Stéphane Demri
Denis Lugiez

Complexity of Modal Logics with Presburger Constraints

Research Report LSV-08-25

September 2008

Laboratoire Spécification et Vérification

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

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 EXML 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 EXML 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 reduction from Sheaves logic SL into EXML that allows us to establish that SL satisfiability is also PSPACE-complete, significantly improving the best known upper bound.

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

1 Introduction

Logics for XML documents. In order to query XML documents with Presburger and/or regular constraints, logical and automata-based formalisms

[1] This is a complete and corrected version of [DL06]

have 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, Presburger 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 [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].

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,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 [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, 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 EXML 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]). The exact relationships between EXML, SL and the logic from [SSMH04] are provided in the paper. Our main result states that EXML 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,Tob00] (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]. Our algorithm can be viewed as the optimal composition between an algorithm that transforms an EXML 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 [ZL03,SSMH04]. Indeed, as by-products of the complexity results about EXML, we show that

- there is a logarithmic space reduction from Sheaves logic SL [resp. the fixpoint free fragment of the main logic from [SSMH04] (herein called SSMH)] into EXML.
- SL [resp. SSMH] satisfiability 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 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].

Plan of the paper. In Section 2, we introduce the extended modal logic EXML 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 following 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 EXML

2.1 Definition

Given countably infinite sets $\mathrm{AP} = \{p_1, p_2, \ldots\}$ 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

- $p \in \mathrm{AP}$, $\mathtt{R} \in \Sigma$,
- $b, k, c \in \mathbb{N}$, $a \in \mathbb{Z}$,
- $\sim \in \{<, >, =\}$,
- $\mathcal{A}$ 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 maximal number of imbrications of the symbol $\sharp$ 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 $|\phi|$.

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 $\Sigma_{i=1}^{i=n} a_i \sharp^{\mathtt{R}_i}\phi_i$. 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 EXML is a structure $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<^{\mathtt{R}}_{nd})_{nd\in T, \mathtt{R}\in\Sigma}, l\rangle$ where

- T is the set of nodes (possibly infinite),

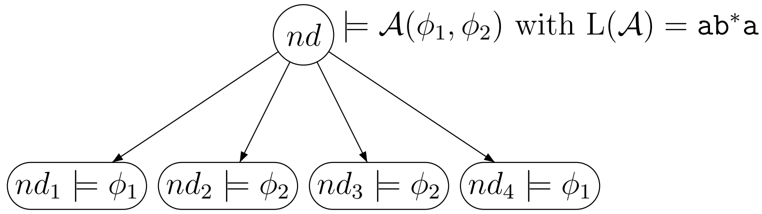

Fig. 1. Semantics for regular constraints

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

In the rest of the paper, we write $R_{\mathtt{R}}(nd) = nd_1 < \ldots < nd_\alpha$ to mean that

$$R_{\mathtt{R}}(nd) \stackrel{\text{def}}{=} \{nd' \in T : \langle nd, nd' \rangle \in R_{\mathtt{R}}\} = \{nd_1, \ldots, nd_\alpha\},$$

and $nd_1 <^{\mathtt{R}}_{nd} \ldots <^{\mathtt{R}}_{nd} nd_\alpha$. Given a finite-branching binary relation $R \subseteq T \times T$, we write $R^\sharp(nd)$ to denote the cardinal of the set $\{nd' \in T : \langle nd, nd' \rangle \in R\}$. The satisfaction relation $\models$ is inductively defined below where $\mathcal{M}$ is a model for EXML 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 $\mathcal{M}, nd \models \phi_2$,
- $\mathcal{M}, nd \models \Sigma_i a_i \sharp^{\mathtt{R}_i} \phi_i \sim b$ iff $\Sigma_i a_i R^\sharp_{\mathtt{R}_i, \phi_i}(nd) \sim b$ with $R_{\mathtt{R}_i, \phi_i} = \{\langle nd', nd'' \rangle \in T \times T : \langle nd', nd'' \rangle \in R_{\mathtt{R}_i},$ and $\mathcal{M}, nd'' \models \phi_i\}$,
- $\mathcal{M}, nd \models \Sigma_i a_i \sharp^{\mathtt{R}_i} \phi_i \equiv_k c$ iff there is $n \in \mathbb{N}$ such that $\Sigma_i a_i R^\sharp_{\mathtt{R}_i, \phi_i}(nd) = nk + c$,
- $\mathcal{M}, nd \models \mathcal{A}(\mathtt{R}, \phi_1, \ldots, \phi_n)$ iff there is $\mathtt{a}_{i_1} \cdots \mathtt{a}_{i_\alpha} \in \mathrm{L}(\mathcal{A})$ such that $R_{\mathtt{R}}(nd) = nd_1 < \ldots < nd_\alpha$ and for every $j \in \{1, \ldots, \alpha\}$, $\mathcal{M}, nd_j \models \phi_{i_j}$.

Observe that constraints of the form $\Sigma_i a_i \sharp^{\mathtt{R}_i} \phi_i \equiv_k c$ 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.

