Complexity of Modal Logics with Presburger Constraints

Stéphane Demri, Denis Lugiez

Complexity of Modal Logics with Presburger Constraints [1]

Stéphane Demri [a]

[a] *LSV, ENS Cachan, CNRS, INRIA Saclay*
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: `demri@lsv.ens-cachan.fr`

Denis Lugiez [b]

[b] *LIF*
UMR 6166 Aix-Marseille Université – CNRS
39, rue Joliot-Curie, F-13453 Marseille Cedex 13
email: `lugiez@lif.univ-mrs.fr`

Abstract

We introduce the extended modal logic EML with regularity constraints and full Presburger constraints on the number of children that generalize graded modalities, also known as number restrictions in description logics. We show that EML satisfiability is only PSPACE-complete by designing a Ladner-like algorithm. This extends a well-known and non-trivial PSPACE upper bound for graded modal logic. Furthermore, we provide a detailed comparison with logics that contain Presburger constraints and that are dedicated to query XML documents. As an application, we provide a logarithmic space reduction from a variant of Sheaves logic SL into EML that allows us to establish that its satisfiability problem is also PSPACE-complete, significantly improving the best known upper bound.

Key words: modal logic, Ladner-like algorithm, arithmetical constraint, regularity constraint, computational complexity

1 Introduction

Logics for XML documents. In order to query XML documents with arithmetical and regular constraints, logical and automata-based formalisms have

[1] This is an extended version of [DL06]

been recently introduced [SSMH04,ZL06,BT05,OTTR05,SSM07] leading to various expressiveness and complexity results about logics and specialized tree automata. As usual, XML documents are viewed as labeled, unranked ordered trees. For instance, a logic with fixpoint operators, arithmetical and regularity constraints is introduced in [SSMH04] and shown decidable with an exponential time complexity, which improves results for description logics with qualified number restrictions [CG05]. At the same period, the sister logic SL ("Sheaves Logic") is shown decidable in [ZL06, Section 4.4] (see also [ZL03]) with a non-elementary decision procedure. The more expressive logic GDL is however shown undecidable in [ZL06] since GDL can express properties about disjoint sequences of children, as done also in Separation Logic (see e.g. [Rey02]). More generally, designing modal logics for semistructured data, either for tree-like models [Mar03,ABD$^+$05] or for graph-like models [ADdR03,BCT04] has been a fruitful approach since it allows to reuse known technical machineries adapted to special purpose formalisms. A temporal logic with counting can be also found in [MR03] but it has been introduced for other purposes, namely to characterize the expressive power of MSO in which second-order quantifications are over paths.

Our motivation. The main goal of this work is to introduce a modal logic allowing Presburger constraints (more general than those in graded modal logics [BC85,Tob01,PS04] or description logics [HB91,HST00,CG05]) and with regularity constraints as in the logical formalisms from [Wol83,ZL03,SSMH04] but with a satisfiability problem that can be solved in polynomial space. This would refine decidability and complexity results from [Tob01,SSMH04,ZL06]. Such an hypothetical logic would be much more helpful than the minimal modal logic K that is also known to be PSPACE-complete [Lad77] but K has not the ability to express such complex arithmetical and regularity constraints. With such requirements, fixpoint operators are out of the game since modal μ-calculus is already EXPTIME-complete. Similarly, Presburger constraints should be in a normal form since full Presburger logic has already a complexity higher than 2EXPTIME, see e.g. [FR74,Ber77]. It is worth observing that as far as memory ressources are concerned, no EXPTIME-complete problem is known to be solved in polynomial space. Hence, the potential difference between EXPTIME-completeness and PSPACE-completeness remains, so far, a significant gap in practice for running algorithms (PSPACE and EXPTIME have not been proved to be distinct classes).

Our contribution. We consider an Extended Modal Logic EML with full Presburger constraints on the number of children and with regularity constraints. It is a minor variant of either the fixpoint free fragment of [SSMH04] or the Sheaves Logic SL [ZL06] (extending also Presburger Modal logic from [Dem03a]). Relationships between EML, SL and the logic from [SSMH04] are provided in the paper. Our main result states that EML satisfiability is in PSPACE. The complexity upper bound is proved with a Ladner-like algorithm, see the original one in [Lad77] and this is strongly related to tableaux methods, see e.g. [Fit83,Gor99]. Such an algorithm can be also advantageously viewed as a specialized depth-first strategy to find proofs in an analytic proof system. Our results generalize what is known about graded modal logic [Fin72,BC85,Tob01] (including also the majority logic from [PS04]) and apart from its larger scope, we believe our proof is also much more transparent. A different approach introduced in [SP06] provides similar algorithms for graded modal logic and majority logic. Our proof uses the fact that it is simple to characterize the Parikh image of regular images in terms of semilinear sets (see [SSMH04,SSM07]) and systems of linear equations admit *small* solutions [Pap81], see also [BT76]. The use of small solutions for such systems goes back to [Rac78] in which the boundedness problem for vector addition systems is shown in EXPSPACE by taking advantage of small solutions to generate small paths. Our algorithm can be viewed as the optimal composition between an algorithm that transforms an EML formula into a Presburger tree automata and an algorithm that tests emptiness for these peculiar Presburger tree automata. This provides us new and non-trivial PSPACE complexity upper bounds that are not direct consequences of [SSMH04] since composing a polynomial space reduction with a polynomial space test does not imply the existence of a direct polynomial space test for the composition. For example, runs of linearly-bounded alternating Turing machines can be computed in polynomial space and testing if a run is accepting can be done in polynomial space in the size of the run. However, since APSPACE = EXPTIME, it is unlikely that the composition can be done in PSPACE. Additionally, our algorithm substantially refines results from [ZL06,SSMH04]. Indeed, as by-products of the complexity results about EML, we show that

- there is a logarithmic space reduction from a slight variant of Sheaves logic SL [resp. the fixpoint free fragment of the main logic from [SSMH04] (herein called SSMH)] into EML.
- the satisfiability problem for this variant of SL [resp. SSMH] is PSPACE-complete.
- the logic $\mathtt{PDL}_{\mathtt{tree}}$ from [ABD[+]05] is undecidable when extended with Presburger constraints. Modalities in $\mathtt{PDL}_{\mathtt{tree}}$ are quite rich since they allow us to navigate more freely in tree models, for instance sibling relations are present.

The complexity upper bounds are established via a logspace reduction whereas the PSPACE lower bound is proved by reducing satisfiability for the modal logic K (with modal operators $\Box$ and $\Diamond$) restricted to the truth constants as the only atomic formulae and characterized by the class of all the Kripke structures or equivalently by the class of all finite trees. Indeed, PSPACE-hardness of this very K fragment is already known [Hem01].

Plan of the paper. In Section 2, we introduce the extended modal logic EML and we show why it is safe for the satisfiability problem to restrict ourselves to finite labeled, unranked ordered trees with a unique label on transitions (using rather standard arguments). Section 3.1 contains preliminary definitions and results for the forthcoming algorithm. The Ladner-like algorithm is presented in Section 3.2 whereas its correctness and complexity are analyzed in Sections 3.3 and 3.4, respectively. The appendix A contains the proof that the branching factor of models can be bounded, essentially adapting developements from [SSM07]. In Section 4, we compare our result with related work and it is the opportunity to establish complexity results about SL and SSMH. Section 5 concludes the paper and states a few open problems.

2 Extended Modal Logic EML

2.1 Definition

Given countably infinite sets $\mathrm{AT} = \{p_1, p_2, \ldots\}$ of propositional variables and $\Sigma = \{\mathtt{R}_1, \mathtt{R}_2, \ldots\}$ of relation symbols, we define the set of formulae and terms inductively as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid t \sim b \mid t \equiv_k c \mid \mathcal{A}^{\mathtt{R}}(\phi_1, \ldots, \phi_n)$$

$$t ::= a \times \sharp^{\mathtt{R}}\phi \mid t + a \times \sharp^{\mathtt{R}}\phi,$$

where

- $p \in \mathrm{AT}$, $\mathtt{R} \in \Sigma$,
- $b, c \in \mathbb{N}$, $k \in \mathbb{N} \setminus \{0, 1\}$, $a \in \mathbb{Z} \setminus \{0\}$,
- $\sim \in \{<, >, =\}$,
- $\mathcal{A}$ is a nondeterministic finite-state automaton over an n-letter alphabet $\Sigma_{\mathcal{A}}$ in which the letters are linearly ordered $\Sigma_{\mathcal{A}} = \mathtt{a}_1, \ldots, \mathtt{a}_n$. The language accepted by $\mathcal{A}$ is denoted by $\mathrm{L}(\mathcal{A})$.

We write $|\phi|$ to denote the size of the formula ϕ with some reasonably succinct encoding and $\mathrm{md}(\phi)$ to denote the "modal degree" of ϕ defined as the greatest number of nested occurrences of $\sharp$ and automata-based operators in ϕ. We also write $sub(\phi)$ to denote the set of subformulae of ϕ. We assume that the cardinal of $sub(\phi)$ is bounded by $|\phi|$.

An expression of form $\sharp^{\mathtt{R}}\phi$ should be understood as a variable in a Presburger arithmetic formula interpreted as the number of immediate $\mathtt{R}$-successors satisfying the formula ϕ. A term of the form $a_1 \times \sharp^{\mathtt{R}_1}\phi_1 + \ldots + a_m \times \sharp^{\mathtt{R}_m}\phi_m$ is abbreviated by $\sum_{i=1}^{i=n} a_i \sharp^{\mathtt{R}_i}\phi_i$. Because of the presence of Boolean operators and quantifier-elimination for Presburger arithmetic (first-order theory of $\langle \mathbb{N}, <, = \rangle$), any kind of Presburger constraints can be expressed in this formalism, maybe less concisely with respect to an analogous language with quantifiers. We assume in the following that the automata are encoded reasonably succinctly and the elements in $\mathbb{Z}$ are represented with a binary encoding.

A model $\mathcal{M}$ for EML is a structure $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<_s^{\mathtt{R}})_{s\in T, \mathtt{R}\in\Sigma}, l \rangle$ where

- T is the set of nodes (possibly infinite),
- $(R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}$ is a family of binary relations in $T \times T$ such that for all $\mathtt{R} \in \Sigma$ and $s \in T$, the set $\{s' \in T : \langle s, s' \rangle \in R_{\mathtt{R}}\}$ is finite (finite-branching),
- each relation $<_s^{\mathtt{R}}$ is a total ordering on the $R_{\mathtt{R}}$-successors of s,
- $l : T \rightarrow 2^{\mathrm{AT}}$ is the valuation function where 2^{AT} denotes the powerset of AT.

At this stage, a model is not a tree-like structure but we shall argue later why we can restrict ourselves to such structures, using standard arguments from modal logics. In the rest of the paper, we write $R_{\mathtt{R}}(s) = s_1 < \ldots < s_\alpha$ to mean that $R_{\mathtt{R}}(s) \stackrel{\text{def}}{=} \{s' \in T : \langle s, s' \rangle \in R_{\mathtt{R}}\} = \{s_1, \ldots, s_\alpha\}$ and $s_1 <_s^{\mathtt{R}} \ldots <_s^{\mathtt{R}} s_\alpha$. Given a finite-branching binary relation $R \subseteq T \times T$, we write $R^\sharp(s)$ to denote the cardinal of the set $\{s' \in T : \langle s, s' \rangle \in R\}$ and R^* to denote the reflexive and transitive closure relation of R. The satisfaction relation $\models$ is inductively defined below where $\mathcal{M}$ is a model for EML and $s \in T$:

- $\mathcal{M}, s \models p$ iff $p \in l(s)$,
- $\mathcal{M}, s \models \neg\phi$ iff not $\mathcal{M}, s \models \phi$,
- $\mathcal{M}, s \models \phi_1 \wedge \phi_2$ iff $\mathcal{M}, s \models \phi_1$ and $\mathcal{M}, s \models \phi_2$,
- $\mathcal{M}, s \models \sum_i a_i \sharp^{\mathtt{R}_i} \phi_i \sim b$ iff $\sum_i a_i R^\sharp_{\mathtt{R}_i, \phi_i}(s) \sim b$ with $R_{\mathtt{R}_i, \phi_i} = \{\langle s', s'' \rangle \in T \times T : \langle s', s'' \rangle \in R_{\mathtt{R}_i},$ and $\mathcal{M}, s'' \models \phi_i\}$,
- $\mathcal{M}, s \models \sum_i a_i \sharp^{\mathtt{R}_i} \phi_i \equiv_k c$ iff there is $n \in \mathbb{N}$ such that $\sum_i a_i R^\sharp_{\mathtt{R}_i, \phi_i}(s) = nk + c$,
- The relation $\mathcal{M}, s \models \mathcal{A}^{\mathtt{R}}(\phi_1, \ldots, \phi_n)$ holds when the finite sequence of children of the node s induces a finite pattern from $\mathrm{L}(\mathcal{A})$. There is a correspondence between the letters $\mathtt{a}_1, \ldots, \mathtt{a}_n$ from the alphabet of $\mathcal{A}$ and the argument formulae $\phi_1, \ldots, \phi_n$ (below each letter $\mathtt{a}_i$ is associated with the argument formula ϕ_i). More precisely, $\mathcal{M}, s \models \mathcal{A}^{\mathtt{R}}(\phi_1, \ldots, \phi_n)$ iff there is $\mathtt{a}_{i_1} \cdots \mathtt{a}_{i_\alpha} \in \mathrm{L}(\mathcal{A})$ such that
  · $R_{\mathtt{R}}(s) = s_1 < \ldots < s_\alpha$,
  · for every $j \in \{1, \ldots, \alpha\}$, $\mathcal{M}, s_j \models \phi_{i_j}$.

Observe that constraints of the form $\sum_i a_i \sharp^{\mathtt{R}_i} \phi_i \equiv_k c$ can be expressed by regularity constraints but less concisely because of the binary encoding of integers. Moreover, these constraints are included so that by withdrawing regularity constraints we still obtain arithmetical constraints that have the expressive power of Presburger arithmetic.

Figure 1 illustrates the semantics of automata-based formulae.

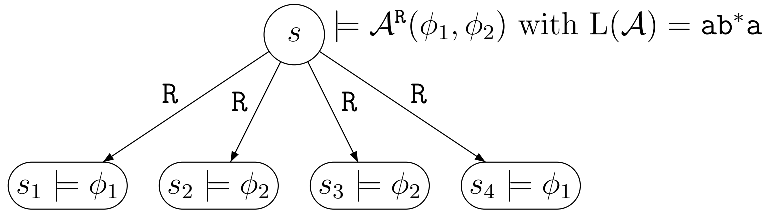

Fig. 1. Semantics for regular constraints

The automata in EML are used exactly as those defining temporal operators in extended temporal logic ETL [Wol83]. The modal operator $\Diamond$ (see e.g. [BdRV01]) is defined by $\Diamond\phi \approx \sharp^{\mathtt{R}}\phi \geqslant 1$ (and dually $\Box\phi \approx \sharp^{\mathtt{R}}\neg\phi = 0$) whereas the formula $\Diamond_{\geqslant n}\phi$ from graded modal logic is defined by $\Diamond_{\geqslant n}\phi \approx \sharp^{\mathtt{R}}\phi \geqslant n$. A basic example of what EML can express and graded modal logic cannot is that "there are twice more children satisfying p than children satisfying q" which can be stated by $\sharp^{\mathtt{R}}p - 2\sharp^{\mathtt{R}}q = 0$. Similarly, as in [PS04], one can express that "more than half of children satisfies the formula ϕ" with the formula $2\sharp\phi - \sharp\top > 0$.

