$ *over* $\mathcal{T}$ *s.t. for each input node* $w \in \pi$, *there is exactly one node* y *of* r *reading* w, *i.e., such that* $Lab_r(y) = (q, w)$ *for some state* q*; moreover,* $q \in \mathrm{Q}_\ell$.

Let $\Sigma = 2^{\mathrm{AP}}$, $p \in \mathrm{AP}$, $\mathcal{A} = \langle \Sigma, \mathsf{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$ be an HFTA that is nondeterministic in one path, and Q_ℓ be the existential component such that $q_I \in \mathrm{Q}_\ell$. We consider the HFTA $\exists^{\mathsf{C}} p.\mathcal{A} = \langle 2^{\mathrm{AP} \setminus \{p\}}, \mathsf{Q}, \delta', q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega \rangle$, where the transition function δ' is defined as follows for all $q \in \mathrm{Q}$ and $a \in 2^{\mathrm{AP} \setminus \{p\}}$: $\delta'(q, a) = \delta(q, a)$ if $q \notin \mathrm{Q}_\ell$, and $\delta'(q, a) = \delta(q, a) \vee \delta(q, a \cup \{p\})$ otherwise. Hence, on all the states which are not in the existential component Q_ℓ, $\exists^{\mathsf{C}} p.\mathcal{A}$ behaves as $\mathcal{A}$. On the states in Q_ℓ, the projection automaton guesses whether in the simulated run of $\mathcal{A}$, proposition p marks the current input node or not, and proceeds according to the guess and the transition function of $\mathcal{A}$. By Definition 6.3, we easily obtain the following result.

▶ **Proposition 6.4.** *Let* $\mathcal{A}$ *be an* HFTA *over* 2^{AP} *that is nondeterministic in one path and* $p \in \mathrm{AP}$. *Then,* $\mathsf{L}(\exists^{\mathsf{C}} p.\mathcal{A}) = \exists^{\mathsf{C}} p.\mathsf{L}(\mathcal{A})$.

From HFTA to HFTA that are nondeterministic in one path. We now show that HFTA can be effectively translated into equivalent HFTA that are nondeterministic in one path. We first establish a preliminary result on the one-step logic $\mathrm{FOE}_1^+(\mathrm{Q})$ for a given non-empty set Q.

▶ **Definition 6.5.** *Let θ be a first-order* Q*-constraint and θ_s be a first-order* $(\mathrm{Q}\cup 2^{\mathrm{Q}})$*-constraint which is* 2^{Q}*-functional in one direction. We say that θ_s simulates θ if the following hold:*

- *for each minimal model* (S, I) *of θ and for each $s \in \mathrm{S}$,* $(\mathrm{S}, \mathrm{I}[s \to \{\mathrm{I}(s)\}])$ *is a model of θ_s;*
- *for each minimal model* (S, I) *of θ_s, let $s \in \mathrm{S}$ be the unique element in* S *such that* $\mathrm{I}(s)$ *is of the form* $\{\mathrm{Q}'\}$ *for some* $\mathrm{Q}' \in 2^{\mathrm{Q}}$*. Then, the pair* $(\mathrm{S}, \mathrm{I}[s \to \mathrm{Q}'])$ *is a model of θ;*

where the mappings $\mathrm{I}[s \to \{\mathrm{I}(s)\}]$ *and* $\mathrm{I}[s \to \mathrm{Q}']$ *are defined in the obvious way.*

Since each first-order Q-constraint is effectively equivalent to a disjunction of basic formulae [12], we easily obtain the following result.

