

UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Quantifying over Trees in Monadic Second-Order Logic

This is the peer reviewed (post-print) version of the following article:

Original
Quantifying over Trees in Monadic Second-Order Logic / Benerecetti, M., Bozzelli, L., Mogavero, F., Peron, A.. - ELETTRONICO. - (2023), pp. 1-13. (ACM/IEEE Symposium on Logic in Computer Science, LICS 2023 Boston 26 Giugno 2023 -29 Giugno 2023) [10.1109/LICS56636.2023.10175832].

Availability:
This version is available at: 11368/3057819 since: 2023-09-11T12:31:14Z
Publisher:
IEEE

Published
DOI:10.1109/LICS56636.2023.10175832
Terms of use:

Some rights reserved. More details in the PostPrint rights section below.

License:
Digital Rights Management not defined

Publisher copyright

-.-

(Article begins on next page)

08 November 2026

arTs Università degli Studi di Trieste
Archivio della ricerca – postprint

Quantifying over Trees in Monadic Second-Order Logic

Massimo Benerecetti, Laura Bozzelli, Fabio Mogavero
Università degli Studi di Napoli Federico II
emails: {massimo.benerecetti,laura.bozzelli,fabio.mogavero}@unina.it

Adriano Peron
Università degli Studi di Trieste
email: adriano.peron@units.it

Abstract*—Monadic Second-Order Logic** (MSO) **extends *First-Order Logic (FO) **with variables ranging over sets and quantifications over those variables. We introduce and study *Monadic Tree Logic*** (MTL)**, a fragment of** MSO **interpreted on infinite-tree models, where the sets over which the variables range are arbitrary subtrees of the original model. We analyse the expressiveness of** MTL **compared with variants of** MSO **and** MPL**, namely** MSO **with quantifications over paths. We also discuss the connections with temporal logics, by providing non-trivial fragments of the *Graded* μ-**CALCULUS **that can be embedded into** MTL **and by showing that** MTL **is enough to encode temporal logics for reasoning about strategies with** FO**-definable goals.**

I. Introduction

The study of *Monadic Second-Order Logic* (MSO), namely *First-Order Logic* (FO) extended with variables and quantifications ranging over sets, has attracted a lot of attention over the years, mainly because of its high expressive power and nice computational properties, particularly when interpreted over words and trees. This, in turn, makes it a good fit as a formal framework for reasoning about *regular languages* and *computational systems* in general, whose set of possible dynamic evolutions is often represented as a tree structure.

A seminal result in the field was originally provided by Büchi [11], who proved the equivalence between the monadic second-order logic of one successor with variables ranging over finite sets and *finite-state automata on finite words* [48], which he exploited to devise a decision procedure for that logic. The result was then extended to the case of variables ranging over arbitrary sets and *finite-state automata on infinite words* in [12], [13]. Rabin [46] later proved the decidability of MSO interpreted over binary trees, by means of an automata-theoretic characterisation of the expressive power of logic on these structures. This result was then extended by Walukiewicz [51], [52], who provided a general framework for investigating MSO by means of a class of automata that captures the expressive power of MSO on trees with arbitrary (finite and infinite) branching degree. By exploiting Rabin's result, Muller and Schupp [43], [44] have shown that MSO is decidable for graphs with bounded tree-width, while Courcelle [20], [19], [21] has conducted a quite extensive study of MSO on graphs in connection with both graph theory and complexity theory.

Variants of MSO over tree models have also been studied. *Weak Monadic Second-Order Logic* (WMSO) is one such variant, where the second-order variables can only range over finite sets. An automata-theoretic characterisation of WMSO on binary trees has been proposed by Rabin [47] to show that WMSO is strictly less expressive than MSO on this class of structures. Automata for WMSO on these trees have been proposed in [41], [42], where *weak alternating tree automata* have been introduced. It has been shown, moreover, that WMSO is equivalent to the alternation-free fragment of *Modal* μ-CALCULUS [33], when direct access to the left and right children of a node is available [3]. A deeper analysis of this connection, when bisimulation invariant fragments are considered, has been recently carried out in [26], [14], [15].

Another noteworthy variant is *Monadic Path Logic* [28], [29] (MPL), where second-order variables are restricted to range over paths. This restriction makes it strictly less expressive than full MSO over trees. The interest attracted by MPL resides in the fact that many temporal logics, most notably CTL*, can be embedded into MPL [29], [50], [39], [40].

One needs to jump, however, directly to MSO in order to be able to capture the full *Modal* μ-CALCULUS [32] and its *graded extension* [34]. One reason for this is that one can express path properties in μ-CALCULUS that are not expressible in FOL and may be witnessed by *non-connected* (*i.e.*, *non-convex*) sets of nodes. One such property is the one true at a node t if all the nodes at even positions from t onward along some path satisfy a given atomic proposition p. A μ-CALCULUS formula that collects all those witnesses is, for instance, $\nu X.\,(p \wedge \Diamond\Diamond X)$, where the modal formula $\Diamond\varphi$ holds at a node if one of its successors satisfies φ. The nodes of the tree which occur at even positions along some path are not connected to one another and they are all witnesses of the property. More importantly, those non-connected witnesses depend on one another, in the sense that, for a node t to witness the property, some node two steps further ahead must also witness it and removing p from it would prevent t from becoming a witness. The possible non-connectedness of the witnesses is what makes the MSO ability to quantify over sets of, possibly non-connected, nodes an essential feature. This contrasts, for instance, with the property true at a node t when there exists a path from it and a prefix π of that path where q holds at the last node and p holds at all the previous ones. This can be expressed in CTL* by the formula $\mathtt{E}(p\,\mathtt{U}\,q)$. In this case, indeed, if a node t satisfies the property, all the nodes along the witnessing prefix up to the node witnessing q satisfy it as well. Non-connected witnesses may exist for this property too, however, they are all independent from each other, in the sense that removing the property from a prefix π (*e.g.*, by removing p and q from the labelling of the connected witnesses forming π) would bear no consequences for the other witnesses outside π. All properties expressible in CTL* are of

this kind and quantifying over paths, which are connected sets of nodes, suffices to capture them all in MPL.

It turns out that a similar connectedness property holds true even for more expressive logics than CTL*, such as languages for reasoning about strategies and games, like *Alternating-Time Temporal Logic* (ATL*) [1], [2], *Strategy Logic* (SL) [16], [17], [38], [36], [37], and *Substructure Temporal Logic* (STL*) [5], [6]. For instance, it has been shown that STL*, an extension of CTL* that implicitly allows for quantification over subtrees, is strictly more expressive than CTL*, since the latter is bisimulation invariant, while the former is not. By means of subtree quantifications, STL* is able to model strategies and, therefore, to encode games with FO-definable goals.

Guided by these observations, it appears natural to consider the seemingly missing fragment of MSO in which quantifications range over subtrees. In this work, we study precisely this semantic restriction, that we call *Monadic Tree Logic*, MTL for short, interpreted over non-blocking trees. We provide a full picture of the expressive power of the logic, together with some variants that restrict the range of the second-order variables to finite trees only, giving rise to Weak MTL (WMTL), and to infinite ones only, leading to Co-Weak MTL (coWMTL), and compare them to the corresponding variants of MSO and MPL. Each variant turns out to be strictly less expressive than the corresponding MSO variant and strictly more expressive than the analogous MPL variant. We also show that MTL (*resp.*, MSO, MPL) is equivalent to coWMTL (*resp.*, coWMSO, coWMPL) and more expressive than WMTL (*resp.*, WMSO, WMPL), when interpreted over finitely-branching trees. Interestingly enough, though, the situation changes drastically when we interpret MSO and MTL on arbitrary trees that allow for infinitely-branching degrees. In this case, the co-weak variants of each logic strictly contain both the corresponding full and weak versions, while the latter two become incomparable.

The second part of the article analyses the connections between variants of MTL and temporal logics. More specifically, we identify a non-trivial fragment of the Graded μ-CALCULUS, called *One-Step Graded* μ-CALCULUS (Gμ-CALCULUS[1S]), that can be captured by MTL and can express properties of trees that cannot be expressed in MPL. We also show that the alternation-free fragment of Gμ-CALCULUS[1S] (AFGμ-CALCULUS[1S]) can be captured by WMTL and not by WMPL. Finally, we provide an encoding of STL* into MTL, showing that quantification over trees is powerful enough to reason about games with FO-definable goals.

II. BACKGROUND

Let $\mathbb{N}$ be the set of natural numbers. For a finite or infinite word w over some alphabet, $|w|$ denotes the length of w ($|w| = \infty$ if w is infinite) and for all $0 \leq i < |w|$, $w(i)$ is the $(i+1)$-th letter of w.

Trees. A tree T is a subset of $\mathbb{N}^*$ such that there is an element $\tau_0 \in T$, called the *root of* T, so that:

- for each $\tau \in T$, τ is of the form $\tau_0 \cdot \tau'$ for some $\tau' \in \mathbb{N}^*$;
- for all $\tau, \tau' \in \mathbb{N}^*$, if $\tau_0 \cdot \tau \in T$ and τ' is a prefix of τ, then $\tau_0 \cdot \tau' \in T$.

Elements of T are called *nodes*. For $\tau \in T$, a *child* of τ in T is a node in T of the form $\tau \cdot n$ for some $n \in \mathbb{N}$. A *descendant* of τ in T is a node in T of the form $\tau \cdot \tau'$, for some $\tau' \in \mathbb{N}^*$. A *subtree of* T is a subset of T which is a tree. The *subtree of* T *rooted at a node* $\tau \in T$ is the tree consisting of all the descendants of τ in T. A *forest of* T is a union of subtrees of T. A *path* of T is a subtree π of T that 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 viewed as a word over T, in accordance with the total ordering in π induced by the child relation. A tree T is said to be:

- *finitely-branching* if each node in T has finitely many children in T (and *infinitely-branching*, otherwise);
- *blocking* if some node in T has no children in T (and *non-blocking*, otherwise);
- a *chain* if it has a unique maximal path from the root (each node has at most one child). Note that a path of a tree corresponds to a chain.
- a *complete binary tree* if each node has exactly two children;
- *dense* if it contains a subtree T' such that each node of T' has a descendant in T' having at least two distinct children in T'.

Note that a dense tree has an uncountable number of infinite paths or, equivalently, contains a complete binary tree as minor. Dense trees correspond to thick trees in [10], [30].

Labelled trees and Kripke trees. For an alphabet Σ, a Σ-labelled tree is a pair $\mathcal{T} = (T, Lab)$ consisting of a tree and a labelling $Lab : T \mapsto \Sigma$ assigning to each node in T a symbol in Σ. Note that the dynamic behaviour of a system starting from an initial state can be modelled by a 2^{AP}-labelled tree, where AP is a finite set of atomic propositions. A node in the tree T represents a state of the system and the root corresponds to the initial state. The maximal paths in the tree starting from the root correspond to the possible executions of the system from the initial state. Moreover, a node of a tree is labelled by elements in AP, representing the atomic predicates that hold at the given state of the computation. Since we consider labelled trees modelling the dynamic behaviour of *reactive* systems and for these systems the executions are in general infinite, we will restrict the interpretation of the considered logics to labelled trees which are non-blocking. A non-blocking tree T is infinite, and maximal paths in T are infinite as well.

Given a finite set AP of atomic propositions, a *Kripke tree* over AP is a non-blocking 2^{AP}-labelled tree.

Relative Expressiveness. Let $\mathcal{M}$ be a set of models, and L and L' be two logical languages interpreted over models in $\mathcal{M}$. Given two formulas $\varphi \in L$ and $\varphi' \in L'$, we say that φ and φ' are *equivalent* if for each model $M \in \mathcal{M}$, M satisfies φ *iff* M satisfies φ'. The language L *is subsumed by* L', denoted $L \leq L'$, if each formula in L has an equivalent formula in L'. The language L *is strictly less expressive than* L, written $L < L'$, if $L \leq L'$ and there is a L'-formula which has no equivalent in L.

Two logics L and L' *are expressively incomparable*, denoted by $L \not\sim L'$, if both $L \not\leq L'$ and $L' \not\leq L$. Finally, two logics L and L' are *expressively equivalent*, denoted $L \equiv L'$, if both $L \leq L'$ and $L' \leq L$.

