Modal Logics with Composition on Finite Forests: Expressivity and Complexity

Bartosz Bednarczyk
TU Dresden & University of Wrocław

Stéphane Demri
LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay

Raul Fervari
FAMAF, Universidad Nacional de Córdoba & CONICET

Alessio Mansutti
LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay

Abstract

We study the expressivity and complexity of two modal logics interpreted on finite forests and equipped with standard modalities to reason on submodels. The logic ML($|$) extends the modal logic K with the composition operator $|$ from ambient logic, whereas ML($*$) features the separating conjunction $*$ from separation logic. Both operators are second-order in nature. We show that ML($|$) is as expressive as the graded modal logic GML (on trees) whereas ML($*$) is strictly less expressive than GML. Moreover, we establish that the satisfiability problem is Tower-complete for ML($*$), whereas it is (only) AExp$_{\text{Pol}}$-complete for ML($|$), a result which is surprising given their relative expressivity. As by-products, we solve open problems related to sister logics such as static ambient logic and modal separation logic.

CCS Concepts: • **Theory of computation → Modal and temporal logics**.

Keywords: modal logic on trees, separation logic, static ambient logic, graded modal logic, expressive power, complexity

1 Introduction

The ability to quantify over substructures to express properties of a model is often instrumental to perform modular and local reasoning. Two well-known examples are provided

by separation logics [28, 35, 42], dedicated to reasoning on pointer programs, and ambient (or more generally, spatial) logics [9, 12, 14, 19], dedicated to reasoning on disjoint data structures. In the realm of modal logics dedicated to knowledge representation, submodel reasoning remains a key ingredient to express the dynamics of knowledge and belief, as done in the logics of public announcement [5, 33, 37], sabotage modal logics [4], refinement modal logics [11] and relation-changing logics [1–3]. Though the models may be of different nature (e.g. memory states for separation logics, epistemic models for logics of public announcement or finite edge-labelled trees for ambient logics), all those logics feature composition operators that enable to compose or decompose substructures in a very natural way.

From a technical point of view, reasoning about submodels requires a global analysis, unlike the local approach for classical modal and temporal logics (typically based on automata techniques [47, 48]). This makes the comparison between those formalisms quite challenging and often limited to a superficial analysis on the different classes of models and composition operators. For instance, the composition operator $|$ in ambient logics decomposes a tree into two disjoint pieces such that once a node has been assigned to one submodel, all its descendants belong to the same submodel. Instead, the separating conjunction $*$ from separation logic decomposes the memory states into two disjoint memory states. Obviously, these and other well-known operators are closely related but no uniform framework investigates exhaustively their relationships in terms of expressive power.

Most of these logics can be easily encoded in monadic second-order logic MSO (or in second-order modal logics [24, 30]). Complexity-wise, if models are tree-like structures, we can then infer decidability thanks to the celebrated Rabin's theorem [40]. However, most likely, this does not produce the best decision procedures when it comes to solving simple reasoning tasks (e.g. the satisfiability problem of MSO is Tower-complete [43]). Thus, relying on MSO as a common umbrella to capture and understand the differences between those logical formalisms is often not satisfactory.

Our motivations. Our intention in this work is to provide an in-depth comparison between the composition operator $|$ from static ambient logic [12] and the separating conjunction $*$ from separation logics [42] by identifying a common

ground in terms of logical languages and models. As a consequence, we are able to study the effects of having these operators as far as expressivity and complexity are concerned. We aim at defining two logics whose only differences rest on their use of | and ∗ syntactically and semantically (by considering the adequate composition operation). To do so, we pick as our common class of models, the Kripke-style finite trees (actually finite forests, so that the class is closed under taking submodels), which provides an ubiquitous class of structures, extremely well-studied in computer science. For the underlying logical language (i.e. apart from | or ∗), we advocate the use of the standard modal logic K (i.e. to have Boolean connectives and the standard modality ◇) so that the main operations on the models amount to quantify over submodels or to move along the edges. This framework is sufficiently fundamental to give us the possibility to take advantage of model theoretical tools from modal logics [6, 8, 20]. The benefits of settling a common ground for comparison may lead to further comparisons with other logics and new results.

Our contributions. We introduce ML(|) and ML(∗), two logics interpreted on Kripke-style forest models, equipped with the standard modality ◇, and respectively with the composition operator | from static ambient logic [12] and with the separating conjunction ∗ from separation logic [42]. Both logical formalisms can state non-trivial properties about submodels, but the binary modalities | and ∗ operate differently: whereas ∗ is able to decompose the models at any depth, | is much less permissive as the decomposition is completely determined by what happens at the level of the children of the current node. We study their expressive power and complexity, obtaining surprising results. We show that ML(|) is as expressive as the graded modal logic GML [6, 45] whereas ML(∗) is strictly less expressive than GML. Interestingly, this latter development partially reuses the result for ML(|), hence showing how our framework allows us to transpose results between the two logics. To show that GML is strictly more expressive than ML(∗), we define Ehrenfeucht-Fraïssé games for ML(∗). In terms of complexity, the satisfiability problem for ML(|) is shown AExp_{Pol}-complete[1], interestingly the same complexity as for the refinement modal logic RML [11] handling a quantifier over refinements (generalising the submodel construction). The AExp_{Pol} upper bound follows from an exponential-size model property, whereas the lower bound is by reducing the satisfiability problem for an AExp_{Pol}-complete team logic [27]. Much more surprisingly, although ML(∗) is strictly less expressive than ML(|), its complexity is much higher (not even elementary). Precisely, we show that the satisfiability problem for ML(∗) is Tower-complete. The Tower upper bound is a consequence of [40], whereas hardness is shown by reduction from a Tower-complete tiling problem, adapting substantially the

---

[1] Problems in AExp_{Pol} are decidable by an alternating Turing machine working in exponential-time and using polynomially many alternations [10].

Tower-hardness proof from [7] for second-order modal logic K on finite trees. To conclude, we get the best of our results on ML(|) and ML(∗) to solve several open problems. We relate ML(|) with an intensional fragment of static ambient logic SAL(|) from [12] by providing polynomial-time reductions between their satisfiability problems. Consequently, we establish AExp_{Pol}-completeness of SAL(|), refuting hints from [12, Section 6]. Similarly, we show that the modal separation logic MSL($◇^{-1}$, ∗) from [21] is Tower-complete. *Omitted proofs can be found in the technical report on ArXiv.*

2 Preliminaries

In this section, we introduce the logics ML(|) and ML(∗) interpreted on tree-like structures equipped with operators to split the structure into disjoint pieces. Due to the presence of such operators, we are required to consider a class of models that is closed under submodels, which we call Kripke-style finite forests (or finite forests for short).

Let AP be a countably infinite set of *atomic propositions*. A *(Kripke-style) finite forest* is a triple $\mathfrak{M} = (W, R, V)$ where W is a non-empty finite set of *worlds*, $V : \text{AP} \to \mathcal{P}(W)$ is a *valuation* and $R \subseteq W \times W$ is a binary relation whose inverse R^{-1} is functional and acyclic. Then, in particular the graph described by (W, R) is a finite collection of disjoint finite trees (where R encodes the child relation).