The automata in EXML 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 EXML 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 of EXML is satisfiable whenever there exist a model $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<^{\mathtt{R}}_{nd})_{nd\in T, \mathtt{R}\in\Sigma}, l\rangle$ and $nd \in T$ such that $\mathcal{M}, nd \models \phi$.

We write RML [resp. PML] to denote the restriction of EXML without Presburger constraints [resp. without regularity constraints]. For instance, PML is already more expressive than formalisms defined in [Tob00,PS04,KRM05].

2.2 Equivalence Between Graphs, Trees and Finite Trees

Even though EXML 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* EXML *formula ϕ, ϕ is satisfiable iff ϕ is satisfiable in a model $\mathcal{M}$ such that for all relation symbols $\mathtt{R}$ occurring in ϕ and $nd \in T$, the restriction of $\langle T, R_{\mathtt{R}}\rangle$ to $R^*_{\mathtt{R}}(nd)$ is a tree.*

Proof. Suppose that ϕ has a EXML model $\mathcal{M} = \langle T, (R_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<^{\mathtt{R}}_{nd})_{nd\in T, \mathtt{R}\in\Sigma}, l\rangle$ and a 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 orderings. The model $\mathcal{M}' = \langle T', (S_{\mathtt{R}})_{\mathtt{R}\in\Sigma}, (<'^{\mathtt{R}}_{nd})_{nd\in T', \mathtt{R}\in\Sigma}, l'\rangle$ is defined as follows:

- T' is the set of finite non-empty sequences of the form $nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_k\ nd_k$,
- $(nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_n\ nd_n)\ S_{\mathtt{R}}\ (nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_n\ nd_n\ \mathtt{R}_{n+1}\ nd_{n+1})$ iff $\langle nd_n, nd_{n+1}\rangle \in R_{\mathtt{R}}$ and $\mathtt{R} = \mathtt{R}_{n+1}$,
- $l'(nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_n\ nd_n) = l(nd_n)$ for every $nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_n\ nd_n \in T'$,
- each ordering $<'^{\mathtt{R}}_{nd'}$ is the one induced by $<^{\mathtt{R}}_{nd''}$ by considering the last element nd'' of the sequence nd'.

One can show that for every $nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_n\ nd_n \in T'$ and EXML formula ψ, $\mathcal{M}', nd\ \mathtt{R}_1\ nd_1 \ldots \mathtt{R}_n\ nd_n \models \psi$ 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* EXML *formula ϕ, ϕ is satisfiable iff ϕ is satisfiable in a model $\mathcal{M}$ such that T is finite and for all relation symbols $\mathtt{R}$ occurring in ϕ and $nd \in T$, the restriction of $\langle T, R_{\mathtt{R}}\rangle$ to $R^*_{\mathtt{R}}(nd)$ is a tree.*

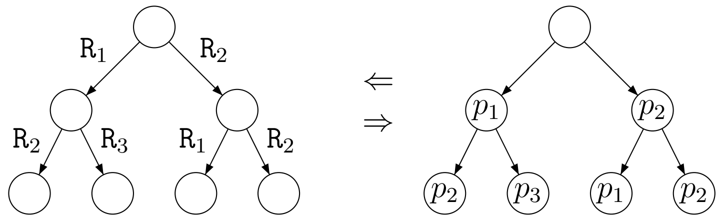

Fig. 2. Elimination of relation symbols

2.3 Restriction to One Relation

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

Lemma 3 *For every* EXML *formula ϕ, one can compute in logspace an* EXML *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.

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 $\mathcal{A}(\mathtt{R}_i, \psi_1, \ldots, \psi_m)$ 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.

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 $\langle T, R, (<_{nd})_{nd \in T}, l \rangle$.

3 An Algorithm for EXML Satisfiability

In this section, we show that EXML 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]).

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 $\mathrm{cl}(X)$ 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 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* EXML *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* $\mathrm{cl}(n, \phi)$, $\psi \in \mathrm{cl}(n+1, \phi)$,
- *for all* $n \in \mathbb{N}$ *and* $\mathcal{A}(\phi_1, \ldots, \phi_m) \in \mathrm{cl}(n, \phi)$, $\{\phi_1, \ldots, \phi_m\} \subseteq \mathrm{cl}(n+1, \phi)$.

In the sequel, we consider EXML formulae ϕ such that for every n such that $\mathrm{cl}(n,\phi) \neq \emptyset$, 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 EXML formula, one can compute an equivalent EXML 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 EXML 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 Appendix A) in the main algorithm. This is analogous to the requirement to check maximal consistency at the propositional level but not EXML satisfiability at once. It is the adequate construction of locally consistent sets that will guarantee that the initial set of formulae is EXML 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.

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

Let X be an n-locally consistent subset of $\mathrm{cl}(n,\phi)$. The set X is encoded as follows. To each subformula ψ in $\mathrm{cl}(n,\phi)$ 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

$\mathrm{cl}(n, \phi)$, 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 $\mathrm{cl}(n, \phi)$, we associate a value c' in $\{0, \ldots, K-1\}$ 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-locally consistent subset of $\mathrm{cl}(n, \phi)$ can be encoded with $\mathcal{O}(|\phi|^2)$ bits.