Counting-CTL*. We recall syntax and semantics of Counting-CTL* (CCTL* for short [40]), which extends the classic branching-time temporal logic CTL* [24] by counting operators. The syntax of CCTL* is given by specifying inductively the set of *state formulas* φ and the set of *path formulas* ψ over a given finite set AP of atomic propositions:

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

where $a \in$ AP, X and U are the standard "next" and "until" temporal modalities, E is the existential path quantifier, and $\mathtt{D}^n$ is the counting operator with $n \in \mathbb{N}$. The language of CCTL* consists of the state formulas of CCTL*. We also use the standard shorthands $\mathtt{A}\varphi \triangleq \neg\mathtt{E}\neg\varphi$ ("universal path quantifier") and $\mathtt{F}\,\psi \triangleq \top\,\mathtt{U}\,\psi$ ("eventually").

The semantics is given *w.r.t.* Kripke trees $\mathcal{T} = (T, Lab)$ (over AP). For a node τ of T, a path π of T, and $0 \leq i < |\pi|$, the satisfaction relations $\mathcal{T}, \tau \models \varphi$, for state formulas φ (meaning that φ holds at node τ of $\mathcal{T}$), and $\mathcal{T}, \pi, i \models \psi$, for path formulas ψ (meaning that ψ holds at position i of the path π in $\mathcal{T}$), are inductively defined as follows (Boolean connectives are treated as usual):

$\mathcal{T}, \tau \models a$	$\Leftrightarrow a \in Lab(\tau)$;
$\mathcal{T}, \tau \models \mathtt{E}\psi$	$\Leftrightarrow \mathcal{T}, \pi, 0 \models \psi$ for some path π of T starting at node τ
$\mathcal{T}, \tau \models \mathtt{D}^n\varphi$	$\Leftrightarrow$ there are at least n distinct children τ' of τ in T such that $\mathcal{T}, \tau' \models \varphi$
$\mathcal{T}, \pi, i \models \varphi$	$\Leftrightarrow \mathcal{T}, \pi(i) \models \varphi$
$\mathcal{T}, \pi, i \models \mathtt{X}\,\psi$	$\Leftrightarrow i+1 < \|\pi\|$ and $\mathcal{T}, \pi, i+1 \models \psi$
$\mathcal{T}, \pi, i \models \psi_1\,\mathtt{U}\,\psi_2$	$\Leftrightarrow$ for some $i \leq j < \|\pi\|$: $\mathcal{T}, \pi, j \models \psi_2$ and $\mathcal{T}, \pi, k \models \psi_1$ for all $i \leq k < j$.

A Kripke tree $\mathcal{T}$ satisfies a state formula φ, written $\mathcal{T} \models \varphi$, if $\mathcal{T}, \tau_0 \models \varphi$, where τ_0 is the root of $\mathcal{T}$. Given a non-blocking tree T, we write $T \models \varphi$ to mean that $T, Lab_\emptyset \models \varphi$, where $Lab_\emptyset(\tau) = \emptyset$ for all $\tau \in T$.

We also consider two semantic variants of CCTL*, that we call *Weak* CCTL* (WCCTL*) and *coWeak* CCTL* (coWCCTL*), where the path quantifiers E and A range over finite paths and infinite paths, respectively, starting from the current node. Standard CTL* [24] is the syntactical fragment of coWCCTL* where the counting operators are not allowed.

III. Monadic Tree Logic

We start by defining in Section III-A the three main logics we shall consider: MSO, MTL, and MPL. The three languages do not differ at the syntactic level, but only on the range of quantification of the second-order variables. For convenience, though, we provide a unified language (MSOL), where the second-order quantifiers are decorated with a symbol α that explicitly indicates the domain of the quantified variable: S for sets, T for trees, and P for paths.

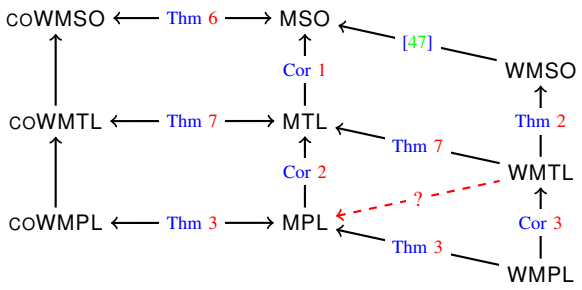


Figure 1: Expressiveness results for MSOL over finitely-branching Kripke trees.

This section is mainly devoted to the analysis of the expressiveness of the various semantic fragments of MSOL interpreted over non-blocking trees. In Section III-B, we compare the expressiveness of MPL, MTL, and MSO in the general case, where second-order variables are interpreted over both finite and infinite paths, subtrees, and sets, respectively, of the considered model tree. Then, in Sections III-C and III-D, we consider similar expressiveness issues for the *weak semantic variant* (second-order variables range over finite paths, finite subtrees, and finite sets, respectively) and the *co-weak semantic variant* (second-order variables range over infinite paths, infinite subtrees, and infinite sets, respectively).

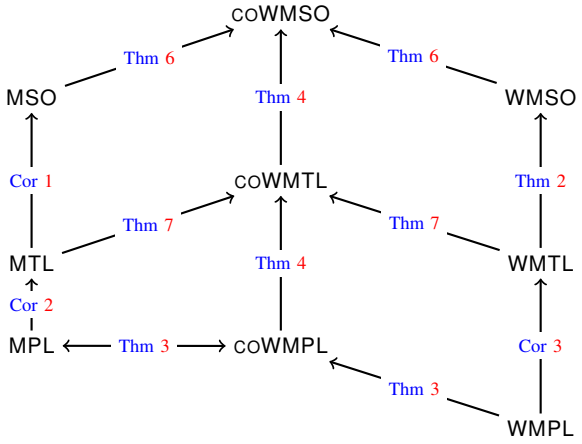


Figure 2: Expressiveness results for MSOL over arbitrary Kripke trees.

Finally, in Section III-E we consider expressiveness issues of weak semantics versus co-week semantics. The complete picture of results is summarised in Figure 1 for the case of finitely-branching Kripke trees and in Figure 2 for the general case. The two figures have to be interpreted as follows. An edge connecting two logics has the following meaning: if the edge has a single arrow, then the target logic is more expressive than the source; otherwise, the two logics are expressively equivalent. If there is no edge between two distinct logics and no relation is deducible by the other edges, then the two logics are expressively incomparable. A red edge decorated with a question mark indicates a currently open question.

A. Monadic Second-Order Logics

For a given finite set AP of atomic propositions, MSOL is a language defined over the signature $\{\leq\} \cup \{P_a \mid a \in \mathrm{AP}\}$, where second-order quantification is restricted to monadic predicates, $\leq$ is a binary predicate, and P_a is a monadic predicate for each $a \in \mathrm{AP}$.

Definition 1 (MSOL Syntax)**.** *Given a finite set* AP *of atomic propositions, a finite set* Vr_1 *of first-order variables (or* node *variables), and a finite set* Vr_2 *of second-order variables (or* set *variables), the syntax of* Monadic Second-Order/Tree/Path Logic *(*MSO/MTL/MPL*, for short) is the set of formulae built according to the following grammar, where* $p \in \mathrm{AP}$*,* $x, y \in \mathrm{Vr}_1$*, and* $X \in \mathrm{Vr}_2$*:*

$$\varphi := P_a(x) \mid x \leq y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \mid \exists^{\alpha} X.\, \varphi,$$

where $p \in \mathrm{AP}$*,* $x, y \in \mathrm{Vr}_1$*,* $X \in \mathrm{Vr}_2$*, and* α *is* S *for* MSO*,* T *for* MTL*, and* P *for* MPL*.*

Note that MSO (*resp.*, MTL, MPL) corresponds to the syntactical fragment of MSOL where the second-order existential quantification takes only the form $\exists^{\mathtt{S}}$ (*resp.*, $\exists^{\mathtt{T}}$, $\exists^{\mathtt{P}}$). We also exploit the standard logical connectives $\vee$ and $\rightarrow$ as abbreviations, the universal first-order quantifier $\forall x$, defined as $\forall x.\varphi \triangleq \neg\exists x.\neg\varphi$, and the universal second-order quantifier $\forall^{\alpha} X$, defined as $\forall^{\alpha} X.\varphi \triangleq \neg\exists^{\alpha} X.\neg\varphi$. We may also make use of the shorthands (i) $x = y$ for $x \leq y \wedge y \leq x$, (ii) $x < y$ for $x \leq y \wedge \neg(y \leq x)$; (iii) $\exists x \in X.\, \varphi$ for $\exists x.\, (x \in X \wedge \varphi)$, and (iv) $\forall x \in X.\, \varphi$ for $\forall x.\, (x \in X \rightarrow \varphi)$.

As usual, a *free variable* of a formula φ is a variable occurring in φ that is not bound by a quantifier. A *sentence* is a formula with no free variables. The language of MSOL consists of its sentences. We also consider the first-order fragment FO of MSOL, where second-order quantifiers and second-order variables are not allowed.

Semantics of MSOL**.** Formulas of MSOL are interpreted over Kripke trees over AP. A Kripke tree $\mathcal{T} = (T, Lab)$ induces the relational structure with domain T, where the binary predicate $\leq$ corresponds to the descendant relation in T, and P_a denotes the set $\{\tau \in T : a \in Lab(\tau)\}$ of a-labelled nodes.

Let us fix a Kripke tree $\mathcal{T} = (T, Lab)$ over AP. A *first-order valuation for the tree* T is a mapping $\mathcal{V}_1 : \mathrm{Vr}_1 \mapsto T$ assigning to each first-order variable a node of T. A *second-order valuation for the tree* T is a mapping $\mathcal{V}_2 : \mathrm{Vr}_2 \mapsto 2^T$ assigning to each second-order variable a subset of T.

Definition 2 (MSOL Semantics)**.** *Given a* MSOL *formula* φ*, a Kripke tree* $\mathcal{T} = (T, Lab)$ *over* AP*, a first-order valuation* $\mathcal{V}_1$ *for* T*, and a second-order valuation* $\mathcal{V}_2$ *for* T*, the satisfaction relation* $\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \varphi$*, meaning that* $\mathcal{T}$ *satisfies the formula* φ *under the valuations* $\mathcal{V}_1$ *and* $\mathcal{V}_2$*, is defined as follows (the treatment of Boolean connectives is standard):*

$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models P_a(x) \Leftrightarrow a \in Lab(\mathcal{V}_1(x))$;
$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models x \leq y \Leftrightarrow \mathcal{V}_1(y)$ *is a descendant of* $\mathcal{V}_1(x)$ *in* T;
$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models x \in X \Leftrightarrow \mathcal{V}_1(x) \in \mathcal{V}_2(X)$;
$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \exists x.\varphi \Leftrightarrow T, Lab, \mathcal{V}_1[x \mapsto \tau], \mathcal{V}_2 \models \varphi$ *for some* $\tau \in T$;
$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \exists^{\mathtt{S}} X.\varphi \Leftrightarrow \mathcal{T}, \mathcal{V}_1, \mathcal{V}_2[X \mapsto S] \models \varphi$ *for some set of nodes* $S \subseteq T$;
$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \exists^{\mathtt{T}} X.\varphi \Leftrightarrow \mathcal{T}, \mathcal{V}_1, \mathcal{V}_2[X \mapsto T'] \models \varphi$ *for some subtree* T' *of* T;
$\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \exists^{\mathtt{P}} X.\varphi \Leftrightarrow \mathcal{T}, \mathcal{V}_1, \mathcal{V}_2[X \mapsto \pi] \models \varphi$ *for some path* π *of* T.

where for a first-order valuation $\mathcal{V}_1$ for T, a node $\tau \in T$, and a first-order variable $x \in T$, $\mathcal{V}_1[x \mapsto \tau]$ denotes the first-order valuation defined as follows: $\mathcal{V}_1[x \mapsto \tau](x) = \tau$ and $\mathcal{V}_1[x \mapsto \tau](y) = \mathcal{V}_1(y)$ if $y \neq x$. The meaning of notation $\mathcal{V}_2[X \mapsto S]$, for a second-order valuation $\mathcal{V}_2$ for T, a set $S \subseteq T$, and a second-order variable X is similar.

Note that the satisfaction relation $\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \varphi$, for fixed $\mathcal{T}$ and φ, depends only on the values assigned by $\mathcal{V}_1$ and $\mathcal{V}_2$ to the variables occurring free in φ. In particular, if φ is a sentence, we say that $\mathcal{T}$ *satisfies* φ, written $\mathcal{T} \models \varphi$, if $\mathcal{T}, \mathcal{V}_1, \mathcal{V}_2 \models \varphi$ for some valuations $\mathcal{V}_1$ and $\mathcal{V}_2$. In this case, we also say that $\mathcal{T}$ is a model of φ. A non-blocking tree T satisfies a sentence φ, written $T \models \varphi$, if $(T, Lab_{\emptyset}) \models \varphi$, where $Lab_{\emptyset}$ assigns to each T-node the empty set.

