Eliminating recursion in the μ-calculus*

Martin Otto

RWTH Aachen

Consider the following problem: given a formula of the modal μ-calculus, decide whether this formula is equivalently expressible in basic modal logic. It is shown that this problem is decidable, in fact in deterministic exponential time. The decidability result can be obtained through a model theoretic reduction to the monadic second-order theory of the complete binary tree, which by Rabin's classical result is decidable, albeit of non-elementary complexity. An improved analysis based on tree automata yields an exponential time decision procedure.

1. Introduction

The propositional μ-calculus L_μ has, since its introduction in its present form in [12], emerged as one of the major logical formalism that can deal with interesting aspects of the dynamic and temporal behaviour of processes or programs. As such, it comprises the expressive power of several other well developed logical formalism for reasoning about transition systems, among them computation tree logic CTL and propositional dynamic logic PDL. Conceptually and model theoretically the μ-calculus is a modal logic. It extends propositional modal logic ML by a least fixed point operation, and it shares with basic modal logic the crucial semantic property of being invariant under bisimulation. The least fixed point construct, which is essentially second-order in nature, adds to modal logic a powerful, yet tractable form of recursion. It is this aspect of recursion that boosts the expressiveness of the μ-calculus in allowing it to express truly dynamic features of transition systems that go far beyond the more static and local properties expressible in basic modal logic. Liveness, safety or termination conditions are typical examples of L_μ-definable properties. Given the broad applicability of the μ-calculus and its fragments for specification and model checking uses, it is natural to consider the issue whether a formalization of some supposedly interesting condition on transition systems requires the use of the recursive features of the μ-calculus in an essential way. It can be that although some given L_μ-specification syntactically involves μ-constructs, it is logically equivalent to a much simpler, static and local assertion in basic modal logic.

Similar questions about the possibility to eliminate recursion have of course been asked and investigated in other contexts. The so-called *boundedness problem* for Datalog programs, which arises in connection with the issue of database query optimization, is a case in point. In the context of classical model theory,

* This is an extension of the original submission; the EXPTIME result, based on an automata theoretic analysis, is new here. Moreover, the original model theoretic approach has been simplified. I am very grateful to Moshe Vardi for having, with his comments on an earlier version, inspired these improvements.

boundedness was originally introduced and studied by Barwise and Moschovakis [1]. Several results in the first-order context and for the applications to Datalog query optimization show the boundedness problem to be undecidable even in very restricted settings [7, 9, 11]. Probably the strongest known exception concerns monadic Datalog (or the boundedness of simple monadic fixed points over existential first-order formulae without equality or negation), shown to be decidable in [3].

In the present paper we propose a proof that the eliminability of recursion from arbitrarily nested L_μ-formulae does indeed constitute a decidable problem.

Main Theorem *The following problem is decidable, in fact even in* EXPTIME*: given a formula in the modal μ-calculus, decide whether this formula can equivalently be expressed in plain modal logic.*

By way of interpreting this result in a somewhat wider context, we recall how ML and L_μ are characterized as exactly the bisimulation-invariant fragments of first-order logic and monadic second-order logic, respectively, see [2] and [10].

Theorem (van Benthem) *A first-order formula $\varphi(x)$ is equivalent to a formula of* ML *if and only if the class of its models,* $\mathrm{Mod}(\varphi)$*, is closed under bisimulation.*

Theorem (Janin, Walukiewicz) *A monadic second-order formula $\varphi(x)$ is equivalent to a formula of* L_μ *if and only if* $\mathrm{Mod}(\varphi)$ *is closed under bisimulation.*

In view of these characterizations one may rephrase our main result as follows: given bisimulation-invariance, the distinction between first-order and true monadic second-order becomes decidable. This provides a nice analogy between the bisimulation-invariant scenario and the much more limited scenario of word structures. For word structures the corresponding distinction is known to be decidable due to the classical results of Büchi, Elgot, Trakhtenbrot and Schützenberger, McNaughton, Papert, since it coincides with star-freeness of regular languages, see [15].

Turning back to the related issue of boundedness, we may look at the boundedness problem for modal logic as a special case of our decision problem: given a formula $\varphi(X)$ of modal logic in which X occurs only positively, decide whether there is some $n \in \mathbb{N}$ such that the least fixed point $\mu_X\varphi(X)$ associated with $\varphi(X)$ is always reached within n iterations. By a straightforward variation of a theorem of Barwise and Moschovakis [1], $\varphi(X)$ is bounded if and only if $\mu_X\varphi(X)$ is ML-definable. Thus, our main theorem implies in particular the decidability of the boundedness problem for modal formula. This contrasts sharply with the undecidability of the boundedness problem for two-variable first-order logic as established in [11], and adds to the comparative study of modal versus two-variable logics which has emerged in related research, see e.g. [17, 8].

Following preparations in Section 2, we shall complete in Section 3 the proof of the decidability claim of the main theorem by a reduction to S2S based on a bounded branching property for the issue of ML-expressibility. In the final Section 4 we then present an alternative automata theoretic analysis which furthermore yields the EXPTIME result.