A formula ϕ is satisfiable whenever there exist a model $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<^{\mathtt{R}}_s)_{s\in T, \mathtt{R}\in\Sigma}, l\rangle$ and $s \in T$ such that $\mathcal{M}, s \models \phi$.

Examples of formulae. We present below a few more examples of properties that can be expressed in EML.

- The number of children obtained with relation $\mathtt{R}_1$ and satisfying p is equal to the number of children obtained with relation $\mathtt{R}_2$ and satisfying q:

$$\sharp^{\mathtt{R}_1}p = \sharp^{\mathtt{R}_2}q$$

- The number of children obtained with the relation $\mathtt{R}$ is even:

$$\sharp^{\mathtt{R}}\top \equiv_2 0 \quad \text{or} \quad \mathcal{A}^{\mathtt{R}}(\top) \text{ with } \mathrm{L}(\mathcal{A}) = (\mathtt{a}\cdot\mathtt{a})^*$$

- For all the nodes of distance at most N obtained with the relation $\mathtt{R}'$, the number of children obtained with the relation $\mathtt{R}$ and satisfying p is strictly greater than the number of those satisfying $\neg p$:

$$\bigwedge_{i=0}^{N} \overbrace{\Box\ldots\Box}^{i \text{ times}}(\sharp^{\mathtt{R}}p > \sharp^{\mathtt{R}}\neg p),$$

  with $\Box\psi \stackrel{\text{def}}{=} \sharp^{\mathtt{R}'}\neg\psi = 0$.

Even though EML models are defined from general Kripke structures (apart from the fact that they are finite-branching), we show below that we can restrict ourselves to finite unranked ordered trees. Given a finite set of relation symbols $X = \{\mathtt{R}_1, \ldots, \mathtt{R}_n\}$, $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<_s^{\mathtt{R}})_{s\in T,\mathtt{R}\in\Sigma}, l\rangle$ is said to be a tree model with respect to X iff the restriction of $\mathcal{M}$ to $\bigcup_i R_{\mathtt{R}_i}$ is a tree.

Lemma 1 *For every* EML *formula ϕ, ϕ is satisfiable iff ϕ is satisfiable in a finite tree model with respect to the set of relation symbols occurring in ϕ.*

Proof. Suppose that ϕ has a EML model $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<_s^{\mathtt{R}})_{s\in T,\mathtt{R}\in\Sigma}, l\rangle$ and a node $s \in T$ such that $\mathcal{M}, s \models \phi$. We build a model $\mathcal{M}'$ satisfying the tree condition by unfolding $\mathcal{M}$ in the standard way. However, it remains to define the corresponding linear orderings. The model $\mathcal{M}' = \langle T', (S_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<_s^{\prime\mathtt{R}})_{s\in T',\mathtt{R}\in\Sigma}, l'\rangle$ is defined as follows:

- T' is the set of finite non-empty sequences of the form $s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_k\ s_k$,
- $(s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_n\ s_n)\ S_{\mathtt{R}}\ (s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_n\ s_n\ \mathtt{R}_{n+1}\ s_{n+1})$ iff $\langle s_n, s_{n+1}\rangle \in R_{\mathtt{R}}$ and $\mathtt{R} = \mathtt{R}_{n+1}$,
- $l'(s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_n\ s_n) = l(s_n)$ for every $s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_n\ s_n \in T'$,
- each ordering $<_{s'}^{\prime\mathtt{R}}$ is the one induced by $<_{s''}^{\mathtt{R}}$ by considering the last element s'' of the sequence s'.

One can show that for every $s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_n\ s_n \in T'$ and EML formula ψ, $\mathcal{M}', s\ \mathtt{R}_1\ s_1 \ldots \mathtt{R}_n\ s_n \models \psi$ iff $\mathcal{M}, s_n \models \psi$. In particular $\mathcal{M}', s \models \phi$. Since the formula tree of ϕ is finite and, arithmetical or regular constraints only speak about direct successors, we can truncate $\mathcal{M}'$ in order to obtain a finite model satisfying ϕ. $\square$

2.3 Restriction to One Relation

Additionally, one relation symbol suffices as a consequent of the result below.

Lemma 2 *For every* EML *formula ϕ, one can compute in logspace an* EML *formula ϕ' with a unique relation symbol $\mathtt{R}$ such that ϕ is satisfiable on finite trees iff ϕ' is satisfiable on finite trees.*

Proof. Let $\mathtt{R}_1, \ldots, \mathtt{R}_n$ be the relation symbols occurring in ϕ. To each $\mathtt{R}_i$, we associate a new propositional variable p_i. Intuitively, "p_i" holds true whenever the (backward) transition leading to the parent node is labelled by $\mathtt{R}_i$. The only relation symbol used in ϕ' will be $\mathtt{R}$. Figure 2.3 illustrates this type of transformation.

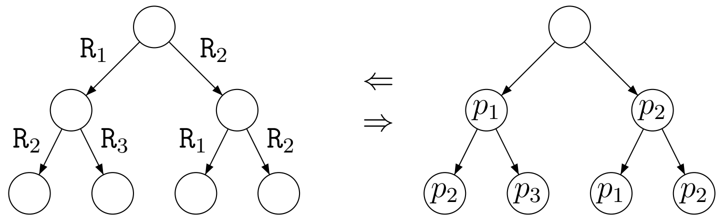

Fig. 2. Elimination of relation symbols

The formula ϕ' is the conjunction $\phi'_1 \wedge \phi'_2$ where

- ϕ'_1 states that a unique p_i holds true at each non-root node:

$$\phi'_1 = \bigwedge_{i=1}^{|\phi|} \overbrace{\Box \ldots \Box}^{i \text{ times}} \big(\bigvee_{j \in \{1,\ldots,n\}} (p_j \wedge \bigwedge_{l \in \{1,\ldots,n\} \setminus \{j\}} \neg p_l) \big)$$

  with $\Box \psi \stackrel{\text{def}}{=} \sharp^{\mathtt{R}} \neg \psi = 0$,
- the formula ϕ'_2 is obtained from ϕ by replacing each occurrence of $\sharp^{\mathtt{R}_i}\psi$ by $\sharp^{\mathtt{R}}(p_i \wedge \psi)$, and each occurrence of $\mathcal{A}_i^{\mathtt{R}}(\psi_1, \ldots, \psi_m)$ by $(\mathcal{A}')^{\mathtt{R}}(\neg p_i, p_i \wedge \psi_1, \ldots, p_i \wedge \psi_m)$ where $\mathcal{A}'$ is defined as follows. If the alphabet of $\mathcal{A}$ is $\Sigma = \{a_1, \ldots, a_m\}$, the alphabet of $\mathcal{A}'$ is $\Sigma' = \{a_0\} \uplus \Sigma$ and $\mathrm{L}(\mathcal{A}') = \{\sigma \in (\Sigma')^* : \sigma^{\setminus a_0} \in \mathrm{L}(\mathcal{A})\}$ where $\sigma^{\setminus a_0}$ is obtained from σ by erasing all occurrences of the new letter a_0. $\mathcal{A}'$ can be computed in logspace in the size of $\mathcal{A}$ by adding self-loops.

One can check that ϕ is satisfiable iff ϕ' is satisfiable. $\Box$

In the rest of the paper, we assume that Σ is a singleton set $\{\mathtt{R}\}$, we write $\mathcal{A}(\phi_1, \ldots, \phi_n)$ instead of $\mathcal{A}^{\mathtt{R}}(\phi_1, \ldots, \phi_n)$ and $\sharp\phi_i$ instead of $\sharp^{\mathtt{R}}\phi_i$. Models are simply written as tuples $\langle T, R, (<_s)_{s \in T}, l \rangle$. Furthermore, without any loss of generality, we assume that formulae are satisfied at the root node of models.

3 An Algorithm for EML Satisfiability

In this section, we show that EML satisfiability can be solved in polynomial space by using a Ladner-like algorithm [Lad77] and an analysis about constraint systems using in some place a crucial argument from the proof of [SSM07, Claim 7.3]. The original algorithm [Lad77] is designed for the modal logics K and S4 and an extension to tense logic can be found in [Spa93] (see also other extensions for multimodal logics in [Dem03b]).

We define below a notion of closure à la Fischer-Ladner [FL79] for finite sets of formulae. Intuitively, the closure $\mathrm{cl}(X)$ of X contains all the formulae useful to evaluate the truth of formulae in X.

Definition 1 *Let X be a finite set of formulae.* $\mathrm{cl}(X)$ *is the smallest set of formulae such that*

- $X \subseteq \mathrm{cl}(X)$, $\mathrm{cl}(X)$ *is closed under subformulae,*
- *if* $\psi \in \mathrm{cl}(X)$, *then* $\neg\psi \in \mathrm{cl}(X)$ *(we identify* $\neg\neg\psi$ *with* ψ*),*
- *if* $t \sim b \in \mathrm{cl}(X)$, *then* $t \sim' b \in \mathrm{cl}(X)$ *for every* $\sim' \in \{<, >, =\}$,
- *let K be the least common multiple (lcm) of all the constants k occurring in subformulae of the form $t \equiv_k c$. If $t \equiv_k c \in \mathrm{cl}(X)$, then $t \equiv_K c' \in \mathrm{cl}(X)$ for every $c' \in \{0, \ldots, K-1\}$.*

A set X of formulae is said to be closed iff $\mathrm{cl}(X) = X$. Observe that $\mathrm{card}(\mathrm{cl}(X))$ is exponential in $\mathrm{card}(X)$, which is usually not a good start to establish a polynomial space upper bound. Nevertheless, consistent sets of formulae that are satisfiable contain exactly one formula from $\{t \equiv_K c : c \in \{0, \ldots, K-1\}\}$ for each constraint $t \equiv_k c'$ in X. Hence, as explained below, encoding consistent sets will require only polynomial space.

We refine the notion of closure by introducing a new parameter n: the distance from the root node to the current node where the formulae are evaluated. Each set $\mathrm{cl}(n, \phi)$ is therefore a subset of $\mathrm{cl}(\{\phi\})$.

Definition 2 *Let ϕ be an* EML *formula. For $n \in \mathbb{N}$,* $\mathrm{cl}(n, \phi)$ *is the smallest set such that:*

- $\mathrm{cl}(0, \phi) = \mathrm{cl}(\{\phi\})$, *for every $n \in \mathbb{N}$,* $\mathrm{cl}(n, \phi)$ *is closed,*
- *for all $n \in \mathbb{N}$ and $\sharp\psi$ occurring in some formula of* $\mathrm{cl}(n, \phi)$, *we have* $\psi \in \mathrm{cl}(n+1, \phi)$,
- *for all $n \in \mathbb{N}$ and* $\mathcal{A}(\phi_1, \ldots, \phi_m) \in \mathrm{cl}(n, \phi)$, *we have* $\{\phi_1, \ldots, \phi_m\} \subseteq \mathrm{cl}(n+1, \phi)$.

In the sequel, we consider EML formulae ϕ such that for every n, we have $\mathrm{cl}(n, \phi) \neq \emptyset$, the lcm of all the constants k occurring in subformulae from $\mathrm{cl}(n, \phi)$ of the form $t \equiv_k c$ is equal to the lcm of all k occurring in ϕ. Without any loss of generality, we also assume that $\equiv_K$ does not occur in ϕ. Given an EML formula, one can compute an equivalent EML formula satisfying the above requirements by at most doubling its size.

We are only interested in subsets of $\mathrm{cl}(n, \phi)$ whose conjunction of its elements is EML satisfiable. A necessary condition to be satisfiable is to be consistent locally, i.e. at the propositional level and at the level of arithmetical constraints. As far as these latter constraints are concerned, we are more interested to introduce a notion of consistency that allows a polynomial space encoding of consistent sets than to guarantee that the Presburger constraints in a given set are indeed satisfiable. This latter property is checked with constraint systems (see Appendix A) in the main algorithm. This is analogous to the requirement to check maximal consistency at the propositional level but not EML satisfiability at once. It is the adequate construction of locally consistent sets that will guarantee that the initial set of formulae is EML satisfiable.

Definition 3 *A set $X \subseteq \mathrm{cl}(n, \phi)$ is said to be n-locally consistent iff the conditions below hold:*

- *if $\neg\psi \in \mathrm{cl}(n, \phi)$, then $\neg\psi \in X$ iff $\psi \notin X$,*
- *if $\psi_1 \wedge \psi_2 \in \mathrm{cl}(n, \phi)$, then $\psi_1 \wedge \psi_2 \in X$ iff $\psi_1, \psi_2 \in X$,*
- *if $t \sim b \in \mathrm{cl}(n, X)$ then there is a unique $\sim' \in \{<, >, =\}$ such that $t \sim' b \in X$,*
- *if $t \equiv_k c \in \mathrm{cl}(n, X)$, then there is a unique $c' \in \{0, \ldots, K-1\}$ such that $t \equiv_K c' \in X$,*
- *if $t \equiv_k c \in \mathrm{cl}(n, X)$, then $\neg t \equiv_k c \in X$ iff there is $c' \in \{0, \ldots, K-1\}$ such that $t \equiv_K c' \in X$ and not $c' \equiv_k c$,*
- *if $t \sim b \in \mathrm{cl}(n, X)$ then $\neg t \sim b \in X$ iff there is $\sim' \in \{<, >, =\} \setminus \{\sim\}$ such that $t \sim' b \in X$.*

The last condition is obviously a consequence of the two first ones, but we prefer to keep it for the sake of clarity. Observe that given an EML model $\mathcal{M}$ and a node s, the set of subformulae $\{\psi \in \mathrm{cl}(n, \phi) : \mathcal{M}, s \models \psi\}$ is n-locally consistent and it behaves as a *type* for the node s. Moreover, in the above definition, maximal consistency is required and this will simplify a few technical developements.

Lemma 3 *Let ϕ be a* EML *formula and $n \in \mathbb{N}$.*

(I) *Every n-locally consistent set has cardinal at most $2 \times |\phi|$ and can be encoded with a polynomial amount of bits with respect to $|\phi|$.*
(II) $\mathrm{cl}(|\phi|, \phi) = \emptyset$.
(III) *Given a set $X \subseteq \mathrm{cl}(0, \phi)$ of cardinal at most $2 \times |\phi|$ and $n \in \mathbb{N}$, one can decide in polynomial-time in $|\phi|$ whether X is n-locally consistent.*

Proof. (I) By Definition 2, $\mathrm{cl}(n, \phi) \subseteq \mathrm{cl}(\phi)$. Let $X \subseteq \mathrm{cl}(n, \phi)$ be an n-locally consistent set.