We define $R(w) \overset{\text{def}}{=} \{w' \in W \mid (w, w') \in R\}$. Worlds in $R(w)$ are understood as *children* of w. We inductively define R^n: $R^0 \overset{\text{def}}{=} \{(w, w) \mid w \in W\}$; $R^{n+1} \overset{\text{def}}{=} \{(w, w'') \mid \exists w' (w, w') \in R^n$ and $(w', w'') \in R\}$. R^+ denotes the transitive closure of R.

We define operators that chop a finite forest. It should be noted that these operators, as well as the resulting logics, can be cast under the umbrella of the logic of bunched implications BI [25, 39], with the exception that we do not explicitly require them to have an identity element (as enforced on the multiplicative operators of BI, see [25]). Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}_i = (W_i, R_i, V_i)$ (for $i \in \{1, 2\}$) be three finite forests.

The separation logic composition. We introduce the binary operator + that performs the disjoint union at the level of parent-child relation. Formally,

$$\mathfrak{M} = \mathfrak{M}_1 + \mathfrak{M}_2 \overset{\text{def}}{\Leftrightarrow} R_1 \uplus R_2 = R, W_1 = W_2 = W, V_1 = V_2 = V.$$

This is the composition used in separation logic [21, 42]. The figure below depicts possible instances for $\mathfrak{M}, \mathfrak{M}_1$ and $\mathfrak{M}_2$.
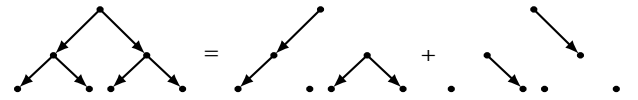
The ambient logic composition. We introduce the operator $+_w$, where $w \in W$, that constraints further +:

$$\mathfrak{M} = \mathfrak{M}_1 +_w \mathfrak{M}_2 \overset{\text{def}}{\Leftrightarrow} \mathfrak{M} = \mathfrak{M}_1 + \mathfrak{M}_2 \text{ and } R_i^+(w') = R^+(w')$$
$$\text{holds for all } i \in \{1, 2\} \text{ and } w' \in R_i(w).$$

$\mathfrak{M}$ is a disjoint union between $\mathfrak{M}_1$ and $\mathfrak{M}_2$ except that, as soon as $w' \in R_i(w)$, the whole subtree of w' in R belongs

to $\mathfrak{M}_i$, like the composition in ambient logic [12]. Below, we illustrate a model decomposed with $+_w$.
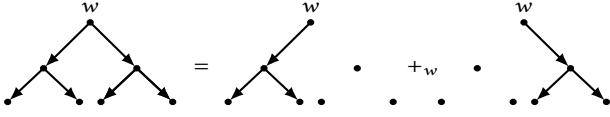
We say that $\mathfrak{M}_1$ is a *submodel* of $\mathfrak{M}$, written $\mathfrak{M}_1 \sqsubseteq \mathfrak{M}$ if there is $\mathfrak{M}_2$ such that $\mathfrak{M} = \mathfrak{M}_1 + \mathfrak{M}_2$.

Modal logics on trees. The logic ML($|$) enriches the modal logic K (a.k.a. ML) with a binary connective $|$, called *composition operator*, that admits submodel reasoning via the operator $+_w$. Similarly, ML($*$) enriches ML with the connective $*$, called *separating conjunction* (or *star*) that admits submodel reasoning via the operator $+$. Both connectives $|$ and $*$ are understood as binary modalities. As we show throughout the paper, ML($|$) and ML($*$) are strongly related to the graded modal logic GML [20]. For conciseness, let us define all these logics by considering formulae that contain all of their ingredients. These formulae are built from

$$\varphi := \top \mid p \mid \varphi \wedge \varphi \mid \neg \varphi \mid \Diamond \varphi \mid \Diamond_{\geq k} \varphi \mid \varphi * \varphi \mid \varphi | \varphi,$$

where $p \in \mathrm{AP}$ and $k \in \mathbb{N}$ (encoded in binary). A *pointed forest* $(\mathfrak{M}, w)$ is a finite forest $\mathfrak{M} = (W, R, V)$ together with a world $w \in W$. The satisfaction relation $\models$ is defined as follows (standard clauses for $\wedge$, $\neg$ and $\top$ are omitted):

$\mathfrak{M}, w \models p \qquad \Leftrightarrow w \in V(p)$;

$\mathfrak{M}, w \models \Diamond \varphi \qquad \Leftrightarrow$ there is $w' \in R(w)$ s.t. $\mathfrak{M}, w' \models \varphi$;

$\mathfrak{M}, w \models \Diamond_{\geq k} \varphi \Leftrightarrow |\{w' \in R(w) \mid \mathfrak{M}, w' \models \varphi\}| \geq k$;

$\mathfrak{M}, w \models \varphi_1 * \varphi_2 \Leftrightarrow$ there are $\mathfrak{M}_1, \mathfrak{M}_2$ s.t. $\mathfrak{M} = \mathfrak{M}_1 + \mathfrak{M}_2$, $\mathfrak{M}_1, w \models \varphi_1$ and $\mathfrak{M}_2, w \models \varphi_2$;

$\mathfrak{M}, w \models \varphi_1 | \varphi_2 \Leftrightarrow$ there are $\mathfrak{M}_1, \mathfrak{M}_2$ s.t. $\mathfrak{M} = \mathfrak{M}_1 +_w \mathfrak{M}_2$, $\mathfrak{M}_1, w \models \varphi_1$ and $\mathfrak{M}_2, w \models \varphi_2$.

The formulae $\varphi \Rightarrow \psi$, $\varphi \vee \psi$ and $\bot$ are defined as usual. We use the following standard abbreviations: $\Box \varphi \stackrel{\text{def}}{=} \neg \Diamond \neg \varphi$, $\Diamond_{\leq k} \varphi \stackrel{\text{def}}{=} \neg \Diamond_{\geq k+1} \varphi$ and $\Diamond_{=k} \varphi \stackrel{\text{def}}{=} \Diamond_{\geq k} \varphi \wedge \Diamond_{\leq k} \varphi$. We write size$(\varphi)$ to denote the *size* of φ with a tree representation of formulae and with a reasonably succinct encoding of atomic formulae. Besides, we write md(φ) to denote the *modal degree* of φ understood as the maximal number of nested unary modalities (i.e. $\Diamond$ or $\Diamond_{\geq k}$) in φ. Similarly, the *graded rank* gr(φ) of φ is defined as max$(\{k \mid \Diamond_{\geq k} \psi \in \mathrm{subf}(\varphi)\} \cup \{0\})$, where subf$(\varphi)$ is the set of all the subformulae of φ.

Given the formulae φ and ψ, $\varphi \equiv \psi$ denotes that φ and ψ are *logically equivalent*; i.e., for every pointed forest $(\mathfrak{M}, w)$, $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}, w \models \psi$. For instance ($k \geq 1$ and $p \in \mathrm{AP}$):

1. $\Diamond \varphi \equiv \Diamond_{\geq 1} \varphi$;   2. $(\Box\Box\bot | \Box\Box\bot) \not\equiv (\Box\Box\bot * \Box\Box\bot)$;

3. $\Diamond_{\geq k}\, p \equiv \underbrace{\Diamond p * \cdots * \Diamond p}_{k \text{ times}}$;   4. $\Diamond_{\geq k}\, \varphi \equiv \underbrace{\Diamond\varphi | \cdots | \Diamond\varphi}_{k \text{ times}}$.

