



HAL
open science

Presburger Modal Logic Is PSPACE-Complete

Stéphane Demri, Denis Lugiez

► **To cite this version:**

Stéphane Demri, Denis Lugiez. Presburger Modal Logic Is PSPACE-Complete. IJCAR 2006 - Automated Reasoning, Third International Joint Conference, Ulrich Furbach; Natarajan Shankar, Aug 2006, Seattle, United States. pp.541-556, 10.1007/11814771_44 . hal-03201487

HAL Id: hal-03201487

<https://hal.science/hal-03201487>

Submitted on 18 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Presburger Modal Logic is PSPACE-complete

Stéphane Demri¹ and Denis Lugiez²

¹ Laboratoire Spécification et Vérification
CNRS & INRIA Futurs projet SECSI & ENS Cachan
demri@lsv.ens-cachan.fr

² Laboratoire d'Informatique Fondamentale, Marseille
UMR 6166 CNRS-Université de Provence
lugiez@lif.univ-mrs.fr

Abstract. We introduce a Presburger modal logic PML 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 PML 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 show that satisfiability for Sheaves Logic SL is PSPACE-complete, improving significantly its best known upper bound.

1 Introduction

Logics for XML documents. In order to query XML documents with Presburger and/or regular constraints, logical and automata-based formalisms have been recently introduced [ZL06,SSMH04,BT05] 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 fixed-point operators, Presburger and regularity constraints is shown EXPTIME-complete in [SSMH04], improving results for description logics with qualified number restrictions [CG05]. At the same period, the sister logic SL (“Sheaves Logic”) is shown decidable in [ZL03]. The more expressive logic GDL is however shown undecidable in [ZL06] since GDL can express properties about disjoint sequences of children. 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.

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,Tob00] or description logics [HST00,CG05]) and with regularity constraints as in the logical formalisms from [Wol83,ZL03,SSMH04] but with a satisfiability problem in polynomial space which would refine decidability and complexity results from [Tob00,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 Presburger and regularity constraints. With such requirements, fixed-point 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. It is worth observing that as far as memory resources 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.

Our contribution. We consider a Presburger modal logic PML with full Presburger constraints on the number of children and with regularity constraints. It is a minor variant of either the fixed-point free fragment of [SSMH04] or the Sheaves Logic SL [ZL06]. The exact relationships between PML, SL and the logic from [SSMH04] are provided in the paper. We show that the satisfiability problem is PSPACE-complete (only the binary representation for integers is used). The complexity upper bound is proved with a Ladner-like algorithm, see the original one in [Lad77] and strongly related tableaux methods in [Gor99]. This result generalizes what is known about graded modal logic [Fin72,BC85,Tob00] and apart from its larger scope, we believe our proof is also much more transparent. Even though some of the bounds used in our algorithm are obtained by a careful analysis of proofs from [SSMH04], our algorithm can be viewed as the optimal composition between an algorithm that transforms a PML formula into a Presburger tree automata and an algorithm that tests emptiness for these peculiar Presburger tree automata. This provides a new and non-trivial PSPACE complexity upper bound that is not a direct consequence 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 $\text{APSPACE} = \text{EXPTIME}$, it is unlikely that the composition can be done in PSPACE. Additionally, our algorithm substantially refines results from [ZL03,SSMH04]. Indeed, as by-products of the PSPACE-completeness of PML, we show that SL satisfiability [ZL06] is PSPACE-complete, the fixed-point free fragment of the main logic from [SSMH04] is also PSPACE-complete and the logic PDL_{tree} from [ABD⁺05] is undecidable when extended with Presburger constraints. 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 restricted to the only truth constants as 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].

Omitted proofs can be found in the report [DL06].

2 Presburger Modal Logic

Given countably infinite sets $\text{AP} = \{p_1, p_2, \dots\}$ of propositional variables and $\Sigma = \{\mathbf{R}_1, \mathbf{R}_2, \dots\}$ of symbol relations, we define the set of formulae and terms

inductively as follows: $\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid t \sim b \mid t \equiv_k c \mid \mathcal{A}(\mathbf{R}, \phi_1, \dots, \phi_n)$ and $t ::= a \times \sharp^{\mathbf{R}}\phi \mid t + a \times \sharp^{\mathbf{R}}\phi$, where $p \in \text{AP}$, $\mathbf{R} \in \Sigma$, $b, k, c \in \mathbb{N}$, $a \in \mathbb{Z}$, $\sim \in \{<, >, =\}$ and \mathcal{A} is a nondeterministic finite-state automaton over an n -letter alphabet $\Sigma_{\mathcal{A}}$ in which the letters are linearly ordered $\Sigma_{\mathcal{A}} = \mathbf{a}_1, \dots, \mathbf{a}_n$. The language accepted by \mathcal{A} is denoted by $L(\mathcal{A})$. We write $|\phi|$ to denote the size of the formula ϕ with some reasonably succinct encoding and $\text{md}(\phi)$ to denote the “modal degree” of ϕ defined as the maximal number of imbrications of the symbol \sharp in ϕ .

A term of the form $a_1 \times \sharp^{\mathbf{R}_1}\phi_1 + \dots + a_m \times \sharp^{\mathbf{R}_m}\phi_m$ is abbreviated by $\Sigma_i a_i \sharp^{\mathbf{R}_i}\phi_i$. Because of the presence of Boolean operators and quantifier-elimination for Presburger arithmetic, 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 PML is a structure $\mathcal{M} = \langle T, (R_{\mathbf{R}})_{\mathbf{R} \in \Sigma}, (\prec_{nd}^{\mathbf{R}})_{nd \in T}, l \rangle$ where

- T is the set of nodes (possibly infinite),
- $(R_{\mathbf{R}})_{\mathbf{R} \in \Sigma}$ is a family of binary relations in $T \times T$ such that for all $\mathbf{R} \in \Sigma$ and $nd \in T$, the set $\{nd' \in T : \langle nd, nd' \rangle \in R_{\mathbf{R}}\}$ is finite (finite-branching),
- each relation $\prec_{nd}^{\mathbf{R}}$ is a total ordering on the $R_{\mathbf{R}}$ -successors of nd ,
- $l : T \rightarrow 2^{\text{AP}}$ is the valuation function.