Basic predicates expressible in MSOL**.** In the following we define some useful standard predicates which can be expressed in MSOL by using only first-order quantification:

- the second-order binary predicates $X \subset Y$, $X \subseteq Y$, $X = Y$, $X \neq Y$ (of expected meaning). For instance, $X \subseteq Y$ can be expressed as $\forall x.\, (x \in X \rightarrow x \in Y)$;
- the child relation is definable in MSOL by the binary predicate $\mathtt{child}(x, y) \triangleq x < y \wedge \neg\exists z.\, (x < z \wedge z < y)$;
- $\mathtt{path}(X)$ (*resp.*, $\mathtt{path}_{\infty}(X)$, *resp.*, $\mathtt{path}_{f}(X)$) capturing the subsets of the given tree which are paths (*resp.*, infinite paths, *resp.*, finite paths). For example:

$$\mathtt{path}(X) \triangleq \forall x \in X. \forall y \in X. ((x \leq y \vee y \leq x) \wedge ((x < y \wedge \neg\mathtt{child}(x, y)) \rightarrow \exists z \in X. (x < z \wedge z < y)).$$

- the property of being a tree is captured by the second-order predicate:

$$\mathtt{tree}(X) \triangleq \exists x. \forall y. (x \leq y) \wedge \forall x \in X. \forall y \in X. ((x < y \wedge \neg\mathtt{child}(x, y)) \rightarrow \exists z \in X. (x < z \wedge z < y)).$$

Weak and coWeak semantic variants. We also consider the *Weak semantics variants* of MSO, MTL, and MPL, denoted WMSO, WMPL and WMTL, respectively. In WMSO, second-order variables are interpreted as *finite* sets of the given tree. Similarly, in WMTL and WMPL, second-order quantification ranges over *finite* subtrees and *finite* paths, respectively, of the given tree. Finally, we consider the *coWeak semantics variants* of MSO, MTL, and MPL (written COWMSO, COWMPL and COWMTL, respectively) where second-order variables are interpreted as infinite sets of the given tree (infinite paths and infinite subtrees in the case of COWMPL and COWMTL, respectively).

B. Expressiveness under Full Quantifications

In this section, we compare the expressiveness of the logics MPL, MTL, and MSO. We prove that MTL strictly lies

between MPL and MSO even in the case of *finitely-branching setting* (*i.e.*, over the class of finitely-branching Kripke trees). First, we show that MTL is strictly less expressive than MSO. In fact, the result already holds over the class of 2^{AP}-labelled infinite chains. For this subclass of Kripke trees, quantification over trees reduces to quantification over paths, which in turn can be simulated by first-order quantification [13]. Thus, since FO $<$ MSO even over infinite chains, we obtain the following result.

Proposition 1. *Over 2^{AP}-labelled infinite chains, it holds that:*

- MTL $\equiv$ WMTL $\equiv$ coWMTL $\equiv$ FO $<$ MSO*, and*
- *by [13],* MSO $\equiv$ WMSO $\equiv$ coWMSO.

Since the class of chains can be trivially captured in FO, by Proposition 1, the following hold.

Proposition 2. *For all $L \in \{\mathsf{MSO}, \mathsf{WMSO}, \mathsf{coWMSO}\}$ and $L' \in \{\mathsf{MTL}, \mathsf{WMTL}, \mathsf{coWMTL}\}$, $L \not\leq L'$ even in the finitely-branching setting.*

Clearly, MTL is subsumed by MSO (the predicate $\mathtt{tree}(X)$ can be expressed in MSO). Therefore, by Proposition 2, we obtain the desired result.

Corollary 1. MTL $<$ MSO *even in the finitely-branching setting.*

Next, we show that MPL is strictly less expressive than MTL. Evidently, MPL is subsumed by MTL (path quantification can be simulated by tree quantification and first-order quantification). In order to show that MTL is not subsumed by MPL, we prove that the density property (characterising the class of dense non-blocking trees) is definable in MTL but not in MPL. The density property can be expressed in MTL as:

$$\exists^{\mathrm{T}} X.\, \forall x \in X.\, \exists x_1 \in X.\, \exists x_2 \in X.\\ (x < x_1 \wedge x < x_2 \wedge \neg\, x_1 \leq x_2 \wedge \neg\, x_2 \leq x_1).$$

To prove that the density property cannot be expressed in MPL, we need a preliminary result that generalises the known expressiveness equivalence between coWMPL and coWCCTL* [40]. The easy translation of coWCCTL* into coWMPL can be trivially adapted to show that every CCTL* (*resp.*, WCCTL*) formula can be translated in linear time into an equivalent MPL (*resp.*, WMPL) formula. By adapting the compositional argument in [40] for showing that coWMPL is subsumed by coWCCTL*, we obtain the following result.

Proposition 3. MPL $\equiv$ CCTL* *and* WMPL $\equiv$ WCCTL*.

For each $n \geq 1$, we define two non-empty classes ND_n and D_n of non-blocking finitely-branching trees such that the following holds for each $n > 1$:

- ND_n contains only isomorphic trees which does *not* satisfy the density property;
- D_n contains only isomorphic trees which satisfy the density property;
- no state formula φ in CCTL* with size smaller than n distinguishes the classes ND_n and D_n, *i.e.*, for all $T \in D_n$ and $T' \in ND_n$, $T \models \varphi$ *iff* $T' \models \varphi$.

Thus, by Proposition 3, it follows that the logic MPL cannot capture the density property.

In the following, the size $|\varphi|$ of a CCTL* formula φ is defined as the length of the string for representing φ, where we assume that the natural numbers k in the counting operators $\mathtt{D}^k$ are encoded in unary. In particular, $|\mathtt{D}^k\varphi| = k + 1 + |\varphi|$. The classes ND_n and D_n are defined by induction on $n \geq 1$:

- ND_1 and D_1 coincide and consist of the infinite chains;
- for each $n > 1$, ND_n is the smallest set of non-blocking trees T satisfying the following conditions:
  - the root of T, called ND_n*-node*, has exactly $n\cdot(n-1)+1$ distinct children
    $$\tau_{1,1}, \ldots, \tau_{1,n}, \ldots, \tau_{n-1,1}, \ldots, \tau_{n-1,n}, \tau_n;$$
  - for all $\ell \in [1, n-1]$, the subtrees rooted at the children $\tau_{\ell,1}, \ldots, \tau_{\ell,n}$ are in ND_ℓ;
  - the subtree rooted at τ_n is in ND_n.
- for each $n > 1$, D_n is the smallest set of non-blocking trees T satisfying the following conditions:
  - the root of T, called D_n*-node*, has exactly $n\cdot(n-1)+2$ distinct children
    $$\tau_{1,1}, \ldots, \tau_{1,n}, \ldots, \tau_{n-1,1}, \ldots, \tau_{n-1,n}, \tau_n, \tau'_n;$$
  - for all $\ell \in [1, n-1]$, the subtrees rooted at the children $\tau_{\ell,1}, \ldots, \tau_{\ell,n}$ are in ND_ℓ (note ND_ℓ and not D_ℓ);
  - the subtrees rooted at τ_n and τ'_n are in D_n.

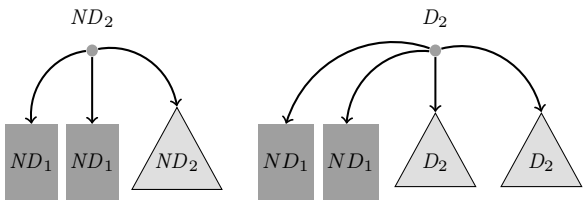


Figure 3: The classes of trees ND_n and D_n for $n = 2$.

By construction, the following holds.

Lemma 1. *For all $n > 1$, the trees in D_n are dense, while those in ND_n are not.*

Proof. Let $n > 1$, $T \in D_n$, and S be the subset of T consisting only of D_n-nodes. By construction, S is a complete binary tree. Hence, T is dense. Next, we show that for all $k \geq 1$ and $T \in ND_k$, T is not dense. Hence, the result follows. The proof is by induction on k. The case $k = 1$ is obvious since, in this case, T is an infinite chain. Now, let $k > 1$. We assume that T is dense, and derive a contradiction. Hence, there is a subtree S of T satisfying the following condition:

- *reachability invariance*; every node of S has a descendant in S having at least two distinct children in S.

By construction, one of the following two conditions occurs:

- there is $1 \leq h < k$ and $T' \in ND_h$ such that $S \subseteq T'$. By the induction hypothesis, T' is not dense, and we derive a contradiction;
- for some node τ of S, the subtree S' of S rooted at node τ is a subset of a tree T' in ND_h for some $1 \leq h < k$. Note that like S, S' satisfies the reachability invariance

condition. Conversely, by the induction hypothesis, T' is not dense, reaching a contradiction. □

We can show that any CCTL* state formula φ does not distinguishes the classes D_n and ND_m for all $n, m \geq |\varphi|$. The proof is by structural induction on the size of φ. In the evaluation of the temporal and counting modalities, we need to compare representatives of the classes D_i and ND_j for (possibly distinct) indexes i and j satisfying the invariant $i, j \geq |\psi|$, where ψ is the currently processed subformula of φ.

Lemma 2. *Let φ be a* CCTL* *state formula. Then, for all $m, n > 1$ such that $\min(m, n) \geq |\varphi|$ and for all $T, T' \in D_n \cup D_m \cup ND_n \cup ND_m$, the following holds: $T \models \varphi \Leftrightarrow T' \models \varphi$.*

Thus, by Lemmata 1–2 and the equivalence MPL ≡ CCTL*, the following holds.

Theorem 1. *The density property is not expressible in* MPL *even in the finitely-branching setting.*

Since the density property can be expressed in MTL, and MPL is trivially subsumed by MTL, by Corollary 1 and Theorem 1, we obtain the following expressiveness hierarchy for the logics MPL, MTL, and MSO.

Corollary 2. MPL < MTL < MSO *even in the finitely-branching setting.*

C. Expressiveness under Weak Quantifications

In this section, we compare the expressiveness of the weak semantic variants of MSO, MTL, and MPL. As a main result, we establish that WMTL strictly lies between WMPL and WMSO even in the finitely-branching setting. Clearly, WMTL is subsumed by WMSO (the requirement that a second-order variable ranging over finite sets captures only finite subtrees of the given tree can be expressed by using only first-order quantification). Thus, by Proposition 2, the following holds.

Theorem 2. WMTL < WMSO *even in the finitely-branching setting.*

Next, we show that WMPL < WMTL. First, we observe that since quantification over finite paths can be expressed in FO, WMPL and FO are expressively equivalent and WMPL ≤ WMTL. In order to show that WMTL is more expressive than WMPL in the general case, we consider the *infinitely-branching property* requiring that a tree is infinitely-branching. This property can be expressed in MTL under the weak semantics as $\neg\forall x.\, \exists^{\mathsf{T}} X.\, \forall y.\, (\mathtt{child}(x, y) \rightarrow y \in X)$. On the other hand, it is known by [49] that every satisfiable MSO formula is satisfied by a finitely-branching Kripke tree. Thus, since WMPL, MPL and MTL are subsumed by MSO, the following holds.

Proposition 4. *The infinitely-branching property is not definable in the logics* MSO, MTL, MPL, *and* WMPL.

In order to show that WMTL is not subsumed by WMPL even in the finitely-branching setting, we show that for each atomic proposition a, the property (called *a-acceptance*) expressed by the CTL* formula $\mathtt{AF}\, a$ is not definable in WMPL. Note that the a-acceptance property captures the Kripke trees such that each infinite path from the root visits a node labeled by a. In the finitely-branching setting, this property can be expressed in WMTL by requiring that there is a finite *tree-prefix* T_a of the given tree T such that (i) each leaf of T_a is labeled by proposition a, and (ii) for each non-leaf node $\tau \in T_a$, each child of τ in T is child of τ in T_a as well.

In the following, we show that the a-acceptance property is not expressible in WMPL. By Proposition 3, it suffices to prove that a-acceptance is not definable in WCCTL*. show that the CTL* formula $\mathtt{AF}\, a$ (called

For a WCCTL* formula ψ, we say that ψ is *balanced* if:

- for each subformula $\psi_1 \,\mathtt{U}\, \psi_2$ of ψ, it holds that $|\psi_1| = |\psi_2|$;
- for each subformula $\mathtt{E}\theta$ of ψ, θ is of the form $\theta_1 \wedge \theta_2$ with $|\theta_1| = |\theta_2|$.