Before defining the main algorithm in Section 3.2, let us introduce the notion of M-bounded models. Let ϕ be an EXML formula, M be a natural number and $\mathcal{M}$ be a finite tree model such that $\mathcal{M}, nd \models \phi$ for some node nd. We say that $\langle \mathcal{M}, nd \rangle$ is M-bounded for ϕ iff for every node nd' of distance d from nd, the cardinal of $R(nd')$ 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 $d \geqslant |\phi|$.

3.2 The Algorithm

We define the function SAT such that ϕ is EXML satisfiable in some M-bounded model iff there is $X \subseteq \mathrm{cl}(0, \phi)$ such that X is 0-locally consistent and $\mathrm{SAT}(\phi, X, 0)$ has a computation that returns `true`. Indeed, the function $\mathrm{SAT}(X, \phi, d)$ defined in Figure 3 is parameterized by some natural number M (see the step (guess-number-children)). We shall fix later the value M that will be only exponential in $|\phi|$ (see Lemma 8).

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. We also write $Y_1, \ldots, Y_N$ the $(d+1)$-locally consistent sets (in the worst case N is exponential in $|\phi|$). In Figure 3, N is denoted by $nb(d+1)$. If we guess a set Y_x that contains some unsatisfiable formula (wrt M-bounded models) then $\mathrm{SAT}(Y_x, \phi, d+1)$ has no accepting computation which also induces a non accepting computation for $\mathrm{SAT}(X, \phi, d)$. Moreover, we check on the fly that the regularity constraints hold true as also done in [SSMH04]. In particular, we visit on the fly the automata obtained by the subset construction in order to check negative regularity constraints.

The algorithm described in SAT 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 (but its correctness proof is indeed a kind of completeness proof),
- the graph of recursive calls (here for SAT) induces a tree model for the argument formula. Since EXML models are precisely trees, we get the EXML

`function` SAT(X, ϕ, 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 SAT$(Y_x, \phi, d+1)$ then `abort`;
  - (3) for every $\psi \in \mathrm{cl}(d+1, \phi)$ that is not a periodicity constraint, if $\psi \in Y_x$, then $C_\psi := C_\psi + 1$;
  - (4) for every $\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$,
    - (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 $\Sigma_i a_i \sharp\psi_i \sim b \in X$, if $\Sigma_i a_i \times C_{\psi_i} \sim b$ does not hold, then `abort`,
  - (2) for every $\Sigma_i a_i \sharp\psi_i \equiv_k c \in X$, if $\Sigma_i a_i \times C_{\psi_i} \equiv_k c$ does not hold, then `abort`,
  - (3) for every $\mathcal{A}(\psi_1, \ldots, \psi_\alpha) \in X$, 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. 3. Satisfiability algorithm

model for free. For other (modal) logics, a model is usually obtained by applying to the tree a closure operator in order to get a model of the logic under consideration. The nature of this closure depends strongly on the conditions on the models (symmetry, transitivity, etc.). We do not consider this type of extensions in this paper.

3.3 Complexity Analysis

Firstly, we characterize the space needed to run SAT.

Lemma 5 *For all 0-locally consistent sets X, and computations of* $\mathrm{SAT}(\phi, X, 0)$

- *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 logspace in M.*

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

Proof. By Lemma 4, 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). 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 polynomial-time 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 $|\phi|$).

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

Proof. Assume that $\mathrm{SAT}(X, \phi, 0)$ has a computation $\mathcal{C}$ that returns `true`. Let us build a EXML model $\mathcal{M} = \langle T, R, (<_{nd})_{nd \in T}, l \rangle$ for which there is $nd \in T$ such that for every $\psi \in X$, we have $\mathcal{M}, nd \models \psi$ iff $\psi \in X$.

An execution of $\mathrm{SAT}(Y, \phi, d)$ that returns true can be represented by a finite sequence of pairs composed of a control state and the value of variables (here integers). We call it an accepting computation. Because of the very nature of the function SAT (see Lemma 5), it is not difficult to show that $\mathrm{SAT}(Y, \phi, d)$ has a finite number of accepting computations (doubly exponential in $|\phi|$) and therefore we write SC to denote the finite set of accepting computations for $\mathrm{SAT}(Y, \phi, d)$ with $d \leqslant |\phi|$ and Y is a d-locally consistent subset of $\mathrm{cl}(d, \phi)$. An accepting computation for $\mathrm{SAT}(Y, \phi, d)$ is said to be of depth d.

Let N be $|\phi|$. We write $seq_{M,N}$ to denote the sequences of length at most $N+1$ with elements in $\{1,\ldots,max_{0\leqslant d\leqslant|\phi|}nb(d+1)\times M\}$. For each $\sigma \in seq_{M,N}$, we write $\sigma(i)$ to denote the ith element of σ if it exists.

We define T as the set of tuples in $2^{\mathrm{cl}(\{\phi\})} \times \{0,\ldots,|\phi|\} \times seq_{M,N} \times SC^2$ such that $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle \in T$ iff

- $\mathcal{C}_1$ is an accepting computation for $\mathrm{SAT}(Y,\phi,d)$.
- σ is of length d,
- at some point of $\mathcal{C}_2$, $\mathrm{SAT}(Y,\phi,d)$ is called providing the execution $\mathcal{C}_1$ and $\mathcal{C}_2$ is of level $d-1$ if $d>0$.