2. Basic definitions and Preliminaries

Tree structures. Since the μ-calculus and modal logic satisfy the tree model property, it will throughout suffice to consider tree structures rather than arbitrary Kripke structures. We shall also restrict attention to the notationally simpler case in which only a single binary transition relation (accessibility relation) E is present. For us, therefore, a tree structure of type $\tau = \{P_1, \ldots, P_l\}$ is a structure $\mathfrak{A} = (A, E^{\mathfrak{A}}, P_1^{\mathfrak{A}}, \ldots, P_l^{\mathfrak{A}}, 0^{\mathfrak{A}})$ where $(A, E^{\mathfrak{A}}, 0^{\mathfrak{A}})$ is a tree with root $0^{\mathfrak{A}}$ and the $P_i^{\mathfrak{A}}$ are subsets A. Here (A, E) being a tree with root a means that a is the unique element of zero in-degree w.r.t. E and that every element is reachable from a on a unique directed E-path (whose length is the *height* of that element). The following classes of tree structures will be important:

(i) $\mathcal{T}[\tau]$ consisting of all tree structures of type τ;

(ii) $\mathcal{T}_n[\tau] \subseteq \mathcal{T}[\tau]$ consisting of those tree structures whose branching is bounded by n;

(iii) $\mathcal{T}_{n;m}[\tau] \subseteq \mathcal{T}[\tau]$ consisting of those tree structures whose branching is bounded by n in all nodes of height less than m.

For an element a of a tree structure $\mathfrak{A}$ we denote by $\langle a \rangle$ the elements of the subtree rooted at a, by $\mathfrak{A} \upharpoonright \langle a \rangle$ that subtree itself. By $\langle a \rangle^m$ we denote the set of elements whose height in $\mathfrak{A} \upharpoonright \langle a \rangle$ is at most m, by $\mathfrak{A} \upharpoonright \langle a \rangle^m$ the induced subtree rooted a. We denote by $E^{\mathfrak{A}}[a]$ the set of immediate E-successors of a in $\mathfrak{A}$.

Prunings and end extensions. If $\mathfrak{A} \subseteq \mathfrak{B}$ and both $\mathfrak{A}$ and $\mathfrak{B}$ are tree structures we call $\mathfrak{A}$ a *pruning* of $\mathfrak{B}$ to stress the view that $\mathfrak{A}$ is obtained from $\mathfrak{B}$ through cutting away subtrees. Note that $\mathfrak{A} \subseteq \mathfrak{B}$ for tree structures implies that $A \subseteq B$ is an *initial* subset in the tree $\mathfrak{B}$, i.e. $0^{\mathfrak{B}} \in A$ and A is $E^{\mathfrak{B}}$-connected.

We say that $\mathfrak{A}$ is a *finite pruning* of $\mathfrak{B}$ if there is some $n \in \mathbb{N}$ such that for all $a \in A$ whose distance from the root is at least n, $E^{\mathfrak{A}}[a] = E^{\mathfrak{B}}[a]$.

$\mathfrak{B}$ is an *end extension* of the tree $\mathfrak{A}$, $\mathfrak{A} \subseteq_{\text{end}} \mathfrak{B}$, if $\mathfrak{A} \subseteq \mathfrak{B}$ and if $E^{\mathfrak{B}}[a] = E^{\mathfrak{A}}[a]$ for all interior nodes (non-leaves) a of $\mathfrak{A}$.

Propositional modal logic. We write ML for *propositional modal logic*. ML$[\tau]$ for $\tau = \{P_1, \ldots, P_l\}$ has the following formulae: each P_i is a formula; ML$[\tau]$ is closed under Boolean connectives $\wedge, \neg$ (and $\vee$, which, however, we regard as defined); and if φ is a formula, then so are $\Diamond\varphi$ (and dually $\Box\varphi$, which again we regard as defined). The semantics is defined over Kripke structures in the natural way. $(\mathfrak{A}, a) \models P_i$ if $a \in P_i^{\mathfrak{A}}$; the Boolean connectives behave as usual; and $(\mathfrak{A}, a) \models \Diamond\varphi$ if there is some $a' \in E^{\mathfrak{A}}[a]$ for which $(\mathfrak{A}, a') \models \varphi$. If $\mathfrak{A}$ is a tree structure of the appropriate type, with root $0^{\mathfrak{A}}$, we simply write $\mathfrak{A} \models \varphi$ for $(\mathfrak{A}, 0^{\mathfrak{A}}) \models \varphi$. I.e. we always regard the root as the distinguished element of a tree structure unless otherwise specified.

The propositional μ-calculus. L_μ augments the syntax and semantics of ML by means of a monadic least fixed point constructor. A formula $\varphi(X)$ is *positive* in the monadic second-order variable X, if X only occurs in the scope of an even number of negations. In this case $\varphi(X)$ induces a monotone operator on subsets P of τ-structures $\mathfrak{A}$ according to $P \mapsto \{a \in A \mid (\mathfrak{A}, a) \models \varphi(P)\}$.

Owing to its monotonicity, this operator has a *least fixed point* $[\mu_X\varphi(X)]^{\mathfrak{A}}$, which may also be obtained as the limit of the monotone sequence of its *stages*, $P_\alpha = \{a \in A \mid (\mathfrak{A}, a) \models \varphi(\bigcup_{\beta<\alpha} P_\beta)\}$. Now $\mu_X\varphi(X)$ is itself a formula of L_μ with semantics according to $(\mathfrak{A}, a) \models \mu_X\varphi(X)$ if $a \in [\mu_X\varphi(X)]^{\mathfrak{A}}$.

The following observation will be useful in the analysis of L_μ-formulae. It follows from monotonicity considerations.