(a) For each subformula $\psi \in \mathrm{cl}(n, \phi)$ with Boolean outermost connective, either $\psi \in X$ or $\neg\psi \in X$.
(b) For each atomic subformula $\psi \in \mathrm{cl}(n, \phi)$, either $\psi \in X$ or $\neg\psi \in X$.
(c) For each atomic subformulae $t \sim b \in \mathrm{cl}(n, \phi)$, either $\neg t \sim b \in X$ and $t \sim' b$

for some $\sim' \in \{<, >, =\} \setminus \{\sim\}$ or $t \sim b \in X$,

(d) For each atomic subformulae $t \equiv_k c \in \mathrm{cl}(n, \phi)$, there is a unique $c' \in \{0, \ldots, K-1\}$ such that $t \equiv_K c' \in X$.

These are the only ways to obtain subformulae in X. Each subformulae in ϕ contributes to at most one formula in X except the subformulae of the form $t \equiv_k c$, that can contribute to at most two formulae in X (with the additional subformulae of the form $t \equiv_K c' \in X$). Consequently, the cardinal of X is bounded by $2 \times |\phi|$. Each subformula with Boolean outermost connective, each atomic subformula and their negations can be encoded with 1 bit. Similarly, each atomic formula of the form $t \sim' b$ can be encoded with 2 bits. Finally, each subformula of the form $t \equiv_K c'$ can be encoded with $\mathcal{O}(log(K))$ bits, that is at most quadratic in $|\phi|$. Indeed, K is at most $k_1 \times \cdots \times k_u$ where each k_i occurs in some atomic formula of the form $t \equiv_{k_i} c_i$ in ϕ. Hence, each n-locally consistent subset of $\mathrm{cl}(n, \phi)$ can be encoded with $\mathcal{O}(|\phi|^2)$ bits.

(II) We define the modal degree of a finite set of formulae as the maximal modal degree among the modal degrees of all formulae belonging to the set. By convention, the modal degree of the empty set is zero. By Definitions 1 and 2, $\mathrm{md}(\mathrm{cl}(0, \phi)) = \mathrm{md}(\phi) < |\phi|$. Moreover, for each $n \in \mathbb{N}$ such that $\mathrm{md}(\mathrm{cl}(n, \phi)) > 0$, we have $\mathrm{md}(\mathrm{cl}(n+1, \phi)) < \mathrm{md}(\mathrm{cl}(n, \phi))$. One can also observe that whenever $\mathrm{md}(\mathrm{cl}(n, \phi)) = 0$, we have $\mathrm{cl}(n, \phi) = \emptyset$. This allows us to conclude that $\mathrm{cl}(|\phi|, \phi) = \emptyset$.

(III) First, observe that since $\mathrm{cl}(n, \phi) = \emptyset$ for $n > |\phi|$, n can be represented with a binary encoding with no harm. By building the formula tree of ϕ, it is possible to compute the formulae in $\mathrm{cl}(n, \phi)$ whose outermost connective is Boolean as well as the atomic formulae from $sub(\phi)$ that are also in $\mathrm{cl}(n, \phi)$. Such a computation mainly depends on the modal depth of the subformula occurrences in the tree. An analogous analysis can be done with elements of $\mathrm{cl}(n, \phi)$ that are of the form $t \sim' b$. This allows to check the conditions (a)–(c) above. Finally, a visit of the formula tree also allows to decide which terms should occur in subformulae of the form $t \equiv_K c$ in $\mathrm{cl}(n, \phi)$. It remains then to check that if $t \equiv_k c \in \mathrm{cl}(n, X)$, then $\neg t \equiv_k c \in X$ implies there is $c' \in \{0, \ldots, K-1\}$ such that $t \equiv_K c' \in X$ and not $c' \equiv_k c$, which can be performed in polynomial-time in $|\phi|$. Similarly, one needs to check that $t \equiv_k c \in X$ implies there is $c' \in \{0, \ldots, K-1\}$ such that $t \equiv_K c' \in X$ and $c' \equiv_k c$. $\square$

Before defining the main algorithm in Section 3.2, let us introduce the notion of M-bounded models.

Definition 4 *Let ϕ be an* EML *formula, M be a natural number and $\mathcal{M}$ be a finite tree model such that $\mathcal{M}, s \models \phi$ for some node s. We say that $\langle \mathcal{M}, s \rangle$ is M-bounded for ϕ iff for every node s' of distance d from s, the cardinal of*

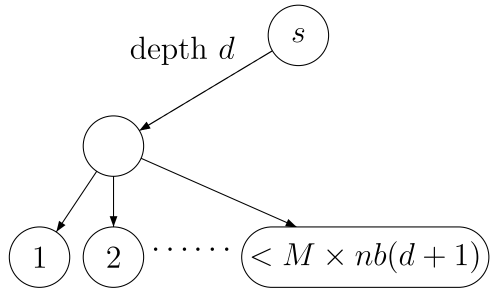

Fig. 3. A schema for M-boundedness

$R(s')$ is bounded by $nb(d+1) \times M$ where $nb(d+1)$ is the number of distinct $(d+1)$-locally consistent sets (with respect to ϕ).

Observe that $nb(d+1)$ is exponential in $|\phi|$ in the worst case and $nb(d+1) = \emptyset$ as soon as $d \geqslant |\phi|$. Figure 3 presents a schematic illustration for M-boundedness.

3.2 An Algorithm for M-Bounded Satisfiability

We define the function SAT such that ϕ is EML satisfiable in some M-bounded model iff there is $X \subseteq \mathrm{cl}(0, \phi)$ such that X is 0-locally consistent and $\mathrm{SAT}(X, 0)$ has a computation that returns `true`. Indeed, the function $\mathrm{SAT}(X, d)$ defined in Figure 4 is parameterized by some natural number M (see the step (guess-number-children)) and by the formula ϕ. These two parameters should be understood as global variables. We shall fix later the value M that will be only exponential in $|\phi|$ (see Lemma 7 and Appendix A).

The first argument X is intended to be a subset of $\mathrm{cl}(d, \phi)$. SAT is a non-deterministic algorithm but it can be defined as a deterministic one by enumerating possibilities instead of guessing, in the standard way. The $(d+1)$-locally consistent sets are denoted by Y_i for some $1 \leqslant i \leqslant nb(d+1)$.

A call $\mathrm{SAT}(X, d)$ performs the following actions. First it checks whether X is d-locally consistent and if the modal degree is zero, then it returns `true` in case of d-locally consistency. In order to check that X is satisfiable, children of the node are guessed from left to right (providing an ordering of the successors) and during the guess, auxiliary variables are updated. For each subformula ψ, there is a counter C_ψ and its current value contains the current number of children that should satisfy ψ. Similarly, regularity constraints use auxiliary variables. For each subformula in X whose outermost connective is automata-based, we introduce a variable that encodes the current state in the automaton. At the end of the guess of the children, this variable should be equal to a final state of the automaton. By contrast, for each subformula in X whose outermost connective is the negation of some automata-based formula, we introduce a variable that encodes the set of states that could be reached so far in the automaton (simulating a subset construction of the underlying automaton). At the end of the guess of the children, this variable should not contain any final state of the automaton. Checking regularity constraints on-the-fly as done herein goes back to [SSMH04]. After guessing at most $M \times nb(d+1)$ children, there is a final checking that verifies that the regularity constraints and the arithmetical constraints are satisfied. For instance, an atomic formula of the form $\sum a_i \sharp \psi_i \sim b$ will lead to verify whether $\sum a_i \sharp C_{\psi_i} \sim b$ holds true. For each child, we guess in fact a $(d+1)$-locally consistent set Y, which allows us to update all the auxiliary variables. However, we apply recursively $\mathrm{SAT}(Y, d+1)$ to ensure that not only Y is $(d+1)$-locally consistent but also that Y is satisfiable. Hence, if we guess a set Y that contains some unsatisfiable formula with respect to M-bounded models then $\mathrm{SAT}(Y, d+1)$ has no accepting computation which also induces a non accepting computation for $\mathrm{SAT}(X, d)$.

The algorithm SAT described in Figure 4 is a typical example of Ladner-like algorithm, see e.g. similar algorithms in [Lad77,Spa93,Dem03b]. Indeed,

- it does not rely on any machinery such as automata or tableaux/sequent proof systems for checking satisfiability,
- the graph of recursive calls (here for SAT) induces a tree model for the argument formula. Since EML models are precisely trees, we get the EML model for free.

3.3 Complexity Analysis

Firstly, we characterize the space needed to run SAT.

Lemma 4 *For all 0-locally consistent sets X, and computations of* $\mathrm{SAT}(X, 0)$

- *the recursive depth is linear in $|\phi|$,*
- *each call requires space polynomial in the sum of*
  - · *the space for encoding 0-locally consistent sets*
  - · *and log(M).*

Consequently, only polynomial space is required when M is exponential in $|\phi|$.

`function` $\mathrm{SAT}(X, d)$
(consistency) if X is not d-locally consistent then `abort`;
(base case) if X contains only propositional formulae then return `true`;
(witnesses)

- **(initialization-counters)** for every $\psi \in \mathrm{cl}(d+1, \phi)$ that is not a periodicity constraint of the form $t \equiv_K c$, $C_\psi := 0$;
- **(initialization-states)** for every $\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$, $q_{\mathcal{A}(\psi_1,\ldots,\psi_\alpha)} := q_0$ for some initial state q_0 of $\mathcal{A}$;
- **(initialization-states-complement)** for every $\neg\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$, $Z_{\neg\mathcal{A}(\psi_1,\ldots,\psi_\alpha)} := I$ where I is the set of initial states of $\mathcal{A}$;
- **(guess-number-children)** guess NB in $\{0, \ldots, nb(d+1) \times M\}$;
- **(guess-children-from-left-to-right)** for $i = 1$ to NB do
  1. guess $x \in \{1, \ldots, nb(d+1)\}$;
  2. if not $\mathrm{SAT}(Y_x, d+1)$ then `abort`;
  3. for every $\psi \in \mathrm{cl}(d+1, \phi)$ different from some $t \equiv_K c$, if $\psi \in Y_x$, then $C_\psi := C_\psi + 1$;
  4. for every $\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$,
     (a) guess a transition $q_{\mathcal{A}(\psi_1,\ldots,\psi_\alpha)} \xrightarrow{a_i} q'$ in $\mathcal{A}$ with $\Sigma_{\mathcal{A}} = \mathtt{a}_1, \ldots, \mathtt{a}_\alpha$;
     (b) if $\psi_i \in Y_x$, then $q_{\mathcal{A}(\psi_1,\ldots,\psi_\alpha)} := q'$, otherwise `abort`;
  5. for every $\neg\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$, $Z_{\neg\mathcal{A}(\psi_1,\ldots,\psi_\alpha)} := \{q : \exists\ q' \in Z_{\neg\mathcal{A}(\psi_1,\ldots,\psi_\alpha)}, q' \xrightarrow{\mathtt{a}_i} q, \psi_i \in Y_x\}$;
- **(final-checking)**
  1. for every $\sum_i a_i \sharp \psi_i \sim b \in X$, if $\sum_i a_i \times C_{\psi_i} \sim b$ does not hold, then `abort`,
  2. for every $\sum_i a_i \sharp \psi_i \equiv_k c \in X$, if $\sum_i a_i \times C_{\psi_i} \equiv_k c$ does not hold, then `abort`,
  3. for every $\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$, if $q_{\mathcal{A}(\psi_1,\ldots,\psi_\alpha)}$ is not a final state of $\mathcal{A}$, then `abort`;
  4. for every $\neg\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$, if $Z_{\neg\mathcal{A}(\psi_1,\ldots,\psi_\alpha)}$ contains a final state of $\mathcal{A}$, then `abort`;
- **(return-true)** return `true`.

Fig. 4. Satisfiability algorithm

Proof. By Lemma 3, the size of the stack of recursive calls to SAT is at most $|\phi|$ since $\mathrm{cl}(|\phi|, \phi) = \emptyset$. In the function SAT, the steps (consistency), (base case), (initialization-counters), (initialization-states) and (initialization-states-complement) can be obviously checked in polynomial time in ϕ (and therefore in polynomial space), see e.g. Lemma 3(III). In the step (guess-children-from-left-to-right), one needs a counter to count at most until $nb(d+1) \times M$. A polynomial amount of bits in $|\phi| + log(M)$ suffices. All the non-recursive instructions in (guess-children-from-left-right) can be done in time polynomial in $|\phi| + log(M)$. Since at the end of the step (guess-children-from-left-right), the values of the counters are less than or equal to $nb(d+1) \times M$, checking the points 1. and 2. in (final-checking) can be done in polynomial space in $|\phi| + log(M)$ (remember that the encoding of constants a_i, b and c and k are already in linear space in $|\phi|$). $\square$

3.4 Correctness

After having characterized the space needed to run the algorithm, it remains to prove that it is correct as far as the M-bounded models are concerned.

Lemma 5 *If for some* $X \subseteq \mathrm{cl}(0,\phi)$, $\mathrm{SAT}(X,0)$ *has a computation that returns* `true` *and* $\phi \in X$*, then* ϕ *is* EML *satisfiable in some* M*-bounded model.*

Proof. Assume that $\mathrm{SAT}(X,0)$ has an accepting computation with $\phi \in X$. Let us build an EML model $\mathcal{M} = \langle T, R, (<_s)_{s\in T}, l\rangle$ for which there is $s \in T$ such that for every $\psi \in X$, we have $\mathcal{M}, s \models \psi$ iff $\psi \in X$.

From an accepting computation of $\mathrm{SAT}(X,0)$, we consider the following finite ordered tree $\langle T, R, (<_s)_{s\in T}, L\rangle$ that corresponds to the calls tree of $\mathrm{SAT}(X,0)$.

- $\langle T, R, (<_s)_{s\in T}\rangle$ is a finite ordered tree,
- for each $s \in T$, $L(s) = \langle Y, d\rangle$ for some d-consistent set Y,
- the root node s_0 is labelled by $\langle X, 0\rangle$,
- for each node s with $s_1 <_s \cdots <_s s_n$, the call related to $l(s)$ recursively calls SAT with the respective arguments $l(s_1)$, …, $l(s_n)$ and in this very ordering.

The model $\mathcal{M}$ we are looking for, is precisely $\mathcal{M} = \langle T, R, (<_s)_{s\in T}, l\rangle$ for which $l(s) = Y \cap \mathrm{AT}$ where $L(s) = \langle Y, d\rangle$ for each s.

By structural induction on ψ, we shall show that for all $s \in T$ with $L(s) = \langle Y, d\rangle$, for all $\psi \in \mathrm{cl}(d,\phi)$, we have $\psi \in Y$ iff $\mathcal{M}, s \models \psi$. Consequently, we then get $\mathcal{M}, s_0 \models \phi$. The case when ψ is a propositional variable is by definition of l.