Elements of T are not encoded in the most economic way since Y and d can be computed from $\mathcal{C}_1$ but introducing explicitly these objects simplifies the presentation. Indeed, the model $\mathcal{M}$ is of doubly exponential size in $|\phi|$ but we shall show below how to simply reduce it as an exponential size M-bounded model.

Let us define the binary relations:

- $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle\ R\ \langle Y', d', \sigma', \mathcal{C}'_1, \mathcal{C}'_2\rangle$ iff
(1) $d' = d+1$, $\mathcal{C}_1 = \mathcal{C}'_2$,
(2) in $\mathcal{C}_1$, $\mathrm{SAT}(Y',\phi,d+1)$ has an accepting computation $\mathcal{C}'_1$ and this computation is done at least $\sigma'(d+1)$ times,
(3) for all i, $1 \leqslant i \leqslant d$ implies $\sigma'(i) = \sigma(i)$.
- $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle <_{\langle Z,d,\sigma,\mathcal{C}''_1,\mathcal{C}''_2\rangle} \langle Y', d', \sigma', \mathcal{C}'_1, \mathcal{C}'_2\rangle$ iff
(1) $\mathcal{C}''_1 = \mathcal{C}_2 = \mathcal{C}'_2$,
(2) $\langle Z, d, \sigma, \mathcal{C}''_1, \mathcal{C}''_2\rangle R \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle$ and $\langle Z, d, \sigma, \mathcal{C}''_1, \mathcal{C}''_2\rangle R \langle Y', d', \sigma', \mathcal{C}'_1, \mathcal{C}'_2\rangle$,
(3) the $\sigma(d)$th call to $\mathrm{SAT}(Y,\phi,d)$ in $\mathcal{C}_2$ is before the $\sigma'(d)$th call to $\mathrm{SAT}(Y',\phi,d)$ in $\mathcal{C}_2$ (in the step (guess-children-from-left-to-right)).

The labeling function $l : T \to 2^{\mathrm{AP}}$ is simply defined by $l(\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle) \stackrel{\mathrm{def}}{=} \mathrm{AP} \cap Y$. R is indeed a finite tree and each relation $<_{\langle Z,d,\sigma,\mathcal{C}''_1,\mathcal{C}''_2\rangle}$ is a linear ordering on the set of R-successors of $\langle Z, d, \sigma, \mathcal{C}''_1, \mathcal{C}''_2\rangle$. So $\mathcal{M}$ is a EXML model.

By structural induction on ψ, we shall show that for all $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle \in T$, $\psi \in \mathrm{cl}(d,\phi)$, $\psi \in Y$ iff $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle \models \psi$. Consequently, we then get $\mathcal{M}, \langle X, 0, \epsilon, \mathcal{C}, \mathcal{C}\rangle \models \phi$ where $\mathcal{C}$ is an accepting computation for $\mathrm{SAT}(X,\phi,0)$. 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 $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle \in T$, if $\psi \in \mathrm{cl}(d,\phi)$, then $\psi \in Y$ iff $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2\rangle \models \psi$.

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 $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \in T$ such that $\psi \in \mathrm{cl}(d, \phi)$. By definition of T, $\mathcal{C}_1$ is an accepting computation of $\mathrm{SAT}(Y, \phi, d)$. If $\psi \in Y$, then each call in $\mathrm{SAT}(Y_{x_1}, \phi, d+1)$, ..., $\mathrm{SAT}(Y_{nb}, \phi, d+1)$ has an accepting computation (providing the computations $\mathcal{C}_1^1, \ldots, \mathcal{C}_1^{nb}$). Hence the children of $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle$ in $\mathcal{M}$ are the following (from left to right):

$$\langle Y_{x_1}, d+1, \sigma_1, \mathcal{C}_1^1, \mathcal{C}_1 \rangle, \ldots, \langle Y_{x_{nb}}, d+1, \sigma_{nb}, \mathcal{C}_1^{nb}, \mathcal{C}_1 \rangle$$

and it is not difficult to show that the steps (initialization-states), (guess-children-from-left-to-right)(4) and (final-checking)(3) guarantee that $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \models \psi$.

If $\psi \notin Y$, 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 $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \not\models \mathcal{A}(\psi_1, \ldots, \psi_k)$.

Case 2: $\psi = \Sigma_{i=1}^{\alpha} a_i \sharp\psi_i \sim b$.
Let $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \in T$ such that $\psi \in \mathrm{cl}(d, \phi)$. By definition of T, $\mathcal{C}_1$ is an accepting computation of $\mathrm{SAT}(Y, \phi, d)$. If $\psi \in Y$, then each call in $\mathrm{SAT}(Y_{x_1}, \phi, d+1)$, ..., $\mathrm{SAT}(Y_{nb}, \phi, d+1)$ has a successful computation (providing the computations $\mathcal{C}_1^1, \ldots, \mathcal{C}_1^{nb}$) and 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 successful computation $\mathcal{C}_1$. Hence the children of $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle$ in $\mathcal{M}$ are the following (from left to right):

$$\langle Y_{x_1}, d+1, \sigma_1, \mathcal{C}_1^1, \mathcal{C}_1 \rangle, \ldots, \langle Y_{x_{nb}}, d+1, \sigma_{nb}, \mathcal{C}_1^{nb}, \mathcal{C}_1 \rangle$$

and it is not difficult to show that the steps (initialization-counters), (guess-children-from-left-to-right)(3) and (final-checking)(1) guarantee that $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \models \psi$.