Observation 1 *If $\varphi(X)$ is positive in X and if $P \subseteq A$ is any stage of $\mu_X\varphi(X)$ over $\mathfrak{A}$, then $(\mathfrak{A}, a) \models \varphi(P) \Leftrightarrow (\mathfrak{A}, a) \models \varphi(P \setminus \{a\})$.*

Relativization. It is useful to associate with classes $\mathcal{C}$ of tree structures of type τ, and with a unary predicate $U \notin \tau$, the class of all those tree structures of type $\tau \,\dot\cup\, \{U\}$ for which the root is an element of the U-part, and for which the tree structure of type τ induced on the largest initial subset contained in the U-part is a member of $\mathcal{C}$. We call this derived class the relativization of $\mathcal{C}$ to U and denote it $\mathcal{C}^U$. It is easy to see that $\mathcal{C}^U$ is ML-definable or L_μ-definable, respectively, if $\mathcal{C}$ is so definable. In fact the following inductively defined U-relativization of L_μ-formulae provides the desired formulae: $\varphi^U = U \wedge \varphi$ for atomic φ; $(\neg\varphi)^U = U \wedge \neg\varphi^U$; $(\varphi_1 \wedge \varphi_2)^U = \varphi_1^U \wedge \varphi_2^U$; $(\Diamond\varphi)^U = U \wedge \Diamond\varphi^U$; $(\mu_X\varphi(X))^U = U \wedge \mu_X\varphi^U(X)$. Note that the translation $\varphi \mapsto \varphi^U$ increases the length only linearly.

Bisimulation. Two tree structures $\mathfrak{A}$ and $\mathfrak{B}$ with roots $0^{\mathfrak{A}}$ and $0^{\mathfrak{B}}$ are *bisimulation equivalent*, $\mathfrak{A} \sim \mathfrak{B}$, if there is an $R \subseteq A \times B$, such that $(0^{\mathfrak{A}}, 0^{\mathfrak{B}}) \in R$ and for all $(c, d) \in R$:

$$c \in P_j^{\mathfrak{A}} \Leftrightarrow d \in P_j^{\mathfrak{B}} \text{ for all } P_j, \quad \text{and} \quad \begin{array}{l} \forall c' \in E^{\mathfrak{A}}[c]\ \exists d' \in E^{\mathfrak{B}}[d] \colon (c', d') \in R, \\ \forall d' \in E^{\mathfrak{B}}[d]\ \exists c' \in E^{\mathfrak{A}}[c] \colon (c', d') \in R. \end{array}$$

We shall also deal with finite approximations of bisimulation equivalence in the form of *n-bisimulation equivalence* $\sim_n$, which for tree structures can be characterized as follows: $\mathfrak{A} \sim_n \mathfrak{B}$ if and only if $\mathfrak{A} \restriction \langle 0^{\mathfrak{A}} \rangle^n \ \sim\ \mathfrak{B} \restriction \langle 0^{\mathfrak{B}} \rangle^n$.

The model theoretic proof of the decidability result in our main theorem relies on a restriction of the issue to some subclass $\mathcal{T}_{n;m}$ of initially n-branching trees. Let us say that some model theoretic condition *holds in restriction to* $\mathcal{T}_{n;*}$ if this condition is true in restriction to $\mathcal{T}_{n;m}$ for some (and hence for all sufficiently large) m.

Lemma 2 *For a bisimulation-closed class $\mathcal{C} \subseteq \mathcal{T}[\tau]$ and for any n the following are equivalent:*
(i) $\mathcal{C}$ is ML*-definable in restriction to $\mathcal{T}_{n;*}[\tau]$.*
(ii) there is some $m \in \mathbb{N}$ such that for any two tree structures $\mathfrak{A}$ and $\mathfrak{A}'$ in $\mathcal{T}_{n;m}$: if $\mathfrak{A} \restriction \langle 0^{\mathfrak{A}} \rangle^m \simeq \mathfrak{A}' \restriction \langle 0^{\mathfrak{A}'} \rangle^m$, then $\mathfrak{A} \in \mathcal{C} \ \Leftrightarrow\ \mathfrak{A}' \in \mathcal{C}$.

Sketch of proof. (i) $\Rightarrow$ (ii) follows directly from the fact that a modal formula of quantifier rank m is insensitive to parts of the structure whose distance from the distinguished node (the root) is greater than m. For (ii) $\Rightarrow$ (i) we observe first that, for any given m, there are only finitely many m-bisimulation classes of τ-trees, each of which is characterized by a single ML-formula of quantifier rank m. So $\mathcal{C}$ is ML-definable over $\mathcal{T}_{n;m}$ as a finite union of ML-definable q-bisimulation types, provided we can show that (ii) and closure of $\mathcal{C}$ under bisimulation together imply that $\mathcal{C}$ is actually m-bisimulation closed over $\mathcal{T}_{n;m}$. Let to this end $\mathfrak{A}, \mathfrak{B} \in \mathcal{T}_{n;m}$, $\mathfrak{B} \sim_m \mathfrak{A}$, $\mathfrak{A} \in \mathcal{C}$; we show that $\mathfrak{B} \in \mathcal{C}$. From $\mathfrak{A} \sim_m \mathfrak{B}$ we may obtain an $\mathfrak{A}' \in \mathcal{T}_{n;m}$, through duplication of subtrees rooted within $\langle 0 \rangle^m$ in $\mathfrak{A}$, with the following properties: $\mathfrak{A}' \sim \mathfrak{A}$ and $\mathfrak{A}' \sim_m \mathfrak{B}$ via a bisimulation R between $\mathfrak{A}' \restriction \langle 0 \rangle^m$ and $\mathfrak{B} \restriction \langle 0 \rangle^m$ that is the graph of a function f from $\mathfrak{A}' \restriction \langle 0 \rangle^m$ onto $\mathfrak{B} \restriction \langle 0 \rangle^m$. Let now $\mathfrak{A}''$ be the result of replacing, for all $c \in A'$ at height m, each $\mathfrak{A}' \restriction \langle c \rangle$ by the corresponding $\mathfrak{B} \restriction \langle f(c) \rangle$. It follows that $\mathfrak{A}'' \sim \mathfrak{B}$, $\mathfrak{A}'' \restriction \langle 0 \rangle^m \simeq \mathfrak{A}' \restriction \langle 0 \rangle^m$, $\mathfrak{A}' \sim \mathfrak{A}$. Now $\mathfrak{A} \in \mathcal{C}$ by assumption, $\mathfrak{A}' \in \mathcal{C}$ by $\sim$-closure, $\mathfrak{A}'' \in \mathcal{C}$ by (ii), and therefore finally $\mathfrak{B} \in \mathcal{C}$ by $\sim$-closure again. $\square$