Induction hypothesis: for all $\psi \in \mathrm{cl}(\phi)$ such that $|\psi| \leqslant n$, for all $s \in T$ with $L(s) = \langle Y, d\rangle$, if $\psi \in \mathrm{cl}(d,\phi)$, then $\psi \in Y$ iff $\mathcal{M}, s \models \psi$.

Let ψ be a formula in $\mathrm{cl}(\phi)$ such that $|\psi| = n+1$. The cases when the outermost connective of ψ is Boolean is a consequence of the d-local consistency of Y and the induction hypothesis. Let us treat the other cases.
Case 1: $\psi = \mathcal{A}(\psi_1, \ldots, \psi_k)$.
Let $s \in T$ with $L(s) = \langle Y, d\rangle$ such that $\psi \in \mathrm{cl}(d,\phi)$. By definition of T, $\mathrm{SAT}(Y,d)$ has an accepting computation. If $\psi \in Y$, then each call in the sequence $\mathrm{SAT}(Y_{x_1}, d+1)$, …, $\mathrm{SAT}(Y_{x_{NB}}, d+1)$ has an accepting computation. Hence the children of s are the following (from left to right) $s_1, \ldots, s_{NB}$ such that $L(s_i) = \langle Y_{x_i}, d+1\rangle$. Then, it is not difficult to show that the

steps (initialization-states), (guess-children-from-left-to-right)(4) and (final-checking)(3) guarantee that $\mathcal{M}, s \models \psi$. If $\psi \not\in Y$, then by consistency of Y, $\neg\mathcal{A}(\psi_1, \ldots, \psi_k) \in Y$ and by following a reasoning as above we also get $\mathcal{M}, s \not\models \mathcal{A}(\psi_1, \ldots, \psi_k)$.

Case 2: $\psi = \sum_{i=1}^{i=\alpha} a_i \sharp \psi_i \sim b$.
Let $s \in T$ such that $L(s) = \langle Y, d\rangle$ and $\psi \in \mathrm{cl}(d, \phi)$. By definition of T, $\mathrm{SAT}(Y, d)$ has an accepting computation. If $\psi \in Y$, then each call in the sequence $\mathrm{SAT}(Y_{x_1}, d+1)$, $\ldots$, $\mathrm{SAT}(Y_{x_{NB}}, d+1)$ has an accepting computation. Moreover, for every $i \in \{1, \ldots, \alpha\}$, there are exactly C_{ψ_i} elements in $Y_{x_1}, \ldots, Y_{x_{NB}}$ that contain ψ_i where C_{ψ_i} is the value of the counter after the step (guess-children-from-left-to-right) in the above-mentioned successful computation for $\mathrm{SAT}(Y, d)$. Hence the children of s in $\mathcal{M}$ are the following (from left to right): $s_1, \ldots, s_{NB}$ with $L(s_i) = \langle Y_{x_i}, d+1\rangle$. It is not difficult to show that the steps (initialization-counters), (guess-children-from-left-to-right)(3) and (final-checking)(1) guarantee that $\mathcal{M}, s \models \psi$.

If $\psi \not\in Y$, then by consistency of Y, there is $\sim' \in \{<, >, =\} \setminus \{\sim\}$ such that $\sum_{i=1}^{\alpha} a_i \sharp \psi_i \sim' b \in X$ by following a reasoning as above this means that $\mathcal{M}, s \models \sum_{i=1}^{i=\alpha} a_i \sharp \psi_i \sim' b$ which entails $\mathcal{M}, s \not\models \sum_{i=1}^{i=\alpha} a_i \sharp \psi_i \sim b$.

Case 3: $\psi = \sum_{i=1}^{i=\alpha} a_i \sharp \psi_i \equiv_k c$.
The proof is similar to the cases 1 and 2.

The current model $\mathcal{M}$ is in exponential size in $|\phi|$ and it is easy to show that $\mathcal{M}$ is M-bounded. $\square$

The converse property holds.

Lemma 6 *If ϕ is* EML *satisfiable in some M-bounded model then for some $X \subseteq \mathrm{cl}(0, \phi)$,* $\mathrm{SAT}(X, 0)$ *has an accepting computation.*

Proof. Assume that ϕ is EML satisfiable in some M-bounded model $\mathcal{M} = \langle T, R, (<_s)_{s\in T}, l\rangle$. So there is $s \in T$ such that $\mathcal{M}, s \models \phi$ and $\langle \mathcal{M}, s\rangle$ is M-bounded. We shall show that whenever $\langle \mathcal{M}', s'\rangle$ is M-bounded and $X = \{\psi \in \mathrm{cl}(d, \phi) : \mathcal{M}', s' \models \psi\}$ for some $d \in \{0, \ldots, |\phi|\}$ and $X \subseteq \mathrm{cl}(d, \phi)$, then $\mathrm{SAT}(X, d)$ has an accepting computation. We recall that X is d-locally consistent. Consequently, we get that $\mathrm{SAT}(\{\psi \in \mathrm{cl}(0, \phi) : \mathcal{M}, s \models \psi\}, 0)$ has an accepting computation.

The proof is by induction on $d_{max} - d$ where d_{max} is the maximal value such that $\mathrm{cl}(d_{max}, \phi) \neq \emptyset$.
Base case: $d = |\phi|$.
Since $\mathrm{cl}(|\phi|, \phi) = \emptyset$, the property holds.
Induction hypothesis: for all $|\phi| \geqslant d' \geqslant n \geqslant 1$, and $X \subseteq \mathrm{cl}(d', \phi)$ such that there exist an EML model $\mathcal{M}' = \langle T', R', (<'_s)_{s\in T'}, l'\rangle$ and $s' \in T'$ verifying

$X = \{\psi \in \mathrm{cl}(d', \phi) : \mathcal{M}', s' \models \psi\}$ and $\langle \mathcal{M}', s' \rangle$ is M-bounded, $\mathrm{SAT}(X, d')$ has an accepting computation.

Let $d' = n - 1$ and X be a subset of $\mathrm{cl}(d', \phi)$ for which there exist an EML model $\mathcal{M}' = \langle T', R', (<'_s)_{s \in T'}, l' \rangle$ $nd' \in T'$ verifying $X = \{\psi \in \mathrm{cl}(d', \phi) : \mathcal{M}', nd' \models \psi\}$ and $\langle \mathcal{M}', s' \rangle$ is M-bounded. The set X is therefore d'-locally consistent and EML satisfiable, i.e. $\bigwedge_{\psi \in X} \psi$ is EML satisfiable.

For $i \in \{1, \ldots, nb(d'+1)\}$, we write n_i to denote the number of children nd'' of nd' such that $Y_i = \{\psi \in \mathrm{cl}(d+1, \phi) : \mathcal{M}, s'' \models \psi\}$. Since $\mathcal{M}'$ is M-bounded, $\sum_i n_i \leqslant nb(d'+1) \times M$. This is sufficient to establish that $\mathrm{SAT}(X, d')$ has an accepting computation. Indeed, the step (consistency) is successful because X is d'-locally consistent. The guessed number NB is obviously $n_1 + \cdots + n_{n_{nb(d'+1)}}$ and each set Y_i is guessed n_i times in the step (guess-children-from-left-to-right). Additionally, the order in which the sets Y_i are guessed is precisely given by the ordering of the children of the root of $\mathcal{M}'$. Since $\mathcal{M}'$ is a model for X, for every $i \in \{1, \ldots, nb(d'+1)\}$, if $n_i \neq 0$, then the set Y_i is satisfiable in some M-bounded model. By the induction hypothesis, $\mathrm{SAT}(Y_i, d'+1)$ returns `true`. Each passage to (guess-children-from-left-to-right)(4,5) as well as the passage to (final-checking) are successful steps because the numbers of children is computed from $\mathcal{M}'$. Consequently, $\mathrm{SAT}(X, d')$ has an accepting computation. $\Box$

So, we have established that a formula ϕ is EML satisfiable in a M-bounded model iff for some $X \subseteq \mathrm{cl}(0, \phi)$, $\mathrm{SAT}(X, \phi, 0)$ has a computation that returns `true`.

3.5 A Sharp Bound for EML *Models*

In this section, we state the following lemma which provides a bound on the size of minimal models for EML. The proof of the lemma is partly based on [SSM07] and Appendix A provides all technical details that are missing in the proof sketch below.

Lemma 7 *There is a polynomial $p(\cdot)$ such that for every formula ϕ, ϕ is* EML *satisfiable iff ϕ is satisfiable in some $2^{p(|\phi|)}$-bounded model.*

Proof.(sketch) Given a d-locally consistent set X, the main part of the proof consists in building a Boolean combination of arithmetical constraints, say $\mathcal{S}_X$, such that X is EML satisfiable iff $\mathcal{S}_X$ has a solution. In the system $\mathcal{S}_X$, the atomic constraints are of the form $\sum_j a_j \times x_{i_j} = b$ ($a_j \in \mathbb{Z}$, $b \in \mathbb{N}$, $x_{i,j}$ is a variable). By [Pap81] (see also [BT76]), a finite set $\mathcal{S}$ of atomic constraints has a positive solution iff there is a positive solution such that all the coefficients are bounded by

$$n \times (ma)^{2m+1}$$

where n is the number of variables, a is the maximal absolute value among constants in $\mathcal{S}$ and m is the number of atomic constraints in $\mathcal{S}$. In the system $\mathcal{S}_X$, to each $(d+1)$-locally consistent set Y_i, there is a variable x_i counting how many children of the node satisfying X satisfy Y_i. Hence, small solutions for $\mathcal{S}_X$ would imply that the number of children for that node satisfying X is bounded. The statement of the lemma is then obtained by applying such a reasoning at any depth of the tree model.

Let us briefly sketch how the system $\mathcal{S}_X$ is built. For each $\psi \in \mathrm{cl}(d+1, \phi)$, we write t_ψ to denote the sum $\sum_{\{i \,:\, \psi \in Y_i\}} x_i$. Remember that the $(d+1)$-locally consistent sets are denoted by $Y_1, \ldots, Y_{nb(d+1)}$ and each variable x_i is related to the number of children satisfying the subformulae in Y_i. First, we require that the sum of the x_i's such that Y_i is not EML satisfiable is zero. This constraint will be checked on-the-fly in SAT by using recursive calls. Moreover, an atomic formula $\sum_i a_i \# \psi_i = b \in X$ leads to the atomic constraint $\sum_i a_i t_{\psi_i} = b$ in $\mathcal{S}_X$. The other arithmetical constraints are treated in a similar way. The automata-based formulae in X or the negation of such formulae in X are treated in the following way. Let $\mathcal{A}_1(\ldots), \ldots, \mathcal{A}_l(\ldots), \neg\mathcal{A}'_1(\ldots), \ldots, \neg\mathcal{A}'_{l'}(\ldots) \in X$ be such formulae in X for which the argument subformulae are in $\{\psi_1, \ldots, \psi_P\} \subseteq \mathrm{cl}(d+1, \phi)$. We consider the enriched alphabet $\Sigma = \{Y_1, \ldots, Y_{nb(d+1)}\}$ made of $(d+1)$-locally consistent sets. Using the subset construction for finite-state automata, one can build an exponential-size automaton $\mathcal{B}$ over the alphabet Σ such that for every $w = Y_{j_1} \cdots Y_{j_\alpha} \in \Sigma^*$, $w \in \mathrm{L}(\mathcal{A})$ iff the conditions below hold:

- For all i, there exist formulae $\psi_1 \in Y_{j_1}, \ldots, \psi_\alpha \in Y_{j_\alpha}$ such that $\psi_1 \cdots \psi_\alpha \in \mathrm{L}(\mathcal{A}_i)$.
- For all i, there are no formulae $\psi_1 \in Y_{j_1}, \ldots, \psi_\alpha \in Y_{j_\alpha}$ such that $\psi_1 \cdots \psi_\alpha \in \mathrm{L}(\mathcal{A}'_i)$.

The set of atomic constraints obtained from $\mathcal{B}$ is obtained from the characterization of its Parikh image, that is a subset of $\mathbb{N}^{|\Sigma|}$. A Parikh image of a finite word built over the alphabet Σ is a tuple in $\mathbb{N}^{|\Sigma|}$ that contains for each letter in Σ, its number of occurrences. The Parikh image of a language is defined as a set of such tuples in $\mathbb{N}^{|\Sigma|}$ obtained from the Parikh image of words from the language. We recall that given a finite-state automaton $\mathcal{A} = \langle \Sigma, Q, \delta, I, F \rangle$, its Parikh image $\pi(\mathrm{L}(\mathcal{A}))$ is a finite union of linear sets $\{\sigma_0 + \sum_{i=1}^{m} y_i \sigma_i : y_i \geqslant 0\}$. By [SSMH04], we can enforce that each σ_j is in $\{0, \ldots, |Q|\}^{|\Sigma|}$ and m is bounded by $(|Q|+1)^{|\Sigma|}$. So, $\pi(\mathrm{L}(\mathcal{B}))$ is equal to some union $\mathrm{L}_1 \cup \cdots \cup \mathrm{L}_m$ with $\mathrm{L}_i = \{\sigma_0 + \sum_{j=1}^{h} \; y_j \sigma_j : y_j \geqslant 0\}$. Each σ_j is in $\{0, \ldots, |Q'|\}^{|\Sigma|}$ where Q' is the set of states for $\mathcal{B}$ and h is bounded by $(|Q'|+1)^{|\Sigma|} \leqslant 2^{p(|\phi|) \times 2^{|\phi|}}$. The set of atomic constraints obtained from $\mathcal{B}$ is then

$$\begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_{|\Sigma|} \end{pmatrix} = \sigma_0 + \sum_{j=1}^{h} y_j \sigma_j .$$

It contains $|\Sigma| + h$ variables and it admits a (small) solution whose values are at most doubly exponential in $|\phi|$. This is too large for the bound we aimed to. In order to get a constraint system with small solutions of adequate size, we take advantage of [SSM07, Claim 7.3]. Full details are provided in Appendix A. $\square$

By Lemmas 4, 5, 6 and 7 (and PSPACE-hardness of modal logic K), we obtain the main result of the paper. Indeed, M can be choosen exponential in $|\phi|$.

Theorem 1 EML *satisfiability is* PSPACE-*complete.*

PSPACE-hardness follows from the fact that $\Diamond$ can be encoded as a simple regularity constraint, whence the reduction from modal logic K.

4 Complexity results for similar logics

In this section, we compare EML with other logics dealing with Presburger constraints. We clarify the relationships between EML and the different logics from [ZL06,SSMH04,ABD+05] and to state some new PSPACE-completeness and undecidability results.

4.1 Graded Modal Logics

Graded modal logics are obviously the modal ancestors of EML where the formulae with Presburger constraints are of the form $\Diamond_{\geqslant n}\phi$, are considered, see e.g. the early works [Fin72,BC85,Cer90,vdH92,vdHdR95].