If $\psi \notin Y$, then by consistency of Y, there is $\sim' \in \{<, >, =\} \setminus \{\sim\}$ such that $\Sigma_{i=1}^{\alpha} a_i \sharp\psi_i \sim' b \in X$ by following a reasoning as above this means that $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \models \Sigma_{i=1}^{\alpha} a_i \sharp\psi_i \sim' b$ which is actually equivalent to $\mathcal{M}, \langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle \not\models \Sigma_{i=1}^{\alpha} a_i \sharp\psi_i \sim b$.

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

The current model $\mathcal{M}$ is in double exponential size in $|\phi|$ and it is easy to show that $\mathcal{M}$ is M-bounded. One can obtain the exponential size model property by considering a unique accepting computation for each triple $\langle Y, d, \sigma \rangle$ if some node $\langle Y, d, \sigma, \mathcal{C}_1, \mathcal{C}_2 \rangle$ belongs to $\mathcal{M}$ ($\mathcal{C}_1$ belongs irrelevant). Additionally, one can get rid of $\mathcal{C}_2$ by replacing $\mathcal{C}_2$ by the triple $\langle Y', d', \sigma' \rangle$ such that $\mathcal{C}_2$ is an accepting computation for $\mathrm{SAT}(Y', \phi, d')$. In that way, we get exactly a set of nodes of exponential size in $|\phi|$. Moreover, this model is still M-bounded.

The converse property holds.

Lemma 7 *If ϕ is* EXML *satisfiable in some M-bounded model then for some $X \subseteq \mathrm{cl}(0, \phi)$,* $\mathrm{SAT}(X, \phi, 0)$ *has a computation that returns* `true`*.*

Proof. Assume that ϕ is EXML satisfiable in some M-bounded model. So there is an EXML model $\mathcal{M} = \langle T, R, (<_{nd})_{nd \in T}, l \rangle$ and $nd \in T$ such that $\mathcal{M}, nd \models \phi$ and $\langle \mathcal{M}, nd \rangle$ is M-bounded. We shall show that for all $d \in \{0, \ldots, |\phi|\}$ and $X \subseteq \mathrm{cl}(d, \phi)$ such that there exist an EXML model $\mathcal{M}' = \langle T', R', (<'_{nd})_{nd \in T'}, l' \rangle$ $nd' \in T'$ verifying $X = \{\psi \in \mathrm{cl}(d, \phi) : \mathcal{M}', nd' \models \psi\}$ and $\langle \mathcal{M}', nd' \rangle$ is M-bounded, $\mathrm{SAT}(X, \phi, d)$ has an accepting computation. Consequently, we get that $\mathrm{SAT}(\{\psi \in \mathrm{cl}(0, \phi) : \mathcal{M}, nd \models \psi\}, \phi, 0)$ 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 1: $d = |\phi|$.
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 EXML model $\mathcal{M}' = \langle T', R', (<'_{nd})_{nd \in T'}, l' \rangle$ and $nd' \in T'$ verifying $X = \{\psi \in \mathrm{cl}(d', \phi) : \mathcal{M}', nd' \models \psi\}$ and $\langle \mathcal{M}', nd' \rangle$ is M-bounded, $\mathrm{SAT}(X, \phi, d')$ has an accepting computation.

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

For $i \in \{1, \ldots, N\}$ (N is the number of $d'+1$-locally consistent sets), we write n_i to denote the number of children nd'' of nd' such that $Y_i = \{\psi \in \mathrm{cl}(d+1, \phi) : \mathcal{M}, nd'' \models \psi\}$. Since $\mathcal{M}'$ is M-bounded, $\Sigma_i n_i \leqslant nb(d+1) \times M$. This is sufficient to establish that $\mathrm{SAT}(X, \phi, d')$ 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, $\mathrm{SAT}(Y_i, \phi, d'+1)$ 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, $\mathrm{SAT}(X, \phi, d')$ has an accepting computation.

So, we have established that a formula ϕ is EXML 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`. Moreover, we can establish the following lemma partly based on [SSM07] and whose proof can be found in Appendix A.

Lemma 8 *There is a polynomial $p(\cdot)$ such that for every formula ϕ, ϕ is* EXML *satisfiable iff ϕ is satisfiable in some $2^{p(|\phi|)}$-bounded model.*

By Lemmas 5, 6, 7 and 8 (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 EXML *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 EXML with other logics dealing with Presburger constraints. This is the opportunity to clarify the relationships between EXML 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 EXML where the formulae with Presburger constraints are of the form $\Diamond_{\geqslant n}\phi$ and the like, 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 [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 complexity result about EXML 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,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 this is 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 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 EXML models whereas the mode of representation for regular languages and semilinear sets is the same as for EXML. 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 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 $t \sim b$ or $t \equiv_k c$ with $t = \Sigma a_i x_i$.