Proving the inexpressiveness result of a-acceptance for balanced WCCTL* state formulas allows us to state it for any WCCTL* state formula, since (by using conjunctions of $\top$) a WCCTL* state formula can be trivially converted into an equivalent balanced WCCTL* state formula.

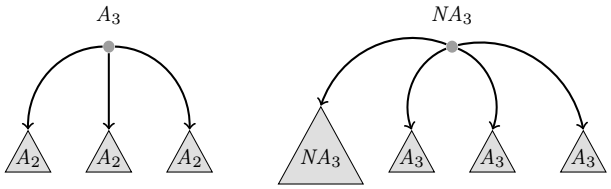


Figure 4: The classes of Kripke trees A_n and NA_n for $n = 3$.

Let $\mathrm{AP} = \{a\}$. For each $n \geq 1$, we define two non-empty classes A_n and NA_n of *finitely-branching* Kripke trees over AP such that the following holds for each $n \geq 1$:

- A_n contains only isomorphic Kripke trees which satisfy the a-acceptance property;
- NA_n contains only isomorphic Kripke trees which does *not* satisfy the a-acceptance property;
- no balanced state formula φ in WCCTL* with size smaller than n distinguishes the classes NA_n and A_n, *i.e.*, for all $(T, Lab) \in A_n$ and $(T', Lab') \in NA_n$, $(T, Lab) \models \varphi$ iff $(T', Lab') \models \varphi$.

The classes A_n and NA_n are defined by induction on $n \geq 1$:

- A_1 consist of the labeled infinite chains where each node is labeled by proposition a (a-node or A_1-node);
- for each $n > 1$, A_n is the smallest set of Kripke trees (T, Lab) satisfying the following conditions:
  - the root of T, called *A_n-node*, has empty label and exactly n distinct children $\tau_1, \ldots, \tau_n$;
  - for all $\ell \in [1, n]$, the Kripke subtree rooted at the child τ_i is in A_{n-1}.
- for each $n \geq 1$, NA_n is the smallest set of Kripke trees (T, Lab) satisfying the following conditions:
  - the root of T, called *NA_n-node*, has empty label and exactly $n + 1$ distinct children $\tau_0, \tau_1, \ldots, \tau_n$;

– for all $\ell \in [1, n]$, the Kripke subtree rooted at the child τ_i is in A_n;
– the Kripke subtree rooted at τ_0 is in NA_n.

Note that since Kripke trees are infinite, it makes sense that a Kripke tree (T, Lab) and its subtree rooted at a child of the root are isomorphic. Hence, the class NA_n is well defined.

Let $n \geq 1$. By construction, for each Kripke tree in A_n, each infinite path from the root has a suffix visiting only a-nodes. On the other hand, for each Kripke tree in NA_n, there is an infinite path from the root visiting only nodes with empty label (in particular, NA_n-nodes). Hence, the following holds.

Lemma 3. *For all $n \geq 1$, the Kripke trees in A_n satisfy the a-acceptance property, while the Kripke trees in NA_n not.*

We can show that any balanced WCCTL* state formula φ does not distinguish the classes A_n and NA_n for all $n \geq |\varphi|$.

Lemma 4. *Let φ be a balanced WCCTL* state formula. Then for all $n > |\varphi|$, $(T, Lab) \in NA_n$ and $(T', Lab') \in A_n$, it holds that $(T, Lab) \models \varphi$ if and only if $(T', Lab') \models \varphi$.*

Thus, by Lemmata 3 and 4 and Proposition 3, it follows that the a-acceptance property cannot be expressed in WMPL. Moreover, note that MPL $\equiv$ coWMPL. This is because (i) quantification over finite paths can be expressed in FO, and (ii) the requirement that a path is infinite can be defined in MPL by using only first-order quantifications. Thus, since a-acceptance can be expressed in MPL, we easily obtain the following result.

Theorem 3. *No WMPL formula can express the a-acceptance property. Moreover, it holds that WMPL $\equiv$ FO, MPL $\equiv$ coWMPL, and WMPL $<$ MPL, even in the finitely-branching setting.*

Since WMTL can express the infinitely-branching property and, in the finitely-branching setting, the a-acceptance property too, by Theorem 2, Proposition 4, and Theorem 3, we obtain the following expressiveness hierarchy for weak variants.

Corollary 3. WMPL $<$ WMTL $<$ WMSO *even in the finitely-branching setting.*

D. Expressiveness under coWeak Quantifications

In this section, we establish an expressiveness hierarchy for the coWeak versions of the considered logics MPL, MTL, and MSO similar to the one for the corresponding Weak versions.

Theorem 4. coWMPL $<$ coWMTL $<$ coWMSO *even in the finitely-branching setting.*

Proof. Evidently, coWMPL $\leq$ coWMTL (quantification over infinite paths can be simulated by quantification over infinite trees and first-order quantification). Moreover, coWMTL $\leq$ coWMSO (the requirement that a second-order variable in coWMSO captures only infinite subtrees of the given tree can be expressed by using only first-order quantification). Since coWMPL $\equiv$ MPL (Theorem 3), by Theorem 1, coWMPL cannot express the density property even in the finitely-branching setting. On the other hand, the density property is expressible in coWMTL. Indeed, the MTL formula used in Section III-B for expressing the density property is equivalent to its coWeak semantics variant. Moreover, by Proposition 2, coWMSO is not subsumed by coWMTL. Hence, the result directly follows. □

E. Weak Quantifications versus coWeak Quantifications

In this section, we compare the logics MTL and MSO with their corresponding coWeak and Weak semantics variants.

It is known by [10], [30] that the density property cannot be expressed in WMSO even in the finitely-branching setting. Thus, being WMTL $\leq$ WMSO, the previous inexpressiveness result holds for WMTL as well. On the other hand, we have seen in Section III-B that the density property can be instead expressed in MTL (hence, in MSO as well). Moreover, in Section III-C, we have proved that the infinitely-branching property can be expressed in WMTL (hence, in WMSO too) but not in MSO and MTL (see Proposition 4). It follows that over arbitrary Kripke trees, WMTL and MTL (*resp.*, WMSO and MSO) are expressively incomparable.

However, in the finitely-branching setting, it is known that WMSO is subsumed by MSO. Indeed, in this setting, the predicate $\mathtt{fin}(X)$ capturing the finite sets of the given tree can be expressed in MSO by the formula

$$\exists^{\mathsf{S}} Y.\, \big(\mathtt{tree}(Y) \wedge X \subseteq Y \wedge \neg \exists^{\mathsf{S}} Z.\, (Z \subseteq Y \wedge \mathtt{path}_{\infty}(Z)\big).$$

Moreover, in the finitely-branching setting, assuming that X is interpreted as a subtree of the given tree, the predicate $\mathtt{fin}(X)$, can be defined in MTL as $\neg \exists^{\mathsf{T}} Y.\, (Y \subseteq X \wedge \mathtt{path}_{\infty}(Y))$. Thus, by Proposition 2, we obtain the following result.

Theorem 5. MSO $\not\sim$ WMSO, MTL $\not\sim$ WMTL, MSO $\not\sim$ WMTL, *and* MTL $\not\sim$ WMSO. *In the finitely-branching setting,* WMSO $<$ MSO, WMTL $<$ MTL, *and* MTL $\not\sim$ WMSO.

Now, we show that second-order quantification over finite sets can be simulated in coWMSO by using the following characterisation of the finite subsets of a non-blocking tree.

Lemma 5. *Let T be a non-blocking tree and $S \subseteq T$. Then, S is finite* iff *the following condition is fulfilled:*

(*) there exist an infinite tree $T_\infty \subseteq T$, an infinite forest $F_\infty \subseteq T$, and an infinite set $Y_\infty \subseteq T$ such that:
– *T_∞ is finitely-branching;*
– *$F_\infty \subseteq Y_\infty \subseteq T_\infty$;*
– *for each infinite path π of T_∞, there is a suffix of π which visits only nodes of F_∞;*
– *$S = T_\infty \setminus Y_\infty$.*

Proof. If Condition $(*)$ is satisfied, then S is contained in the tree T_f obtained from T_∞ by removing all the nodes of the forest F_∞. Since T_∞ is finitely-branching and each infinity path of T_∞ eventually visits only nodes of F_∞, it follows that T_f is finite. Hence, S is finite as well.

Now, assume that S is finite. Since T is non-blocking, there must be a finite subtree T_f of T such that $S \subseteq T_f \setminus L$, where L is the set of leaves of T_f. For each $\tau \in L$, let π_τ any

infinite path of T starting at node τ (since T is not-blocking such a path π_τ exists). Let F_∞ be the infinite forest given by $\bigcup_{\tau \in F} \pi_\tau$. Define $T_\infty \triangleq T_f \cup F_\infty$ and $Y_\infty = (T_f \setminus S) \cup F_\infty$. Evidently, Condition $(*)$ is fulfilled. □

We can easily express in coWMSO that an infinite subset of a not-blocking tree T is a tree (*resp.*, a forest), and that an infinite tree is finitely-branching. In particular, assuming that a set variable Z is interpreted as an infinite tree T_∞, the property that T_∞ is finitely-branching can be expressed in coWMSO by the formula $\neg\exists x.\, \exists^{\mathsf{S}} X.\, [X \subseteq Z \wedge x \in X \wedge \forall y \in X.\, (y = x \vee \mathtt{child}(x, y))]$. Thus, by Lemma 5, we easily deduce that both MSO and WMSO are subsumed by coWMSO.

Moreover, the class of infinitely-branching trees can be captured in coWMSO by the formula $\exists x.\, \exists X^{\mathsf{S}}.\, [x \in X \wedge \forall y \in X.\, (y = x \vee \mathtt{child}(x, y))]$. Hence, by Proposition 4, it follows that coWMSO is in general more expressive than MSO. However, in the finitely-branching setting, since the predicate $\mathtt{fin}(X)$ is definable in MSO, we have MSO $\equiv$ coWMSO. Thus, being the density property definable both in MSO and coWMSO but not in WMSO, we obtain the following result.

Theorem 6. MSO $<$ coWMSO *and* WMSO $<$ coWMSO. *In the finitely-branching setting,* MSO $\equiv$ coWMSO *and* WMSO $<$ coWMSO.

Now, let us consider the coWeak semantics variant of MTL. We first show that both MTL and WMTL are subsumed by coWMTL. In other terms, second-order quantification over finite trees can be simulated in coWMTL. We exploit the following characterisation of the finite subtrees of a given not-blocking tree.

Lemma 6. *Let T be a not-blocking tree and T' be a subtree of T. Then, T' is finite* iff *the following condition holds:* $(*)$ *there is a node $\tau \in T$ and an infinite tree $T_\infty \subseteq T$ such that:*

- *$\tau \in T_\infty$ and T_∞ is finitely-branching;*
- *each infinite path π of T_∞ visits some strict descendant of τ in T;*
- *$T' = T_\infty \setminus \{\tau' \in T \mid \tau < \tau'\}$.*

Proof. Evidently, Condition $(*)$ entails that T' is finite. Vice versa, assume that T' is finite, and let τ be any leaf node of T'. Since T is non-blocking, there exists an infinite path π_τ of T starting at node τ. Define $T_\infty \triangleq T' \cup \pi_\tau$. It easily follows that Condition $(*)$ is fulfilled. □

By Lemma 6, we deduce the following result.

Proposition 5. MTL $\leq$ coWMTL *and* WMTL $\leq$ coWMTL.

The infinitely-branching property can be expressed in coWMTL by the formula $\exists x.\, \exists^{\mathsf{T}} X.\, [x \in X \wedge \forall y \in X.\, (y = x \vee \mathtt{child}(x, y))]$. Hence, by Propositions 4 and 5, it follows that coWMTL is in general more expressive than MTL. However, in the finitely-branching setting, assuming that X is interpreted as a subtree of the given tree, the predicate $\mathtt{fin}(X)$ can be defined in MTL. Hence, in this setting, MTL $\equiv$ coWMTL. Thus, being the density property definable both in MTL (see Section III-B) and coWMTL (see the proof of Theorem 4) but not in WMTL and WMSO, by Proposition 2, we obtain the following result.

Theorem 7. MTL $<$ coWMTL, WMTL $<$ coWMTL, MSO $\not\sim$ coWMTL, *and* WMSO $\not\sim$ coWMTL. *In the finitely-branching setting,* MTL $\equiv$ coWMTL, WMTL $<$ coWMTL, *and* WMSO $\not\sim$ coWMTL.