In the rest of the paper, we write $R_{\mathbf{R}}(nd) = nd_1 < \dots < nd_{\alpha}$ to mean that $R_{\mathbf{R}}(nd) \stackrel{\text{def}}{=} \{nd' \in T : \langle nd, nd' \rangle \in R_{\mathbf{R}}\} = \{nd_1, \dots, nd_{\alpha}\}$, and $nd_1 \prec_{nd}^{\mathbf{R}} \dots \prec_{nd}^{\mathbf{R}} nd_{\alpha}$. Given a finite-branching binary relation $R \subseteq T \times T$, we write $R^{\sharp}(q)$ to denote the cardinal of the set $\{q' \in T : \langle q, q' \rangle \in R\}$. The satisfaction relation is inductively defined below where \mathcal{M} is a model for PML and $nd \in T$:

- $\mathcal{M}, nd \models p$ iff $p \in l(nd)$; $\mathcal{M}, nd \models \neg\phi$ iff not $\mathcal{M}, nd \models \phi$,
- $\mathcal{M}, nd \models \phi_1 \wedge \phi_2$ iff $\mathcal{M}, nd \models \phi_1$ and $\mathcal{M}, nd \models \phi_2$,
- $\mathcal{M}, nd \models \Sigma_i a_i \sharp^{\mathbf{R}_i}\phi_i \sim b$ iff $\Sigma_i a_i R_{\mathbf{R}_i, \phi_i}^{\sharp}(nd) \sim b$ with $R_{\mathbf{R}_i, \phi_i} = \{\langle nd', nd'' \rangle \in T \times T : \langle nd', nd'' \rangle \in R_{\mathbf{R}_i}, \text{ and } \mathcal{M}, nd'' \models \phi_i\}$,
- $\mathcal{M}, nd \models \Sigma_i a_i \sharp^{\mathbf{R}_i}\phi_i \equiv_k c$ iff there is $n \in \mathbb{N}$ such that $\Sigma_i a_i R_{\mathbf{R}_i, \phi_i}^{\sharp}(nd) = nk + c$,
- $\mathcal{M}, nd \models \mathcal{A}(\mathbf{R}, \phi_1, \dots, \phi_n)$ iff there is $\mathbf{a}_{i_1} \dots \mathbf{a}_{i_{\alpha}} \in L(\mathcal{A})$ such that $R_{\mathbf{R}}(nd) = nd_1 < \dots < nd_{\alpha}$ and for every $j \in \{1, \dots, \alpha\}$, $\mathcal{M}, nd_j \models \phi_{i_j}$.

The automata in PML are used exactly as those defining temporal operators in extended temporal logic [Wol83]. The modal operator \diamond (see e.g. [BdRV01]) is defined by $\diamond\phi \approx \sharp^{\mathbf{R}}\phi \geq 1$ (and dually $\square\phi \approx \sharp^{\mathbf{R}}\neg\phi = 0$) whereas formula $\diamond_{\geq n}\phi$ from graded modal logic is defined by $\diamond_{\geq n}\phi \approx \sharp^{\mathbf{R}}\phi \geq n$. A basic example of what PML 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 $\sharp^{\mathbf{R}}p - 2\sharp^{\mathbf{R}}q = 0$.

A formula ϕ of PML is satisfiable whenever there exist a model $\mathcal{M} = \langle T, (R_{\mathbf{R}})_{\mathbf{R} \in \Sigma}, (\prec_{nd}^{\mathbf{R}})_{nd \in T}, l \rangle$ and $nd \in T$ such that $\mathcal{M}, nd \models \phi$. Even though PML 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.

Lemma 1. *For every PML formula ϕ , ϕ is satisfiable iff ϕ is satisfiable in a model \mathcal{M} such that for all relation symbols R occurring in ϕ and $nd \in T$, the restriction of $\langle T, R_R \rangle$ to $R_R^*(nd)$ is a tree.*

Proof. Suppose that ϕ has a PML model $\mathcal{M} = \langle T, (R_R)_{R \in \Sigma}, (\prec_{nd}^R)_{nd \in T}, l \rangle$ and state $nd \in T$ such that $\mathcal{M}, nd \models \phi$. 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 ordering. The model $\mathcal{M}' = \langle T', (S_R)_{R \in \Sigma}, (\prec_{nd}^R)_{nd \in T'}, l' \rangle$ is defined as follows:

- T' is the set of finite non-empty sequences of the form $nd R_1 nd_1 \dots R_k nd_k$,
- $(nd R_1 nd_1 \dots R_n nd_n) S_R (nd R_1 nd_1 \dots R_n nd_n R_{n+1} nd_{n+1})$ iff $\langle nd_n, nd_{n+1} \rangle \in R_R$ and $R = R_{n+1}$,
- $l'(nd R_1 nd_1 \dots R_n nd_n) = l(nd_n)$ for every $nd R_1 nd_1 \dots R_n nd_n \in T'$,
- each ordering \prec_{nd}^R is the one induced by \prec_{nd}^R by considering the last element nd of the sequence nd' .

One can show that for every $nd R_1 nd_1 \dots R_n nd_n \in T'$ and PML formula ψ , $\mathcal{M}', nd R_1 nd_1 \dots R_n nd_n \models \psi$ iff $\mathcal{M}, nd_n \models \psi$. In particular $\mathcal{M}, \langle nd \rangle \models \phi$.