▶ **Proposition 6.6.** *Let θ be a first-order* Q*-constraint. Then, one can construct a first-order* $(\mathrm{Q} \cup 2^{\mathrm{Q}})$*-constraint θ_s which is* 2^{Q}*-functional in one direction and simulates θ.*

Fix an HFTA $\mathcal{A} = \langle \Sigma, \mathrm{Q}, \delta, q_I, \mathsf{H}, \mathsf{H}_\exists, \Omega\rangle$ with $\mathsf{H} = \langle \mathrm{Q}_1, \ldots, \mathrm{Q}_n\rangle$. We construct in two stages an equivalent HFTA $\mathtt{Sim}(\mathcal{A})$ that is nondeterministic in one path. First, by a kind of powerset construction, we construct an automaton $\mathcal{A}_{\mathtt{PATH}}$ that is nondeterministic in one path but the acceptance condition of the existential component $Pow_{\mathcal{A}}$ containing the initial state is not a Büchi condition but an ω-regular set over the infinite sequences on $Pow_{\mathcal{A}}$. In the second stage of the construction, we show how the ω-regular condition can be converted into a standard Büchi condition by equipping the "macro" states in $Pow_{\mathcal{A}}$ with additional information. Intuitively, given an input tree (T, Lab) accepted by $\mathcal{A}$, the automaton $\mathcal{A}_{\mathtt{PATH}}$ simulates the behaviour of $\mathcal{A}$ along an accepting run r over (T, Lab) by guessing an infinite path π of the input tree from the root and proceeding as follows:

- in the input nodes $w \notin \pi$, $\mathcal{A}_{\mathtt{PATH}}$ simply simulates the behaviour of $\mathcal{A}$ along r;
- in the input nodes $w \in \pi$, $\mathcal{A}_{\mathtt{PATH}}$ keeps track in its "macro" state (a state in the existential component $Pow_{\mathcal{A}}$) of the states of $\mathcal{A}$ associated with the copies of $\mathcal{A}$ that read w along r. Thus, in the run of $\mathcal{A}_{\mathtt{PATH}}$, there is a unique infinite path ν from the root associated with the guessed input path π, and ν "collects" the set of parallel paths ν_r of the simulated run of $\mathcal{A}$ associated with the input path π. In order to check the acceptance condition on the individual parallel paths ν_r, an infinite sequence of "macro" states ρ must allow to distinguish the individual infinite paths over Q grouped by ρ. Thus, like in [82], a "macro" state associated with an input node w is a set of pairs (q_p, q): the pair (q_p, q) represents a copy of $\mathcal{A}$ in state q at node w along the simulated run r which has been generated by a copy of $\mathcal{A}$ in state q_p reading the parent node of w in the input tree.

Formally, we denote by $Pow_{\mathcal{A}}$ the subset of $2^{\mathrm{Q}\times\mathrm{Q}}$ consisting of the sets of pairs (q, q') of $\mathcal{A}$-states such that the order of q' is equal or lower than the order of q (*order requirement*). A $Pow_{\mathcal{A}}$*-path* ν is an infinite word $\nu = \mathrm{P}_0\mathrm{P}_1\ldots$ over $Pow_{\mathcal{A}}$ such that the following conditions are fulfilled: (i) $\mathrm{P}_0 = \{(q_I, q_I)\}$ (*initialisation*), and (ii) for all $i \geq 0$ and $(q_i, q_{i+1}) \in \mathrm{P}_{i+1}$, there is an element of P_i of the form (q_{i-1}, q_i) (*consecution*). An $\mathcal{A}$*-path* of ν is a maximal (possibly finite) non-empty sequence $q_0q_1\ldots$ of $\mathcal{A}$-states such that $(q_{i-1}, q_i) \in \mathrm{P}_i$ for all $1 \leq i < |\nu|$. The $Pow_{\mathcal{A}}$-path ν is $\mathcal{A}$-accepting if all infinite $\mathcal{A}$-paths of ν satisfy the parity condition Ω of $\mathcal{A}$. The automaton $\mathcal{A}_{\mathtt{PATH}}$ is then given by $\mathcal{A}_{\mathtt{PATH}} = \langle \Sigma, \mathrm{Q} \cup Pow_{\mathcal{A}}, \delta_{\mathtt{PATH}}, \{(q_I, q_I)\}, \mathsf{H}_{\mathtt{PATH}}, \mathsf{H}_\exists \cup \{Pow_{\mathcal{A}}\}, \Omega\rangle$ where $\mathsf{H}_{\mathtt{PATH}} = \langle \mathrm{Q}_1, \ldots, \mathrm{Q}_n, Pow_{\mathcal{A}}\rangle$ (the existential component $Pow_{\mathcal{A}}$ has highest order) and $\delta_{\mathtt{PATH}}$ is defined as follows:

- for all $q \in \mathrm{Q}$ and $a \in \Sigma$, $\delta_{\mathtt{PATH}}(q, a) = \delta(q, a)$;

- for all $\mathrm{P} \in Pow_{\mathcal{A}}$ and $a \in \Sigma$, if P is empty, then $\delta_{\mathtt{PATH}}(\mathrm{P}, a) = \exists x. \, \mathrm{P}(x)$; otherwise, let us consider the first-order $(\mathrm{Q} \times \mathrm{Q})$-constraint θ given by $\bigwedge_{(q_p,q) \in \mathrm{P}} \delta_q(q, a)$, where $\delta_q(q, a)$ is obtained from $\delta(q, a)$ by replacing each predicate $q'(y)$ occurring in $\delta(q, a)$ with $(q, q')(y)$. By Proposition 6.6, one can construct a first-order $((\mathrm{Q} \times \mathrm{Q}) \cup Pow_{\mathcal{A}})$-constraint θ_s which is $Pow_{\mathcal{A}}$-functional in one direction and simulates θ. Then, $\delta_{\mathtt{PATH}}(\mathrm{P}, a)$ is obtained from θ_s by replacing each predicate $(q, q')(y)$ occurring in θ_s associated with an element of $\mathrm{Q} \times \mathrm{Q}$ with q'. Note that $\delta_{\mathtt{PATH}}(\mathrm{P}, a)$ satisfies the first-order existential requirement and is $Pow_{\mathcal{A}}$-functional in one direction.

Note that in the definition of $\mathcal{A}_{\mathtt{PATH}}$, no acceptance condition is defined for the macro states in $Pow_{\mathcal{A}}$ (the parity condition Ω inherited by $\mathcal{A}$ is defined only on the states in Q). By construction and Proposition 6.6, for each run r of $\mathcal{A}_{\mathtt{PATH}}$ over an input (T, Lab) and every infinite path π of r starting at the root, either π is associated with a $Pow_{\mathcal{A}}$-path ν (in this case, we say that π is accepting if ν is accepting) or π gets trapped into some non-transient component of $\mathcal{A}$ (in this case, acceptance of π is determined by the parity condition Ω). We denote by $\mathsf{L}(\mathcal{A}_{\mathtt{PATH}})$ the set of input trees (T, Lab) such that there is a run of $\mathcal{A}_{\mathtt{PATH}}$ over (T, Lab) whose infinite paths starting at the root are all accepting. By construction and Proposition 6.6, we easily deduce the following crucial result.

▶ **Lemma 6.7.** *$\mathcal{A}_{\mathtt{PATH}}$ is nondeterministic in one path and $\mathsf{L}(\mathcal{A}_{\mathtt{PATH}}) = \mathsf{L}(\mathcal{A})$.*

Construction of the Automaton $\mathtt{Sim}(\mathcal{A})$. Let F_{B} (resp., $\mathrm{F}_{\mathsf{coB}}$) be the set of states in the existential (resp., universal) components of $\mathcal{A}$ having even (resp., odd) color. Fix a $Pow_{\mathcal{A}}$-path ν. By the order requirement, each infinite $\mathcal{A}$-path of ν gets trapped into an existential or universal component of $\mathcal{A}$. Thus, by the hesitant acceptance requirement of HFTA, the $Pow_{\mathcal{A}}$-path ν is $\mathcal{A}$-accepting if and only if for each infinite $\mathcal{A}$-*path* π of ν, the following holds: if π gets trapped into an existential component, then π visits *infinitely* many times some state in F_{B} (*Büchi condition*); otherwise (i.e., π gets trapped into an universal component), π visits *finitely* many times all the states in $\mathrm{F}_{\mathsf{coB}}$ (*coBüchi condition*).