The modal logic ML is the logic restricted to formulae with the unique modality $\Diamond$ [8]. Similarly, the graded modal logic GML is restricted to the *graded modalities* $\Diamond_{\geq k}$ [20]. We

introduce the modal logics ML($|$) and ML($*$), which are restricted to the suites of modalities $(\Diamond, |)$ and $(\Diamond, *)$, respectively. The two equivalences (3) and (4) already shed some light on ML($|$) and ML($*$): the two logics are similar when it comes to their formulae of modal degree one.

Lemma 2.1. *Let φ be a formula in* ML($|$) *with* md$(\varphi) \leq 1$. *Then,* $\varphi \equiv \varphi[\,|\leftarrow *\,]$ *where* $\varphi[\,|\leftarrow *\,]$ *is the formula in* ML($*$) *obtained from φ by replacing every occurrence of $|$ by $*$.*

However, as shown by the non-equivalence (2), it is unclear how the two logics compare when it comes to formulae of modal degree greater than one. Indeed, since $\mathfrak{M} = \mathfrak{M}_1 +_w \mathfrak{M}_2$ implies $\mathfrak{M} = \mathfrak{M}_1 + \mathfrak{M}_2$, but not vice-versa, the separating conjunction $*$ is more permissive than the operator $|$. However, further connections between the two operators can be easily established. Let us introduce the auxiliary operator $\blacklozenge$ defined as $\blacklozenge\varphi \stackrel{\text{def}}{=} \varphi * \Box\bot$. Formally,

$$(W, R, V), w \models \blacklozenge\varphi \Leftrightarrow \text{ there is } R' \subseteq R \text{ s.t. } R'(w) = R(w)$$
$$\text{and } (W, R', V), w \models \varphi.$$

Similar operators are studied in [2, 4, 11]. We show that $\blacklozenge$ and $|$ are sufficient to capture $*$ (essential property for Section 5).

Lemma 2.2. *Let $\varphi, \psi \in$ GML. We have $\varphi * \psi \equiv \blacklozenge(\varphi | \psi)$.*

Unlike $|$, when $*$ splits a finite forest $\mathfrak{M}$ into $\mathfrak{M}_1$ and $\mathfrak{M}_2$, it may disconnect in both submodels worlds that are otherwise reachable, from the current world, in $\mathfrak{M}$. Applying $\blacklozenge$ before $|$ allows us to imitate this behaviour. Indeed, even though $|$ preserves reachability in either $\mathfrak{M}_1$ or $\mathfrak{M}_2$, $\blacklozenge$ deletes part of $\mathfrak{M}$, making some world inaccessible. This way of expressing the separating conjunction allows us to reuse some methods developed for ML($|$) in order to study ML($*$).

The logic QK^t. Both ML($|$) and ML($*$) can be seen as fragments of the logic QK^t, which in turn is known to be a fragment of monadic second-order logic on trees [7]. The logic QK^t extends ML with second-order quantification and is interpreted on finite trees. Its formulae are defined according to the following grammar:

$$\varphi := p \mid \Diamond\varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists p\, \varphi.$$

Given $\mathfrak{M} = (W, R, V)$ and $w \in W$, the satisfaction relation $\models$ of ML is extended as follows:

$\mathfrak{M}, w \models \exists p\, \varphi$ iff $\exists W' \subseteq W$ s.t. $(W, R, V[p \leftarrow W']), w \models \varphi$.

One can show logspace reductions from ML($|$) and ML($*$) to QK^t, by simply reinterpreting the operators $*$ and $|$ as restrictive forms of second-order quantification, and by relativising $\Diamond$ to appropriate propositional symbols in order to capture the notion of submodel (details are omitted).

Satisfiability problem. The *satisfiability problem* for a logic $\mathfrak{L}$, written Sat$(\mathfrak{L})$, takes as input a formula φ in $\mathfrak{L}$ and checks whether there is a pointed forest $(\mathfrak{M}, w)$ such that $\mathfrak{M}, w \models \varphi$.

Note that any $\mathfrak{L}$ among ML, GML, ML($|$) or ML($*$) has the tree model property, i.e. any satisfiable formula is also satisfied in some tree structure. The problems Sat(ML) and

Sat(GML) are known to be PSpace-complete, see e.g. [8, 29, 44, 45], and therefore Sat(ML($\mid$)) and Sat(ML($*$)) are PSpace-hard. As an upper bound, by Rabin's theorem [40], the satisfiability problem for QK^t is decidable in Tower, which transfers directly to Sat(ML($\mid$)) and Sat(ML($*$)).

Expressive power. Given two logics $\mathfrak{L}_1$ and $\mathfrak{L}_2$, we say that $\mathfrak{L}_2$ is *at least as expressive as* $\mathfrak{L}_1$ (written $\mathfrak{L}_1 \preceq \mathfrak{L}_2$) whenever for every formula φ of $\mathfrak{L}_1$, there is a formula ψ of $\mathfrak{L}_2$ such that $\varphi \equiv \psi$. $\mathfrak{L}_1 \approx \mathfrak{L}_2$ denotes that $\mathfrak{L}_1$ and $\mathfrak{L}_2$ are *equally expressive*, i.e. $\mathfrak{L}_1 \preceq \mathfrak{L}_2$ and $\mathfrak{L}_2 \preceq \mathfrak{L}_1$. Lastly, $\mathfrak{L}_1 \prec \mathfrak{L}_2$ denotes that $\mathfrak{L}_2$ is *strictly more expressive* than $\mathfrak{L}_1$, i.e. $\mathfrak{L}_1 \preceq \mathfrak{L}_2$ and $\mathfrak{L}_1 \not\approx \mathfrak{L}_2$. The equivalence (1) recalls us that ML $\prec$ GML [20]. From the equivalence (4), we get GML $\preceq$ ML($\mid$).

3 ML($\mid$): Expressiveness and Complexity

In this section, we study the expressive power of ML($\mid$) and the complexity of Sat(ML($\mid$)). We show constructively that ML($\mid$) $\preceq$ GML, hence proving ML($\mid$) $\approx$ GML. Next, we show that Sat(ML($\mid$)) is AExp$_{\text{Pol}}$-complete. The upper bound is achieved by proving an exponential-size model property. The lower bound is by reduction from the satisfiability problem for propositional team logic [27, Thm. 4.9].

3.1 ML($\mid$) is not more expressive than GML

Establishing ML($\mid$) $\preceq$ GML amounts to show that given φ_1, φ_2 in GML, one can construct ψ in GML such that $\varphi_1 \mid \varphi_2 \equiv \psi$. For instance, a simple case analysis yields the equivalence $(p \vee \Diamond_{\geq 3} r) \mid (q \vee \Diamond_{\leq 5} q) \equiv (p \vee \Diamond_{\geq 3} r)$. With this property, the general algorithm consists in iteratively replacing innermost subformulae of the form $\varphi_1 \mid \varphi_2$ by a counterpart in GML, allowing us to eliminate all the occurrences of $\mid$ and obtain an equivalent formula in GML. The base case involves subformulae φ_1 and φ_2 in ML (a fragment of GML).