Since the formula tree of every formula is finite and Presburger or regular constraints only speak about direct successors, we can establish the result below.

Lemma 2. *For every PML formula ϕ , ϕ is satisfiable iff ϕ is satisfiable in a model \mathcal{M} such that T is finite and for all relation symbols R occurring in ϕ and $nd \in T$, the restriction of $\langle T, R_R \rangle$ to $R_R^*(nd)$ is a tree.*

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

Lemma 3. *For every PML formula ϕ , one can compute in logspace a PML formula ϕ' with a unique relation symbol R such that ϕ is satisfiable on finite trees iff ϕ' is satisfiable on finite trees.*

In the rest of the paper, we assume that Σ is a singleton set $\{R\}$, we write $\mathcal{A}(\phi_1, \dots, \phi_n)$ instead of $\mathcal{A}(R, \phi_1, \dots, \phi_n)$ and $\# \phi_i$ instead of $\#^R \phi_i$. Models are written as tuples $\langle T, R, (\prec_{nd})_{nd \in T}, l \rangle$.

3 An Optimal Algorithm for PML Satisfiability

In this section, we show that PML satisfiability can be solved in polynomial space by using a Ladner-like algorithm [Lad77]. The original algorithm [Lad77] is designed for the modal logics K and S4, see a tense extension in [Spa93].

3.1 Consistent Sets of Formulae

We define below a notion of closure à la Fisher-Ladner [FL79] for finite sets of formulae. Intuitively, the closure $\text{cl}(X)$ 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. $\text{cl}(X)$ is the smallest set of formulae such that

- $X \subseteq \text{cl}(X)$, $\text{cl}(X)$ is closed under subformulae,
- if $\psi \in \text{cl}(X)$, then $\neg\psi \in \text{cl}(X)$ (we identify $\neg\neg\psi$ with ψ),
- if $t \sim b \in \text{cl}(X)$, then $t \sim' b \in \text{cl}(X)$ for every $\sim' \in \{<, >, =\}$,
- let K be the lcm of all the constants k occurring in subformulae of the form $t \equiv_k c$. Without any loss of generality, we can assume that \equiv_K does not occur in ϕ . If $t \equiv_k c \in \text{cl}(X)$, then $t \equiv_K c' \in \text{cl}(X)$ for every $c' \in \{0, \dots, K-1\}$.

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

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

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

We are only interested in subsets of $\text{cl}(n, \phi)$ whose conjunction of its elements is PML satisfiable. A necessary condition to be satisfiable is to be consistent locally, i.e. at the propositional level and at the level of Presburger 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 below) in the main algorithm. This is analogous to the requirement to check maximal consistency at the propositional level but not PML satisfiability at once.

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

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

Lemma 4. *Let ϕ be a PML 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) $\text{cl}(|\phi|, \phi) = \emptyset$.*

Let X be an n -consistent subset of $\text{cl}(n, \phi)$. The set X is encoded as follows. To each subformula ψ in $\text{cl}(n, \phi)$ that is neither a periodicity constraint of the form $t \equiv_K c$, nor a constraint of the form $t \sim b$, we associate a bit encoding whether ψ belongs to X . To each formula of the form $t \sim b$ in $\text{cl}(n, \phi)$, we associate a value \sim' in $\{<, >, =\}$ encoding the fact that $t \sim' b$ belongs to X . Analogously, to each formula of the form $t \equiv_K c$ in $\text{cl}(n, \phi)$, we associate a value c' in $\{0, \dots, K-1\}$ encoding the fact that $t \equiv_K c'$ belongs to X . This unique c' requires $\mathcal{O}(|\phi|)$ bits to be encoded. Hence, each n -consistent subset of $\text{cl}(n, \phi)$ can be encoded with $\mathcal{O}(|\phi|^2)$ bits.

3.2 Constraint Systems

In this section, we explain how a consistent set induces solutions from numerical constraint systems based on the Presburger and regularity constraints. A constraint system \mathcal{S} over the set of variables $\{x_1, \dots, x_n\}$ is a Presburger formula built over $\{x_1, \dots, x_n\}$ 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 $\bar{x} \in \mathbb{N}^n$ such that $\bar{x} \models \mathcal{S}$ in Presburger arithmetic. We base our analysis on the following lemma, which follows from a result of Papadimitriou [Pap81].

Lemma 5. *Let \mathcal{S} be a constraint system over $\{x_1, \dots, x_n\}$. \mathcal{S} has a positive solution iff there is a positive solution s.t. 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} .*

Given a PML formula ϕ and an n -locally consistent set X , we associate a constraint system \mathcal{S}_X as follows. The number of $(n+1)$ -locally consistent sets is bounded by $nb(n+1) \stackrel{\text{def}}{=} 2^{p_1(|\phi|)}$ for some polynomial $p_1(\cdot)$ and we denote below such sets by $Y_1, \dots, Y_{nb(n+1)}$. The system \mathcal{S}_X contains the variables $x_1, \dots, x_{nb(n+1)}$. To each formula $\psi \in \text{cl}(n+1, \phi)$ 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 wlog 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:

- $\sum Y_i$ is not satisfiable $x_i = 0$,
- if $\sum_i a_i \# \phi_i = b \in X$, then we add $\sum_i a_i t_{\phi_i} = b$,
- if $\sum_i a_i \# \phi_i < b \in X$, then we add $\sum_i a_i t_{\phi_i} + y = b - 1$ where y is new,
- if $\sum_i a_i \# \phi_i > b \in X$, then we add $\sum_i a_i t_{\phi_i} - y = b + 1$ where y is new,
- if $\sum_i a_i \# \phi_i \equiv_K c \in X$, then we add $\sum_i a_i t_{\phi_i} - Ky = c$ where y is new,
- if $\mathcal{A}_1(\phi_1^1, \dots, \phi_{n_1}^1), \dots, \mathcal{A}_l(\phi_1^l, \dots, \phi_{n_l}^l)$ and $\neg \mathcal{A}'_1(\psi_1^1, \dots, \psi_{m_1}^1), \dots, \neg \mathcal{A}'_{l'}(\psi_1^{l'}, \dots, \psi_{m_{l'}}^{l'})$ are all the automaton-based formulae in X , then we add

the Presburger formula of the form

$$\bigvee \exists \dots y_k^{i,j} \dots z_k^{i,j} \dots \left(\bigwedge_{i,j} (t_{\phi_i^j} = a_0^{i,j} + \sum_k y_k^{i,j} a_k^{i,j}) \wedge \left(\bigwedge_{i,j} t_{\psi_i^j} = b_0^{i,j} + \sum_k z_k^{i,j} b_k^{i,j} \right) \right)$$

such that each disjunct has at most $2^{p_1(|\phi|)} + 2^{2 \times |\phi|^2}$ variables and the absolute values of the coefficients $a_k^{i,j}$ and $b_k^{i,j}$ are bounded by $2^{2 \times |\phi|}$.

The positive solutions form the Parikh image of the language $L(\mathcal{A}_1) \cap \dots \cap L(\mathcal{A}_{n_1}) \cap -L(\mathcal{A}'_1) \cap \dots \cap -L(\mathcal{A}'_{m_1})$ over the alphabet $Y_1, \dots, Y_{nb(n+1)}$. A transition $q \xrightarrow{a_i} q'$ in $\mathcal{A}_1(\phi_1^1, \dots, \phi_{n_1}^1)$ is read as a concise representation for the transitions of the form $q \xrightarrow{Y} q'$ with $\phi_i^1 \in Y$. The existence of such a formula is a consequence of the proof of [SSMH04, Theorem 3] and the proof of [SSMH04, Theorem 6]. Indeed, computing the minimal and deterministic automaton for the product of $\mathcal{A}_1, \dots, \mathcal{A}_l, \mathcal{A}'_1, \dots, \mathcal{A}'_{l'}$ over the alphabet $\{Y_1, \dots, Y_{nb(n+1)}\}$ produces a constraint system of dimension $nb(n+1)$ with doubly exponential number of variables and coefficients bounded by an exponential value in $|\phi|$ [VSS05]. However, the constraints induced by the automaton-based formulae involve subformulae of ϕ and the latter system can be reduced to a system with only an exponential amount of variables using some combinatorial argument from [SSMH04, Theorem 6].

If we restrict ourselves to the fragment of PML with at most k regularity constraints per formulae and *deterministic* automata, say $\text{PML}_k^{\text{det}}$, then an automaton \mathcal{A} accepting $L(\mathcal{A}_1) \cap \dots \cap L(\mathcal{A}_{n_1}) \cap -L(\mathcal{A}'_1) \cap \dots \cap -L(\mathcal{A}'_{m_1})$ ($n_1 + m_1 \leq k$) can be built in polynomial-time in $|\phi|$ and then a Presburger formula for the Parikh image of $L(\mathcal{A})$ in linear-time in \mathcal{A} using [VSS05, Theorem 4]. In the case PML is studied in full generality instead of a specific $\text{PML}_k^{\text{det}}$, one needs to take advantage of this huge disjunction. The number of disjuncts may be (double) exponential but each disjunct will satisfy the good size properties to get PSPACE. The construction of \mathcal{S}_X is done in the way that allows to state the result below:

Lemma 6. *Let ϕ be a PML formula, $d \in \{0, \dots, |\phi|\}$ and X be a d -locally consistent of formulae. Then, X is PML satisfiable iff \mathcal{S}_X has a positive solution.*

Because of the Presburger formula introduced by the regular constraints, the system \mathcal{S}_X can be viewed as a disjunction of constraint systems \mathcal{S}' such that the number of variables in \mathcal{S}' is at most exponential in $|\phi|$, the number of atomic constraint in \mathcal{S}' is polynomial in $|\phi|$ and the maximal absolute value among the constants occurring in \mathcal{S}' is at most exponential in $|\phi|$. By Lemma 5, if \mathcal{S}' has a solution, then each value can be encoded in polynomial space in $|\phi|$. Consequently, if \mathcal{S}_X has a solution, then each value can be encoded in polynomial space in $|\phi|$. We write M to denote the maximal value for all the systems \mathcal{S}_X from the d -locally consistent set of ϕ with $d \leq |\phi|$. M is actually in $\mathcal{O}(2^{p_2(|\phi|)})$.

3.3 The Algorithm

We define the function SAT such that ϕ is PML satisfiable iff there is $X \subseteq \text{cl}(0, \phi)$ such that X is 0-locally consistent and $\text{SAT}(\phi, X, 0)$ has a computation that

returns **true**. The function $\text{SAT}(X, \phi, d)$ is defined in Fig. 1. The first argument X is intended to be a subset of $\text{cl}(d, \phi)$. 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.