Such logics have been extended to fit more specific motivations, giving epistemic logics [vdHM91] and description logics (see e.g. [HB91,CG05]) with graded modalities. It is only in [Tob01] that minimal graded modal logic, counterpart of the modal logic K, is shown decidable in PSPACE, various decidability results being earlier established in a systematic way in [Cer94]. Our complexity result about EML extends the main result from [Tob01]. Various extensions of known logics by adding graded modalities have been considered and undecidability is often obtained because the ability to count allows sometime to encode a grid, see e.g. [BP04]. However, the EXPTIME-completeness of graded μ-calculus [KSV02] remains a major complexity result. Furthermore, there exist various attempts to encode concisely logics with counting into logics with no explicit counting mechanism, see e.g. [OSH96,MP97,Kaz04], but none of them implies a PSPACE upper bound, even for the poor minimal graded modal logic counterpart of K. Modal-like logics with more expressive Presburger constraints on the number of children can be found in [SSMH04,ZL06,SSM07] and are the subject of the two next sections.

4.2 Sheaves Logic

4.2.1 Definition

In this section, we recall the syntax and semantics of the Sheaves Logic SL [ZL06, Section 4.4] that is shown decidable in [ZL06] with a non-elementary algorithm. For the sake of uniformity, we adopt a presentation of SL models similar to the one for EML models whereas the mode of representation for regular languages and semilinear sets is the same as for EML. Hence, regular languages are represented by finite-state automata (instead of regular expressions in [ZL06] that are less concise) and arithmetical constraints are represented by quantifier-free Presburger formulae as in EML (instead of Presburger formulae in [ZL06] that are much more concise). We admit that our choice of representations for such objects is crucial to obtain the forthcoming PSPACE upper bound and we thought that it is fair to use the same encodings as in EML. Apart from the mode representation, the logic presented below differs from the one in [ZL06, Section 4.4] since herein we allow Boolean operators at the level of element formulae (denoted by E) as done for document formulae (denoted by D).

The major difference between SL and EML rests on the evaluation of quantifier-free Presburger formulae. In EML, in order to evaluate a Boolean combination of atomic formulae of the form either $\sum_i a_i \sharp\phi_i \sim b$ or $\sum_i a_i \sharp\phi_i \equiv_k c$, each successor node can contribute to the interpretation of more than one expression of the form $\sharp\phi_i$. By contrast, in order to evaluate the analogous formula in SL (see below the formulae of the form $\exists x_1, \ldots, x_p : \sum_i a_i x_i \sim b : x_1 E_1 \& \cdots \& x_p E_p$), each successor node contributes to the interpretation of exactly one analogous expression of the form $\sharp\phi_i$, namely x_i.

The element and document formulae are inductively defined as follows:

- $E := \alpha[D] \mid \delta \mid \neg E \mid E \wedge E \mid \texttt{true}$,
- $D := \mathcal{A}(E_1, \ldots, E_p) \mid \exists x_1, \ldots, x_p : \phi(x_1, \ldots, x_p) : x_1 E_1 \& \cdots \& x_p E_p \mid \texttt{true} \mid \neg D \mid D \wedge D$,

where

- α belongs to a countably infinite set TAGS of tags,
- δ belongs to a countably infinite set DATATYPES of datatypes, disjoint from TAGS,
- $\mathcal{A}$ is a nondeterministic finite-state automaton over an p-letter alphabet $\Sigma_{\mathcal{A}}$ in which the letters are linearly ordered $\Sigma_{\mathcal{A}} = \mathtt{a}_1, \ldots, \mathtt{a}_p$.
- $\phi(x_1, \ldots, x_p)$ is a Boolean combination of Presburger formulae built over the variables $x_1, \ldots, x_p$ of the form either $t \sim b$ with $\sim \in \{<, >, =\}$ or $t \equiv_k c$ with $t = \sum_i a_i x_i$.

A model $\mathcal{M}$ for SL is a structure $\mathcal{M} = \langle T, R, (<_s)_{s \in T}, l \rangle$ where

- T is a finite set of states,
- $\langle T, R \rangle$ is a tree and each $<_s$ is a total ordering on $R(s)$,
- $l : T \rightarrow \mathrm{TAGS} \cup \mathrm{DATATYPES}$ is a labeling function such that
  - · for every $s \in T$, if s is a leaf of $\langle T, R \rangle$ then $l(s) \in \mathrm{DATATYPES}$,
  - · for every $s \in T$, if s is not a leaf of $\langle T, R \rangle$ then $l(s) \in \mathrm{TAGS}$.

The satisfaction relation $\models$ is inductively defined below where $\mathcal{M}$ is a model for SL and $s \in T$ (we omit the clauses for Boolean operators):

- $\mathcal{M}, s \models \delta$ iff $\delta = l(s)$,
- $\mathcal{M}, s \models \alpha[D_1 \wedge D_2]$ iff $\mathcal{M}, s \models \alpha[D_1]$ and $\mathcal{M}, s \models \alpha[D_2]$,
- $\mathcal{M}, s \models \alpha[\neg D]$ iff $\alpha = l(s)$ and not $\mathcal{M}, s \models \alpha[D]$,
- $\mathcal{M}, s \models \alpha[\mathtt{true}]$ iff $\alpha = l(s)$,
- $\mathcal{M}, s \models \alpha[\exists x_1, \ldots, x_p : \phi(x_1, \ldots, x_p) : x_1 E_1 \& \cdots \& x_p E_p]$ iff $\alpha = l(s)$, $R(s) = s_1 < \cdots < s_k$, and there exist $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, s_j \models E_{i_j}$ and $[x_1 \leftarrow n_1, \ldots, x_p \leftarrow n_p] \models \phi(x_1, \ldots, x_p)$ with $n_i = \mathrm{card}(\{s \in \{1, \ldots, k\} : i_s = i\})$,
- $\mathcal{M}, s \models \alpha[\mathcal{A}(E_1, \ldots, E_p)]$ iff $\alpha = l(s)$, $R(s) = s_1 < \cdots < s_k$, and there is $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, s_j \models E_{i_j}$ and $\mathtt{a}_{i_1} \cdots \mathtt{a}_{i_k} \in \mathrm{L}(\mathcal{A})$ with $\Sigma_{\mathcal{A}} = \mathtt{a}_1, \ldots, \mathtt{a}_p$.

As said earlier, the major difference with the semantics of EML (see also [SSMH04]) is that in Presburger constraints each child counts *only once*.

4.2.2 PSPACE-completeness

Let ϕ be an SL formula with tags $\{\alpha_1, \ldots, \alpha_n\}$ and datatypes $\{\delta_1, \ldots, \delta_{n'}\}$. We define a EML formula ϕ' built over the propositional variables (plus others, see below)

$$AP = \{p_{\alpha_1}, \ldots, p_{\alpha_n}, p_{\alpha_{new}}\} \cup \{p_{\delta_1}, \ldots, p_{\delta_{n'}}, p_{\delta_{new}}\}.$$

Given an EML formula φ, we write $\forall^m \varphi$ as an abbreviation for $\bigwedge_{i=0}^{m} \overbrace{\Box \ldots \Box}^{i \text{ times}} \varphi$. The formula ϕ' is defined as the conjunction $\phi'_{val} \wedge t(\phi)$ where $t(\phi)$ is defined recursively on the structure of ϕ and ϕ'_{val} states constraints about the valuation of datatypes and tags in SL models. For each document formula of the form $D = \exists x_1 \cdots x_p : \phi(x_1, \ldots, x_p) : x_1 E_1 \ \& \cdots \& \ x_p E_p$ in ϕ, we introduce new propositional variables $p_D^1, \ldots, p_D^p$.

The formula ϕ'_{val} is defined as the conjunction of the formulae below

- $\forall^{|\phi|} \bigvee_{p \in AP} (p \wedge \bigwedge_{q \in AP \setminus \{p\}} \neg q) \ \wedge \ \overbrace{\forall^{|\phi|} (\Diamond \mathtt{true} \Rightarrow \bigvee_{\alpha \in \{\alpha_1, \ldots, \alpha_n, \alpha_{new}\}} p_\alpha)}^{\text{internal nodes labeled by tags}}$
- $\underbrace{\forall^{|\phi|} (\Box \mathtt{false} \Rightarrow \bigvee_{\delta \in \{\delta_1, \ldots, \delta_{n'}, \delta_{new}\}} p_\delta)}_{\text{leaves labeled by datatypes}}$
- $\forall^{|\phi|} (\bigwedge_{D \text{ is of the form } \exists \ldots E_p} \ (\bigwedge_{i \neq j \in \{1, \ldots, p\}} \neg (p_D^i \wedge p_D^j) \wedge (p_D^i \Rightarrow t(E_i))))$.

where $|\phi|$ is the size of ϕ (an optimal construction would consider $\mathrm{md}(\phi)$) and t is the reduction from SL formulae to EML formulae defined below.

- t is homomorphic for Boolean operators and $t(\mathtt{true}) = \mathtt{true}$,
- $t(\alpha_i[D]) = p_{\alpha_i} \wedge t(D)$, $t(\delta_i) = p_{\delta_i}$,
- $t(\mathcal{A}(E_1, \ldots, E_p)) = \mathcal{A}(t(E_1), \ldots, t(E_p))$,
- $t(\exists x_1 \cdots x_p : \phi(x_1, \ldots, x_p) : x_1 E_1 \ \& \cdots \& \ x_p E_p)$ equals the formula below:

  $$\phi(x_1, \ldots, x_p)[x_1 \leftarrow \sharp(p_D^1), \ldots, x_p \leftarrow \sharp(p_D^p)] \wedge \neg \sharp(\neg p_D^1 \wedge \cdots \wedge \neg p_D^p) > 0.$$

  where $\phi(x_1, \ldots, x_p)[x_1 \leftarrow \sharp(p_D^1), \ldots, x_p \leftarrow \sharp(p_D^p)]$ is obtained from $\phi(x_1, \ldots, x_p)$ by replacing each occurrence of x_i by $\sharp(p_D^i)$.

New propositional variables need to be introduced and a constraint on them needs to be stated because in SL in Presburger constraints each child can count only once. It is not difficult to show that t is sound.

Lemma 8 *t is a logspace reduction such that ϕ is satisfiable iff ϕ' is satisfiable.*

Proof. First, suppose that ϕ is SL satisfiable. There exist an SL model $\mathcal{M} = \langle T, R, (<_s)_{s \in T}, l \rangle$ and $s \in T$ such that $\mathcal{M}, s \models \phi$. Let $\mathcal{M}'$ be the EML model $\mathcal{M}' = \langle T', R', (<'_s)_{s \in T'}, l' \rangle$ defined by:

- $\langle T', R', (<'_s)_{s \in T'} \rangle = \langle T, R, (<_s)_{s \in T} \rangle$,
- for every $s \in T'$, $p_{l(s)} \in l'(s)$. Moreover, $l'(s)$ may contain other propositional variables of the form p_D^i as explained below. Let $D = \exists x_1 \cdots x_p : \phi(x_1, \ldots, x_p) : x_1 E_1 \ \& \cdots \& \ x_p E_p$ be a document formula occurring in ϕ.

If $\mathcal{M}, s \models \alpha[D]$, then by definition $R(s) = s_1 < \cdots < s_k$, and there are $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, s_j \models E_{i_j}$ and $[x_1 \leftarrow n_1, \ldots, x_p \leftarrow n_p] \models \phi(x_1, \ldots, x_p)$ with $n_i = \mathrm{card}(\{s \in \{1, \ldots, k\} : i_s = i\})$. So for every j, we require that $p_D^{i_j} \in l'(s_j)$.
If $\mathcal{M}, s \not\models \alpha[D]$ and $\mathcal{M}, s \models \alpha[\exists x_1 \cdots x_p : \neg\phi(x_1, \ldots, x_p) : x_1 E_1 \,\&\cdots\&\, x_p E_p]$, then by definition $R(s) = s_1 < \cdots < s_k$, and there are $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, s_j \models E_{i_j}$ and $[x_1 \leftarrow n_1, \ldots, x_p \leftarrow n_p] \models \neg\phi(x_1, \ldots, x_p)$ with $n_i = \mathrm{card}(\{s \in \{1, \ldots, k\} : i_s = i\})$. So for every j, we require that $p_D^{i_j} \in l'(s_j)$.

If $\mathcal{M}, s \not\models \alpha[D]$ and $\mathcal{M}, s \not\models \alpha[\exists x_1 \cdots x_p : \neg\phi(x_1, \ldots, x_p) : x_1 E_1 \,\&\cdots\&\, x_p E_p]$, then this means either there is one child of s, say s', satisfies none of the E_i or $l(s) \neq \alpha$. So, we require that none of the p_D^is belongs to $l(s'')$ for $s'' \in R(s)$.

By structural induction, one can show that $\mathcal{M}', s \models t(\phi)$.

Now suppose that $\phi'_{val} \wedge t(\phi)$ is EML satisfiable. There exist a EML model $\mathcal{M} = \langle T, R, (<_s)_{s \in T}, l\rangle$ and $s \in T$ such that $\mathcal{M}, s \models \phi$. Let $\mathcal{M}'$ be the SL model $\mathcal{M}' = \langle T', R', (<'_s)_{s \in T'}, l'\rangle$ defined by:

- $\langle T', R', (<'_s)_{s \in T'}\rangle = \langle T, R, (<_s)_{s \in T}\rangle$,
- for every $s \in T'$, $l'(s) = \beta$ where β is the unique element of TAGS $\cup$ DATATYPES such that $p_\beta \in l(s)$ ($l(s)$ may contain other propositional variables of the form p^i_{new}). Unicity is guaranteed by the satisfaction of ϕ'_{val}.

It is easy to show that $\mathcal{M}', s \models \phi$. $\square$

Consequently, SL is in PSPACE which contrasts with the non-elementary complexity of the decision procedure from [ZL06].

Proposition 1 *SL satisfiability problem is* PSPACE*-complete.*

Proof. It remains to establish the PSPACE-hardness of SL. This can be done by reducing the satisfiability problem for minimal modal logic K with no propositional variable but with logical constant `true` and `false` that is already PSPACE-complete [Hem01]. We can even restrict ourselves to negation-free formulae. Let us define a reduction t' from this fragment of modal logic K into SL:

- $t'(\mathtt{true}) = \mathtt{true}$, $t'(\mathtt{false}) = \neg\mathtt{true}$,
- $t'(\phi \wedge \phi') = t'(\phi) \wedge t'(\phi')$,
- $t'(\phi \vee \phi') = \neg(\neg t'(\phi) \wedge \neg t'(\phi'))$,
- $t'(\Diamond\phi) = \alpha[\exists\, x : x \geqslant 1 : x\ t'(\phi)]$,
- $t'(\Box\phi) = \alpha[\exists\, x : x = 0 : x\ \neg t'(\phi)] \vee \delta$.

α is a tag (always the same) and δ is a datatype (always the same). One can show that the negation-free formula ϕ (with no propositional variable) is K satisfiable iff $t'(\phi)$ is SL satisfiable.