Let us provide a few definitions. Let φ be a formula in GML. We write $\mathsf{max}_{\mathsf{PC}}(\varphi)$ to denote the set of subformulae ψ of φ that are maximal and modality-free, i.e.

1. ψ is modality-free: it does not contain modalities $\Diamond_{\geq k}$ and one of its occurrences is not in the scope of $\Diamond_{\geq k}$;
2. ψ is maximal: one of its occurrences does not belong to a larger modality-free subformula of φ.

For instance, $\mathsf{max}_{\mathsf{PC}}((p \vee \Diamond_{\geq 3} r) \wedge (q \vee p)) = \{p, q \vee p\}$. Similarly, $\mathsf{max}_{\mathsf{GM}}(\varphi)$ denotes the set of subformulae ψ of φ such that ψ is of the form $\Diamond_{\geq k} \psi'$ and one of its occurrences in φ is not in the scope of graded modalities $\Diamond_{\geq k}$. For instance,

$\mathsf{max}_{\mathsf{GM}}((p \vee \Diamond_{\geq 3} r) \wedge (q \vee \Diamond_{\geq 5} \Diamond_{\geq 2} q)) = \{\Diamond_{\geq 3} r, \Diamond_{\geq 5} \Diamond_{\geq 2} q\}$.

Every formula φ in GML is a Boolean combination of formulae from $\mathsf{max}_{\mathsf{PC}}(\varphi) \cup \mathsf{max}_{\mathsf{GM}}(\varphi)$. Lastly, φ is in *good shape* if the properties (1) and (2) below hold:

1. $\mathsf{max}_{\mathsf{PC}}(\varphi) \subseteq \{\bot, \top\}$. Consequently, every propositional variable in φ occurs in the scope of a graded modality;
2. For all $\Diamond_{\geq k} \psi, \Diamond_{\geq k'} \psi'$ in $\mathsf{max}_{\mathsf{GM}}(\varphi)$ with $\psi \neq \psi'$, the conjunction $\psi \wedge \psi'$ is unsatisfiable.

Let φ_1 and φ_2 be GML formulae. First, we show that when $\varphi_1 \wedge \varphi_2$ is in good shape, there is a GML formula ψ such that $\varphi_1 \mid \varphi_2 \equiv \psi$. To do so, we take a slight detour through Presburger arithmetic (PA), see e.g. [26, 38]. Given two formulae φ_1, φ_2 in GML, we will characterise the formula $\varphi_1 \mid \varphi_2$ by using arithmetical constraints for the number of successors. Then, we will take advantage of basic properties of PA in order to eliminate quantifiers, and obtain a GML formula. Below, the variables x, y, z, . . ., possibly decorated and occurring in formulae, are from PA and therefore they are interpreted by natural numbers.

Let φ be in GML s.t. $\mathsf{max}_{\mathsf{PC}}(\varphi) \subseteq \{\top, \bot\}$ and $\{\psi_1, \ldots, \psi_n\}$ contains the set $\{\psi \mid \Diamond_{\geq k} \psi \in \mathsf{max}_{\mathsf{GM}}(\varphi)\}$. We define formulae in PA that state constraints about the number of children satisfying a formula ψ_j. The variable x_j is intended to be interpreted as the number of children satisfying ψ_j. We write $\varphi^{\mathsf{PA}}(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ to denote the arithmetical formula obtained from φ by replacing with $\mathsf{x}_j \geq k$ every occurrence of $\Diamond_{\geq k} \psi_j$ that it is not in the scope of a graded modality. For instance, assuming that $\varphi = \Diamond_{\geq 5} (p \wedge q) \vee \neg \Diamond_{\geq 4} \neg p$, the expression $\varphi^{\mathsf{PA}}(\mathsf{x}_1, \mathsf{x}_2)$ denotes the formula $\mathsf{x}_1 \geq 5 \vee \neg(\mathsf{x}_2 \geq 4)$.

Let φ_1, φ_2 be GML formulae such that $\varphi_1 \wedge \varphi_2$ is in good shape and $\{\psi_1, \ldots, \psi_n\} = \{\psi \mid \Diamond_{\geq k} \psi \in \mathsf{max}_{\mathsf{GM}}(\varphi_1 \wedge \varphi_2)\}$. We consider the formula $[\varphi_1, \varphi_2]^{\mathsf{PA}}$ in PA defined below:

$$[\varphi_1, \varphi_2]^{\mathsf{PA}} \stackrel{\text{def}}{=} \exists \mathsf{y}_1^1, \mathsf{y}_1^2, \ldots, \mathsf{y}_n^1, \mathsf{y}_n^2 \ (\textstyle\bigwedge_{j=1}^n \mathsf{x}_j = \mathsf{y}_j^1 + \mathsf{y}_j^2) \wedge$$
$$\varphi_1^{\mathsf{PA}}(\mathsf{y}_1^1, \ldots, \mathsf{y}_n^1) \wedge \varphi_2^{\mathsf{PA}}(\mathsf{y}_1^2, \ldots, \mathsf{y}_n^2).$$

The formula $[\varphi_1, \varphi_2]^{\mathsf{PA}}$ states that there is a way to divide the children in two distinct sets and each set allows to satisfy φ_1^{PA} or φ_2^{PA}, respectively. As PA admits quantifier elimination [16, 38, 41], there is a quantifier-free formula χ equivalent to $[\varphi_1, \varphi_2]^{\mathsf{PA}}$ and its free variables are among $\mathsf{x}_1, \ldots, \mathsf{x}_n$. A priori, the atomic formulae of χ may not be of the simple form $\mathsf{x}_j \geq k$ (e.g. 'modulo constraints' or constraints of the form $\sum a_i \mathsf{x}_j \geq k$ may be involved). However, if the atomic formulae of χ are restricted to expressions of the form $\mathsf{x}_j \geq k$, then we write χ^{GML} to denote the GML formula obtained from χ by replacing every occurrence of $\mathsf{x}_j \geq k$ by $\Diamond_{\geq k} \psi_j$.

Lemma 3.1. *Let* φ_1, φ_2 *be in* GML *such that* $\varphi_1 \wedge \varphi_2$ *is in good shape.* $[\varphi_1, \varphi_2]^{\mathsf{PA}}$ *is equivalent to a quantifier-free* PA *formula* χ *whose atomic formulae are only of the form* $\mathsf{x}_j \geq k$. *Moreover,* $\varphi_1 \mid \varphi_2 \equiv \chi^{\mathsf{GML}}$ *and* $\mathsf{gr}(\chi^{\mathsf{GML}}) \leq \mathsf{gr}(\varphi_1) + \mathsf{gr}(\varphi_2)$.