It is easy to see that condition (ii) of the lemma is further equivalent to the following condition on expansions of the complete n-ary tree: *There is a finite initial subset V of the complete n-branching tree T_n such that for all initial $W \subseteq V$ and all $P_1, \dots, P_l \subseteq W$: either all end extensions of $(T_n, P_1, \dots, P_l) \restriction W$ are in $\mathcal{C}$, or none is.* Omitting the straightforward application of well-known interpretation techniques, we note that this condition is expressible in monadic second-order over the complete n-branching tree, provided $\mathcal{C}$ itself is monadic second-order definable (as is clearly the case for L_μ-definable classes). But by Rabin's famous theorem [14], the monadic second-order theories of the complete n-branching trees are all decidable — actually uniformly in n, since all these theories are uniformly interpretable in that of the complete binary tree, S2S. Thus we have the following for the restriction of the ML-expressibility issue to classes $\mathcal{T}_{n;*}$.

Proposition 3 *The following decision problem is decidable via reduction to* S2S*, uniformly in n and in the vocabulary of φ: given $\varphi \in \mathrm{L}_\mu$ and $n \in \mathbb{N}$, decide whether there is a formula $\psi \in \mathrm{ML}$ that is equivalent to φ in restriction to $\mathcal{T}_{n;*}$.*

3. Prunings, preservation, and decidability via S2S

Prunings offer a canonical means to govern the branching degree of tree structures. The idea is to associate with each node a set of properties which are relevant for its direct E-successors, and to consider those prunings, which – at each node a – retain sufficiently many immediate successors so as to still realize the same relevant properties in the remaining successors of a.

Definition 4 Let Γ be a set of classes of tree structures of type τ. A pruning $\mathfrak{B} \subseteq \mathfrak{A}$ is called *Γ-elementary*, if for all $b \in B$ and for all $\mathcal{C} \in \Gamma$: if there is some $a \in E^{\mathfrak{A}}[b]$ such that $\mathfrak{A} \restriction \langle a \rangle \in \mathcal{C}$, then there also is an $a \in E^{\mathfrak{B}}[b]$ such that $\mathfrak{A} \restriction \langle a \rangle \in \mathcal{C}$.

Observation 5 *Given Γ and $\mathfrak{A}$, there is a Γ-elementary pruning of $\mathfrak{A}$, whose branching degree is bounded by $|\Gamma|$.*

Definition 6 Let Γ be a set of classes of tree structures of type τ. A class $\mathcal{C}_0 \subseteq \mathcal{T}[\tau]$ is *finitely preserved with respect to Γ* if for all Γ-elementary finite prunings $\mathfrak{A} \subseteq \mathfrak{B}$, $\mathfrak{B} \in \mathcal{C}_0$ iff $\mathfrak{A} \in \mathcal{C}_0$. Γ is a *preservation set* if each $\mathcal{C} \in \Gamma$ is finitely preserved w.r.t. Γ.

These notions apply in the context of the well-known small branching property for L_μ which plays a role in satisfiability considerations, see e.g. [13, 5] and compare the so-called Fischer-Ladner closure of φ from [5, 6] and others.

Proposition 7 *Any L_μ-definable class $\mathcal{C} = \mathrm{Mod}(\varphi)$ is a member of some preservation set Γ_φ of L_μ-definable classes, whose size is bounded by the length of φ.*

Sketch of proof. By induction with respect to the structure of the defining formula $\varphi \in \mathrm{L}_\mu$. We first argue that $\mathrm{Mod}(\varphi)$ is finitely preserved w.r.t. some Γ_φ consisting of $\mathrm{Mod}(\varphi)$ and fewer than $|\varphi|$ other L_μ-definable classes; identifying these classes with their defining L_μ-formulae, we regard Γ_φ as a subset of L_μ, with $\varphi \in \Gamma_\varphi$. The atomic case is obvious with $\Gamma_\varphi = \{\varphi\}$; negation is dealt with by putting $\Gamma_{\neg\varphi} = \Gamma_\varphi \cup \{\neg\varphi\}$ (the complement of $\mathcal{C}$ is finitely preserved w.r.t. Γ if $\mathcal{C}$ itself is). Similarly we may put $\Gamma_{\varphi_1 \wedge \varphi_2} = \Gamma_{\varphi_1} \cup \Gamma_{\varphi_2} \cup \{\varphi_1 \wedge \varphi_2\}$. Modal quantification is covered in $\Gamma_{\Diamond\varphi} = \Gamma_\varphi \wedge \{\Diamond\varphi\}$.