It is known that coBüchi alternating word automata (AWA) over infinite words can be converted in quadratic time into equivalent Büchi AWA by means of the so called *ranking construction* [46]. We adapt the ranking construction and the Miyano-Hayashi construction [53] (for converting a Büchi AWA into an equivalent Büchi NWA) for providing a characterisation of acceptance of $Pow_{\mathcal{A}}$-paths ν by a classical Büchi condition on an extension of ν obtained by adding to each macro-state visited by ν additional finite-state information. Hence, we obtain the following result.

▶ **Theorem 6.8.** *For the given* HFTA *$\mathcal{A}$, one can construct an* HFTA $\mathtt{Sim}(\mathcal{A})$ *that is nondeterministic in one path and such that* $\mathsf{L}(\mathtt{Sim}(\mathcal{A})) = \mathsf{L}(\mathcal{A})$.

By Theorem 6.8 and Proposition 6.4, we obtain the following result.

▶ **Corollary 6.9.** *The class of* HFTA *is effectively closed under chain projection.*

An HFTA with transition function δ is in *normal form* if over existential (resp., universal) components Q_ℓ, $\delta(q, a)$ (resp., the dual of $\delta(q, a)$) is Q_ℓ-functional in one direction for all $q \in \mathrm{Q}_\ell$ and $a \in \Sigma$. Since the constructions for the Boolean language operations and the construction for the closure under chain projection (Theorem 6.8 and Proposition 6.4) preserve the normal form, we deduce the following result.

▶ **Theorem 6.10.** *Given an* MCL *sentence φ, one can construct an* HFTA *$\mathcal{A}_\varphi$ in normal form such that* $\mathsf{L}(\mathcal{A}_\varphi) = \mathsf{L}(\varphi)$.

We exploit the normal form for showing that CDL (or, equivalently, the class of symmetric HGTA) provides a characterisation of the bisimulation-fragment of MCL. It is known [82, 12] that for each FTA $\mathcal{A}$, one can construct a symmetric FTA $\mathcal{A}_S$ such that if $\mathsf{L}(\mathcal{A})$ is bisimulation-closed, then $\mathcal{A}$ and $\mathcal{A}_S$ accept the same tree-language. By adapting the construction given in [82, 12], we can show that a similar result holds for HFTA in normal form versus symmetric HGTA. Hence, by Theorems 5.8 and 5.9 and Theorem 6.10, we deduce the following result.

▶ **Theorem 6.11.** *The bisimulation-invariant fragment of* MCL, CDL, *and the class of symmetric* HGTA *are expressively equivalent in a constructive way.*

7 Conclusion

This work provides automata-theoretic characterisations of branching-time temporal logics, mainly focusing on CTL* and CDL, the latter being a syntactic variant of the already known ECTL*. Specifically, we prove the equivalence between the symmetric variant of classic ranked Hesitant Tree Automata (HTA) and both CDL and the bisimulation-invariant fragment of Monadic Chain Logic (MCL). The full MCL, instead, is proved equivalent to a first-order variant of HTAs. In addition, we close a longstanding gap in the expressiveness landscape of branching-time logics, by providing an automata-theoretic characterisation of CTL*. This is obtained via a generalisation to tree-languages of the notion of counter-freeness, originally introduced in the context of word languages. The generalisation essentially decomposes an HTA into a number of counter-free word automata, one for each level of the state decomposition of the HTA. This decomposition, however, works correctly only when the HTA satisfies the additional property of mutual-exclusion. The property requires that different sets of automaton states, active at the same time on a given node of the input tree, must accept different subtrees. Both mutual-exclusion and counter-freeness seem to be essential to capture a meaningful notion of counter-freeness for tree automata. Together these results bring the expressiveness landscape for branching-time temporal logics to almost the same level as their linear-time counterparts.