Additional expressiveness results. By the results established so far, in order to have a complete picture about the expressiveness comparison between the considered logics, we have to compare MPL with WMTL and WMSO. It is known that in the finitely-branching setting, CTL* is not subsumed by WMSO. This follows from [22], [23], [25], where the authors prove that the formula $\mathtt{E}\,\mathtt{GF}\,p$ cannot be expressed in AFμ-Calculus, and [3], where it is shown that AFμ-Calculus is equivalent to WMSO. Thus, being CTL* $\leq$ MPL and WMTL $\leq$ WMSO, and since the infinitely-branching property can be expressed in WMTL and WMSO but not in MPL (Proposition 4), by Proposition 2, the following holds.

Theorem 8. MPL $\not\sim$ WMTL *and* MPL $\not\sim$ WMSO. *In the finitely-branching setting,* MPL $\not\leq$ WMTL *and* MPL $\not\sim$ WMSO.

It remains an open question whether in the finitely-branching setting, WMTL is subsumed by MPL or not.

IV. Connections with Temporal Logics

The results in the previous section show that MTL is a non-trivial fragment of MSO that strictly contains MPL. So the question of its relationship with temporal logics becomes worthy of investigation. To this end, we first identify a new fragment of the *Graded* μ-Calculus (Gμ-Calculus) [34] and its *alternation-free variant* (AFGμ-Calculus), whose semantics can be encoded in MTL and WMTL, respectively. To the best of our knowledge, this is the first non-trivial example (*i.e.*, not subsumed by other temporal formalisms) of modal fixpoint logics, whose semantics does not require the full power of set quantifications. We then present a translation of *Substructure Temporal Logic* (STL*) [5], [6] into coWMTL, which shows that the latter is powerful enough to reason about games with FO-definable goals and to encode several verification problems, such as *reactive synthesis* [45] and *module checking* [35].

A. One-Step Graded μ-Calculus

As observed by Wolper [53], there are simple ω-regular properties that cannot be expressed in classic temporal logics, while they are easily expressible in Kozen's Modal μ-Calculus [33]. One of the simplest examples is the existence of a path in a Kripke tree where a given atomic proposition p holds true at all even positions along it: $\nu X.(p \wedge \Diamond\Diamond X)$. As we already observed in the Introduction, this formula is witnessed by non-connected set of nodes and each witness depends on the one two steps ahead in the path, due to the double nesting of the modal operator $\Diamond$ preceding the

fixpoint variable. This contrasts with classic temporal logic formulae, whose translation into the Modal μ-CALCULUS does not require multiple nestings of the modal operators in front of the fixpoint variables. For instance, the CTL formula $\mathtt{E}(p\,\mathtt{U}\,q)$ is equivalent to $\mu X.(q \vee (p \wedge \Diamond X))$ and a single modality separates the fixpoint operator from its variable.

On the basis of these observations, it seems natural to conjecture that preventing multiple nestings of modalities over fixpoint variables suffices to write formulae whose dependent sets of witnesses are always connected to each other, while the non-connected ones are independent from one another. This "independence" property seems crucial for the existence of an encoding into MTL, which can only predicate over connected sets of nodes and, therefore, cannot talk about non-connected sets. In the following, we prove this conjecture, by first introducing the *one-step fragment* of Graded μ-CALCULUS [34], an extension of the Modal μ-CALCULUS with graded (*i.e.*, counting) modalities [27], and then showing a direct translation of its semantics into MTL.

Definition 3. *The* One-Step Fragment of Gμ-CALCULUS *(*Gμ-CALCULUS[1S]*) is the set of formulae built accordingly to the following context-sensitive grammar, where* $\mathrm{Z}, \mathrm{O} \subseteq \mathrm{Vr}_2$*,* $X \in \mathrm{Z}$*, and* $p \in \mathrm{AP}$*:*

$$\begin{aligned}\varphi_{\mathrm{Z,O}} &:= \bot \mid \top \mid p \mid \neg\varphi_{\mathrm{Z,O}} \mid \varphi_{\mathrm{Z,O}} \wedge \varphi_{\mathrm{Z,O}} \mid \varphi_{\mathrm{Z,O}} \vee \varphi_{\mathrm{Z,O}} \\ &\quad \mid \Diamond_{\geq k}\, \varphi_{\mathrm{O},\emptyset} \mid \Box_{<k}\, \varphi_{\mathrm{O},\emptyset} \mid X \mid \vartheta_{\emptyset,\emptyset}; \\ \vartheta_{\mathrm{Z,O}} &:= \mu X.\, \vartheta_{\mathrm{Z}\cup\{X\},\mathrm{O}\cup\{X\}} \mid \nu X.\, \vartheta_{\mathrm{Z}\cup\{X\},\mathrm{O}\cup\{X\}} \mid \varphi_{\mathrm{Z,O}}.\end{aligned}$$

$\Phi_{\mathrm{Z,O}}$ (resp., $\Theta_{\mathrm{Z,O}}$) *denotes the set of formulae described by the first (*resp.*, second) rule called base (*resp.*, fixpoint) formulae, where every occurrence of a variable is positive (*i.e.*, within the scope of an even number of negations). Formulae from* $\Phi_{\emptyset,\emptyset}$ *and* $\Theta_{\emptyset,\emptyset}$ *are called* sentences*.*

The two sets Z and O of fixpoint variables, called *zero-* and *one-step variables*, respectively, identify the only free variables that can occur in a Gμ-CALCULUS[1S] formula. Specifically, the variables in Z can be used out of the scope of any modalities, while those in O need to occur inside a single nesting of a modality. No nesting of modalities is allowed before reaching a fixpoint variable from the corresponding fixpoint operator.

Examples of Gμ-CALCULUS[1S] sentences are the encodings $\mu X.\,(q \vee (p \wedge \Diamond_{\geq 1} X))$ and $\nu X.\,\mu Y.\,(p \wedge \Diamond_{\geq 1} X) \vee (\Diamond_{\geq 1} Y)$ of the CTL and CTL* state formulae $\mathtt{E}(p\,\mathtt{U}\,q)$ and $\mathtt{E}\,\mathtt{GF}\,p$, respectively. Another example of formula from the set $\Theta_{\{Y\},\{Z\}}$ is $\mu X.(X \vee Y) \vee \Diamond_{\geq 2}(X \vee Z \wedge \Box_{<1} p)$. On the contrary, neither $\Diamond_{\geq 1} Y$ nor $\Diamond_{\geq 1}\Box_{\geq 1} Z$ belong to $\Theta_{\{Y\},\{Z\}}$. In the first case, indeed, Y is not a one-step variable, so it is not allowed in the scope of a modality. It is still, however, a Gμ-CALCULUS[1S] formula, since it belongs, *e.g.*, to $\Theta_{\emptyset,\{Y\}}$. The second formula, instead, does not belong at all to the one-step fragment, since the variable Z occurs in the scope of two nested modalities.

The semantics of Gμ-CALCULUS[1S] is completely standard (see, *e.g.*, [34] for the full definition). Given a Kripke tree $\mathcal{T}$ and a set of variables V, let $\mathrm{Asg}_{\mathcal{T}}(\mathrm{V})$ be the set of assignments mapping each variable in V to some set of nodes of $\mathcal{T}$. For every Gμ-CALCULUS[1S] formula $\vartheta \in \Theta_{\mathrm{Z,O}}$, Kripke tree $\mathcal{T}$, and assignment $\chi \in \mathrm{Asg}_{\mathcal{T}}(\mathrm{Z}\cup\mathrm{O})$ over the free variables from $\mathrm{Z} \cup \mathrm{O}$, the denotation $[\![\vartheta]\!]^{\mathcal{T}}_{\chi}$ is defined recursively on the structure of ϑ. Here, we only report the cases for the counting modalities and fixpoint operators, where $\mathsf{post}(w)$ denotes, as usual, the set of children of the node $w \in \mathrm{T}$:

- $[\![\Diamond_{\geq k}\, \varphi]\!]^{\mathcal{T}}_{\chi} \triangleq \left\{ w \in \mathrm{T} \;\middle|\; |\mathsf{post}(w) \cap [\![\varphi]\!]^{\mathcal{T}}_{\chi}| \geq k \right\}$;
- $[\![\Box_{<k}\, \varphi]\!]^{\mathcal{T}}_{\chi} \triangleq \left\{ w \in \mathrm{T} \;\middle|\; |\mathsf{post}(w) \setminus [\![\varphi]\!]^{\mathcal{T}}_{\chi}| < k \right\}$;
- $[\![\mu X.\, \vartheta]\!]^{\mathcal{T}}_{\chi} \triangleq \bigcap \left\{ \mathrm{W} \subseteq \mathrm{T} \;\middle|\; [\![\vartheta]\!]^{\mathcal{T}}_{\chi[X\mapsto \mathrm{W}]} \subseteq \mathrm{W} \right\}$;
- $[\![\nu X.\, \vartheta]\!]^{\mathcal{T}}_{\chi} \triangleq \bigcup \left\{ \mathrm{W} \subseteq \mathrm{T} \;\middle|\; \mathrm{W} \subseteq [\![\vartheta]\!]^{\mathcal{T}}_{\chi[X\mapsto \mathrm{W}]} \right\}$.

The satisfaction relation $\mathcal{T} \models \varphi$ between a Kripke tree and sentence holds when the root of $\mathcal{T}$ belongs to $[\![\varphi]\!]$.

Let us first observe that Gμ-CALCULUS[1S] is able to express the density property of trees by means of the sentence

$$\varphi_{Den} \triangleq \nu X.\, \mu Y.\, (\Diamond_{\geq 2}\, X) \vee (\Diamond_{\geq 1}\, Y).$$

Indeed, if $\mathcal{T} \models \varphi_{Den}$, the root of $\mathcal{T}$ belongs to the denotation $\Delta \triangleq [\![\varphi_{Den}]\!]^{\mathcal{T}}_{\varnothing}$. By the semantics of greatest-fixpoint, we have $[\![\mu Y.\, (\Diamond_{\geq 2}\, X) \vee (\Diamond_{\geq 1}\, Y)]\!]^{\mathcal{T}}_{\{X\mapsto\Delta\}} = \Delta$, obtained by evaluating the least-fixpoint subformula $\mu Y.\, (\Diamond_{\geq 2}\, X) \vee (\Diamond_{\geq 1}\, Y)$ on the assignment mapping the variable X to the entire denotation $\Delta \subseteq \mathrm{T}$. This equality implies that every node v in Δ satisfies one of the following: *(a)* v has at least two distinct children in Δ; *(b)* v is able to reach a node in Δ that satisfies Property *(a)*. Hence, Δ precisely identifies the set of nodes that form a subtree each of whose nodes has two distinct strict descendants in Δ. Therefore, $\mathcal{T}$ enjoys the density property. On the other hand, if $\mathcal{T}$ enjoys the density property, there exists a set $\mathrm{M} \subseteq \mathrm{T}$ of nodes corresponding to an infinite binary-tree minor of $\mathcal{T}$. Consider the denotation $\Delta' \triangleq [\![\mu Y.\, (\Diamond_{\geq 2}\, X) \vee (\Diamond_{\geq 1}\, Y)]\!]^{\mathcal{T}}_{\{X\mapsto\mathrm{M}\}}$ of least-fixpoint subformula $\mu Y.\, (\Diamond_{\geq 2}\, X) \vee (\Diamond_{\geq 1}\, Y)$ of φ_{Den} for the assignment mapping the greatest-fixpoint variable X to M. Due to the definition of the set M, every node in it can reach at least two distinct nodes in M as well. Thus, $\mathrm{M} \subseteq \Delta'$, which implies that $\mathrm{M} \subseteq [\![\varphi_{Den}]\!]^{\mathcal{T}}_{\varnothing}$, by the semantics of greatest-fixpoint. Moreover, the root of $\mathcal{T}$ belongs to Δ', since it can reach any node in M. Therefore, $\mathcal{T} \models \varphi_{Den}$ and we have the following result.

Theorem 9. *The density property is expressible in* Gμ-CALCULUS[1S].

As an immediate corollary of Theorem 9, jointly with Theorem 1 and the observation made in [10], [30] on the inability of WMSO to characterise the class of dense trees, we obtain the following expressiveness relation.

Corollary 4. Gμ-CALCULUS[1S] $\not\leq$ MPL *and* Gμ-CALCULUS[1S] $\not\leq$ WMSO.