Finally, let $\varphi = \mu_X\psi(X)$. Note that $\Gamma_{\psi(X)}$ is a set of formulae in a vocabulary involving X as a basic proposition. Let $\Gamma[\mu_X\psi/X]$ be the result of substituting $\mu_X\psi(X)$ for every occurrence of X in all formulae in Γ. We claim that $\Gamma_\varphi = \Gamma_{\psi(X)}[\mu_X\psi/X] \cup \{\varphi\}$ is good for φ. We have to show that φ itself as well as any other member of Γ_φ is finitely preserved w.r.t. Γ_φ. We first argue for formulae other than φ, i.e. for $\chi[\mu_X\psi/X]$ for $\chi(X) \in \Gamma_{\psi(X)}$. Under the assumption that $\mu_X\psi(X)$ itself is preserved, preservation for $\chi[\mu_X\psi/X]$ is inherited from the corresponding preservation property of Γ_ψ: it corresponds to the special case of the latter in which X happens to be interpreted as $\mu_X\psi(X)$. It remains to show that $\mu_X\psi(X)$ is finitely preserved w.r.t. Γ_φ. Since we are claiming preservation only w.r.t. finite prunings, we may prove the preservation claim by induction over subtrees and may consider a pruning in just one single node. Assume that $\mathfrak{B} \subseteq \mathfrak{A}$ is a Γ_φ-elementary pruning obtained from $\mathfrak{A}$ through deletion of subtrees rooted in elements $a' \in E^{\mathfrak{A}}[a]$. Using the assumption that $(\mathfrak{B}, b) \models \mu_X\psi(X)$ iff $(\mathfrak{A}, b) \models \mu_X\psi(X)$, for all $b \in \langle a\rangle^{\mathfrak{B}} \setminus \{a\}$, we find that the pruning $(\mathfrak{B}, [\mu_X\psi]^{\mathfrak{A}} \setminus \{a\}, a) \subseteq (\mathfrak{A}, [\mu_X\psi]^{\mathfrak{A}} \setminus \{a\}, a)$ is $\Gamma_{\psi(X)}$-elementary. This gives the desired preservation of φ at a, through an application of Observation 1. □

Lemma 8 *Let $\mathcal{C} \subseteq \mathcal{T}[\tau]$ be a member of a preservation set Γ. If $\mathcal{C}$ is* ML*-definable in restriction to $\mathcal{T}_{n;*}$ for $n = 2 \cdot |\Gamma|$, then $\mathcal{C}$ is* ML*-definable over $\mathcal{T}[\tau]$.*

Proof. Let $n = 2\cdot|\Gamma|$ and assume that $\mathcal{C}$ is ML-definable in restriction to $\mathcal{T}_{n;m}[\tau]$. Suppose, towards a contradiction, that $\mathcal{C}$ were not ML-definable over $\mathcal{T}[\tau]$. By

Lemma 2, we may therefore find tree structures $\mathfrak{A}$ and $\mathfrak{A}'$ such that $\mathfrak{A} \in \mathcal{C}$, $\mathfrak{A}' \notin \mathcal{C}$, and $\mathfrak{A} \upharpoonright \langle a \rangle^m \simeq \mathfrak{A}' \upharpoonright \langle a' \rangle^m$. W.l.o.g. assume that $\mathfrak{A} \upharpoonright \langle a \rangle^m = \mathfrak{A}' \upharpoonright \langle a' \rangle^m$, and that $\mathfrak{A}$ and $\mathfrak{A}'$ are disjoint beyond height m. Let U and V be monadic predicates not in τ and put $\widehat{\tau} := \tau \cup \{U, V\}$. Let $\widehat{\Gamma} = \{\mathcal{C}^U \mid \mathcal{C} \in \Gamma\} \cup \{\mathcal{C}^V \mid \mathcal{C} \in \Gamma\}$ the set of classes obtained by relativizing those in Γ to U or V, respectively. Note that $|\widehat{\Gamma}| = n$. Let $\mathfrak{B}$ be the tree structure of type $\widehat{\tau}$ obtained from $\mathfrak{A} \cup \mathfrak{A}'$ by putting $U^{\mathfrak{B}} := A$ and $V^{\mathfrak{B}} := A'$. Let $\mathfrak{B}_0 \subseteq \mathfrak{B}$ be a $\widehat{\Gamma}$-elementary finite pruning of $\mathfrak{B}$ such that $\mathfrak{B}_0 \in \mathcal{T}_{n;m}[\widehat{\tau}]$ (cf. Observation 5). Note that in $\mathfrak{B}_0$, U and V still are initial subsets. Let $\mathfrak{A}_0$ be the tree structure of type τ obtained as the restriction of $\mathfrak{B}_0$ to $U^{\mathfrak{B}_0}$, $\mathfrak{A}'_0$ similarly induced by the restriction to $V^{\mathfrak{B}_0}$. Now, $\mathfrak{A}_0$ and $\mathfrak{A}'_0$ are in particular Γ-elementary finite prunings of $\mathfrak{A}$ and $\mathfrak{A}'$, respectively, whence $\mathfrak{A}_0 \in \mathcal{C}$ and $\mathfrak{A}'_0 \notin \mathcal{C}$. Obviously still $\mathfrak{A}_0 \upharpoonright \langle a \rangle^m = \mathfrak{A}'_0 \upharpoonright \langle a' \rangle^m$. But this contradicts the assumption that $\mathcal{C}$ was ML-definable in restriction to $\mathcal{T}_{n;m}[\tau]$, by Lemma 2. □

Corollary 9 *For $\varphi \in \mathrm{L}_\mu$ there is an equivalent formula in* ML *if and only if φ is equivalent to some formula of* ML *in restriction to $\mathcal{T}_{n;*}$ for $n = 2|\varphi|$.*

With Proposition 3, this yields the decidability claim of the main theorem.

4. Tree automata and exponential time complexity

The following is a variant of Lemma 2, which is in fact easier inasmuch as branching degrees are disregarded. Let $\mathcal{C} \subseteq \mathcal{T}[\tau]$, $U \notin \tau$, and $\mathcal{C}^U$ the corresponding relativization. Let $\Delta\mathcal{C}$ be the class of those $\mathfrak{A} \in \mathcal{T}_{\mathrm{fin}}[\tau \cup \{U\}]$ that have two different end extensions $\mathfrak{A} \subseteq_{\mathrm{end}} \mathfrak{B}_i$ such that $\mathfrak{B}_1 \in \mathcal{C}^U$, $\mathfrak{B}_2 \notin \mathcal{C}^U$.