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

- T is a finite set of states,
- $\langle T, R \rangle$ is a tree and each $<_{nd}$ is a total ordering on $R(nd)$,

- $l : T \rightarrow$ TAGS $\cup$ DATATYPES is a labeling function such that
  - for every $nd \in T$, if nd is a leaf of $\langle T, R \rangle$ then $l(nd) \in$ DATATYPES,
  - for every $nd \in T$, if nd is not a leaf of $\langle T, R \rangle$ then $l(nd) \in$ TAGS.

The satisfaction relation $\models$ 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 $\mathcal{M}, nd \models \alpha[D]$,
- $\mathcal{M}, nd \models \alpha[\mathtt{true}]$ iff $\alpha = l(nd)$,
- $\mathcal{M}, nd \models \alpha[\exists x_1, \ldots, x_p : \phi(x_1, \ldots, x_p) : x_1 E_1 \& \cdots \& x_p E_p]$ iff $\alpha = l(nd)$, $R(nd) = nd_1 < \cdots < nd_k$, and there exist $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, nd_j \models E_{i_j}$ and $[x_1 \leftarrow n_1, \ldots, x_p \leftarrow n_p] \models \phi(x_1, \ldots, x_p)$ with $n_i = \mathrm{card}(\{s \in \{1, \ldots, k\} : i_s = i\})$,
- $\mathcal{M}, nd \models \alpha[\mathcal{A}(E_1, \ldots, E_p)]$ iff $\alpha = l(nd)$, $R(nd) = nd_1 < \cdots < nd_k$, and there is $i_1, \ldots, i_k$ such that for every $j \in \{1, \ldots, k\}$, $\mathcal{M}, nd_j \models E_{i_j}$ and $\mathtt{a}_{i_1} \cdots \mathtt{a}_{i_k} \in \mathrm{L}(\mathcal{A})$ with $\Sigma_{\mathcal{A}} = \mathtt{a}_1, \ldots, \mathtt{a}_p$.

A major difference with the semantics of EXML (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 EXML formula ϕ' built over the propositional variables (plus others, see below)

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

Given an EXML 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

- $\forall^{|\phi|} \bigvee_{p \in VP} (p \wedge \bigwedge_{q \in VP \setminus \{p\}} \neg q) \ \wedge \ \overbrace{\forall^{|\phi|} (\Diamond \mathtt{true} \Rightarrow \bigvee_{\alpha \in \{\alpha_1, \ldots, \alpha_n, \alpha_{new}\}} p_\alpha)}^{\text{internal nodes labeled by tags}}$

- $\underbrace{\forall^{|\phi|}(\Box\mathtt{false} \Rightarrow \bigvee_{\delta \in \{\delta_1,\ldots,\delta_{n'},\delta_{new}\}} p_\delta)}_{\text{leaves labeled by datatypes}}$
- $\forall^{|\phi|}(\bigwedge_{D \text{ is of the form } \exists \ldots E_p} (\bigwedge_{i \neq j \in \{1,\ldots,p\}} \neg(p^i_D \wedge p^j_D)) \wedge p^i_D \Rightarrow t(E_i))$.

where $|\phi|$ is the size of ϕ (an optimal construction would consider $\mathrm{md}(\phi)$) and t is the reduction from SL formulae to EXML formulae defined below. Observe that since $\Leftrightarrow$ is not part of the original language, $p^i_D \Leftrightarrow t(E_i)$ uses two copies of $t(E_i)$ but this does not cause any exponential blow-up. Otherwise, this can be easily circumvent by using standard renaming techniques.

- 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:

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

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

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 satisfiable iff ϕ' is satisfiable.*

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

- $\langle T', R', (<'_{nd})_{nd \in T'}\rangle = \langle T, R, (<_{nd})_{nd \in T}\rangle$,
- for every $nd \in T'$, $p_{l(nd)} \in l'(nd)$. Moreover, $l'(nd)$ may contain other propositional variables of the form p^i_D 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 $\mathcal{M}, nd \models \alpha[D]$, then by definition $R(nd) = nd_1 < \cdots < nd_k$, and there are $i_1,\ldots,i_k$ such that for every $j \in \{1,\ldots,k\}$, $\mathcal{M}, nd_j \models E_{i_j}$ 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 $p^{i_j}_D \in l'(nd_j)$.
  If $\mathcal{M}, nd \not\models \alpha[D]$ and $\mathcal{M}, nd \models \alpha[\exists x_1 \cdots x_p : \neg\phi(x_1,\ldots,x_p) : x_1 E_1 \;\&\cdots\&\; x_p E_p]$, then by definition $R(nd) = nd_1 < \cdots < nd_k$, and there are $i_1,\ldots,i_k$ such that for every $j \in \{1,\ldots,k\}$, $\mathcal{M}, nd_j \models E_{i_j}$ 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 $p^{i_j}_D \in l'(nd_j)$.

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

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

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

- $\langle T', R', (<'_{nd})_{nd \in T'} \rangle = \langle T, R, (<_{nd})_{nd \in T} \rangle$,
- for every $nd \in T'$, $l'(nd) = \beta$ where β is the unique element of TAGS $\cup$ DATATYPES such that $p_\beta \in l(nd)$ ($l(nd)$ 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 $\mathcal{M}', nd \models \phi$.