At this point, we can turn to the encoding of the semantics of Gμ-CALCULUS[1S] formulae into MTL. With this aim in mind, we first identify a one-step simulation property enjoyed by the modal base underlying Gμ-CALCULUS[1S]. This property rests on an ordering relation on variable assignments called *one-step simulation*. Given two assignments

- $\mathsf{tr}_x(\bot) \triangleq \bot$;
- $\mathsf{tr}_x(\top) \triangleq \top$;
- $\mathsf{tr}_x(p) \triangleq P_p(x)$;
- $\mathsf{tr}_x(X) \triangleq x \in X$;
- $\mathsf{tr}_x(\neg\varphi) \triangleq \neg\mathsf{tr}_x(\varphi)$;
- $\mathsf{tr}_x(\varphi_1 \odot \varphi_2) \triangleq \mathsf{tr}_x(\varphi_1) \odot \mathsf{tr}_x(\varphi_2)$, $\odot \in \{\wedge, \vee\}$;
- $\mathsf{tr}_x(\Diamond_{\geq k}\,\varphi) \triangleq \exists y_1, \ldots, y_k.\, (\bigwedge_{i \neq j}(y_i \neq y_j) \wedge \bigwedge_{i=1}^{k} \mathtt{child}(x, y_i)) \wedge (\bigwedge_{i=1}^{k} \mathsf{tr}_{y_i}(\varphi))$;
- $\mathsf{tr}_x(\Box_{<k}\,\varphi) \triangleq \forall y_1, \ldots, y_k.\, (\bigwedge_{i \neq j}(y_i \neq y_j) \wedge \bigwedge_{i=1}^{k} \mathtt{child}(x, y_i)) \rightarrow (\bigvee_{i=1}^{k} \mathsf{tr}_{y_i}(\varphi))$;
- $\mathsf{tr}_x(\mu X.\, \vartheta) \triangleq \neg\mathsf{tr}_x(\nu X.\, \mathtt{pnf}(\neg\vartheta[X/\neg X]))$;
- $\mathsf{tr}_x(\nu X.\, \vartheta) \triangleq \exists^{\mathrm{T}} X.\, x \in X \wedge \forall x \in X.\, \mathsf{tr}_x(\vartheta)$.

Table I: Translation function $\mathsf{tr}_x \colon \Theta_{\mathrm{Z,O}} \rightarrow$ MTL from Gμ-CALCULUS[1S] to MTL.

$\chi, \chi' \in \mathrm{Asg}(\mathrm{Z} \cup \mathrm{O})$, we state that χ is *one-step simulated* by χ' *w.r.t.* a set of nodes $\mathrm{W} \subseteq \mathrm{T}$, in symbols $\chi \sqsubseteq_{\mathrm{W}}^{\mathrm{Z,O}} \chi'$, if

- $\chi(X) \cap \mathrm{W} \subseteq \chi'(X)$, for all $X \in \mathrm{Z}$, and
- $\chi(X) \cap \mathsf{post}(\mathrm{W}) \subseteq \chi'(X)$, for all $X \in \mathrm{O}$,

where $\mathsf{post}(\mathrm{W})$ denotes, as usual, the set of children in $\mathcal{T}$ of the nodes in W. Essentially, χ' assigns to any zero-step variable at least as many elements of the context set W as χ and assigns to any one-step variable all the children of nodes in W that χ assigns. The informal reading of $\chi \sqsubseteq_{\mathrm{W}}^{\mathrm{Z,O}} \chi'$ is that χ' contains as much information about W as χ, when the visibility on W is limited to at most one step ahead.

We can now show that the semantics of the base fragment of Gμ-CALCULUS[1S] is monotone *w.r.t.* the one-step simulation relation relativised to the same set of nodes W. Informally, if an assignment χ is simulated by another assignment χ' *w.r.t.* the set of nodes W, then every node from W that belongs to the denotation of a base formula φ *w.r.t.* χ also belongs to the denotation of φ *w.r.t.* χ'. Obviously, this property, ensured by the syntactic restriction on the nesting of modal operators, is enjoyed, *e.g.*, by the one-step formulae $\Diamond_{\geq 1} X$ and $\Box_{<3} X$, but not by the non-one-step formulae $\Diamond_{\geq 1}\Diamond_{\geq 2} X$ and $\Box_{<2}\Diamond_{\geq 1} X$. The *one-step monotonicity* property is stated as follows and can easily be proved by induction on the structure of the base Gμ-CALCULUS[1S] formulae.

Lemma 7. *For every base* Gμ-CALCULUS[1S] *formula* $\varphi \in \Phi_{\mathrm{Z,O}}$*, Kripke tree* $\mathcal{T}$*, set of nodes* $\mathrm{W} \subseteq \mathrm{T}$*, and pair of assignments* $\chi, \chi' \in \mathrm{Asg}_{\mathcal{T}}(\mathrm{Z} \cup \mathrm{O})$ *satisfying* $\chi \sqsubseteq_{\mathrm{W}}^{\mathrm{Z,O}} \chi'$*, it holds that* $[\![\varphi]\!]_{\chi}^{\mathcal{T}} \cap \mathrm{W} \subseteq [\![\varphi]\!]_{\chi'}^{\mathcal{T}}$.

The monotonicity property is at the core of the "independence" property of the semantics of all Gμ-CALCULUS[1S] formulae mentioned above. Indeed, we show that every maximal connected component Δ_w of the denotation $\Delta \triangleq [\![\vartheta]\!]_{\chi}^{\mathcal{T}}$ *w.r.t.* an assignment χ of a formula ϑ rooted at some node $w \in \Delta$ (*i.e.*, a maximal subtree rooted at w and fully contained in Δ) can be also computed by using only the restriction to Δ_w of the interpretation of the fixpoint variables. Essentially, the fact that a node v of Δ_w belongs to Δ is independent of whether any other node outside (the one-step extension of) Δ_w belongs to Δ or not. In other words, disconnected parts of the denotation cannot affect each other.

The maximal connected component of a given set of nodes $\mathrm{W} \subseteq \mathrm{T}$ rooted at a node $w \in \mathrm{T}$ can be defined as

$$\mathrm{W}{\downarrow_w} \triangleq \{v \in \mathrm{W} \mid w \leq v \wedge \forall u \in \mathrm{T}.\, (w \leq u < v) \Rightarrow u \in \mathrm{W}\},$$

while the (Z, O)-restriction of an assignment $\chi \in \mathrm{Asg}_{\mathcal{T}}(\mathrm{Z} \cup \mathrm{O})$ to the (one-step extension of) $\mathrm{W} \subseteq \mathrm{T}$ is defined as

$$(\chi{\restriction_{\mathrm{W}}})(X) \triangleq \begin{cases} \chi(X) \cap \mathrm{W}, & \text{if } X \in \mathrm{Z} \setminus \mathrm{O}; \\ \chi(X) \cap \mathsf{post}(\mathrm{W}), & \text{if } X \in \mathrm{O} \setminus \mathrm{Z}; \\ \chi(X) \cap (\mathrm{W} \cup \mathsf{post}(\mathrm{W})), & \text{otherwise.} \end{cases}$$

The *"independence" property* can be formalised as follows and proved by induction on the nesting of fixpoint operators, where the base case is proved by exploiting Lemma 7.

Lemma 8. *For every fixpoint* Gμ-CALCULUS[1S] *formula* $\vartheta \in \Theta_{\mathrm{Z,O}}$*, Kripke tree* $\mathcal{T}$*, assignment* $\chi \in \mathrm{Asg}_{\mathcal{T}}(\mathrm{Z} \cup \mathrm{O})$*, and node* $w \in \Delta \triangleq [\![\vartheta]\!]_{\chi}^{\mathcal{T}}$*, it holds that* $\Delta_w \triangleq \Delta{\downarrow_w} = [\![\vartheta]\!]_{\chi'}^{\mathcal{T}}{\downarrow_w}$*, where* $\chi' \triangleq \chi{\restriction_{\Delta_w}}$.

Table I reports a translation function $\mathtt{tr} \colon \mathrm{Vr}_1 \rightarrow (\Theta_{\mathrm{Z,O}} \rightarrow$ MTL) turning each Gμ-CALCULUS[1S] formula into an equivalent MTL one. All cases, but those for the fixpoint operators, are standard (see, *e.g.*, [31]) and reported here just for completeness. The real interesting case is the one for the greatest-fixpoint formulae. The idea here is to exploit the "independence" property stated above and reduce the condition for a node w to belong to the greatest-fixpoint to the condition that w belongs to a subtree which is also a post-fixpoint. The translation of least-fixpoint formulae combines the translation for the greatest-fixpoint with the well-known μ-CALCULUS duality property $\mu X.\, \vartheta \equiv \neg\nu X.\, \neg\vartheta[X/\neg X]$ connecting the two fixpoint operators, where $\vartheta[X/\neg X]$ denotes the formula obtained by uniformly replacing each occurrence of the variable X in ϑ with its negation $\neg X$. Note that, since negations between fixpoint operators are not allowed in Gμ-CALCULUS[1S], we transform the formula $\neg\vartheta[X/\neg X]$ into an equivalent one in positive normal form via the auxiliary function $\mathtt{pnf} \colon$ Gμ-CALCULUS $\rightarrow$ Gμ-CALCULUS. In this way, we ensure that, if ϑ is a Gμ-CALCULUS[1S] formula, $\mathtt{pnf}(\neg\vartheta[X/\neg X])$ is a Gμ-CALCULUS[1S] formula as well.

At this point, the following result can be obtained via structural induction, by showing that, for every formula $\vartheta \in \Theta_{\mathrm{Z,O}}$, Kripke tree $\mathcal{T}$, assignment $\chi \in \mathrm{Asg}_{\mathcal{T}}(\mathrm{Z} \cup \mathrm{O})$, and node $w \in \mathrm{T}$, it holds that $w \in [\![\vartheta]\!]_{\chi}^{\mathcal{T}}$ *iff* $\mathcal{T}, \{x \mapsto w\}, \chi \models \mathsf{tr}_x(\vartheta)$.

Theorem 10. Gμ-CALCULUS[1S] $\leq$ MTL.

B. Alternation-Free One-Step Graded μ-CALCULUS

Alternation-Free Modal μ-CALCULUS (AFμ-CALCULUS), namely the fragment of μ-CALCULUS where no alternation of fixpoint operators is allowed, is a quite expressive, still much easier, fragment of Modal μ-CALCULUS that can be encoded in WMSO, when finitely-branching trees are considered [3], [26], [14], [15]. Here we analyse the alternation-free fragment

- $\mathsf{tr}_x(\nu X.\,\vartheta) \triangleq \begin{cases} \neg\mathsf{tr}_x(\mu X.\,\mathit{pnf}(\neg(\vartheta[X/\neg X][Y/\neg Y]))), & \text{if } \vartheta \in \Phi^{\mathtt{AF}}_{\mathrm{Z}',\mathrm{O}'}; \\ \mathsf{tr}_x(\vartheta[X/Y]), & \text{otherwise;} \end{cases}$
- $\mathsf{tr}_x(\mu X.\,\vartheta) \triangleq \begin{cases} \exists^{\mathrm{T}} X.\, x \in X \wedge \forall x \in X.\, \exists^{\mathrm{T}} Y.\, \mathtt{mst}(Y, X, x) \wedge \mathsf{tr}_x(\varphi[X/Y]\!\downarrow_Y), & \text{if } \vartheta \in \Phi^{\mathtt{AF}}_{\mathrm{Z}',\mathrm{O}'}; \\ \mathsf{tr}_x(\vartheta[X/Y]), & \text{otherwise;} \end{cases}$

where $\mathrm{Z}' \triangleq \mathrm{Z} \cup \{Y, X\}$, $\mathrm{O}' \triangleq \mathrm{O} \cup \{Y, X\}$, and $\mathtt{mst}(Y, X, x) \triangleq \forall y.\,(y \in Y) \leftrightarrow (x \leq y \wedge \forall z.\,(x \leq z \leq y) \rightarrow z \in X)$, for some fresh fixpoint variable $Y \in \mathrm{Vr}_2 \setminus \mathsf{free}(\vartheta)$, where the two second-order existential quantifiers range over finite trees.

Table II: Translation function $\mathsf{tr}_x \colon \Theta^{\mathtt{AF}}_{\mathrm{Z,O}} \to \mathsf{WMTL}$ from $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ to WMTL, on finitely-branching trees.

of $\mathsf{G}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ ($\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$) and prove that its semantics can be encoded in WMTL on the same class of trees. Note that $\mathsf{AF}\mu\text{-}\textsc{Calculus}$ is known to be equivalent to WMSO. Hence, dropping the alternation-freeness constraint from $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ would immediately lead us outside WMSO and, therefore, WMTL. We also look at the expressive power of $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ in comparison with WMPL and a graded-on-path extension of CTL, showing that this fragment is an interesting logic on its own.