The bound on $\mathsf{gr}(\chi^{\mathsf{GML}})$ stated in this key lemma is essential to obtain an exponential bound on the smallest model satisfying a formula in ML($\mid$) (see Section 3.2). Thanks to Lemma 3.1, we can show that GML is closed under the operator $\mid$ by reducing the occurrences of this operator to formulae in good shape. In particular, we show that given two arbitrary formulae φ_1 and φ_2 in GML, $\varphi_1 \mid \varphi_2$ is equivalent to a disjunction of formulae of the form $(\psi_1 \mid \psi_2) \wedge \chi$, where χ is a Boolean combination of atomic propositions and $\psi_1 \wedge \psi_2$ is in good shape (hence $\psi_1 \mid \psi_2$ is equivalent to a formula in GML

by Lemma 3.1). This is shown syntactically: atomic propositions are dealt with by propositional reasoning, whereas to produce ψ_1 and ψ_2 we use axioms from GML [6] and rely on the following equivalences:

(guess) $\diamondsuit_{\geq k} \varphi \equiv \diamondsuit_{\geq k} ((\varphi \wedge \psi) \vee (\varphi \wedge \neg\psi))$;

($\diamondsuit_{\geq k}$ dist) if $\varphi \wedge \psi$ unsat., $\diamondsuit_{\geq k}(\varphi \vee \psi) \equiv \bigvee_{k=k_1+k_2}(\diamondsuit_{\geq k_1}\varphi \wedge \diamondsuit_{\geq k_2}\psi)$;

(| dist) $(\varphi \vee \psi) | \chi \equiv (\varphi|\chi) \vee (\psi|\chi)$.

Notice that the conjunction of $\varphi \wedge \psi$ and $\varphi \wedge \neg\psi$ from (guess) is trivially unsatisfiable, allowing us to use ($\diamondsuit_{\geq k}$ dist). As GML is shown to be closed under the operator |, we conclude.

Theorem 3.2. $\mathsf{ML}(\,|\,) \leq \mathsf{GML}$. Therefore, $\mathsf{ML}(\,|\,) \approx \mathsf{GML}$.

To prove $\mathsf{ML}(\,|\,) \leq \mathsf{GML}$, we iteratively put subformulae in good shape and apply Lemma 3.1. This is done several times, potentially causing an exponential blow-up each time a formula is transformed. To provide an optimal complexity upper bound, we need to tame this combinatorial explosion.

3.2 $\mathsf{AExp_{Pol}}$-completeness

In order to show that $\mathsf{Sat}(\mathsf{ML}(\,|\,))$ is in $\mathsf{AExp_{Pol}}$, the main ingredient is to show that given φ in $\mathsf{ML}(\,|\,)$, we build φ' in GML such that $\varphi' \equiv \varphi$ and the models for φ' (if any) do not require a number of children per node more than exponential in $\mathsf{size}(\varphi)$. The proof of Theorem 3.2 needs to be refined to improve the way φ' is computed. In particular, this requires a strategy for the application of the equivalences used to put a formula in good shape.

We need to introduce a few more simple notions. Let φ be a GML formula with $\mathsf{max_{GM}}(\varphi) = \{\diamondsuit_{\geq k_1}\psi_1, \ldots, \diamondsuit_{\geq k_n}\psi_n\}$. We define $\mathsf{bd}(0, \varphi) \stackrel{\text{def}}{=} k_1 + \cdots + k_n$. For all $m \geq 0$, we define $\mathsf{bd}(m+1, \varphi) \stackrel{\text{def}}{=} \max\{\mathsf{bd}(m, \psi) \mid \diamondsuit_{\geq k}\psi \in \mathsf{max_{GM}}(\varphi)\}$. Hence, $\mathsf{bd}(m, \varphi)$ can be understood as the maximal $\mathsf{bd}(0, \psi)$ for some subformula ψ occurring at the modal depth m within φ. We write $\mathsf{max_{bd}}(\varphi)$ for the value $\max\{\mathsf{bd}(m, \varphi) \mid m \in [0, \mathsf{md}(\varphi)]\}$. If φ is satisfiable, we can use $\mathsf{max_{bd}}(\varphi)$ to obtain a bound on the smallest model satisfying it, as stated in Lemma 3.3 below.

Lemma 3.3. Every satisfiable φ in GML is satisfied by a pointed forest with at most $\mathsf{max_{bd}}(\varphi)^{\mathsf{md}(\varphi)+1}$ worlds.

To show that $\mathsf{ML}(\,|\,)$ has the exponential-size model property, we establish that given φ in $\mathsf{ML}(\,|\,)$, there is φ' in GML such that $\varphi' \equiv \varphi$, $\mathsf{md}(\varphi') \leq \mathsf{md}(\varphi)$ and $\mathsf{max_{bd}}(\varphi')$ is exponential in $\mathsf{size}(\varphi)$. First, we consider the fragment F of $\mathsf{ML}(\,|\,)$: $\varphi ::= \diamondsuit_{\geq k} \psi \mid p \mid \varphi|\varphi \mid \varphi \wedge \varphi \mid \neg\varphi$, where $p \in \mathsf{AP}$ and $\diamondsuit_{\geq k} \psi$ is a formula in GML (abusively assumed in $\mathsf{ML}(\,|\,)$ but we know $\mathsf{GML} \leq \mathsf{ML}(\,|\,)$). Given φ in $\mathsf{ML}(\,|\,)$ or in F, we write $\mathsf{cd}(\varphi)$ to denote its *composition degree*, i.e. the maximal number of imbrications of | in φ. We extend the notion of bd to formulae in F, so that $\mathsf{bd}(m, \varphi) = \mathsf{bd}(m, \varphi[\,|\leftarrow \wedge])$, where $\varphi[\,|\leftarrow \wedge]$ is the formula obtained from φ by replacing every occurrence of | by $\wedge$. Similarly, $\mathsf{max_{GM}}(\varphi) \stackrel{\text{def}}{=} \mathsf{max_{GM}}(\varphi[\,|\leftarrow \wedge])$.

Let φ be in F such that $\mathsf{max_{GM}}(\varphi) = \{\diamondsuit_{\geq k_1}\chi_1, \ldots, \diamondsuit_{\geq k_n}\chi_n\}$. The key step to show the exponential-size model property essentially manipulates the formulae in $\mathsf{max_{GM}}(\varphi)$ in order

to produce equivalent formulae $\psi_1, \ldots, \psi_n$, so that for all distinct i and j, $\psi_i \wedge \psi_j$ is in good shape. Moreover, by replacing in φ every $\diamondsuit_{\geq k_i} \chi_i$ with the equivalent formula ψ_i, we only witness an exponential blow-up on $\mathsf{bd}(0, \varphi)$, whereas for every $m > 1$, $\mathsf{bd}(m, \varphi)$ remains polynomially bounded by the bd of the original formula. With the bound on the graded rank found in Lemma 3.1, we derive Lemma 3.4.

Lemma 3.4. Let φ be a formula of the fragment F such that $\mathsf{max_{GM}}(\varphi) = \{\diamondsuit_{\geq k_1}\chi_1, \ldots, \diamondsuit_{\geq k_n}\chi_n\}$ and $\widehat{k} = \max\{k_1, \ldots, k_n\}$. There is a GML formula ψ such that $\varphi \equiv \psi$ and,

1. $\mathsf{md}(\psi) \leq \mathsf{md}(\varphi)$;   2. $\mathsf{bd}(0, \psi) \leq \widehat{k} \times 2^{n+\mathsf{cd}(\varphi)}$;
3. $\mathsf{bd}(1, \psi) \leq n \times \mathsf{bd}(1, \varphi)$;   4. $\forall m \geq 2, \mathsf{bd}(m, \psi) = \mathsf{bd}(m, \varphi)$.