Observe that we do not need to guess values for the auxiliary variables ($y, y_k^{i,j}, z_k^{i,j}$) but their existence is taken into account in the bound M and in the (final-checking) step. Similarly, if we guess a set Y_x that contains some unsatisfiable formula then $\text{SAT}(Y_x, \phi, d+1)$ has no accepting computation which also induces a non accepting computation for $\text{SAT}(X, \phi, d)$. However, the bound M takes into account this type of constraints of unsatisfiable formulae. Moreover, we check on the fly that the regularity constraints hold true. In particular, we visit on the fly the automata obtained by the subset construction in order to check negative regularity constraints.

function $\text{SAT}(X, \phi, d)$

(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 $\psi \in \text{cl}(d+1, \phi)$ that is not a periodicity constraint of the form $t \equiv_K c, C_\psi := 0$;

(initialization-states) for every $\mathcal{A}(\psi_1, \dots, \psi_\alpha) \in X, q_{\mathcal{A}(\psi_1, \dots, \psi_\alpha)} := q_0$ for some initial state q_0 of \mathcal{A} ;

(initialization-states-complement) for every $\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha) \in X, q_{\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha)} := I$ where I is the set of initial states of \mathcal{A} ;

(guess-number-children) guess nb in $\{0, \dots, nb(d+1) \times M\}$;

(guess-children-from-left-to-right) for $i = 1$ to nb do

1. guess $x \in \{1, \dots, nb(n+1)\}$;

2. if not $\text{SAT}(Y_x, \phi, d+1)$ then **abort**;

3. for every $\psi \in \text{cl}(d+1, \phi)$ that is not a periodicity constraint, if $\psi \in Y_x$, then $C_\psi := C_\psi + 1$;

4. for every $\mathcal{A}(\psi_1, \dots, \psi_\alpha) \in X$,

(a) guess a transition $q_{\mathcal{A}(\psi_1, \dots, \psi_\alpha)} \xrightarrow{a_i} q'$ in \mathcal{A} with $\Sigma_{\mathcal{A}} = \mathbf{a}_1, \dots, \mathbf{a}_\alpha$;

(b) if $\psi_i \in Y_x$, then $q_{\mathcal{A}(\psi_1, \dots, \psi_\alpha)} := q'$, otherwise **abort**;

5. for every $\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha) \in X$,

(a) guess a letter \mathbf{a}_i in $\Sigma_{\mathcal{A}} = \mathbf{a}_1, \dots, \mathbf{a}_\alpha$;

(b) if $\psi_i \in Y_x$, then $q_{\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha)} := \{q : \exists q' \in q_{\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha)}, q' \xrightarrow{\mathbf{a}_i} q\}$, otherwise **abort**;

(final-checking)

1. for every $\Sigma_i a_i \# \psi_i \sim b \in X$, if $\Sigma_i a_i \times C_{\psi_i} \sim b$ does not hold, then **abort**,

2. for every $\Sigma_i a_i \# \psi_i \equiv_k c \in X$, if $\Sigma_i a_i \times C_{\psi_i} \equiv_k c$ does not hold, then **abort**,

3. for every $\mathcal{A}(\psi_1, \dots, \psi_\alpha) \in X$, if $q_{\mathcal{A}(\psi_1, \dots, \psi_\alpha)}$ is not a final state of \mathcal{A} , then **abort**;

4. for every $\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha) \in X$, if $q_{\neg\mathcal{A}(\psi_1, \dots, \psi_\alpha)}$ contains a final state of \mathcal{A} , then **abort**;

(return-true) return **true**.

Fig. 1. Satisfiability algorithm

The algorithm described in SAT is a typical example of Ladner-like algorithm, see e.g. similar algorithms in [Lad77, Spa93, Dem03]. Indeed, it does not rely on any machinery such as automata or tableaux/sequent proof systems for checking satisfiability (but its correctness proof is indeed a kind of completeness proof). Moreover, the graph of recursive calls (here for SAT) induces a tree model for the argument formula. Since PML models are precisely trees, we get the PML model for free.

Observe also that comparing our algorithm from the one in [Tob00] for the poorer graded modal logic, our PSPACE upper bound is not based on any specific technique such as the trace technique and Presburger constraints are checked after guessing all the children.

3.4 Complexity Analysis and Correctness

Firstly, we prove that SAT requires only polynomial space.

Lemma 7. *For each 0-locally consistent set X , any computation of $\text{SAT}(\phi, X, 0)$ requires polynomial space in $|\phi|$.*

Then we prove the correctness of the algorithm.

Lemma 8. *A formula ϕ is PML satisfiable iff for some $X \subseteq \text{cl}(0, \phi)$, $\text{SAT}(X, \phi, 0)$ has a computation that returns **true**.*

By Lemmas 7, 8, Savitch's Theorem and PSPACE-hardness of K, we establish our main result.

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

Obviously, PML without regularity constraints is also in PSPACE.

4 Complexity results for similar logics

In this section, we compare PML with other logics with Presburger constraints. This is the opportunity to clarify the relationships between PML and logics from [SSMH04, ZL06, 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 PML where the formulae with Presburger constraints are of the form $\diamond_{\geq n}\phi$ and the like, are considered, see e.g. the early [Fin72, BC85, vdH92, vdHdR95]. Such logics have been extended to fit more specific motivations, giving epistemic logics [vdHM91] and description logics (see e.g. [CG05]) with graded modalities. It is only in [Tob00] 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 result about PML extends the main result from [Tob00]. Various extensions of known logics by adding graded modalities has been considered and undecidability is often obtained because the ability to count is often central to encode a grid, see e.g. [BP04]. However, the EXPTIME-completeness of graded μ -calculus [KSV02] remains a tour de force. Furthermore, there exist various attempts to encode concisely logics with counting into logics with no explicit counting mechanism, see e.g. [OSH96,MP97], but none of them implies a PSPACE upper bound. Modal-like logics with more expressive Presburger constraints on the number of children can be found in [SSMH04,ZL06] and this is the subject of the two next sections.