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 $\mathcal{M}' = \langle T', R', (<'_{nd})_{nd \in T'}, l' \rangle$ is defined as follows:

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

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.

4.3 Fixpoint free SSMH logic

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. Like for SL, definitions are adapted to our presentation to EXML which allows to compare easily the (sometimes minor) differences between EXML, SL and SSMH. The SSMH formulae are inductively defined as follows:

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

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

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 $\mathcal{M} = \langle T, R, (<_{nd)_{nd \in T}}, l\rangle$ where

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

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

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

Unlike SL and like EXML, 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 EXML formula ϕ' built over the propositional variables $VP = \{p_{\alpha_1}, \ldots, p_{\alpha_n}, p_{\alpha_{n+1}}\}$. Let t be a logspace reduction from SSMH formulae to EXML formulae:

- t is homomorphic for Boolean operators and $t(\mathtt{true}) = \mathtt{true}$,
- $t(\alpha\langle\phi(x_1, \ldots, x_p) : x_1\phi_1 \& \cdots \& x_p\phi_p\rangle)$ 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 10 *t is a logspace reduction such that ϕ is satisfiable iff $\forall^{|\phi|} \bigvee_{p \in VP}(p \wedge \bigwedge_{q \in VP \setminus \{p\}} \neg q) \wedge t(\phi)$ is satisfiable.*

The proof is similar (and indeed simpler) than the proof of Lemma 9. We can do better as done for SL.

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:

- $t'(\mathtt{true}) = \mathtt{true}$, $t'(\mathtt{false}) = \neg\mathtt{true}$,
- 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.

4.4 PDL over finite trees

In [ABD+05] a PDL-like logic $\mathtt{PDL}_{\mathtt{tree}}$ 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 $\mathtt{PDL}_{\mathtt{tree}}$ (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 EXML , say $\mathcal{L}$, that is a strict fragment of the logic $\mathtt{PDL_{tree}}$ on which are added Presburger constraints. Hence, the logic $\mathcal{L}$ contains features from both $\mathtt{PDL_{tree}}$ and EXML while being incomparable with them since $\mathcal{L}$ satisfiability will be shown below undecidable.

Given a countably infinite set $\mathrm{AP} = \{p_1, p_2, \ldots\}$ 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 $p \in \mathrm{AP}$, $\mathtt{R} \in \Sigma$, $b \in \mathbb{N}$, $a \in \mathbb{Z}$ 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 EXML strictly contains those of $\mathcal{L}$ (no modulo constraints in $\mathcal{L}$). 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;
- $l : T \rightarrow 2^{\mathrm{AP}}$ 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 EXML 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 $\mathcal{L}$.

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 $\Sigma_i a_i \sharp^{\mathtt{R}_i} \phi_i$ where for some $j \neq j'$, $\mathtt{R}_j \neq \mathtt{R}_{j'}$,
- EXML augmented with the relation symbol $\leftarrow$,
- $\mathtt{PDL}_{\mathtt{tree}}$ augmented with a subclass of Presburger constraints.

The logic obtained by adding $\downarrow^*$ to EXML 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 (use the complexity result of [FL79]).

5 Concluding Remarks

We have shown that EXML satisfiability problem is only PSPACE-complete. This is established by designing a specially tailored Ladner-like algorithm that takes advantage of the constraint systems to be solved from EXML formulae. We improve 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.

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

EXML . For instance, the decidability status of EXML 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].

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.

[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. Technical report, LSV, ENS de Cachan, 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.

[KRM05] V. Kuncak, M. Rinard, and B. Manette. On algorithms and complexity for sets with cardinality constraints. In *Deduction and Applications, 23.-28. October 2005. Dagstuhl Seminar Proceedings 05431 Internationales Begegnungs- und Forschungszentrum (IBFI)*, October 2005.

[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.

[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.

[Tob00] S. Tobies. PSPACE reasoning for graded modal logics. *Journal of Logic and Computation*, 10:1–22, 2000.

[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 8

The proof basically restates the proof of [SSM07, Claim 7.3] in the context of EXML with subsets of $\mathrm{cl}(n, \phi)$. Section A.1 recalls a result due to Papadimitriou [Pap81] about small solutions of constraint systems. 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.

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 $\Sigma_j a_j \times x_{i_j} = b$ 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 11 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 11 *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 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 $\mathcal{S}$.*

Proof.

The system $\mathcal{S}$ can be transformed in disjunctive normal form providing a disjunction of conjunctions with conjuncts of the form either $\Sigma_j a_j \times x_{i_j} = b$ or $\neg(\Sigma_j a_j \times x_{i_j} = b)$. Each disjunct has at most m atomic constraints. Since $\neg(\Sigma_j a_j \times x_{i_j} = b)$ can be rewritten as $(\Sigma_j a_j \times x_{i_j} - y = b+1) \vee (\Sigma_j a_j \times x_{i_j} + y' = b - 1)$, 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.

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. They all belong to $\mathrm{cl}(n+1, \phi)$.