Suppose that ϕ is K satisfiable. So there is a tree model $\mathcal{M} = \langle W, R \rangle$ (no need for labeling) and $w \in W$ such that $\mathcal{M}, w \models \phi$ (the logic K has the finite tree model property). The SL model $\mathcal{M}' = \langle T', R', (<'_s)_{s \in T'}, l' \rangle$ is defined as follows:

- $T' = W$, $R' = R$,
- For every $s \in T'$, $<'_s$ is an arbitrary linear ordering on $<'_s$. These orderings are irrelevant because $t'(\phi)$ has no regularity constraint.
- For every $s \in T'$, if s is a leaf then $l'(s) = \delta$, otherwise $l'(s) = \alpha$.

It is easy to show that $\mathcal{M}', w \models t'(\phi)$. Similarly if $\mathcal{M}$ is a model for $t'(\phi)$, then a model for ϕ is obtained from $\mathcal{M}$ by deleting the labeling and the family of orderings. $\square$

The logics SL and EML interpreted over finite trees cannot be immediately compared because of the presence of tags and datatypes in SL, whence the models are different. The reduction from SL into EML in the proof of Lemma 8 allows to transform an SL model into an EML model. For the other direction, assuming that in SL the nodes are labelled by propositional valuations as in EML, the reduction from EML into SL amounts to being able to express atomic formulae of the form $\sum_{i=1}^{i=\gamma} a_i \sharp p_i \sim b$, which can be captured by the SL formula below:

$$\bigvee_v v[\exists x_0 \cdots x_{2^\gamma - 1} : \sum_{i=1}^{i=\gamma} a_i (\sum_{j : p_i \in X_j} x_j) \sim b : x_0 E_0 \ \& \cdots \& \ x_{2^\gamma - 1} E_{2^\gamma - 1}]$$

where v ranges over all the propositional valuations over a given finite set of propositional variables and, the E_j's and X_j's are defined as follows. Given $j \in \{0, \ldots, 2^\gamma - 1\}$, we write $X_j \subseteq \{p_1, \ldots, p_\gamma, \neg p_1, \ldots, \neg p_\gamma\}$ such that for $1 \leqslant i \leqslant \gamma$, the ith bit of j in binary representation is 1 iff $p_i \in X_j$ iff $\neg p_i \not\in X_j$. The formula E_j is defined as follows:

$$\bigwedge_{\{i:\ i\text{th bit of } j \text{ is } 1\}} p_i \wedge \bigwedge_{\{i:\ i\text{th bit of } j \text{ is } 0\}} \neg p_i$$

Since no node can satisfy two distinct E_j, $E_{j'}$, this allows us to tame the fact that each child counts only once in an arithmetical constraint. However, the translation causes an exponential blowup.

In this section, we recall the syntax and semantics of the fixpoint free fragment of the logic from [SSMH04]. For brevity, we call it SSMH. The full logic in [SSMH04] contains additionnally fixpoint operators and it is a strict extension of modal μ-calculus, see e.g. [BS07]. Like for SL, definitions are adapted to our presentation to EML which allows to compare easily the (sometimes minor) differences between EML, SL and SSMH. The SSMH formulae are inductively defined as follows:

$$\phi ::= \texttt{true} \mid \neg\phi \mid \phi \wedge \phi' \mid \alpha\langle \Phi(x_1, \ldots, x_p) : x_1\phi_1 \& \cdots \& x_p\phi_p\rangle \mid$$

$$\star\langle \Phi(x_1, \ldots, x_p) : x_1\phi_1 \& \cdots \& x_p\phi_p\rangle \mid \alpha\langle \mathcal{A}(\phi_1, \ldots, \phi_p)\rangle \mid \star\langle \mathcal{A}(\phi_1, \ldots, \phi_p)\rangle.$$

where

- α belongs to a countably infinite set TAGS of tags,
- $\mathcal{A}$ is a nondeterministic finite-state automaton over an p-letter alphabet,
- $\Phi(x_1, \ldots, x_p)$ is a Presburger formula as in SL.

A model $\mathcal{M}$ for SSMH is a structure $\mathcal{M} = \langle T, R, (<_{s)_{s \in T}}, l\rangle$ where

- T is a finite set of states,
- $\langle T, R\rangle$ is a tree and each $<_s$ is a total ordering on $R(s)$,
- $l : T \to$ TAGS is a labeling function (no datatypes here).

The satisfaction relation is inductively defined below where $\mathcal{M}$ is a model for SSMH and $s \in T$ (we omit the clauses for Boolean operators):

- $\mathcal{M}, s \models \alpha$ iff $\alpha = l(s)$,
- $\mathcal{M}, s \models \alpha\langle \Phi(x_1, \ldots, x_p) : x_1\phi_1 \& \cdots \& x_p\phi_p\rangle$ iff $\alpha = l(s)$ and $R(s) = s_1 < \cdots < s_k$ and $[x_1 \leftarrow n_1, \ldots, x_p \leftarrow n_p] \models \Phi(x_1, \ldots, x_p)$ where $n_i = \mathrm{card}(\{s \in \{1, \ldots, k\} : \mathcal{M}, s_s \models \phi_i\})$,
- $\mathcal{M}, s \models \star\langle \phi(x_1, \ldots, x_p) : x_1\phi_1 \& \cdots \& x_p\phi_p\rangle$ iff $[x_1 \leftarrow n_1, \ldots, x_p \leftarrow n_p] \models \Phi(x_1, \ldots, x_p)$ where $n_i = \mathrm{card}(\{s \in \{1, \ldots, k\} : \mathcal{M}, s_s \models \phi_i\})$,
- $\mathcal{M}, s \models \alpha\langle \mathcal{A}(\phi_1, \ldots, \phi_p)\rangle$ iff $\alpha = l(s)$, $R(s) = s_1 < \cdots < s_k$ and there is $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, s_j \models \phi_{i_j}$ and $\mathtt{a}_{i_1} \cdots \mathtt{a}_{i_k} \in \mathrm{L}(\mathcal{A})$. (analogous clause for $\star\langle \mathcal{A}(\phi_1, \ldots, \phi_p)\rangle$).

Unlike SL and like EML, a child can count more than once in Presburger constraints. Let ϕ be an SSMH formula with tags $\{\alpha_1, \ldots, \alpha_n\}$. We shall define an EML formula ϕ' built over the propositional variables $AP = \{p_{\alpha_1}, \ldots, p_{\alpha_n}, p_{\alpha_{n+1}}\}$. Let t be a logspace reduction from SSMH formulae to EML formulae:

- t is homomorphic for Boolean operators and $t(\texttt{true}) = \texttt{true}$,

- $t(\alpha\langle\phi(x_1,\ldots,x_p) : x_1\phi_1\&\cdots\&x_p\phi_p\rangle)$ equals

$$p_\alpha \wedge \phi(x_1,\ldots,x_p)[x_1 \leftarrow \sharp t(\phi_1),\ldots,x_p \leftarrow \sharp t(\phi_p)].$$

- $t(\star\langle\phi(x_1,\ldots,x_p) : x_1\phi_1\&\cdots\&x_p\phi_p\rangle)$ equals

$$\phi(x_1,\ldots,x_p)[x_1 \leftarrow \sharp t(\phi_1),\ldots,x_p \leftarrow \sharp t(\phi_p)].$$

- $t(\alpha\langle\mathcal{A}(\phi_1,\ldots,\phi_p)\rangle) = p_\alpha \wedge \mathcal{A}(t(\phi_1),\ldots,t(\phi_p))$,
- $t(\star\langle\mathcal{A}(\phi_1,\ldots,\phi_p)\rangle) = \mathcal{A}(t(\phi_1),\ldots,t(\phi_p))$.

Lemma 9 *t is a logspace reduction such that ϕ is satisfiable iff $\forall^{|\phi|}\bigvee_{p\in AP}(p \wedge \bigwedge_{q\in AP\setminus\{p\}}\neg q) \wedge t(\phi)$ is satisfiable.*

The proof is similar (and indeed simpler) than the proof of Lemma 8. The subformula $\forall^{|\phi|}\bigvee_{p\in AP}(p \wedge \bigwedge_{q\in AP\setminus\{p\}}\neg q)$ guarantees that each node satisfies exactly one atomic proposition from AP. Observe that SSMH and EML has similar expressive power and one can see them as syntactic variants for any bijection between a finite set of propositional valuations and a finite set of tags.

Proposition 2 *SSMH satisfiability problem is* PSPACE*-complete.*

Proof. It remains to establish PSPACE-hardness. We reduce again negation-free fragment of K with no propositional variable to SSMH:

- $t'(\texttt{true}) = \texttt{true}$, $t'(\texttt{false}) = \neg\texttt{true}$,
- t' is homomorphic for Boolean operators,
- $t'(\Diamond\phi) = \star\langle\exists\ x : x \geqslant 1 : x\ t'(\phi)\rangle$,
- $t'(\Box\phi) = \star\langle\exists\ x : x = 0 : x\ \neg t'(\phi)\rangle$.

It is easy to show that ϕ is K satisfiable iff $t'(\phi)$ is SSMH satisfiable. $\Box$

The main differences between SSMH and its extension with fixpoint operators from [SSMH04] is similar to the difference between the modal logic K and the modal μ-calculus. For instance, whenever a formula in SSMH has a tree model, its depth can be polynomially bounded which is not anymore the case when fixpoint operators are added. This partly explains why the satisfiability problem for SSMH is only in PSPACE and for the full logic EXPTIME-complete [SSM07]. Formulae in SSMH can only produce constraints on immediate successors of a node (at distance at most the modal depth of the formula), whereas fixpoint operators allow to express reachability operators that can produce constraints on all descendant nodes.

In [ABD+05] a PDL-like logic $\mathtt{PDL_{tree}}$ is introduced where models are finite, labeled ordered trees and the four atomic relations are: left-sibling, right-sibling, mother-of and daughter-of. Other relations can be generated with standard "program operators" (iteration, test, union and composition). There is no (full) Presburger constraints in $\mathtt{PDL_{tree}}$ (except the obvious ones derived from the standard modal operators) but regularity constraints can be stated thanks to the interplay between the program operators and the atomic relations. $\mathtt{PDL_{tree}}$ satisfiability is shown EXPTIME-complete in [ABD+05]. It is not difficult to show that, on the model of the undecidability proof for [ZL06, Proposition 1], adding Presburger constraints to $\mathtt{PDL_{tree}}$ leads to undecidability. We provide below an undecidability proof for a logic sharing features from $\mathtt{PDL_{tree}}$ and EML, say $\mathcal{L}$, that is a strict fragment of the logic $\mathtt{PDL_{tree}}$ on which are added Presburger constraints. Hence, the logic $\mathcal{L}$ is an hybrid version of $\mathtt{PDL_{tree}}$ and EML without subsuming any of them. Nevertheless, below, the satisfiability problem for $\mathcal{L}$ will be shown undecidable mainly because of the ability to compare for each node, its number of siblings with its number of children. Hence, as illustrated below, combining this type of comparisons with the ability to access to all descendant nodes, makes the logic too expressive to retain decidability.

Given a countably infinite set $\mathrm{AT} = \{p_1, p_2, \ldots\}$ of propositional variables and $\Sigma = \{\downarrow, \downarrow^*, \rightarrow, \rightarrow^*, \leftarrow, \leftarrow^*, \uparrow, \uparrow^*\}$ a set of relation symbols, we define the set of formulae and terms inductively as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid t \sim b \quad t ::= a \times \sharp^{\mathtt{R}}\phi \mid t + a \times \sharp^{\mathtt{R}}\phi$$

where $p \in \mathrm{AT}$, $\mathtt{R} \in \Sigma$, $b \in \mathbb{N}$, $a \in \mathbb{Z}\backslash\{0\}$ and $\sim \in \{<, >, =\}$. The programs from $\mathtt{PDL_{tree}}$ are much richer than Σ because iteration, test, union and composition are present in $\mathtt{PDL_{tree}}$. Similarly, the Presburger constraints from EML strictly contains those of $\mathcal{L}$, as $\mathcal{L}$ has no modulo constraints. A model $\mathcal{M}$ for $\mathcal{L}$ is a structure

$$\mathcal{M} = \langle T, R_{\downarrow}, R_{\downarrow^*}, R_{\rightarrow}, R_{\rightarrow^*}, R_{\leftarrow}, R_{\leftarrow^*}, R_{\uparrow}, R_{\uparrow^*}, l\rangle$$

where

- $\langle T, R_{\downarrow}, R_{\rightarrow}\rangle$ is a finite ordered tree with $R_{\downarrow}$ and $R_{\rightarrow}$ are child-of and right-sibling relations, respectively;
- $l : T \rightarrow 2^{\mathrm{AT}}$ is the valuation function,
- for every $\mathtt{R} \in \{\downarrow, \rightarrow, \leftarrow, \uparrow\}$, $R^*_{\mathtt{R}} = R_{\mathtt{R}^*}$ ($R^*_{\mathtt{R}}$ is the reflexive and transitive closure of $R_{\mathtt{R}}$), $R_{\rightarrow} = R^{-1}_{\leftarrow}$ and $R_{\uparrow} = R^{-1}_{\downarrow}$,

The satisfaction relation is inductively defined as for EML except this time the models are finite ordered trees.

Proposition 3 *The satisfiability problem for $\mathcal{L}$ is undecidable.*

Proof. The proof is by reducing the halting problem for 2-counter machine. A 2-counter machine M consists of two counters C_1 and C_2, and a sequence of $n \geqslant 1$ instructions. The Lth instruction is written as one of the following:

L : $C_i = C_i + 1$; goto L'.
L : if $C_i = 0$ then goto L' else $C_i = C_i - 1$; goto L''.

We represent the configurations of M by triples $\langle L, c_1, c_2 \rangle$ where $1 \leqslant L \leqslant n$, $c_1 \geqslant 0$ and $c_2 \geqslant 0$. A computation of M is a finite sequence of related configurations, starting with the initial configuration $\langle 1, 0, 0 \rangle$. The halting problem can be stated as the existence of a finite sequence of related configurations that reaches the instruction 1 in at least one step. We build a formula ϕ of $\mathcal{L}$ such that M halts iff ϕ is satisfiable in $\mathcal{L}$.