4.2 Sheaves Logic

Definition In this section, we recall the syntax and semantics of the Sheaves Logic SL that is shown decidable in [ZL03,ZL06] with a non-elementary algorithm. For the sake of uniformity, we adopt a presentation of SL models similar to the one for PML models whereas the mode of representation for regular languages and semilinear sets is the same as for PML. Indeed, the choice of representations for such objects may induce sometimes complexity gaps because of the different conciseness of the formalisms. Similarly, we allow Boolean operators at the level of element formulae (denoted by E) as done for document formulae (denoted by D). The element formulae are inductively defined as follows:

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

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 $\Sigma_{\mathcal{A}} = \mathbf{a}_1, \dots, \mathbf{a}_p$.
- $\phi(x_1, \dots, x_p)$ is a Boolean combination of Presburger formulae built over the variables x_1, \dots, x_p of the form either $t \sim b$ or $t \equiv_k c$ with $t = \sum a_i x_i$.

A model \mathcal{M} for SL is a structure $\mathcal{M} = \langle T, R, (\prec_{nd})_{nd \in T}, l \rangle$ where T is a finite set of states, $\langle T, R \rangle$ is a tree and each \prec_{nd} is a total ordering on $R(nd)$ and $l : T \rightarrow \text{TAGS} \cup \text{DATATYPES}$ is a labeling function such that for every $nd \in T$, if nd is a leaf of $\langle T, R \rangle$ then $l(nd) \in \text{DATATYPES}$ and for every $nd \in T$, if nd is not a leaf of $\langle T, R \rangle$ then $l(nd) \in \text{TAGS}$. The satisfaction relation is inductively defined below where \mathcal{M} is a model for SL and $nd \in T$ (we omit the clauses for Boolean operators):

- $\mathcal{M}, nd \models \delta$ iff $\delta = l(nd)$,
- $\mathcal{M}, nd \models \alpha[D_1 \wedge D_2]$ iff $\mathcal{M}, nd \models \alpha[D_1]$ and $\mathcal{M}, nd \models \alpha[D_2]$,

- $\mathcal{M}, nd \models \alpha[\neg D]$ iff $\alpha = l(nd)$ and not $\mathcal{M}, nd \models \alpha[D]$,
- $\mathcal{M}, nd \models \alpha[\mathbf{true}]$ iff $\alpha = l(nd)$,
- $\mathcal{M}, nd \models \alpha[\exists x_1, \dots, x_p : \phi(x_1, \dots, x_p) : x_1 E_1 \& \dots \& x_p E_p]$ iff $\alpha = l(nd)$, $R(nd) = nd_1 < \dots < nd_k$, and there exist i_1, \dots, i_k such that for every $j \in \{1, \dots, k\}$, $\mathcal{M}, nd_j \models E_{i_j}$ and $[x_1 \leftarrow n_1, \dots, x_p \leftarrow n_p] \models \phi(x_1, \dots, x_p)$ with $n_i = \text{card}(\{l \in \{1, \dots, k\} : i_l = i\})$,
- $\mathcal{M}, nd \models \alpha[\mathcal{A}(E_1, \dots, E_p)]$ iff $\alpha = l(nd)$, $R(nd) = nd_1 < \dots < nd_k$, and there is i_1, \dots, i_k such that for every $j \in \{1, \dots, k\}$, $\mathcal{M}, nd_j \models E_{i_j}$ and $\mathbf{a}_{i_1} \dots \mathbf{a}_{i_k} \in \mathbb{L}(\mathcal{A})$ with $\Sigma_{\mathcal{A}} = \{\mathbf{a}_1, \dots, \mathbf{a}_p\}$.

A major difference with the semantics of PML (see also [SSMH04]) is that for SL in Presburger constraints each child counts only once.

PSPACE-completeness Let ϕ be an SL formula with tags $\{\alpha_1, \dots, \alpha_n\}$ and datatypes $\{\delta_1, \dots, \delta_{n'}\}$. We define a formula ϕ' built over the propositional variables (plus others, see below) $VP = \{p_{\alpha_1}, \dots, p_{\alpha_n}, p_{\alpha_{new}}\} \cup \{p_{\delta_1}, \dots, p_{\delta_{n'}}, p_{\delta_{new}}\}$.

Given a PML φ , we write $\forall^n \varphi$ as an abbreviation for $\bigwedge_{i=0}^n \overbrace{\square \dots \square}^{i \text{ times}} \varphi$. 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 $D = \exists x_1 \dots x_p : \phi(x_1, \dots, x_p) : x_1 E_1 \& \dots \& x_p E_p$ in ϕ , we introduce new propositional variables p_D^1, \dots, p_D^p .

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

$$\begin{aligned} & \forall^{|\phi|} \bigvee_{p \in VP} (p \wedge \bigwedge_{q \in VP \setminus \{p\}} \neg q) \wedge \overbrace{\forall^{|\phi|} (\diamond \mathbf{true} \Rightarrow \bigvee_{\alpha \in \{\alpha_1, \dots, \alpha_n, \alpha_{new}\}} p_{\alpha})}^{\text{internal nodes labeled by tags}} \wedge \\ & \quad \forall^{|\phi|} (\square \mathbf{false} \Rightarrow \underbrace{\bigvee_{\delta \in \{\delta_1, \dots, \delta_{n'}, \delta_{new}\}} p_{\delta}}_{\text{leaves labeled by constants of datatypes}}) \wedge \\ & \quad \forall^{|\phi|} \bigwedge_{D \text{ is of the form } \exists \dots E_p} \bigvee_{i \in \{1, \dots, p\}} (p_D^i \wedge \bigwedge_{j \in \{1, \dots, p\} \setminus \{i\}} \neg p_D^j) \end{aligned}$$

where $|\phi|$ is the size of ϕ . Actually, an optimal construction would consider $\text{md}(\phi)$. Let t be reduction from SL formulae to PML formulae:

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

$$\phi(x_1, \dots, x_p)[x_1 \leftarrow \#(p_D^1 \wedge t(E_1)), \dots, x_p \leftarrow \#(p_D^p \wedge t(E_p))]$$

where $\phi(x_1, \dots, x_p)[x_1 \leftarrow \#(p_D^1 \wedge t(E_1)), \dots, x_p \leftarrow \#(p_D^p \wedge t(E_p))]$ is obtained from $\phi(x_1, \dots, x_p)$ by replacing each occurrence of x_i by $\#(p_D^i \wedge t(E_i))$.