First, let us build automata $\mathcal{B}_1$, ..., $\mathcal{B}_l$ over the alphabet $\Sigma = \{Y_1, \ldots, Y_N\}$ where $Y_1, \ldots, Y_N$ are the only $(n+1)$-locally consistent sets (N is exponential in $|\phi|$). For $i \in \{1, \ldots, l\}$, $\mathcal{B}_i$ 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 $\mathcal{B}'_1$, ..., $\mathcal{B}'_{l'}$ from the automata $\mathcal{A}'_1$, ..., $\mathcal{A}'_{l'}$. We write $\mathcal{B}'^{\neg}_1$, ..., $\mathcal{B}'^{\neg}_{l'}$ to denote the complement automata obtained, for instance, by the powerset construction.

Hence, we can define a product automaton $\mathcal{B}$ obtained by synchronizing $\mathcal{B}'_1$, ..., $\mathcal{B}_l$, $\mathcal{B}'_1$, ..., $\mathcal{B}'_{l'}$ 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 $w = Y_1 \cdots Y_\alpha \in \Sigma^*$, $w \in \mathrm{L}(\mathcal{B})$ iff the conditions below hold true.
  - For $i \in \{1, \ldots, l\}$, there are $\psi_1 \in Y_1$, ..., $\psi_\alpha \in Y_\alpha$ such that $\psi_1 \cdots \psi_\alpha \in \mathrm{L}(\mathcal{A}_i)$.
  - For $i \in \{1, \ldots, l'\}$, there are no $\psi_1 \in Y_1, \ldots, \psi_\alpha \in Y_\alpha$ such that $\psi_1 \cdots \psi_\alpha \in \mathrm{L}(\mathcal{A}'_i)$.

The Parikh image of $\mathrm{L}(\mathcal{B})$, subset of $\mathbb{N}^N$ and denoted by $\pi(\mathrm{L}(\mathcal{B}))$, is a finite union $\mathrm{L}_1 \cup \cdots \cup \mathrm{L}_m$ of linear sets $\mathrm{L}_i = \{\sigma_0 + \Sigma_{j=1}^h y_j\sigma_j : y_j \geqslant 0\}$ where each σ_j is in $\{0, \ldots, |Q'|\}^N$ 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 11), if the constraint system

$$\begin{pmatrix} z_1 \\ z_2 \\ \dots \\ z_N \end{pmatrix} = \sigma_0 + \Sigma_{j=1}^{h} y_j \sigma_j$$

made of $N + h$ variables and N 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 equations

We write $H : \mathbb{N}^N \to \mathbb{N}^P$ to denote the homomorphism such that

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

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 $\{H(\sigma_0) + \Sigma_{j=1}^{h} y_j H(\sigma_j) : y_j \geqslant 0\}$. 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

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

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

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

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 12 *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 EXML 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 $t_\psi = \Sigma_{i, \psi \in Y_i} x_i$. 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:

- $\Sigma_{Y_i \text{ is not satisfiable}}\, x_i = 0$,
- if $\Sigma_i a_i \sharp\phi_i = b \in X$, then we add $\Sigma_i a_i t_{\phi_i} = b$,
- if $\Sigma_i a_i \sharp\phi_i < b \in X$, then we add $\Sigma_i a_i t_{\phi_i} + y = b - 1$ where y is a new variable,
- if $\Sigma_i a_i \sharp\phi_i > b \in X$, then we add $\Sigma_i a_i t_{\phi_i} - y = b + 1$ where y is a new variable,
- if $\Sigma_i a_i \sharp\phi_i \equiv_K c \in X$, then we add $\Sigma_i a_i t_{\phi_i} - Ky = c$ 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 11 and 12, if $\mathcal{S}_X$ has solutions, then $\mathcal{S}_X$ has solutions with values bounded by some M exponential in $|\phi|$. We write M the maximal value amongst all the values obtained for the different depths n between 0 and $|\phi|$.

The proof of Lemma 8 is then a simple consequence of Lemma 13 below.

Lemma 13 *Let ϕ be a* EXML *formula, $d \in \{0, \ldots, |\phi|\}$ and X be a d-locally consistent set of formulae. Then, X is* EXML *satisfiable iff $\mathcal{S}_X$ has a positive solution.*

Proof. It is easy to check that if X is EXML 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 EXML model $\mathcal{M} = \langle T, R, (<_{nd})_{nd \in T}, l \rangle$ 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 EXML model $\mathcal{M}_i = \langle T_i, R_i, (<^i_{nd})_{nd \in T_i}, l_i \rangle$ and $nd_i \in T_i$ such that $\mathcal{M}_i, nd_i \models Y_i$. $\mathcal{M}$ is built from n_1 copies of $\mathcal{M}_1$, ..., $n_{nb(d+1)}$ copies of $\mathcal{M}_{nb(d+1)}$ by adding R-transitions between the root nd of T (a new state) and all the nd_i's of all copies. Moreover $l(nd) = \mathrm{AP} \cap X$. 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 nd 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 be a d-locally consistent of formulae is a d-locally consistent set, one can easily show that $\mathcal{M}, nd \models X$. 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 $\mathcal{S}_X$.