As usual, we use the standard notations:

$$\langle \mathtt{R} \rangle \phi \stackrel{\text{def}}{=} \sharp^{\mathtt{R}} \phi > 0 \qquad [\mathtt{R}]\phi \stackrel{\text{def}}{=} (\sharp^{\mathtt{R}} \neg \phi = 0).$$

A computation $\langle q_1, c_1, d_1 \rangle, \ldots, \langle q_t, c_t, d_t \rangle$ is encoded as a finite ordered tree of depth $t+1$ over the propositional variables $1, \ldots, n, n+1, n+2$. The variable $n+1$ [resp. $n+2$] is related to the counter C_1 [resp. C_2]. The root is labelled by no propositional variable (valuation $\{\}$) and the leftmost branch is the following sequence of valuations:

$$\{\}, \{q_1\}, \ldots, \{q_t\}.$$

Each node labelled by $\{q_i\}$ on that special branch has $c_i + d_i$ right-siblings with the following valuations

$$\overbrace{\{n+1\}, \ldots, \{n+1\}}^{c_i \text{ times}}, \overbrace{\{n+2\}, \ldots, \{n+2\}}^{d_i \text{ times}}.$$

The formula ϕ is defined as the conjunction of the following formulae and enforces the above encoding of computations:

- Initial configuration:

$$\neg(1 \vee \cdots \vee n+2) \wedge \langle \downarrow \rangle (1 \wedge \overbrace{(\sharp^{\rightarrow^*} n+1 = 0)}^{C_1 = 0} \wedge \overbrace{(\sharp^{\rightarrow^*} n+2 = 0)}^{C_2 = 0}).$$

- Unicity of the labelling of the nodes:

$$[\downarrow][\downarrow^*](\bigvee_{1 \leqslant i \leqslant n+2} (i \wedge \bigwedge_{i' \neq i} \neg i')).$$

- The instruction counter is the leftmost child:

$$[\downarrow][\downarrow^*]((\bigvee_{1\leqslant i\leqslant n} i) \Leftrightarrow \neg\langle\leftarrow\rangle\top).$$

- Encoding of C_1 is strictly before the encoding of C_2:

$$[\downarrow^*](n+1 \Rightarrow (\sharp^{\leftarrow^*} n+2 = 0)).$$

- Instruction L: $C_1 = C_1 + 1$; goto L'.

$$[\downarrow^*](L\wedge\langle\downarrow\rangle\top \Rightarrow \langle\downarrow\rangle[L'\wedge\overbrace{(\sharp^{\downarrow} n+1 - \sharp^{\rightarrow^*} n+1 = 1)}^{C_1:=C_1+1}\wedge\overbrace{(\sharp^{\downarrow} n+2 - \sharp^{\rightarrow^*} n+2 = 0)}^{C_2 \text{ is unchanged}})].$$

- Instruction L: if $C_1 = 0$ then goto L' else $C_1 = C_1 - 1$; goto L''.

$$[\downarrow^*](L \wedge \langle\downarrow\rangle\top \wedge \overbrace{(\sharp^{\rightarrow^*} n+1 = 0)}^{C_1=0} \Rightarrow$$

$$\langle\downarrow\rangle L' \wedge \overbrace{(\sharp^{\downarrow} n+1 - \sharp^{\rightarrow^*} n+1 = 0)}^{C_1 \text{ is unchanged}} \wedge \overbrace{(\sharp^{\downarrow} n+2 - \sharp^{\rightarrow^*} n+2 = 0)}^{C_2 \text{ is unchanged}})\wedge$$

$$[\downarrow^*](L \wedge \langle\downarrow\rangle\top \wedge \overbrace{\neg(\sharp^{\rightarrow^*} n+1 = 0)}^{C_1\neq 0} \Rightarrow$$

$$\langle\downarrow\rangle L'' \wedge \overbrace{(\sharp^{\rightarrow^*} n+1 - \sharp^{\downarrow} n+1 = 1)}^{C_1:=C_1-1} \wedge \overbrace{(\sharp^{\downarrow} n+2 - \sharp^{\rightarrow^*} n+2 = 0)}^{C_2 \text{ is unchanged}})$$

- The instruction 1 is reached after at least one step: $\langle\downarrow\rangle\langle\downarrow\rangle\langle\downarrow^*\rangle 1$.

Then, it is easy to show that M halts iff ϕ is satisfiable in $\mathcal{L}$. $\square$

If we modify the models by allowing infinite trees with finite-branching, satisfiability becomes Σ_1^1-hard by reducing the recurring problem for nondeterministic 2-counter machines [AH94, Lemma 8]. The formulae built in the proof of Proposition 3 are specific since only the relation symbols from $\{\downarrow^*, \downarrow, \rightarrow^*, \leftarrow\}$ are used. The decidability status of the following logics is still open:

- restriction of $\mathcal{L}$ to formulae with no subformula of the form $\sum_i a_i \sharp^{\mathtt{R}_i}\phi_i$ where for some $j \neq j'$, $\mathtt{R}_j \neq \mathtt{R}_{j'}$ (forbidding for instance the comparison of the number of siblings with the number of children),
- EML on finite trees with the relation symbols $\downarrow$ (as before) and the left-sibling relation $\leftarrow$,
- $\mathtt{PDL}_{\mathtt{tree}}$ augmented with a subclass of Presburger constraints.

The logic obtained by adding $\downarrow^*$ to EML is a fragment of the logic SSMH extended with fixpoints, for which satisfiability is shown decidable in [SSMH04]. Actually, this fragment is already EXPTIME-hard, even if we use only trivial regularity and Presburger constraints (by using the complexity result of [FL79]).

5 Concluding Remarks

In this paper, we have shown that the satisfiability problem for the logic EML is only PSPACE-complete. We improve previous results, for instance from [Tob01,SSMH04,ZL06], and we give a PSPACE bound for the satisfiability of a logic that generalizes many previous logics. The proof to obtain the PSPACE upper bound is established by designing a specially tailored Ladner-like algorithm and by using reasoning on constraint systems from [SSMH04]. The logic EML is therefore a remarkable example of modal logics with a reasonable complexity that combines counting abilities and regularity constraints, which are useful features for applications ranging from query language for XML documents to knowledge representation.

We plan to investigate decidable fragments of $\mathtt{PDL}_{\mathtt{tree}}$ augmented with Presburger constraints on the numbers of children that are more expressive than EML. For instance, the decidability status of EML extended with the left-sibling relation (and therefore with an enriched class of arithmetic constraints) is open.

Acknowledgment We would like to thank Anca Muscholl and Helmut Seidl for their explanations about some parts of the paper [SSM07] and the anonymous referees for helpful comments and suggestions on previous versions.

References

[ABD+05] L. Afanasiev, P. Blackburn, I. Dimitriou, B. Gaiffe, E. Goris, M. Marx, and M. de Rijke. PDL for ordered trees. *Journal of Applied Non-Classical Logics*, 15(2):115–135, 2005.

[ADdR03] N. Alechina, S. Demri, and M. de Rijke. A modal perspective on path constraints. *Journal of Logic and Computation*, 13(6):939–956, 2003.

[AH94] R. Alur and T. Henzinger. A really temporal logic. *Journal of the Association for Computing Machinery*, 41(1):181–204, 1994.

[BC85] M. Fattorosi Barnaba and F. De Caro. Graded modalities. *Studia Logica*, 44(2):197–221, 1985.

[BCT04] N. Bidoit, S. Cerrito, and V. Thion. A first step towards modeling semistructured data in hybrid multimodal logic. *Journal of Applied Non-Classical Logics*, 14(4):447–475, 2004.

[BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[Ber77] L. Berman. Precise bounds for Presburger arithmetic and the reals with addition: Preliminary report. In *FOCS'77*, pages 95–99, 1977.

[BP04] P. Bonatti and A. Peron. On the undecidability of logics with converse, nominals, recursion and counting. *Artificial Intelligence*, 158(1):75–96, 2004.

[BS07] J. Bradfield and C. Stirling. Modal mu-calculi. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, pages 721–756. Elsevier, 2007.

[BT76] I. Borosh and L. Treybig. Bounds on positive integral solutions of linear diophantine equations. *American Mathematical Society*, 55:299–304, 1976.

[BT05] I. Boneva and J.M. Talbot. Automata and logics for unranked and unordered trees. In *RTA'05*, volume 3467 of *Lecture Notes in Computer Science*, pages 500–515, 2005.

[Cer90] C. Cerrato. General canonical models for graded normal logics. *Studia Logica*, 49(2):242–252, 1990.

[Cer94] C. Cerrato. Decidability by filtrations for graded normal logics (graded modalities V). *Studia Logica*, 53(1):61–73, 1994.

[CG05] D. Calvanese and G. De Giacomo. Expressive description logics. In *Description Logics Handbook*, pages 178–218. Cambridge University Press, 2005.

[Dem03a] S. Demri. (modal) logics for semistructured data (bis). In *Third Workshop on Methods for Modalities, Nancy*, 2003. Invited talk. Slides available at `www.lsv.ens-cachan/~demri`.

[Dem03b] S. Demri. A polynomial space construction of tree-like models for logics with local chains of modal connectives. *Theoretical Computer Science*, 300(1–3):235–258, 2003.

[DL06] S. Demri and D. Lugiez. Presburger Modal Logic is PSPACE-complete. In *IJCAR'06*, volume 4130 of *Lecture Notes in Computer Science*, pages 541–556. Springer, 2006.

[Fin72] K. Fine. In so many possible worlds. *Notre Dame Journal of Formal Logic*, 13(4):516–520, 1972.

[Fit83] M. Fitting. *Proof methods for modal and intuitionistic logics*. D. Reidel Publishing Co., 1983.

[FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[FR74] M. Fischer and M. Rabin. Super-exponential complexity of Presburger arithmetic. In *Complexity of Computation*, volume 7 of *SIAM-AMS proceedings*, pages 27–42. AMS, 1974.

[Gor99] R. Goré. Tableaux methods for modal and temporal logics. In *Handbook of Tableaux Methods*, pages 297–396. Kluwer, 1999.

[HB91] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *KR'91*, pages 335–346, 1991.

[Hem01] E. Hemaspaandra. The complexity of poor man's logic. *Journal of Logic and Computation*, 11(4):609–622, 2001.

[HST00] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic SHIQ. In *CADE'00*, volume 1831 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2000.

[Kaz04] Y. Kazakov. A polynomial translation from the two-variable guarded fragment with number restrictions to the guarded fragment. In *JELIA'04*, volume 3229 of *Lecture Notes in Computer Science*, pages 372–384. Springer, 2004.

[KSV02] O. Kupferman, U. Sattler, and M.Y. Vardi. The complexity of the graded μ-calculus. In *CADE'02*, volume 2392 of *Lecture Notes in Computer Science*, pages 423–437, 2002.

[Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, 1977.

[Mar03] M. Marx. XPath and Modal Logics of finite DAG's. In *TABLEAUX'03*, volume 2796 of *Lecture Notes in Artificial Intelligence*, pages 150–164. Springer, 2003.

[MP97] A. Montanari and A. Policriti. A set-theoretic approach to automated deduction in graded modal logics. In *IJCAI'97*, pages 196–201, 1997.

[MR03] F. Moller and A. Rabinovich. Counting on CTL*: on the expressive power of monadic path logic. *Information and Computation*, 184(1):147–159, 2003.

[OSH96] H.J. Ohlbach, R. Schmidt, and U. Hustadt. Translating graded modalities into predicate logics. In H. Wansing, editor, *Proof theory of modal logic*, pages 253–291. Kluwer, 1996.

[OTTR05] H. Ohsaki, J.M. Talbot, S. Tison, and Y. Roos. Monotone AC-tree automata. In *LPAR'05*, volume 3835 of *Lecture Notes in Artificial Intelligence*, pages 337–351. Springer, 2005.

[Pap81] Chr. Papadimitriou. On the complexity of integer programming. *Journal of the Association for Computing Machinery*, 28(4):765–768, 1981.

[PS04] E. Pacuit and S. Salame. Majority logic. In *KR'04*, pages 598–605. AAAI Press, 2004.

[Rac78] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978.

[Rey02] J.C. Reynolds. Separation logic: a logic for shared mutable data structures. In *LICS'02*, pages 55–74. IEEE, 2002.

[SP06] L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. In *LICS'06*, pages 231–240. IEEE, 2006.

[Spa93] E. Spaan. The complexity of propositional tense logics. In M. de Rijke, editor, *Diamonds and Defaults*, pages 287–309. Kluwer, 1993.

[SSM07] H. Seidl, Th. Schwentick, and A. Muscholl. Counting in trees. *Texts in Logic and Games*, 2:575–612, 2007.

[SSMH04] H. Seidl, Th. Schwentick, A. Muscholl, and P. Habermehl. Counting in trees for free. In *ICALP'04*, volume 3142 of *Lecture Notes in Computer Science*, pages 1136–1149, 2004.

[Tob01] S. Tobies. PSPACE reasoning for graded modal logics. *Journal of Logic and Computation*, 11:85–106, 2001.

[vdH92] W. van der Hoek. On the semantics of graded modalities. *Journal of Applied Non-Classical Logics*, 2(1):81–123, 1992.

[vdHdR95] W. van der Hoek and M. de Rijke. Counting objects. *Journal of Logic and Computation*, 5(3):325–345, 1995.

[vdHM91] W. van der Hoek and J.-J. Meyer. Graded modalities in epistemic logic. *Logique et Analyse*, 133–134:251–270, 1991.

[Wol83] P. Wolper. Temporal logic can be more expressive. *Information and Computation*, 56:72–99, 1983.

[ZL03] S. Dal Zilio and D. Lugiez. XML schema, tree logic and sheaves automata. In *RTA'03*, volume 2706 of *Lecture Notes in Computer Science*, pages 246–263. Springer, 2003.

[ZL06] S. Dal Zilio and D. Lugiez. XML schema, tree logic and sheaves automata. *Applicable Algebra in Engineering, Communication and Computing*, 17(5):337–377, 2006.

A Proof of Lemma 7

The proof basically restates the proof of [SSM07, Claim 7.3] in the context of EML with subsets of $\mathrm{cl}(n, \phi)$. Section A.1 recalls a result due to Papadimitriou [Pap81] about small solutions of constraint systems, see also [BT76]. Section A.2 explains how the Parikh image of a language defined as an intersection can be characterized from automata (Parikh image of context-free languages are semilinear). Section A.3 shows how to reduce the constraint systems in order to obtain exponential-size small solutions. This is the place where we essentially follow part of the proof of [SSM07, Claim 7.3]. Finally, Section A.4 explains how to build constraint systems from subsets of $\mathrm{cl}(n, \phi)$ and why it allows us to conclude the proof. This appendix completes the proof sketch provided in the body of the paper.

A.1 Constraint Systems

A constraint system $\mathcal{S}$ over the set of variables $\{x_1, \ldots, x_n\}$ is a Presburger formula built over $\{x_1, \ldots, x_n\}$ that is a Boolean combination of atomic constraints of the form $\sum_j a_j \times x_{i_j} = b$ with each $a_j \in \mathbb{Z}$ and $b \in \mathbb{N}$. A positive solution for $\mathcal{S}$ is an element $\overline{x} \in \mathbb{N}^n$ such that $\overline{x} \models \mathcal{S}$ in Presburger arithmetic. We base our analysis on Lemma 10 below, which follows from a result of Papadimitriou [Pap81].

Theorem 2 *[Pap81] Let $\mathcal{S}$ be a constraint system over $\{x_1, \ldots, x_n\}$ made of a single conjunction of atomic constraints. $\mathcal{S}$ has a positive solution iff there is a positive solution such that all the coefficients are bounded by $n \times (ma)^{2m+1}$ where a is the maximal absolute value among the constants occurring in $\mathcal{S}$ and m is the number of atomic constraints in $\mathcal{S}$.*

We have the following corollary.