The *tallness* of a tree structure is the minimum over the heights of the leaves. A class of (finite) tree structures is of *bounded tallness* if there is a uniform finite bound on the tallness of its members.

Lemma 10 *For any bisimulation-closed $\mathcal{C} \subseteq \mathcal{T}[\tau]$, and with the induced classes $\mathcal{C}^U$ and $\Delta\mathcal{C}$ as above, the following are equivalent:*
(i) $\mathcal{C}$ is ML*-definable.*
(ii) there is some $m \in \mathbb{N}$ such that for any two tree structures $\mathfrak{A}$ and $\mathfrak{A}'$ in $\mathcal{T}[\tau]$: if $\mathfrak{A} \upharpoonright \langle 0^{\mathfrak{A}} \rangle^m \simeq \mathfrak{A}' \upharpoonright \langle 0^{\mathfrak{A}'} \rangle^m$, then $\mathfrak{A} \in \mathcal{C} \Leftrightarrow \mathfrak{A}' \in \mathcal{C}$.
(iii) $\Delta\mathcal{C}$ is of bounded tallness.

For L_μ-definable $\mathcal{C}$ we want to view condition (iii) in an automata theoretic setting, having $\Delta\mathcal{C}$ accepted by some suitable tree automaton. Firstly, however, we need to present finite tree structures and some information about their end extensions in a way that fits automata.

A *Λ-labelling* of a (naked) tree $\mathfrak{A} \in \mathcal{T}_{\mathrm{fin}}[\emptyset]$ is a mapping $\lambda \colon A \to \Lambda$. Clearly, any tree structure in $\mathcal{T}_{\mathrm{fin}}[\tau]$ may be coded as a Λ-labelled tree for $\Lambda = \mathcal{P}(\tau)$, by putting $\lambda(a) = \{P_i \in \tau \mid \mathfrak{A}, a \models P_i\}$. A *leaf labelling* on $\mathfrak{A}$ is a labelling defined on the set of leaves of $\mathfrak{A}$ rather than the entire universe. We may regard a finite tree structure of type τ together with a leaf labelling π in alphabet Σ

as a (naked) finite tree with a labelling in the alphabet $\Lambda = \mathcal{P}(\tau) \times (\Sigma \dot\cup \{*\})$, where $\pi(a) \in \mathcal{P}(\tau) \times \{*\}$ for all interior nodes a. It will therefore suffice to deal with the format given by

$$\mathcal{T}^\ell_{\mathrm{fin}}[\Lambda] = \{ T = (\mathfrak{A}, \lambda) \mid \mathfrak{A} \in \mathcal{T}_{\mathrm{fin}}[\emptyset], \lambda \colon A \to \Lambda \text{ a labelling } \}.$$

A *(deterministic leaves-to-root) tree automaton* over $\mathcal{T}^\ell_{\mathrm{fin}}[\Lambda]$ is given as $\mathcal{A} = (Q, \delta)$ where Q is the finite set of states, and δ a transition function of the form $\delta \colon \mathcal{P}(Q) \times \Lambda \to Q$. Its run on $T \in \mathcal{T}^\ell_{\mathrm{fin}}[\Lambda]$ is described by an induced labelling $\rho \colon A \to Q$ defined inductively according to $\rho(a) = \delta\big(\{\rho(a') \mid a' \in E^{\mathfrak{A}}[a]\}, \lambda(a)\big)$. The tree language accepted by $\mathcal{A}$ w.r.t. some $F \subseteq Q$ is $L(\mathcal{A}, F) = \bigcup_{q \in F} L(\mathcal{A}, q)$ where

$$L(\mathcal{A}, q) = \{ T \in \mathcal{T}^\ell_{\mathrm{fin}}[\Lambda] \mid \rho(0) = q \text{ for the run } \rho \text{ of } \mathcal{A} \text{ on } T \}.$$

For an application as an acceptor of trees whose branching degree is bounded by some n (which is the standard format used in the literature), we may of course transcribe δ into a function $\delta^n \colon \bigcup_{m \leqslant n} Q^m \times \Sigma \to Q$. Let $\mathcal{A}^n = (Q, \delta^n)$ be this specialization of $\mathcal{A}$ to trees of n-bounded branching.

Observation 11 *The size of $\mathcal{A} = (Q, \delta)$ is $|\delta| \leqslant |\Lambda| \cdot 2^{|Q|}$. Its restriction $\mathcal{A}^n$ to trees of n-bounded branching is of size $|\delta^n| \leqslant |\Lambda| \cdot |Q|^n$.*

It is one of the main points in our application, though, that the branching should not a priori be bounded. The exponential blow-up between $\mathcal{A}^n$ for fixed n and $\mathcal{A}$ itself is the reason that we shall want to deal with a special subspecies of the above kind of automata, for which the transition function can be given in a more compact format. Assume that $\mathcal{A} = (Q, \delta)$ where Q is of the form $Q \subseteq \mathcal{P}(\Gamma)$ for some finite set Γ, Q closed under union. We say that $\mathcal{A} = (Q, \delta)$ is *of $\cup$-type* w.r.t. Γ if, for all $\boldsymbol{q} \subseteq Q$ and for all $r \in \Lambda$, the value of $\delta(\boldsymbol{q}, r)$ only depends on $\bigcup \boldsymbol{q}$. The following is then straigtforward.