In the proof of Lemma 3.4, a first step essentially consists in applying multiple times (guess) in order to derive, for every $i \in [1, n]$, an equivalence $\diamondsuit_{\geq k_i} \chi_i \equiv \psi_i'$ where

$$\psi_i' \stackrel{\text{def}}{=} \diamondsuit_{\geq k_i} \bigvee_{\mathfrak{f}:[1,n]\to\{\top,\bot\}} \left(\chi_i \wedge [\chi_1]^{\mathfrak{f}(1)} \wedge \cdots \wedge [\chi_n]^{\mathfrak{f}(n)}\right).$$

Here, $[\chi_j]^\top \stackrel{\text{def}}{=} \chi_j$ and $[\chi_j]^\bot \stackrel{\text{def}}{=} \neg\chi_j$. Roughly speaking, in this step, we expand χ_i by considering all the possible truth values for the formulae $\chi_1, \ldots, \chi_n$ (the disjuncts where χ_i is negated can be simply discharged from the disjunction, as they are unsatisfiable). Substituting every $\diamondsuit_{\geq k_i} \chi_i$ by ψ_i' in φ leads to a formula φ' such that $\mathsf{bd}(1, \varphi') \leq n \times \mathsf{bd}(1, \varphi)$ (as in Lemma 3.4) and for every $m \neq 1$, $\mathsf{bd}(m, \varphi') = \mathsf{bd}(m, \varphi)$. Afterwards, we repeatedly apply ($\diamondsuit_{\geq k}$ dist) to ψ_i' and obtain the formula ψ_i satisfying the aforementioned property, i.e. for all distinct i and j, $\psi_i \wedge \psi_j$ is in good shape. With (| dist), this allows us to apply Lemma 3.1 until all the operators | are removed. Besides, replacing every ψ_i' by ψ_i in φ' leads to a formula having the same bd as the formula ψ in Lemma 3.4.

Applying adequately the transformation from Lemma 3.4 to a formula in $\mathsf{ML}(\,|\,)$, i.e. by considering maximal subformulae of the fragment F, allows us to get a logically equivalent GML formula having small models.

Lemma 3.5. Every satisfiable φ in $\mathsf{ML}(\,|\,)$ is satisfied by a pointed forest of size at most exponential in $\mathsf{size}(\varphi)$.

The proof of Lemma 3.5 (relying on Lemma 3.4) consists in showing that for all φ in $\mathsf{ML}(\,|\,)$, there is φ' in GML such that $\varphi' \equiv \varphi$ and $\mathsf{max_{bd}}(\varphi')$ is exponential in $\mathsf{size}(\varphi)$, which is sufficient by Lemma 3.3 to get the exponential-size model property, whence the upper bound $\mathsf{AExp_{Pol}}$.

Theorem 3.6. $\mathsf{Sat}(\mathsf{ML}(\,|\,))$ is in $\mathsf{AExp_{Pol}}$.

The (standard) proof consists in observing that to check the satisfiability status of φ in $\mathsf{ML}(\,|\,)$, first guess a pointed forest of exponential-size (thanks to Lemma 3.5) and check whether it satisfies φ. This can be done in exponential-time using an alternating Turing machine with a linear amount of alternations (between universal states and existential states) by viewing $\mathsf{ML}(\,|\,)$ as a fragment of MSO.

It remains to establish $\mathsf{AExp_{Pol}}$-hardness. We provide a logspace reduction from the satisfiability problem for the team logic PL[~] shown $\mathsf{AExp_{Pol}}$-complete in [27, Thm. 4.9].

PL[~] formulae are defined by the following grammar:
$$\varphi := p \mid \dot{\neg} p \mid \varphi \wedge \varphi \mid {\sim}\varphi \mid \varphi \dot{\vee} \varphi,$$
where $p \in$ AP and the connectives $\dot{\neg}$ and $\dot{\vee}$ are dotted to avoid confusion with those of ML($|$). PL[~] is interpreted on sets of (Boolean) propositional valuations over a finite subset of AP. They are called *teams* and are denoted by $\mathfrak{T}, \mathfrak{T}_1, \ldots$. A model for φ is a team $\mathfrak{T}$ over a set of propositional variables including those occurring in φ and such that $\mathfrak{T} \models \varphi$ with:

$\mathfrak{T} \models p \qquad \Leftrightarrow$ for all $\mathfrak{v} \in \mathfrak{T}$, we have $\mathfrak{v}(p) = \top$;

$\mathfrak{T} \models \dot{\neg} p \qquad \Leftrightarrow$ for all $\mathfrak{v} \in \mathfrak{T}$, we have $\mathfrak{v}(p) = \bot$;

$\mathfrak{T} \models \varphi_1 \dot{\vee} \varphi_2 \Leftrightarrow \exists \mathfrak{T}_1, \mathfrak{T}_2$ s.t. $\mathfrak{T} = \mathfrak{T}_1 \cup \mathfrak{T}_2$, $\mathfrak{T}_1 \models \varphi_1$, $\mathfrak{T}_2 \models \varphi_2$.

The connectives ~ and $\wedge$ are interpreted as the classical negation and conjunction, respectively. Notice that, in the clause for $\dot{\vee}$, the teams $\mathfrak{T}_1$ and $\mathfrak{T}_2$ are not necessarily disjoint.

Let us discuss the reduction from Sat(PL[~]) to Sat(ML($|$)). A direct encoding of a team $\mathfrak{T}$ into a pointed forest $(\mathfrak{M}, w)$ consists in having a correspondence between the propositional valuations in $\mathfrak{T}$ and the propositional valuations of the children of w. This would work fine if there were no mismatch between the semantics for $|$ (disjointness of the children) and the one for $\dot{\vee}$ (disjointness not required). To handle this, when checking the satisfaction of φ in PL[~] with n occurrences of $\dot{\vee}$, we impose that if a propositional valuation occurs among the children of w, then it occurs in least $n + 1$ children. This property must be maintained after applying $\dot{\vee}$ several times, always with respect to the number of occurrences of $\dot{\vee}$ in the subformula of φ that is evaluated. Non-disjointness of the teams is encoded by carefully separating the children of w having identical valuations.