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 9. t is a logspace reduction such that ϕ is SL satisfiable iff ϕ' is PML satisfiable.

So, SL satisfiability is in PSPACE which contrasts with the complexity of the decision procedure from [ZL06].

Proposition 1. SL is PSPACE-complete.

PSPACE-hardness is obtained by reducing modal logic K without propositional variables [Hem01].

4.3 Fixed-point free SSMH logic

In this section, we recall the syntax and semantics of the fixed-point free fragment of the logic from [SSMH04]. For brevity, we call it SSMH. The SSMH formulae are inductively defined as follows:

$$\begin{aligned} \phi ::= & \text{true} \mid \neg\phi \mid \phi \wedge \phi' \mid \alpha\langle\Phi(x_1, \dots, x_p) : x_1\phi_1 \& \dots \& x_p\phi_p\rangle \mid \\ & \star\langle\Phi(x_1, \dots, x_p) : x_1\phi_1 \& \dots \& x_p\phi_p\rangle \mid \alpha\langle\mathcal{A}(\phi_1, \dots, \phi_p)\rangle \mid \star\langle\mathcal{A}(\phi_1, \dots, \phi_p)\rangle. \end{aligned}$$

where α belongs to a countably infinite set TAGS of tags, \mathcal{A} is a nondeterministic finite-state automaton over an p -letter alphabet and $\Phi(x_1, \dots, x_p)$ is a Presburger formula as in SL. A model \mathcal{M} for SSMH is a structure $\mathcal{M} = \langle T, R, (\prec_{nd})_{nd \in T}, l \rangle$ where T is a finite set of states, $\langle T, R \rangle$ is a tree and each \prec_{nd} is a total ordering on $R(nd)$ and, $l : T \rightarrow \text{TAGS}$ is a labeling function (no datatypes here). The satisfaction relation is inductively defined below where \mathcal{M} is a model for SSMH and $nd \in T$ (we omit the clauses for Boolean operators):

- $\mathcal{M}, nd \models \alpha$ iff $\alpha = l(nd)$,
- $\mathcal{M}, nd \models \alpha\langle\Phi(x_1, \dots, x_p) : x_1\phi_1 \& \dots \& x_p\phi_p\rangle$ iff $\alpha = l(nd)$ and $R(nd) = nd_1 < \dots < nd_k$ and $[x_1 \leftarrow n_1, \dots, x_p \leftarrow n_p] \models \Phi(x_1, \dots, x_p)$ where $n_i = \text{card}(\{l \in \{1, \dots, k\} : \mathcal{M}, nd_l \models \phi_i\})$,
- $\mathcal{M}, nd \models \star\langle\Phi(x_1, \dots, x_p) : x_1\phi_1 \& \dots \& x_p\phi_p\rangle$ iff $[x_1 \leftarrow n_1, \dots, x_p \leftarrow n_p] \models \Phi(x_1, \dots, x_p)$ where $n_i = \text{card}(\{l \in \{1, \dots, k\} : \mathcal{M}, nd_l \models \phi_i\})$,
- $\mathcal{M}, nd \models \alpha\langle\mathcal{A}(\phi_1, \dots, \phi_p)\rangle$ iff $\alpha = l(nd)$, $R(nd) = nd_1 < \dots < nd_k$ and there is i_1, \dots, i_k such that for every $j \in \{1, \dots, k\}$, $\mathcal{M}, nd_j \models \phi_{i_j}$ and $\mathbf{a}_{i_1} \dots \mathbf{a}_{i_k} \in L(\mathcal{A})$. (analogous clause for $\star\langle\mathcal{A}(\phi_1, \dots, \phi_p)\rangle$).

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

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

– $t(\alpha(\phi(x_1, \dots, x_p) : x_1\phi_1 \& \dots \& x_p\phi_p))$ equals

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

– $t(\star(\phi(x_1, \dots, x_p) : x_1\phi_1 \& \dots \& x_p\phi_p))$ equals

$$\phi(x_1, \dots, x_p)[x_1 \leftarrow \#t(\phi_1), \dots, x_p \leftarrow \#t(\phi_p)].$$

– $t(\alpha(\mathcal{A}(\phi_1, \dots, \phi_p))) = p_\alpha \wedge \mathcal{A}(t(\phi_1), \dots, t(\phi_p))$,

– $t(\star(\mathcal{A}(\phi_1, \dots, \phi_p))) = \mathcal{A}(t(\phi_1), \dots, t(\phi_p))$.

Lemma 10. t is a logspace reduction s.t. ϕ is SSMH satisfiable iff $\forall^{|\phi|} \bigvee_{p \in VP} (p \wedge \bigwedge_{q \in VP \setminus \{p\}} \neg q) \wedge t(\phi)$ is PML satisfiable.

The proof is similar (and indeed simpler) than the proof of Lemma 9. So, SSMH satisfiability is in PSPACE. We can do better as done for SL.

Proposition 2. SSMH is PSPACE-complete.

4.4 PDL over finite trees

In [ABD⁺05] a PDL-like logic 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 PDL_{tree} but regularity constraints can be stated thanks to the interplay between the program operators and the atomic relations. 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 PDL_{tree} leads to undecidability. We provide below an undecidability proof for a logic sharing features from PDL_{tree} and PML, say \mathcal{L} , that is a strict fragment of the logic PDL_{tree} on which are added Presburger constraints. Hence, the logic \mathcal{L} contains features from both PDL_{tree} and PML while being incomparable with them since \mathcal{L} satisfiability will be shown below undecidable.