Lemma 12 *Let $\mathcal{A} = (Q, \delta)$ be of $\cup$-type w.r.t. Γ and let $|\Gamma| = n$. Then, for any $F \subseteq Q$, $L(\mathcal{A}, F)$ is of bounded tallness iff $L(\mathcal{A}^n, F)$ is of bounded tallness.*

An inspection of typical results for standard (fixed branching) tree automata shows that these carry over to our slightly more general notion of tree automata, with the above notion of size. We refer in particular to M. Vardi's discussion of tree automata and their applications in [16], and to the handbook article [15] by W. Thomas for background.

Theorem 13 *Bounded tallness of $L(\mathcal{A}, F)$ is decidable in time polynomial in the size of $\mathcal{A}$.*

For a fixed finite set $\Gamma \subseteq \mathrm{L}_\mu[\tau]$ let $\mathrm{tp}^\Gamma(\mathfrak{A}) = \{\psi \in \Gamma \mid \mathfrak{A} \models \psi\}$, and $\mathrm{Tp}^\Gamma(\mathfrak{A}, a) = \bigcup_{a' \in E^{\mathfrak{A}}[a]} \mathrm{tp}^\Gamma(\mathfrak{A} \restriction \langle a' \rangle)$. Consider an end extension $\mathfrak{B}$ of a finite tree structure $\mathfrak{A}$ of type τ, $\mathfrak{A} \subseteq_{\mathrm{end}} \mathfrak{B}$. With $\mathfrak{B}$ associate the leaf labelling that maps a leaf a of $\mathfrak{A}$ to $\pi(a) = \mathrm{Tp}^\Gamma(\mathfrak{B}, a)$. If Γ is a preservation set, then the leaf labelling induced by an end extension $\mathfrak{B}$ of $\mathfrak{A}$ fully determines $\mathrm{tp}^{\Gamma}(\mathfrak{B} \restriction \langle a \rangle)$ for all $a \in A$. Indeed, a tree automaton can compute these types from the leaf labelling over $\mathfrak{A}$. Note that $(\mathfrak{A}, \pi)$ is a Σ_{Γ}-leaf-labelled tree structure where $\Sigma_{\Gamma} = \{\mathrm{Tp}^{\Gamma}(\mathfrak{B}, b) \mid \mathfrak{B} \in \mathcal{T}[\tau]\} \subseteq \mathcal{P}(\Gamma)$. If Γ is a set of at most n L_{μ}-formulae whose length is at most n, then this alphabet itself is recognizable in time exponential in n: $p \in \Sigma_{\Gamma}$ if and only if the L_{μ}-formula $\bigwedge_{\psi \in p} \Diamond\psi \wedge \bigwedge_{\psi \in \Gamma \setminus p} \neg\Diamond\psi$ is satisfiable. But L_{μ}-satisfiability is in EXPTIME due to [4]. Note that the necessity to make this labelling alphabet explicit eventually turns our automata theoretic decision procedure into a reduction to L_{μ}-satisfiability. We code $(\mathfrak{A}, \pi)$ as a tree $T \in \mathcal{T}^{\ell}_{\mathrm{fin}}[\Lambda_{\Gamma}]$, where $\Lambda_{\Gamma} \subseteq \mathcal{P}(\tau) \times (\mathcal{P}(\Gamma) \dot{\cup} \{*\})$. Let in this sense $[\mathfrak{A}; \mathfrak{B}]^{\Gamma}$ stand for the $T \in \mathcal{T}^{\ell}_{\mathrm{fin}}[\Lambda_{\Gamma}]$ associated with an end extension $\mathfrak{A} \subseteq_{\mathrm{end}} \mathfrak{B}$. The following extends Proposition 7. The inductive proof, which is an elaboration of that given for Proposition 7 above, is omitted here.

Proposition 14 *For every $\varphi \in \mathrm{L}_{\mu}[\tau]$ there is a preservation set $\Gamma = \Gamma_{\varphi}$ of size $|\Gamma_{\varphi}| \leqslant |\varphi|$ and with $\varphi \in \Gamma_{\varphi}$, and an automaton $\mathcal{A}_{\Gamma}$ with state set $Q_{\Gamma} = \mathcal{P}(\Gamma)$, of $\cup$-type w.r.t. Γ, such that for all $\mathfrak{A} \subseteq_{end} \mathfrak{B}$: $[\mathfrak{A}; \mathfrak{B}]^{\Gamma} \in L(\mathcal{A}_{\Gamma}, q) \Leftrightarrow \mathrm{tp}^{\Gamma}(\mathfrak{B}) = q$.*

We turn to criterion (iii) from Lemma 10. Let $\varphi \in \mathrm{L}_{\mu}[\tau]$ and consider two end extensions $\mathfrak{B}_1$ and $\mathfrak{B}_2$ of $\mathfrak{A} \in \mathcal{T}_{\mathrm{fin}}[\tau \cup \{U\}]$. For $\Delta\mathcal{C}$ we are interested in the case that $\mathfrak{B}_i \models \varphi^U$ and $\mathfrak{B}_2 \not\models \varphi^U$. The $\mathfrak{B}_i$ induce two leaf labellings on $\mathfrak{A}$ w.r.t. $\Gamma = \Gamma_{\varphi^U}$, which we may code into one with a labelling alphabet consisting of the product of the original Σ_{Γ} with itself. This turns the triple $(\mathfrak{A}, \mathfrak{B}_1, \mathfrak{B}_2)$ into a labelled tree

$$[\mathfrak{A}; \mathfrak{B}_1; \mathfrak{B}_2] \in \mathcal{T}^{\ell}_{\mathrm{fin}}[\Lambda^*], \quad \text{where } \Lambda^* = \mathcal{P}(\tau) \times \big((\Sigma_{\Gamma} \times \Sigma_{\Gamma}) \cup \{*\}\big).$$