Let $\Theta^{\mathtt{AF}}_{\mathrm{Z,O}}$ and $\Phi^{\mathtt{AF}}_{\mathrm{Z,O}}$ denote the subsets of $\Theta_{\mathrm{Z,O}}$ and $\Phi_{\mathrm{Z,O}}$, respectively, containing all and only alternation-free formulae.

It is immediate to see that the $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ sentence $\mu X.\, a \vee (\Box_{<1} X) \in \Theta^{\mathtt{AF}}_{\emptyset,\emptyset}$, equivalent to the CTL formula $\mathtt{AF}\, a$, encodes the a-acceptance property.

Theorem 11. *The a-acceptance property is expressible in* $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$.

As an immediate corollary of Theorems 11 and 3, we obtain the following result.

Corollary 5. $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}] \not\leq \mathsf{WMPL}$.

It is well-known that CTL is strictly subsumed by the $\mathsf{AF}\mu\text{-}\textsc{Calculus}$ [22], [18], thanks to the one-step unfolding properties of the temporal operators $\mathtt{U}$ and $\mathtt{R}$. Thus, obviously, $\mathsf{CTL} < \mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ holds as well. We can show, however, a stronger property. In [7], [8], [9], different counting variants of CTL and $\mathsf{CTL}^\star$ than those considered in the previous section have been proposed, called *Graded* CTL (GCTL) and *Graded* $\mathsf{CTL}^\star$ ($\mathsf{GCTL}^\star$), where classic path quantifiers $\mathtt{E}$ and $\mathtt{A}$ are replaced with their graded versions $\mathtt{E}^{\geq k}$ and $\mathtt{A}^{<k}$. These can be read informally as "there are at least k paths" and "all but less than k paths", respectively. Now, Theorem 5.4 of [9] shows that GCTL can be encoded into $\mathsf{G}\mu\text{-}\textsc{Calculus}$ via a generalisation of the classic one-step unfolding properties. A closer inspection of the proof, though, reveals that the translation only uses fixpoint variables within the range of a single modal operators. Hence, the following can be obtained.

Theorem 12. $\mathsf{GCTL} \leq \mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$.

To prove that $\mathsf{AFG}\mu\text{-}\textsc{Calculus}[1\mathtt{S}] \leq \mathsf{WMTL}$ on finitely-branching trees, we first need to introduce some notation and prove some auxiliary properties that hold true even for the full $\mathsf{G}\mu\text{-}\textsc{Calculus}$.

Given a $\mathsf{G}\mu\text{-}\textsc{Calculus}$ formula φ and a variable $X \in \mathrm{Vr}_2$, we denote with $\varphi\!\downarrow_X$ and $\varphi\!\uparrow_X$, called *time-zero suppression*, the formulae obtained from φ by replacing each free occurrence of X not in the scope of a modal operator with $\bot$ and $\top$, respectively. *E.g.*, $(\mu Y.\,(p \vee X) \wedge \Diamond(X \vee Y))\!\downarrow_X = \mu Y.\,(p \vee \bot) \wedge \Diamond_{\geq 1}(X \vee Y)$ and $(X \wedge \mu X.\,(p \vee X) \wedge \Box_{<2} X)\!\uparrow_X = \top \wedge \mu X.\,(p \vee X) \wedge \Box_{<2} X$. Intuitively, the two time-zero suppressions ensure that the membership of a node to the denotation of the resulting formula does not depend on the interpretation of the specified variable at "time-zero".

Proposition 6. *For every* $\mathsf{G}\mu\text{-}\textsc{Calculus}$ *formula φ, variable $X \in \mathsf{free}(\varphi)$, Kripke tree $\mathcal{T} = (\mathrm{T}, \mathit{Lab})$, set of nodes $\mathrm{W} \subseteq \mathrm{T}$, node $w \in \mathrm{T}$, and assignment $\chi \in \mathrm{Asg}_{\mathcal{T}}(\mathsf{free}(\varphi))$, the following holds true:*

a) $[\![\varphi\!\downarrow_X]\!]^{\mathcal{T}}_{\chi} \subseteq [\![\varphi\!\uparrow_X]\!]^{\mathcal{T}}_{\chi}$;
b) $[\![\varphi]\!]^{\mathcal{T}}_{\chi} = [\![\varphi\!\downarrow_X \vee (X \wedge \varphi\!\uparrow_X)]\!]^{\mathcal{T}}_{\chi}$;
c) $w \in [\![\varphi\!\downarrow_X]\!]^{\mathcal{T}}_{\chi[X \mapsto \mathrm{W}]}$ iff $w \in [\![\varphi\!\downarrow_X]\!]^{\mathcal{T}}_{\chi[X \mapsto \mathrm{W} \setminus \{w\}]}$;
d) $w \in [\![\varphi\!\downarrow_X]\!]^{\mathcal{T}}_{\chi[X \mapsto \mathrm{W}]}$ iff $w \in [\![\varphi]\!]^{\mathcal{T}}_{\chi[X \mapsto \mathrm{W} \setminus \{w\}]}$.

Items a-c of the above proposition can easily be obtained by structural induction on the formula φ. In particular, Item a is used in the proof of Item b (see [4, Lemma 9.1.1] for an idea of proof), while Item d is an immediate consequence of Items b and c. It may be interesting to observe that Item b is a generalisation of Shannon's lemma for Boolean function. Moreover, Item c formally states that a node belongs to the denotation of a formula with a variable X suppressed regardless of its membership in the interpretation of that variable.

Essentially, the time-zero suppressions of a formula ϑ are used in the following to make sure that the presence of a node in the denotation of ϑ (typically, the argument of some fixpoint operator) is granted solely by the presence of its descendants and its inclusion in the assignment is indeed redundant. This is crucial to prove the next result, where we characterise the semantics of alternation-free least-fixpoint formulae, by means of finite trees. Specifically, thanks to Kleene's Theorem and the fact that the underlying tree is finitely-branching, the witness for the membership of a node to the denotation of these formulae, once the fixpoint variable is suppressed, is always finite. Moreover, Item d of Proposition 6 ensures that such a witness is indeed a least fixpoint (and not an arbitrary one), since a node belongs to the denotation independently of its membership to the interpretation of the fixpoint variable.

Lemma 9. *For every* $\mathsf{G}\mu\text{-}\textsc{Calculus}[1\mathtt{S}]$ *formula $\vartheta = \mu X_1 \ldots \mu X_k.\, \varphi \in \Theta_{\mathrm{Z,O}}$, with $\varphi \in \Phi_{\mathrm{Z} \cup \{X_1,\ldots,X_k\}, \mathrm{O} \cup \{X_1,\ldots,X_k\}}$, finitely-branching Kripke tree $\mathcal{T} = (\mathrm{T}, \mathit{Lab})$, node $w \in \mathrm{T}$, and assignment $\chi \in \mathrm{Asg}_{\mathcal{T}}(\mathrm{Z} \cup \mathrm{O})$, the following properties are equivalent:*

- $w \in [\![\vartheta]\!]^{\mathcal{T}}_{\chi}$;
- *there exists a finite tree $\mathrm{W} \subseteq \mathrm{T}$ with $w \in \mathrm{W}$ such that $v \in [\![\varphi[X_1/Y, \ldots, X_k/Y]\!\downarrow_Y]\!]^{\mathcal{T}}_{\chi[Y \mapsto \mathrm{W}\downarrow_v]}$, for all $v \in \mathrm{W}$.*

- $\mathtt{tr}_{X,x}(\varphi_1 \mathbb{U}_\phi \varphi_2) \triangleq \exists^{\mathrm{T}} X'.\, (\mathtt{nb}(X') \wedge X' \sqsubset_\phi^X X) \wedge (\mathtt{tr}_{X',x}(\varphi_2) \wedge \forall^{\mathrm{T}} X''.\, (\mathtt{nb}(X'') \wedge X' \sqsubset_\phi^X X'' \sqsubset_\phi^X X) \rightarrow \mathtt{tr}_{X'',x}(\varphi_1))$;
- $\mathtt{tr}_{X,x}(\varphi_1 \mathbb{R}_\phi \varphi_2) \triangleq \forall^{\mathrm{T}} X'.\, (\mathtt{nb}(X') \wedge X' \sqsubset_\phi^X X) \rightarrow (\mathtt{tr}_{X',x}(\varphi_2) \vee \exists^{\mathrm{T}} X''.\, (\mathtt{nb}(X'') \wedge X' \sqsubset_\phi^X X'' \sqsubset_\phi^X X) \wedge \mathtt{tr}_{X'',x}(\varphi_1))$;
- $\mathtt{tr}_{X,x}(\varphi_1 \mathbb{S}_\phi \varphi_2) \triangleq \exists^{\mathrm{T}} X'.\, (\mathtt{nb}(X') \wedge X \sqsubset_\phi^X X') \wedge (\mathtt{tr}_{X',x}(\varphi_2) \wedge \forall^{\mathrm{T}} X''.\, (\mathtt{nb}(X'') \wedge X \sqsubset_\phi^X X'' \sqsubset_\phi^X X') \rightarrow \mathtt{tr}_{X'',x}(\varphi_1))$;
- $\mathtt{tr}_{X,x}(\varphi_1 \mathbb{B}_\phi \varphi_2) \triangleq \forall^{\mathrm{T}} X'.\, (\mathtt{nb}(X') \wedge X \sqsubset_\phi^X X') \rightarrow (\mathtt{tr}_{X',x}(\varphi_2) \vee \exists^{\mathrm{T}} X''.\, (\mathtt{nb}(X'') \wedge X \sqsubset_\phi^X X'' \sqsubset_\phi^X X') \wedge \mathtt{tr}_{X'',x}(\varphi_1))$;

where $\mathtt{nb}(X) \triangleq \forall x \in X.\, \exists y \in X.\, \mathtt{child}(x, y)$ and $Y \sqsubset_\phi^X Z \triangleq (Y \subset Z) \wedge (\forall y \in Y.\, \forall z \in Z.\, (\mathtt{tr}_{X,y}(\phi) \wedge \mathtt{child}(y, z)) \rightarrow z \in Y)$.

Table III: Translation function $\mathtt{tr}_{X,x}\colon \mathsf{STL}^\star \rightarrow \mathsf{MTL}$ from $\mathsf{STL}^\star$ to MTL.

Table II reports the two cases for which the translation function $\mathtt{tr}\colon \mathrm{Vr}_1 \rightarrow (\Theta^{\mathtt{AF}}_{\mathrm{Z,O}} \rightarrow \mathsf{WMTL})$ from AFGμ-CALCULUS[1S] into WMTL differs from the one for the general case. As opposed to the original function, in this case it is the translation of the greatest-fixpoint formulae that is derived from the ones for the least-fixpoint via the duality property. The translation for the latter is then obtained by simply encoding the property stated in the above lemma, where we use the auxiliary formula $\mathtt{mst}(Y, X, x)$ to identify the maximal subtree Y fully included in the witness X and rooted at some given node x. Note also that all least-fixpoint operators are merged together and transformed as a monolithic entity, thanks to the following classic equivalence (see, [4, Proposition 1.3.2]): $\mu X_1 \ldots \mu X_k.\, \varphi \equiv \mu Y.\, \varphi[X_1/Y, \ldots, X_k/Y]$. The expressiveness result reported below is obtained by exploiting the same line of reasoning used for the proof of Theorem 10.

Theorem 13. AFGμ-CALCULUS[1S] $\leq$ WMTL *on finitely-branching trees.*

C. Substructure Temporal Logic

In [5], [6], an extension of CTL*, called Substructure Temporal Logic (STL*), is proposed, with the distinctive feature of being able to implicitly predicate over non-blocking substructures of the underlying Kripke model. When such models are trees, this reduces to reasoning about subtrees. The logic is obtained by adding four (future and past) temporal-like operators $\mathbb{U}$ (until), $\mathbb{R}$ (release), $\mathbb{S}$ (since), and $\mathbb{B}$ (before), called *semilattice operators*, whose interpretation is given relative to the join semilattice induced by the partial order on subtrees.