There are few open questions remaining. In particular, while Theorem 6.11 establishes the equivalence between the bisimulation invariant fragment of MCL and CDL, the precise relationship between CCDL (hence, ECTL*) and full MCL still remains unsettled. In addition, techniques similar to those used in this work may also be applicable to obtain a characterisation of Monadic Tree Logic (MTL), a fragment of MSO where quantified variables range over subtrees [3], and of Substructure Temporal Logic (STL), a temporal logic where one can implicitly predicate over substructure/subtrees [4, 5]. The restriction that variables range over trees, indeed, seem to be tightly connected with the notion of counter-freeness. The difficulty in this case is that counter-free HTAs would not suffice, since both MTL and STL are strictly more expressive than CTL*, and a meaningful definition of decomposition into word automata of a non-hesitant tree automaton is not immediately obvious.

References

1 A. Arnold and D. Niwiński. Fixed Point Characterization of Weak Monadic Logic Definable Sets of Trees. In *Tree Automata and Languages*, pages 159–188. North-Holland, 1992.

2 C. Baier and J.-P. Katoen. *Principles of Model Checking.* MIT Press, 2008.

3 M. Benerecetti, L. Bozzelli, F. Mogavero, and A. Peron. Quantifying over Trees in Monadic Second-Order Logic. In *LICS'23*, pages 1–13. IEEECS, 2023.

4 M. Benerecetti, F. Mogavero, and A. Murano. Substructure Temporal Logic. In *LICS'13*, pages 368–377. IEEECS, 2013.

5 M. Benerecetti, F. Mogavero, and A. Murano. Reasoning About Substructures and Games. *TOCL*, 16(3):25:1–46, 2015.

6 M. Bojańczyk. The Finite Graph Problem for Two-Way Alternating Automata. *TCS*, 3(298):511–528, 2003.

7 U. Boker and Y. Shaulian. Automaton-Based Criteria for Membership in CTL. In *LICS'18*, pages 155–164. ACM, 2018.

8 J.R. Büchi. Weak Second-Order Arithmetic and Finite Automata. *MLQ*, 6(1-6):66–92, 1960.

9 J.R. Büchi. On a Decision Method in Restricted Second-Order Arithmetic. In *ICLMPS'62*, pages 1–11. Stanford University Press, 1962.

10 J.R. Büchi. On a Decision Method in Restricted Second Order Arithmetic. In *Studies in Logic and the Foundations of Mathematics*, volume 44, pages 1–11. Elsevier, 1966.

11 F. Carreiro, A. Facchini, Y. Venema, and F. Zanasi. Weak MSO: Automata and Expressiveness Modulo Bisimilarity. In *CSL'14 & LICS'14*, pages 27:1–27. ACM, 2014.

12 F. Carreiro, A. Facchini, Y. Venema, and F. Zanasi. The Power of the Weak. *TOCL*, 21(2):15:1–47, 2020.

13 F. Carreiro, A. Facchini, Y. Venema, and F. Zanasi. Model Theory of Monadic Predicate Logic with the Infinity Quantifier. *AML*, 61(3-4):465–502, 2022.

14 Y. Choueka. Theories of Automata on ω-Tapes: A Simplified Approach. *JCSS*, 8(2):117–141, 1974.

15 A. Church. Logic, Arithmetics, and Automata. In *ICM'62*, pages 23–35, 1963.

16 E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications: A Practical Approach. In *POPL'83*, pages 117–126. ACM, 1983.

17 E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *TOPLAS*, 8(2):244–263, 1986.

18 E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* MIT Press, 2002.

19 V. Diekert and 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*, pages 261–306. Amsterdam University Press, 2008.