Lemma 10 *Let $\mathcal{S}$ be a constraint system over $\{x_1, \ldots, x_n\}$. $\mathcal{S}$ has a positive solution iff there is a positive solution such that all the coefficients are bounded by $(n + 2 \times m) \times (2 \times m + (a + 1))^{4m+1}$ where a is the maximal absolute value among the constants occurring in $\mathcal{S}$ and m is the number of atomic constraints in $\mathcal{S}$.*

Proof. The system $\mathcal{S}$ can be transformed in disjunctive normal form providing a disjunction of conjunctions with conjuncts of the form either $\sum_j a_j \times x_{i_j} = b$ or $\neg(\sum_j a_j \times x_{i_j} = b)$. Each disjunct has at most m atomic constraints. Since $\neg(\sum_j a_j \times x_{i_j} = b)$ can be rewritten as $(\sum_j a_j \times x_{i_j} - y = b+1) \vee (\sum_j a_j \times x_{i_j} + y' = b - 1)$, we get a disjunction of conjunctions as those in Theorem 2. Here y and y' are new variables. However, the process of replacing negated atomic constraints possibly multiplies by 2 the number of atomic constraints, add at most $2 \times m$ variables, and add one to the maximal absolute value of each conjunction, whence the bound. $\square$

A.2 Product automata over an enriched alphabet

Suppose that the formulae below

$$\mathcal{A}_1(\phi_1^1, \ldots, \phi_{n_1}^1), \ldots, \mathcal{A}_l(\phi_1^l, \ldots, \phi_{n_l}^l), \neg\mathcal{A}'_1(\psi_1^1, \ldots, \psi_{m_1}^1), \ldots, \neg\mathcal{A}'_{l'}(\psi_1^{l'}, \ldots, \psi_{m_{l'}}^{l'})$$

are exactly the automata-based formulae or their negation that occurs in some set $X \subseteq cl(n, \phi)$. Let $\{\psi_1, \ldots, \psi_P\}$ be the subformulae in $sub(\phi)$ that occur as arguments in the above formulae. Necessarily, $\{\psi_1, \ldots, \psi_P\} \subseteq \mathrm{cl}(n+1, \phi)$.

First, let us build automata $\mathcal{B}_1, \ldots, \mathcal{B}_l$ over the alphabet $\Sigma = \{Y_1, \ldots, Y_{nb(n+1)}\}$ where $Y_1, \ldots, Y_{nb(n+1)}$ are the only $(n+1)$-locally consistent sets ($nb(n+1)$ is exponential in $|\phi|$). For every $i \in \{1, \ldots, l\}$, the automata $\mathcal{B}_i$ and $\mathcal{A}_i$ have the same sets of states, initial states and final states and $q \xrightarrow{Y} q'$ in $\mathcal{B}_i$ iff $q \xrightarrow{\psi} q'$ in $\mathcal{A}_i$ for some $\psi \in Y$.

Similarly, we build the automata $\mathcal{B}'_1, \ldots, \mathcal{B}'_{l'}$ from the automata $\mathcal{A}'_1, \ldots, \mathcal{A}'_{l'}$. We write $\mathcal{B}'^{\neg}_1, \ldots, \mathcal{B}'^{\neg}_{l'}$ to denote the complement automata obtained, for instance, by the standard powerset construction.

Hence, we can define a product automaton $\mathcal{B}$ obtained by synchronizing $\mathcal{B}'_1, \ldots, \mathcal{B}_l, \mathcal{B}'_1, \ldots, \mathcal{B}'_{l'}$ over the alphabet Σ satisfying the conditions below:

- The cardinal of the alphabet Σ is bounded by $2^{|\phi|}$ and the set of states Q' has cardinal bounded by $2^{p(|\phi|)}$ for some polynomial $p(\cdot)$.
- For every word $w = Y_{j_1} \cdots Y_{j_\alpha} \in \Sigma^*$, $w \in \mathrm{L}(\mathcal{B})$ iff the conditions below hold true.
  - · For every $i \in \{1, \ldots, l\}$, there are $\psi_1 \in Y_{j_1}, \ldots, \psi_\alpha \in Y_{j_\alpha}$ such that $\psi_1 \cdots \psi_\alpha \in \mathrm{L}(\mathcal{A}_i)$.
  - · For $i \in \{1, \ldots, l'\}$, there are no $\psi_1 \in Y_{j_1}, \ldots, \psi_\alpha \in Y_{j_\alpha}$ such that $\psi_1 \cdots \psi_\alpha \in \mathrm{L}(\mathcal{A}'_i)$.

The Parikh image of $\mathrm{L}(\mathcal{B})$, subset of $\mathbb{N}^{|\Sigma|}$ and denoted by $\pi(\mathrm{L}(\mathcal{B}))$, is a finite union $\mathrm{L}_1 \cup \cdots \cup \mathrm{L}_m$ of linear sets $\mathrm{L}_i = \{\sigma_0 + \sum_{j=1}^{h} y_j \sigma_j : y_j \geqslant 0\}$ where each σ_j is in $\{0, \ldots, |Q'|\}^{|\Sigma|}$ by [SSMH04, Theorem 1]. Consequently, h is bounded by $(|Q'|+1)^{|\Sigma|} \leqslant 2^{p(|\phi|) \times 2^{|\phi|}}$. By Theorem 2 (see also Lemma 10), if the constraint system

$$\begin{pmatrix} z_1 \\ z_2 \\ \dots \\ z_{|\Sigma|} \end{pmatrix} = \sigma_0 + \sum_{j=1}^{h} y_j \sigma_j$$

made of $|\Sigma| + h$ variables and $|\Sigma|$ atomic constraints has solutions, then it admits a (small) solution whose values are at most doubly exponential in $|\phi|$. However, in order to guess such values in polynomial space, we need to improve this double exponential bound to a simple exponential bound in $|\phi|$.

A.3 Reducing the number of variables

Let us pose $N = nb(n+1)$. We write $H : \mathbb{N}^N \to \mathbb{N}^P$ to denote the homomorphism such that

$$H(\begin{pmatrix} n_1 \\ n_2 \\ \dots \\ n_N \end{pmatrix})(i) \stackrel{\text{def}}{=} \sum_{\psi_i \in Y_j} n_j.$$

This map can be naturally extended to sets of tuples. So if the tuple $\begin{pmatrix} n_1 \\ n_2 \\ \dots \\ n_N \end{pmatrix}$ is the Parikh image of the children of a node with respect to the sets of formulae $Y_1, \dots, Y_N$, the tuple $H(\begin{pmatrix} n_1 \\ n_2 \\ \dots \\ n_N \end{pmatrix})$ is the Parikh image with respect to formulae $\psi_1, \dots, \psi_P$. For instance, the number of children satisfying ψ_3 is denoted by $H(\begin{pmatrix} n_1 \\ n_2 \\ \dots \\ n_N \end{pmatrix})(3)$.

By definition of π and $\mathcal{B}$, for every $v \in \mathbb{N}^P$, $v \in H(\pi(\mathrm{L}(\mathcal{B}))$ iff there is $w \in \mathrm{L}(\mathcal{B})$ such that for $j \in \{1, \ldots, P\}$, the cardinal of $\{w(k) : k < |w|, \psi_j \in w(k)\}$ is $v(j)$. Consequently, $v \in H(\pi(\mathrm{L}(\mathcal{B})))$ iff $v \in H(\mathrm{L}_i)$ for some $i \in \{1, \ldots, m\}$. However, $H(\mathrm{L}_i)$ is precisely equal to $\{H(\sigma_0)+\sum_{j=1}^{h} y_j H(\sigma_j) : y_j \geqslant 0\}$. Observe that each $H(\sigma_j)$ has dimension $P \leqslant |\phi|$ and each coefficient is bounded by $N \times 2^{p(|\phi|)\times|\phi|}$. Consequently, the cardinal of the set $\{H(\sigma_j) : 1 \leqslant j \leqslant h\}$ is bounded by $(N \times 2^{p(|\phi|)\times|\phi|} + 1)^{|\phi|}$, which is bounded by $\alpha \leqslant 2^{p_1(|\phi|)}$ for some polynomial $p_1(\cdot)$. Roughly speaking, this entails that there are many images $H(\sigma_j)$ and $H(\sigma_k)$ that are equal with $\sigma_j \neq \sigma_k$. Let $h_1, \ldots, h_\alpha$ be the elements of the above mentioned set. So, (EQUIV) the projections over the components $z_1, \ldots, z_P$ of the solutions of the system

$$(\star) \begin{pmatrix} z_1 \\ z_2 \\ \cdots \\ z_P \end{pmatrix} = H(\sigma_0) + \sum_{j=1}^{\alpha} y_j h_j$$

are exactly the projections over the components $z_1, \ldots, z_P$ of the solutions of the system

$$(\star\star) \begin{pmatrix} z_1 \\ z_2 \\ \cdots \\ z_P \end{pmatrix} = H(\sigma_0) + \sum_{j=1}^{h} y'_j\ H(\sigma_j)$$

Typically, from $(\star\star)$ to $(\star)$, each y_j can be defined as a sum of variables y'_k (with $H(\sigma_k) = h_j$). We recall that a solution of $(\star)$ is a tuple in $\mathbb{N}^{P+\alpha}$ whereas a solution of $(\star\star)$ is a tuple in $\mathbb{N}^{P+h}$. We assume that the P first elements of the tuples correspond to values for $z_1, \ldots, z_P$. We write $\mathcal{S}^\star$ [resp. $\mathcal{S}^{\star\star}$] to denote the disjunction of all the systems of the form $(\star)$ [resp. $(\star\star)$]. There is indeed one disjunct by element from the union $\mathrm{L}_1 \cup \cdots \cup \mathrm{L}_m$. Observe that each disjunct of $\mathcal{S}^\star$ has a polynomial amount of equations, an exponential amount of variables and coefficients are at most exponential in $|\phi|$. The above-mentioned equivalence (EQUIV) can be extended as follows (the proof is by an easy verification).

Lemma 11 *Let $\mathcal{S}'$ is a contraint system with no variable of the form either y_j or y'_j. The two sets below are identical (obtained by projection over the values related to the variables $z_1, \ldots, z_P$):*

(1) $\{v_P \in \mathbb{N}^P : \langle v_P, v\rangle$ is a solution of $\mathcal{S}^\star \wedge \mathcal{S}'\}$.
(2) $\{v_P \in \mathbb{N}^P : \langle v_P, v'\rangle$ is a solution of $\mathcal{S}^{\star\star} \wedge \mathcal{S}'\}$.

Let ϕ be an EML formula and X be a n-locally consistent set. We shall build the system $\mathcal{S}_X$ that contains the variables $x_1, \ldots, x_{nb(n+1)}$. Each x_i is the number of occurrences of "type" Y_i among the children of a node of type X. To each formula $\psi \in \mathrm{cl}(n+1, \phi)$ that is not a periodicity constraint of the form $t \equiv_K c$, we associate the term $t_\psi = \sum_{i, \psi \in Y_i} x_i$. Remember that we have assumed without any loss of generality that formulae of the form $t \equiv_K c$ belongs to the closure sets but are not atomic formulae occurring in ϕ. We shall define $\mathcal{S}_X$ as a conjunction of the constraints below:

- $\sum_{Y_i \text{ is not satisfiable}} x_i = 0$,
- if $\sum_i a_i \sharp \phi_i = b \in X$, then we add $\Sigma_i a_i t_{\phi_i} = b$,
- if $\sum_i a_i \sharp \phi_i < b \in X$, then we add $\Sigma_i a_i t_{\phi_i} + y = b - 1$ where y is a new variable,
- if $\sum_i a_i \sharp \phi_i > b \in X$, then we add $\Sigma_i a_i t_{\phi_i} - y = b + 1$ where y is a new variable,
- if $\sum_i a_i \sharp \phi_i \equiv_K c \in X$, then we add $\sum_i a_i t_{\phi_i} - Ky = c$ where y is a new variable,
- if $\mathcal{A}_1(\phi^1_1, \ldots, \phi^1_{n_1}), \ldots, \mathcal{A}_l(\phi^l_1, \ldots, \phi^l_{n_l})$ and $\neg\mathcal{A}'_1(\psi^1_1, \ldots, \psi^1_{m_1}), \ldots,$ $\neg\mathcal{A}'_{l'}(\psi^{l'}_1, \ldots, \psi^{l'}_{m_{l'}})$ are all the automaton-based formulae in X, then we add the system $\mathcal{S}^\star$ from Section A.3 where each variable z_i is replaced by t_{ψ_i}.

By construction $\mathcal{S}_X$ is equivalent to a disjunction of the form $\bigvee \mathcal{S}_i$ with an exponential amount of disjuncts for which each $\mathcal{S}_i$ has a polynomial amount of equations, an exponential amount of variables and coefficients are at most exponential in $|\phi|$. Hence, by Lemmas 10 and 11, if $\mathcal{S}_X$ has solutions, then $\mathcal{S}_X$ has solutions with values bounded by some M exponential in $|\phi|$. We write M to denote the maximal value amongst all the values obtained for the different depths n between 0 and $|\phi|$.

The proof of Lemma 7 is then a simple consequence of Lemma 12 below.

Lemma 12 *Let ϕ be a* EML *formula, $d \in \{0, \ldots, |\phi|\}$ and X be a d-locally consistent set of formulae. Then, X is* EML *satisfiable iff $\mathcal{S}_X$ has a positive solution.*

Proof. It is easy to check that if X is EML satisfiable, then $\mathcal{S}_X$ has a positive solution. The converse requires a bit more care. Assume that $\mathcal{S}_X$ has a positive solution whose projection over $\{x_1, \ldots, x_{nb(n+1)}\}$ is $\langle n_1, \ldots, n_{nb(d+1)} \rangle$. We build the EML model $\mathcal{M} = \langle T, R, (<_s)_{s \in T}, l \rangle$ as follows. For each $n_i \neq 0$, the set Y_i is satisfiable since $\Sigma_{Y_i \text{ is not satisfiable}}\ n_i = 0$. Hence, there exist a EML model $\mathcal{M}_i = \langle T_i, R_i, (<^i_s)_{s \in T_i}, l_i \rangle$ and $s_i \in T_i$ such that $\mathcal{M}_i, s_i \models Y_i$. $\mathcal{M}$ is built from n_1 copies of $\mathcal{M}_1$, $\ldots$, $n_{nb(d+1)}$ copies of $\mathcal{M}_{nb(d+1)}$ by adding R-

transitions between the root s of T (a new state) and all the s_i's of all copies. Moreover $l(s) = \mathrm{AT} \cap X$. Because $\langle n_1, \ldots, n_{nb(d+1)} \rangle$ is a positive solution of $\mathcal{S}_X$, there is a way to order the children of s so that the constraints of the form either $\mathcal{A}(\psi_1, \ldots, \psi_l)$ or $\neg\mathcal{A}(\psi_1, \ldots, \psi_l)$ in X are also satisfied (this comes by construction of $\mathcal{S}^\star$). Because X is a d-locally consistent set, one can easily show that $\mathcal{M}, s \models X$. This is shown by structural induction and the base case for atomic formulae hold true because $\langle n_1, \ldots, n_{nb(d+1)} \rangle$ is the projection of a positive solution for $\mathcal{S}_X$.