For the sake of space, here we only recall the semantics of the until formulae $\varphi_1 \mathbb{U}_\phi \varphi_2$, which is given for a fixed Kripke-tree model $\mathcal{T}^*$ and one of its non-blocking subtrees $\mathcal{T}$. The satisfaction relation $\mathcal{T} \models^{\mathcal{T}^*} \varphi_1 \mathbb{U}_\phi \varphi_2$ holds if there exists a non-blocking ϕ-preserving strict subtree $\mathcal{T}'$ of $\mathcal{T}$ such that $\mathcal{T}' \models^{\mathcal{T}^*} \varphi_2$ and, for all non-blocking ϕ-preserving trees $\mathcal{T}''$ strictly lying between $\mathcal{T}'$ and $\mathcal{T}$, it holds that $\mathcal{T}'' \models^{\mathcal{T}^*} \varphi_1$. Here, ϕ-preserving means that all the children in $\mathcal{T}$ of the nodes in $\mathcal{T}'$ that satisfy the formula ϕ are kept in $\mathcal{T}'$ as well.

Table III reports the translation function $\mathtt{tr}\colon \mathrm{Vr}_2 \times \mathrm{Vr}_1 \rightarrow (\mathsf{STL}^\star \rightarrow \mathsf{MTL})$ from STL* into MTL that encodes the semantics of the four semilattice operators. The formula $\mathtt{nb}(X)$ encodes the non-blocking property for the tree X, while $Y \sqsubset_\phi^X Z$ encodes the ϕ-preserving subtree relation. The encoding of the remaining part of the logic, which corresponds to CTL*, is the same as the one into MPL proposed in [29] and is not reported here. The comparison between STL* and MPL was left open in [6]. Thanks again to Theorem 1 and the fact that STL* can express the density property [6], the following theorem answers that question.

Theorem 14. $\mathsf{STL}^\star \leq \mathsf{MTL}$ *and* $\mathsf{STL}^\star \not\leq \mathsf{MPL}$.

Section 5 of [6] carries out an analysis of several problems involving reasoning about games. There it is shown that STL* can encode LTL *reactive synthesis* [45], CTL* *module checking* [35], and the solution of turn-based/concurrent zero/non-zero-sum games with LTL, hence FO-definable, goals. Thanks to the above theorem, the same abilities are inherited by MTL.

V. DISCUSSION

We have introduced and studied MTL, a variant of MSO over non-blocking trees, where the domain of the second-order variables is restricted to subtrees of the original structure. An extensive comparison of the expressive power of MTL, and its finite (WMTL) and co-finite (coWMTL) variants against the corresponding variants of MSO and MPL has also been provided. Unlike MSO, which can quantify over non-connected sets of nodes, MTL, much like MPL, is designed to predicate over connected sets of nodes only. As a consequence, while the result that MSO (*resp.*, WMSO, coWMSO) is strictly more expressive than MTL (*resp.*, WMTL, coWMTL) may be not surprising and somewhat expected, far less obvious is the result that the same relationship holds between MTL and MPL. As we have shown, this essentially follows from the ability of MTL to express meaningful properties of trees, such as the density property stipulating that the set of paths of the underlying tree is uncountable or, equivalently, that the tree contains a full binary tree as a minor, which cannot be captured by means of path quantifications only.

These results call for a deeper investigation of the relationships between MTL and temporal logics. Here we have begun this analysis by showing that STL*, an expressive temporal logic able to capture reasoning about strategies and games in a quite general way, can easily be embedded into MTL. This suggests that MTL is indeed enough for reasoning about games with FO-definable goals. A deeper result, though, is obtained by identifying the one-step fragment of the Graded μ-CALCULUS that restricts free variables to occur within the scope of at most a single modal operator. Such a restriction, in turn, essentially prevents the formulae of this language from being able to predicate over non-connected sets. We show, in fact, that this fragment is contained in MTL but cannot be captured by MPL, as it can still express the density property. This study can be viewed as a contribution of interest on its own, since the Modal μ-CALCULUS, although often considered an unfriendly assembly-like specification language, is very important from a practical viewpoint. *Symbolic model-checking*

tools, indeed, exist that compute the denotation of fixpoint expressions over the set of states of the model to verify, see, *e.g.*, [18].

In addition to the question still open of whether WMTL is subsumed by MPL on finitely-branching trees, the most relevant problem still to address is the completeness *w.r.t.* MTL and WMTL of the two one-step fragments (or slight generalisations thereof) of the Gμ-CALCULUS. Particular care will be required for the treatment of the alternation-free fragment on arbitrary-branching trees, where the corresponding fragment of MTL would conceivably not be WMTL but a Noetherian variant [15] of MTL.

REFERENCES

[1] R. Alur, T. Henzinger, and O. Kupferman, "Alternating-Time Temporal Logic," in *FOCS'97*. IEEECS, 1997, pp. 100–109.

[2] ——, "Alternating-Time Temporal Logic," *JACM*, vol. 49, no. 5, pp. 672–713, 2002.

[3] A. Arnold and D. Niwinski, "Fixed Point Characterization of Weak Monadic Logic Definable Sets of Trees," in *Tree Automata and Languages*. North-Holland, 1992, pp. 159–188.

[4] ——, *Rudiments of μ-Calculus*, ser. Studies in Logic and the Foundations of Mathematics. North-Holland, 2001.

[5] M. Benerecetti, F. Mogavero, and A. Murano, "Substructure Temporal Logic," in *LICS'13*. IEEECS, 2013, pp. 368–377.

[6] ——, "Reasoning About Substructures and Games," *TOCL*, vol. 16, no. 3, pp. 25:1–46, 2015.

[7] A. Bianco, F. Mogavero, and A. Murano, "Graded Computation Tree Logic," in *LICS'09*. IEEECS, 2009, pp. 342–351.

[8] ——, "Graded Computation Tree Logic with Binary Coding," in *CSL'10*, ser. LNCS 6247. Springer, 2010, pp. 125–139.

[9] ——, "Graded Computation Tree Logic," *TOCL*, vol. 13, no. 3, pp. 25:1–53, 2012.

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

[11] J. Büchi, "Weak Second-Order Arithmetic and Finite Automata," *MLQ*, vol. 6, no. 1-6, pp. 66–92, 1960.

[12] ——, "On a Decision Method in Restricted Second-Order Arithmetic," in *ICLMPS'62*. Stanford University Press, 1962, pp. 1–11.

[13] ——, "On a Decision Method in Restricted Second Order Arithmetic," in *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1966, vol. 44, pp. 1–11.

[14] F. Carreiro, A. Facchini, Y. Venema, and F. Zanasi, "Weak MSO: Automata and Expressiveness Modulo Bisimilarity," in *CSL'14 & LICS'14*. ACM, 2014, pp. 27:1–27.

[15] ——, "The Power of the Weak," *TOCL*, vol. 21, no. 2, pp. 15:1–47, 2020.

[16] K. Chatterjee, T. Henzinger, and N. Piterman, "Strategy Logic," in *CONCUR'07*, ser. LNCS 4703. Springer, 2007, pp. 59–73.

[17] ——, "Strategy Logic," *IC*, vol. 208, no. 6, pp. 677–693, 2010.

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

[19] B. Courcelle, "The Monadic Second-Order Logic of Graphs, II: Infinite Graphs of Bounded Width," *MST*, vol. 21, no. 4, pp. 187–221, 1989.

[20] ——, "The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs," *IC*, vol. 85, no. 1, pp. 12–75, 1990.

[21] ——, "The Monadic Second-Order Logic of Graphs III: Tree-Decompositions, Minor and Complexity Issues," *RAIROTIA*, vol. 26, no. 3, pp. 257–286, 1992.

[22] E. Emerson, "Model Checking and the Mu-calculus," in *DCFM'96*, ser. DIMACS'31. AMS, 1996, pp. 185–214.

[23] E. Emerson and E. Clarke, "Characterizing Correctness Properties of Parallel Programs Using Fixpoints," in *ICALP'80*, ser. LNCS 85. Springer, 1980, pp. 169–181.

[24] E. Emerson and J. Halpern, ""Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time," *JACM*, vol. 33, no. 1, pp. 151–178, 1986.

[25] E. Emerson and C.-L. Lei, "Efficient Model Checking in Fragments of the Propositional muCalculus," in *LICS'86*. IEEECS, 1986, pp. 267–278.

[26] A. Facchini, Y. Venema, and F. Zanasi, "A Characterization Theorem for the Alternation-Free Fragment of the Modal μ-Calculus," in *LICS'13*. IEEECS, 2013, pp. 478–487.

[27] K. Fine, "In So Many Possible Worlds," *NDJFL*, vol. 13, pp. 516–520, 1972.

[28] Y. Gurevich and S. Shelah, "The Decision Problem for Branching Time Logic," *JSL*, vol. 50, no. 3, pp. 668–681, 1985.

[29] T. Hafer and W. Thomas, "Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree," in *ICALP'87*, ser. LNCS 267. Springer, 1987, pp. 269–279.

[30] T. Idziaszek, M. Skrzypczak, and M. Bojańczyk, "Regular Languages of Thin Trees," *TCS*, vol. 58, no. 4, pp. 614–663, 2016.

[31] 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*, vol. 61, no. 3-4, pp. 247–265, 2004.

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

[33] D. Kozen, "Results on the Propositional muCalculus," *TCS*, vol. 27, no. 3, pp. 333–354, 1983.

[34] O. Kupferman, U. Sattler, and M. Vardi, "The Complexity of the Graded muCalculus," in *CADE'02*, ser. LNCS 2392. Springer, 2002, pp. 423–437.

[35] O. Kupferman, M. Vardi, and P. Wolper, "Module Checking," *IC*, vol. 164, no. 2, pp. 322–344, 2001.

[36] F. Mogavero, A. Murano, G. Perelli, and M. Vardi, "Reasoning About Strategies: On the Model-Checking Problem," *TOCL*, vol. 15, no. 4, pp. 34:1–42, 2014.

[37] ——, "Reasoning About Strategies: On the Satisfiability Problem," *LMCS*, vol. 13, no. 1:9, pp. 1–37, 2017.

[38] F. Mogavero, A. Murano, and M. Vardi, "Reasoning About Strategies," in *FSTTCS'10*, ser. LIPIcs 8. Leibniz-Zentrum fuer Informatik, 2010, pp. 133–144.

[39] F. Moller and A. Rabinovich, "On the Expressive Power of CTL*," in *LICS'99*. IEEECS, 1999, pp. 360–368.

[40] ——, "Counting on CTL*: On the Expressive Power of Monadic Path Logic," *IC*, vol. 184, no. 1, pp. 147–159, 2003.

[41] D. Muller, A. Saoudi, and P. Schupp, "Alternating Automata, the Weak Monadic Theory of Trees and its Complexity," in *ICALP'86*, ser. LNCS 226. Springer, 1984, pp. 275–283.

[42] ——, "Alternating Automata, the Weak Monadic Theory of Trees and its Complexity," *TCS*, vol. 97, no. 2, pp. 233–244, 1992.

[43] D. Muller and P. Schupp, "Pushdown Automata, Graphs, Ends, Second-Order Logic, and Reachability Problems," in *STOC'81*. ACM, 1981, pp. 46–54.

[44] ——, "The Theory of Ends, Pushdown Automata, and Second-Order Logic," *TCS*, vol. 37, pp. 51–75, 1985.

[45] A. Pnueli and R. Rosner, "On the Synthesis of a Reactive Module," in *POPL'89*. ACM, 1989, pp. 179–190.

[46] M. Rabin, "Decidability of Second-Order Theories and Automata on Infinite Trees," *TAMS*, vol. 141, pp. 1–35, 1969.

[47] ——, "Weakly Definable Relations and Special Automata," in *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1970, vol. 59, pp. 1–23.

[48] M. Rabin and D. Scott, "Finite Automata and their Decision Problems," *IBMJRD*, vol. 3, pp. 115–125, 1959.

[49] B. ten Cate and A. Facchini, "Characterizing EF over Infinite Trees and Modal Logic on Transitive Graphs," in *MFCS'11*, ser. LNCS 6907. Springer, 2011, pp. 290–302.

[50] W. Thomas, "On Chain Logic, Path Logic, and First-Order Logic over Infinite Trees," in *LICS'87*. IEEECS, 1987, pp. 245–256.

[51] I. Walukiewicz, "Monadic Second Order Logic on Tree-Like Structures," in *STACS'96*, ser. LNCS 1046. Springer, 1996, pp. 401–413.

[52] ——, "Pushdown Processes: Games and Model-Checking," *IC*, vol. 164, no. 2, pp. 234–263, 2001.

[53] P. Wolper, "Temporal Logic Can Be More Expressive," *IC*, vol. 56, no. 1-2, pp. 72–99, 1983.