Consider now an automaton $\mathcal{A}$ which simulates in parallel two copies of the automaton $\mathcal{A}_{\Gamma}$ of Proposition 14, one working with the first component of the leaf labelling, the other with the second component. The natural way of performing this parallel simulation uses a state set $Q^*_{\Gamma} := Q_{\Gamma} \times Q_{\Gamma}$, so that $\mathcal{A}^*_{\Gamma}$ operates like $\mathcal{A}_{\Gamma}$ in both components. Identifying $\mathcal{P}(\Gamma) \times \mathcal{P}(\Gamma)$ with $\mathcal{P}(\Gamma \times \{1,2\})$, and writing Γ^* for $\Gamma \times \{1,2\}$, we may regard $\mathcal{A}^*_{\Gamma}$ as of $\cup$-type w.r.t. Γ^*.

Now $\Delta\mathcal{C}$ consists of those $\mathfrak{A}$ for which some $[\mathfrak{A}; \mathfrak{B}_1; \mathfrak{B}_2]$ is accepted by $\mathcal{A}^* = \mathcal{A}^*_{\Gamma}$ in some state (q, q') where $\varphi^U \in q$ and $\varphi^U \notin q'$. Therefore, the tallness problem for $\mathcal{A}^*$ turns out to settle ML-expressibility of φ, by Lemma 10:

$$\varphi \text{ is expressible in ML} \quad \Leftrightarrow \quad \begin{array}{l} L(\mathcal{A}^*, F) \text{ is of bounded tallness,} \\ \text{where } F = \{(q, q') \mid \varphi^U \in q, \varphi^U \notin q'\}. \end{array}$$

Note that the size of $\mathcal{A}^*$ is doubly exponential in $|\varphi|$. But by Lemma 12, we may equivalently consider the tallness problem for the automaton $(\mathcal{A}^*)^n$ for $n = |\Gamma^*| = 2 \cdot |\Gamma|$, because $\mathcal{A}^*$ is of $\cup$-type. By Observation 11 and Theorem 13, bounded tallness of $(\mathcal{A}^*)^n$ is decidable in simply exponential time in $|\Gamma|$, and hence in $|\varphi|$. This proves the EXPTIME bound in the main theorem.

This bound is essentially optimal, since there is a straightforward reduction of L_{μ}-satisfiability to ML-expressibility: if $\xi \in \mathrm{L}_{\mu}$ is *not* ML-expressible, and if U and V are not in ξ or φ, then φ is *unsatisfiable* if and only if the formula $\Diamond\varphi^U \wedge \xi^V$ *is* equivalent to a formula in ML (namely to $\perp$).

References

1. J. Barwise and Y.N. Moschovakis, *Global inductive definability*, Journal of Symbolic Logic, 43(3), 1978, pp. 521–534.
2. J.F.A.K. van Benthem, Modal Logic and Classical Logic, Bibliopolis, 1985.
3. S.S. Cosmadakis, H. Gaifman, P.C. Kanellakis, and M.Y. Vardi, *Decidable optimization problems for database logic programs*, Proc. 20th ACM Symp. on Theory of Computing, 1988, pp. 477–490.
4. E.A. Emerson and C. Jutla, *The complexity of tree automata and logics of programs*, Proc. 29thSymp. on Foundations of Computer Science, 1988, pp. 328–337.
5. E.A. Emerson and R. Streett, *An automata theoretic decision procedure for the propositional mu-calculus*, Information and Computation, 81, 1989, pp. 249–264.
6. M.J. Fischer and R.E. Ladner, *Propositional dynamic logic of regular programs*, Journal of Computer and System Sciences, 18, 1979, pp. 194–211.
7. H. Gaifman, H. Mairson, Y. Sagiv, and M.Y. Vardi, *Undecidable optimization problems for database logic problems*, Journal of the Association for Computing Machinery, 40, 1993, pp. 683–713.
8. E. Grädel and M. Otto, *On logics with two variables*, to appear in Theoretical Computer Science.
9. G.G. Hillebrand, P.C. Kanellakis, H.G. Mairson, and M.Y. Vardi, *Undecidable boundedness problems for Datalog programs*, Journal of Logic Programming, 25, 1995, pp. 163–190.
10. D. Janin and I. Walukiewicz, *On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic*, Proc. of 7th Int. Conf. on Concurrency Theory, 1996, pp. 263-277.
11. P.G. Kolaitis and M. Otto, *On the boundedness problem for two-variable first-order logic*, Proc. 13th Symp. on Logic in Computer Science, LICS'98, pp. 513-524.
12. D. Kozen, *Results on the propositional μ-calculus*, Theoretical computer Science, 27, 1983, pp. 333–354.
13. D. Kozen, *A finite model theorem for the propositional μ-calculus*, Studia Logica, 47, 1987, pp. 233–241.
14. M.O. Rabin, *Decidability of second-order theories and automata on infinite trees*, Transactions of the American Mathematical Society, 141, 1969, pp. 1–35.
15. W. Thomas, *Languages, Automata, and Logic*, in Handbook of Formal Languages, vol. 3, G. Rozenberg and A. Salomaa, ed., Springer, 1997, pp. 389–456.
16. M.Y. Vardi, *Automata theory for database theoreticians*, in Theoretical Studies in Computer Science, J.D. Ullman, ed., Academic Press, 1992, pp. 153–180.
17. M.Y. Vardi, *Why is modal logic so robustly decidable*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 31, AMS, 1997, 149–184.