We now formalise the reduction. Assume that we wish to translate φ from PL[~], written with atomic propositions in $P = \{p_1, \ldots, p_m\}$ and containing at most n occurrences of the operator $\dot{\vee}$. We introduce a set $Q = \{q_1, \ldots, q_{n+1}\}$ of auxiliary propositions disjoint from P. The elements of Q are used to distinguish different copies of the same propositional valuation of a team. Thus, with respect to a pointed forest $(\mathfrak{M}, w)$, we require each child of w to satisfy exactly one element of Q. This can be done with the formula
$$\mathsf{uni}(Q) \stackrel{\text{def}}{=} \Box(\bigwedge_{i \neq i' \in [1, n+1]} \neg(q_i \wedge q_{i'}) \wedge \bigvee_{i \in [1, n+1]} q_i).$$
We require that if a child of w satisfies a propositional valuation over (elements in) P, then there are $n + 1$ children satisfying that valuation over P, each of them satisfying a distinct symbol in Q. So, every valuation over P occurring in some child of w, occurs at least in $n + 1$ children of w. However, as the translation of the operator $\dot{\vee}$ modifies the set of copies of a propositional valuation, this property must be extended to arbitrary subsets of Q. Given $\emptyset \neq X \subseteq [1, n+1]$, we require that for all $k \neq k' \in X$, if a children of w satisfies q_k, then there is a child satisfying $q_{k'}$ with the same valuation over P. The formula $\mathsf{cp}(X)$ below does the job:
$$\bigwedge_{k \neq k' \in X} \neg(\Box q_k | (\Diamond_{=1} q_k \wedge \neg(\top | \Diamond_{=1} q_k \wedge \Diamond_{=1} q_{k'} \wedge \bigwedge_{j \in [1, m]} \Diamond p_j \Rightarrow \Box p_j)))).$$

Lastly, before defining the translation map τ, we describe how different copies of the same propositional valuation are split. We introduce two auxiliary choice functions $\mathfrak{c}_1$ and $\mathfrak{c}_2$ that take as arguments $X \subseteq [1, n+1]$, and $n_1, n_2 \in \mathbb{N}$ with $|X| \geq n_1 + n_2$ such that for each $i \in \{1, 2\}$, we have $\mathfrak{c}_i(X, n_1, n_2) \subseteq X, |\mathfrak{c}_i(X, n_1, n_2)| \geq n_i$. Moreover $\mathfrak{c}_1(X, n_1, n_2) \uplus \mathfrak{c}_2(X, n_1, n_2) = X$. The maps $\mathfrak{c}_1$ and $\mathfrak{c}_2$ are instrumental to decide how to split X into two disjoint subsets respecting basic cardinality constraints. The translation map τ is designed as follows ($\emptyset \neq X \subseteq [1, n+1]$):
$$\tau(p, X) \stackrel{\text{def}}{=} \Box((\textstyle\bigvee_{j \in X} q_j) \Rightarrow p);$$
$$\tau(\dot{\neg} p, X) \stackrel{\text{def}}{=} \Box((\textstyle\bigvee_{j \in X} q_j) \Rightarrow \neg p);$$
$$\tau(\varphi_1 \wedge \varphi_2, X) \stackrel{\text{def}}{=} \tau(\varphi_1, X) \wedge \tau(\varphi_2, X); \quad \tau({\sim}\varphi, X) \stackrel{\text{def}}{=} \neg\tau(\varphi, X);$$
$$\tau(\varphi_1 \dot{\vee} \varphi_2, X) \stackrel{\text{def}}{=} (\tau(\varphi_1, X_1) \wedge \mathsf{cp}(X_1)) | (\tau(\varphi_2, X_2) \wedge \mathsf{cp}(X_2)),$$
where **(i)** $|X|$ is greater or equal to the number of occurrences of $\dot{\vee}$ in $\varphi_1 \dot{\vee} \varphi_2$ plus one; **(ii)** given n_1, n_2 such that n_1 (resp. n_2) is the number of occurrences of $\dot{\vee}$ in φ_1 (resp. φ_2) plus one, for each $i \in \{1, 2\}$ we have $\mathfrak{c}_i(X, n_1, n_2) = X_i$.

Lemma 3.7 below guarantees that starting with a linear number of children with the same propositional valuation is sufficient to encode $\dot{\vee}$ within ML($|$).

Lemma 3.7. *Let φ be in PL[~] with n occurrences of $\dot{\vee}$ and built upon $p_1, \ldots, p_m$. Then, φ is satisfiable iff so is*
$$\mathsf{uni}(q_1, \ldots, q_{n+1}) \wedge \mathsf{cp}([1, n+1]) \wedge \tau(\varphi, [1, n+1]).$$

The ML($|$) formula involved in Lemma 3.7 has modal depth one. By Theorem 3.6, Sat(ML($|$)) is AExp_{Pol}-complete even restricted to formulae of modal depth at most one.

Corollary 3.8. Sat(ML($|$)) *is AExp_{Pol}-complete.*

As we show in the next section, the complexity of ML($*$) does not collapse to modal depth one: Sat(ML($*$)) restricted to formulae of modal depth k is exponentially easier than Sat(ML($*$)) restricted to formulae of modal depth $k + 1$.

4 ML($*$) is Tower-complete

We show that Sat(ML($*$)) is Tower-complete, i.e. complete for the class of all problems of time complexity bounded by a tower of exponentials whose height is an elementary function [43]. Given $k, n \geq 0$, we inductively define the tetration function t as $\mathsf{t}(0, n) \stackrel{\text{def}}{=} n$ and $\mathsf{t}(k + 1, n) = 2^{\mathsf{t}(k, n)}$. Intuitively, $\mathsf{t}(k, n)$ defines a tower of exponentials of height k. By k-NExpTime, we denote the class of all problems decidable with a nondeterministic Turing machine (NTM) of working time $O(\mathsf{t}(k, p(n)))$ for some polynomial $p(.)$, on each input of length n. To show Tower-hardness, we design a uniform elementary reduction allowing us to get k-NExpTime-hardness for all k greater than a certain (fixed) integer. In our case, we achieve an exponential-space reduction from the k-NExpTime variant of the tiling problem, for all $k \geq 2$.

The tiling problem $\texttt{Tile}_k$ takes as input a triple $\mathscr{T} = (\mathcal{T}, \mathcal{H}, \mathcal{V})$ where $\mathcal{T}$ is a finite set of tile types, $\mathcal{H} \subseteq \mathcal{T} \times \mathcal{T}$ (resp. $\mathcal{V} \subseteq \mathcal{T} \times \mathcal{T}$) represents the horizontal (resp. vertical)

matching relation, and an initial tile type $c \in \mathcal{T}$. A solution for the instance $(\mathcal{T}, c)$ is a mapping $\tau : [0, t(k, n) - 1] \times [0, t(k, n) - 1] \rightarrow \mathcal{T}$ such that **(first)** $\tau(0, 0) = c$, and

(hor&vert) for all $i \in [0, t(k, n) - 1]$ and $j \in [0, t(k, n) - 2]$,
$(\tau(j, i), \tau(j + 1, i)) \in \mathcal{H}$ and $(\tau(i, j), \tau(i, j + 1)) \in \mathcal{V}$.

The problem of checking whether an instance of Tile_k has a solution is known to be k-NExpTime-complete (see [36]).

The reduction below from Tile_k to Sat(ML($*$)) recycles ideas from [7] to reduce Tile_k to Sat(QK^t). To provide the adequate adaptation for ML($*$), we need to solve two major issues. First, QK^t admits second-order quantification, whereas in ML($*$), the second-order features are limited to the separating conjunction $*$. Second, the second-order quantification of QK^t essentially colours the nodes in Kripke-style structures without changing the frame (W, R). By contrast, the operator $*$ modifies the accessibility relation, possibly making worlds that were reachable from the current world, unreachable in submodels. The Tower-hardness proof for Sat(ML($*$)) becomes then much more challenging: we would like to characterise the position on the grid encoded by a world w by exploiting properties of its descendants (as done for QK^t), but at the same time, we need to be careful and only consider submodels where w keeps encoding the same position. In a sense, our encoding is robust: when the operator $*$ is used to reason on submodels, we can enforce that no world changes the position of the grid that it encodes.