20 C.C. Elgot. Decision Problems of Finite Automata Design and Related Arithmetics. *TAMS*, 98:21–51, 1961.

21 E.A. Emerson and E.M. Clarke. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer, 1982.

22 E.A. Emerson and E.M. Clarke. Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons. *SCP*, 2(3):241–266, 1982.

23 E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time. In *POPL'83*, pages 127–140. ACM, 1983.

24 E.A. Emerson and J.Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *JCSS*, 30(1):1–24, 1985.

25 E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.

26 E.A. Emerson and C.S. Jutla. Tree Automata, muCalculus, and Determinacy. In *FOCS'91*, pages 368–377. IEEECS, 1991.

27 E.A. Emerson, C.S. Jutla, and A.P. Sistla. On Model Checking for the muCalculus and its Fragments. *TCS*, 258(1-2):491–522, 2001.

28 A. Facchini, Y. Venema, and F. Zanasi. A Characterization Theorem for the Alternation-Free Fragment of the Modal μ-Calculus. In *LICS'13*, pages 478–487. IEEECS, 2013.

29 K. Fine. In So Many Possible Worlds. *NDJFL*, 13:516–520, 1972.

30 M.J. Fischer and R.E. Ladner. Propositional Dynamic Logic of Regular Programs. *JCSS*, 18(2):194–211, 1979.

31 D.M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the Temporal Basis of Fairness. In *POPL'80*, pages 163–173. ACM, 1980.
32 G. De Giacomo and M.Y. Vardi. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *IJCAI'13*, pages 854–860. IJCAI' & AAAI Press, 2013.
33 Y. Gurevich and S. Shelah. The Decision Problem for Branching Time Logic. *JSL*, 50(3):668–681, 1985.
34 T. Hafer and W. Thomas. Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree. In *ICALP'87*, LNCS 267, pages 269–279. Springer, 1987.
35 D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
36 D. Janin. *A Contribution to Formal Methods: Games, Logic and Automata*. Habilitation thesis, Université Bordeaux I, Bordeaux, France, 2005.
37 D. Janin and G. Lenzi. On the Relationship Between Monadic and Weak Monadic Second Order Logic on Arbitrary Trees, with Applications to the mu-Calculus. *FI*, 61(3-4):247–265, 2004.
38 D. Janin and I. Walukiewicz. On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic. In *CONCUR'96*, LNCS 1119, pages 263–277. Springer, 1996.
39 H.W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, CA, USA, 1968.
40 R.M. Keller. Formal Verification of Parallel Programs. *CACM*, 19(7):371–384, 1976.
41 S.C. Kleene. Representation of Events in Nerve Nets and Finite Automata. In *Automata Studies*, pages 3–42. Princeton University Press, 1956.
42 D. Kozen. Results on the Propositional muCalculus. *TCS*, 27(3):333–354, 1983.
43 S.A. Kripke. Semantical Considerations on Modal Logic. *APF*, 16:83–94, 1963.
44 O. Kupferman, U. Sattler, and M.Y. Vardi. The Complexity of the Graded muCalculus. In *CADE'02*, LNCS 2392, pages 423–437. Springer, 2002.
45 O. Kupferman and M.Y. Vardi. Freedom, Weakness, and Determinism: From Linear-Time to Branching-Time. In *LICS'98*, pages 81–92. IEEECS, 1998.
46 O. Kupferman and M.Y. Vardi. Weak Alternating Automata are not That Weak. *TOCL*, 2(3):408–429, 2001.
47 O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.
48 R.E. Ladner. Application of Model Theoretic Games to Discrete Linear Orders and Finite Automata. *IC*, 33(4):281–303, 1977.
49 Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems - Specification*. Springer, 1992.
50 Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems - Safety*. Springer, 1995.
51 R. McNaughton. Testing and Generating Infinite Sequences by a Finite Automaton. *IC*, 9(5):521–530, 1966.
52 R. McNaughton and S. Papert. *Counter-Free Automata*. MIT Press, 1971.
53 S. Miyano and T. Hayashi. Alternating Finite Automata on ω-Words. *TCS*, 32(3):321–330, 1984.
54 F. Moller and A.M. Rabinovich. On the Expressive Power of CTL*. In *LICS'99*, pages 360–368. IEEECS, 1999.
55 F. Moller and A.M. Rabinovich. Counting on CTL*: On the Expressive Power of Monadic Path Logic. *IC*, 184(1):147–159, 2003.
56 E.F. Moore. Gedanken-Experiments on Sequential Machines. In *Automata Studies*, pages 129–154. Princeton University Press, 1956.
57 A. Nerode. Linear Automaton Transformations. *PAMS*, 9(4):541–544, 195.
58 D. Perrin. Recent Results on Automata and Infinite Words. In *MFCS'84*, LNCS 176, pages 134–148. Springer, 1984.
59 D. Perrin and J. Pin. First-Order Logic and Star-Free Sets. *JCSS*, 32(3):393–406, 1986.