Given a countably infinite set $\text{AP} = \{p_1, p_2, \dots\}$ of propositional variables and $\Sigma = \{\downarrow, \downarrow^*, \rightarrow, \rightarrow^*, \leftarrow, \leftarrow^*, \uparrow, \uparrow^*\}$ a set of symbol relations, we define the set of formulae and terms inductively as follows: $\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid t \sim b$ and $t ::= a \times \#^R \phi \mid t + a \times \#^R \phi$, where $p \in \text{AP}$, $R \in \Sigma$, $b \in \mathbb{N}$, $a \in \mathbb{Z}$, $\sim \in \{<, >, =\}$. The programs from PDL_{tree} are much richer than Σ because iteration, test, union and composition are present in PDL_{tree} . Similarly, the Presburger constraints from PML strictly contains those of \mathcal{L} . 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; $l : T \rightarrow 2^{\text{AP}}$ is the valuation function,
- for every $R \in \{\downarrow, \rightarrow, \leftarrow, \uparrow\}$, $R_R^* = R_R^*$ (R_R^* is the reflexive and transitive closure of R_R), $R_\rightarrow = R_\leftarrow^{-1}$ and $R_\uparrow = R_\downarrow^{-1}$,

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

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

The proof is by reducing the halting problem for 2-counter machines. 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 $\Sigma_i a_i \#^{R_i} \phi_i$ where for some $j \neq j', R_j \neq R_{j'}$, PML augmented with the relation symbol \leftarrow .

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

5 Concluding Remarks

We have shown that Presburger modal logic that admits in its language full Presburger and regularity constraints has a PSPACE-complete satisfiability problem, that is the same complexity of the modal logic K. This is shown by designing a specially tailored Ladner-like algorithm that takes advantage of the constraint systems to be solved from PML formulae. This improves previous results from [Tob00,SSMH04,ZL06] and paves the way to design querying language for XML documents that can express Presburger and regularity constraints and for which the underlying modal logic is only in PSPACE.

References

- [ABD⁺05] L. Afanasiev, P. Blackburn, I. Dimitriou, B. Gaiffe, E. Goris, M. Marx, and M. de Rijke. PDL for ordered trees. *JANCL*, 15(2):115–135, 2005.
- [ADdR03] N. Alechina, S. Demri, and M. de Rijke. A modal perspective on path constraints. *JLC*, 13(6):939–956, 2003.
- [AH94] R. Alur and T. Henzinger. A really temporal logic. *JACM*, 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. *JANCL*, 14(4):447–475, 2004.
- [BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [BP04] P. Bonatti and A. Peron. On the undecidability of logics with converse, nominals, recursion and counting. *AI*, 158(1):75–96, 2004.
- [BT05] I. Boneva and J.M. Talbot. Automata and logics for unranked and unordered trees. In *RTA '05*, volume 3467 of *LNCS*, pages 500–515, 2005.

- [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.
- [Dem03] S. Demri. A polynomial space construction of tree-like models for logics with local chains of modal connectives. *TCS*, 300(1–3):235–258, 2003.
- [DL06] S. Demri and D. Lugiez. Presburger modal logic is PSPACE-complete. Technical report, LSV, ENS de Cachan, 2006.
- [Fin72] K. Fine. In so many possible worlds. *NDJFL*, 13(4):516–520, 1972.
- [FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *JCSS*, 18:194–211, 1979.
- [Gor99] R. Goré. Tableaux methods for modal and temporal logics. In *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.
- [Hem01] E. Hemaspaandra. The complexity of poor man’s logic. *JLC*, 11(4):609–622, 2001.
- [HST00] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic SHIQ. In *CADE-17*, volume 1831 of *LNCS*, pages 482–496. Springer, 2000.
- [KSV02] O. Kupferman, U. Sattler, and M.Y. Vardi. The complexity of the graded μ -calculus. In *CADE’02*, volume 2392 of *LNCS*, 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 *LNAI*, pages 150–164. Springer, 2003.
- [MP97] A. Montanari and A. Policriti. A set-theoretic approach to automated deduction in graded modal logics. In *IJCAI-15*, pages 196–201, 1997.
- [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.
- [Pap81] Chr. Papadimitriou. On the complexity of integer programming. *JACM*, 28(4):765–768, 1981.
- [Spa93] E. Spaan. The complexity of propositional tense logics. In M. de Rijke, editor, *Diamonds and Defaults*, pages 287–309. Kluwer, 1993.
- [SSMH04] H. Seidl, Th. Schwentick, A. Muscholl, and P. Habermehl. Counting in trees for free. In *ICALP’04*, volume 3142 of *LNCS*, pages 1136–1149, 2004. Long version available at <http://www.mathematik.uni-marburg.de/~tick/>.
- [Tob00] S. Tobies. PSPACE reasoning for graded modal logics. *JLC*, 10:1–22, 2000.
- [vdH92] W. van der Hoek. On the semantics of graded modalities. *JANCL*, 2(1):81–123, 1992.
- [vdHdR95] W. van der Hoek and M. de Rijke. Counting objects. *JLC*, 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.
- [VSS05] K. N. Verma, H. Seidl, and Th. Schwentick. On the complexity of equational Horn clauses. In *CADE’05*, volume 3632 of *LNCS*, pages 337–352, 2005.
- [Wol83] P. Wolper. Temporal logic can be more expressive. *I & C*, 56:72–99, 1983.
- [ZL03] S. Dal Zilio and D. Lugiez. XML schema, tree logic and sheaves automata. In *RTA 2003*, volume 2706 of *LNCS*, 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 (AAECC)*, 2006. To appear.