60 D. Perrin and J. Pin. *Infinite Words*. Pure and Applied Mathematics. Elsevier, 2004.
61 A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57. IEEECS, 1977.
62 A. Pnueli. The Temporal Semantics of Concurrent Programs. *TCS*, 13:45–60, 1981.
63 A. Pnueli and R. Rosner. On the Synthesis of a Reactive Module. In *POPL'89*, pages 179–190. ACM, 1989.
64 M.O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *TAMS*, 141:1–35, 1969.
65 M.O. Rabin. Weakly Definable Relations and Special Automata. In *Studies in Logic and the Foundations of Mathematics*, volume 59, pages 1–23. Elsevier, 1970.
66 M.O. Rabin and D.S. Scott. Finite Automata and their Decision Problems. *IBMJRD*, 3:115–125, 1959.
67 A. Rabinovich. A Proof of Kamp's Theorem. In *CSL'12*, LIPIcs 16, pages 516–527. Leibniz-Zentrum fuer Informatik, 2012.
68 A. Rabinovich. A Proof of Kamp's Theorem. *LMCS*, 10(1):1–16, 2014.
69 R. Rosner. *Modular Synthesis of Reactive Systems*. PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1992.
70 M.P. Schützenberger. On Finite Monoids Having Only Trivial Subgroups. *IC*, 8(2):190–194, 1965.
71 W. Thomas. Star-Free Regular Sets of ω-Sequences. *IC*, 42(2):148–156, 1979.
72 W. Thomas. A Combinatorial Approach to the Theory of ω-Automata. *IC*, 48(3):261–283, 1981.
73 W. Thomas. Logical Aspects in the Study of Tree Languages. In *CAAP'84*, pages 31–50. CUP, 1984.
74 W. Thomas. On Chain Logic, Path Logic, and First-Order Logic over Infinite Trees. In *LICS'87*, pages 245–256. IEEECS, 1987.
75 W. Thomas. Automata on Infinite Objects. In *Handbook of Theoretical Computer Science (vol. B)*, pages 133–191. MIT Press, 1990.
76 B.A. Trakhtenbrot. Finite Automata and the Logic of One-Place Predicates. *AMST*, 59:23–55, 1966.
77 J. van Benthem. *Modal Correspondence Theory*. PhD thesis, University of Amsterdam, Amsterdam, Netherlands, 1977.
78 M.Y. Vardi. Reasoning about The Past with Two-Way Automata. In *ICALP'98*, LNCS 1443, pages 628–641. Springer, 1998.
79 M.Y. Vardi and L.J. Stockmeyer. Improved Upper and Lower Bounds for Modal Logics of Programs: Preliminary Report. In *STOC'85*, pages 240–251. ACM, 1985.
80 M.Y. Vardi and P. Wolper. Yet Another Process Logic. In *LP'83*, LNCS 164, pages 501–512. Springer, 1984.
81 M.Y. Vardi and P. Wolper. Automata-Theoretic Techniques for Modal Logics of Programs. *JCSS*, 32(2):183–221, 1986.
82 I. Walukiewicz. Monadic Second Order Logic on Tree-Like Structures. *TCS*, 275(1-2):311–346, 2002.
83 A. Weinert and M. Zimmermann. Visibly Linear Dynamic Logic. *TCS*, 747:100–117, 2018.
84 P. Wolper. Temporal Logic Can Be More Expressive. *IC*, 56(1-2):72–99, 1983.