4.1 Enforcing $t(j, n)$ children.

Let $\mathfrak{M} = (W, R, V)$ be a finite forest. We consider two disjoint sets of atomic propositions $\mathsf{P} = \{p_1, \ldots, p_n, \mathsf{val}\}$ and $\mathsf{Aux} = \{\mathsf{x}, \mathsf{y}, \mathsf{l}, \mathsf{s}, \mathsf{r}\}$ (whose respective role is later defined). Elements from Aux are understood as *auxiliary* propositions. We call ax-*node* (resp. Aux-*node*) a world satisfying the proposition $\mathsf{ax} \in \mathsf{Aux}$ (resp. satisfying some proposition in Aux). We call t-*node* a world that satisfies the formula $t \overset{\text{def}}{=} \bigwedge_{\mathsf{ax} \in \mathsf{Aux}} \neg \mathsf{ax}$. Every world of $\mathfrak{M}$ is either a t-node or an Aux-node. We say that w' is a t-*child* of $w \in W$ if $w' \in R(w)$ and w' is a t-node. We define the concepts of Aux-child and ax-child similarly.

The key development of our reduction is given by the definition of a formula, of exponential size in $j \geq 1$ and polynomial size in $n \geq 1$, that when satisfied by $(\mathfrak{M}, w)$ forces every t-node in $R^i(w)$, where $0 \leq i < j$, to have exactly $t(j - i, n)$ t-children, each of them encoding a different number in $[0, t(j - i, n) - 1]$. As we impose that w is a t-node, it must have $t(j, n)$ t-children. We assume n to be fixed throughout the section and denote this formula by $\text{type}(j)$. From the property above, if $\mathfrak{M}, w \models \text{type}(j)$ then for all $i \in [1, j-1]$ and all t-nodes $w' \in R^i(w)$ we have $\mathfrak{M}, w' \models \text{type}(j-i)$.

First, let us informally describe how numbers are encoded in the model $(\mathfrak{M}, w)$ satisfying $\text{type}(j)$. Let $i \in [1, j]$. Given a t-node $w' \in R^i(w)$, $\mathfrak{n}_i(w')$ denotes the number encoded by w'. We omit the subscript i when it is clear from the context. When $i = j$, we represent $\mathfrak{n}(w')$ by using the truth

values of the atomic propositions $p_1, \ldots, p_n$. The proposition p_b is responsible for the b-th bit of the number, with the least significant bit being encoded by p_1. For example, for $n = 3$, we have $\mathfrak{M}, w' \models p_3 \wedge p_2 \wedge \neg p_1$ whenever $\mathfrak{n}(w') = 6$. The formula $\text{type}(1)$ forces the parent of w' (i.e. is a t-node in $R^{j-1}(w)$) to have exactly 2^n t-children by requiring one t-child for each possible valuation upon $p_1, \ldots, p_n$. Otherwise, for $i < j$ (and therefore $j \geq 2$), the number $\mathfrak{n}_i(w')$ is represented by the binary encoding of the truth values of val on the t-children of w' which, since $(\mathfrak{M}, w') \models \text{type}(j - i)$, are $t(j - i, n)$ children implicitly ordered by the number they, in turn, encode. The essential property of $\text{type}(j)$ is therefore the following: the numbers encoded by the t-children of a t-node $w'' \in R^i(w)$, represent positions in the binary representation of the number $\mathfrak{n}_i(w'')$. Thanks to this property, the formula $\text{type}(j)$ forces w to have exactly $t(j, n)$ children, all encoding different numbers in $[0, t(j, n) - 1]$. This is roughly represented in the picture below, where "1" stands for val being true whereas "0" stands for val being false.
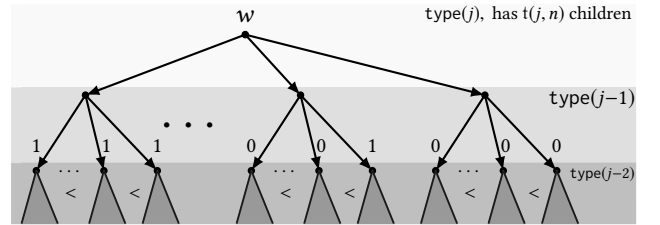
To characterise these trees in ML($*$), we simulate second-order quantification by using Aux-nodes. Informally, we require a pointed forest $(\mathfrak{M}, w)$ satisfying $\text{type}(j)$ to be such that *(i)* every t-node $w' \in R(w)$ has exactly one x-child, and one (different) y-child. These nodes do not satisfy any other auxiliary proposition; *(ii)* for every $i \geq 2$, every t-node $w' \in R^i(w)$ has exactly five Aux-children, one for each $\mathsf{ax} \in \mathsf{Aux}$. We can simulate second-order existential quantification on t-nodes with respect to the symbol $\mathsf{ax} \in \mathsf{Aux}$ by using the operator $*$ in order to remove edges leading to ax-nodes. Then, we evaluate whether a property holds on the resulting model where a t-node "satisfies" $\mathsf{ax} \in \mathsf{Aux}$ if it has a child satisfying ax. To better emphasise the need to move along t-nodes, given a formula φ, we write $\langle t \rangle \varphi$ for the formula $\Diamond(t \wedge \varphi)$. Dually, $[t]\varphi \overset{\text{def}}{=} \Box(t \Rightarrow \varphi)$. $\langle t \rangle^i$ and $[t]^i$ are also defined, as expected.

Let us start to formalise this encoding. Let $j \geq 1$. First, we restrict ourselves to models where every t-node reachable in at most j steps does not have two Aux-children satisfying the same proposition. Moreover, these Aux-nodes have no children and only satisfy exactly one $\mathsf{ax} \in \mathsf{Aux}$. We express this condition with the formula $\text{init}(j)$ below:

$$\boxplus^j \bigwedge_{\mathsf{ax} \in \mathsf{Aux}} \left(\left(t \Rightarrow \neg(\Diamond \mathsf{ax} * \Diamond \mathsf{ax}) \right) \wedge \Box \left(\mathsf{ax} \Rightarrow \Box\bot \wedge \bigwedge_{\mathsf{bx} \in \mathsf{Aux} \setminus \{\mathsf{ax}\}} \neg \mathsf{bx} \right) \right),$$

where $\boxplus^0 \varphi \overset{\text{def}}{=} \varphi$ and $\boxplus^{m+1}\varphi \overset{\text{def}}{=} \varphi \land \Box \boxplus^m (\varphi)$. Notice that if $\mathfrak{M}, w \models \mathrm{init}(j)$ and $\mathfrak{M}' \sqsubseteq \mathfrak{M}$, then $\mathfrak{M}', w \models \mathrm{init}(j)$.

Among the models $((W, R, V), w)$ satisfying $\mathrm{init}(j)$, we define the ones satisfying $\mathrm{type}(j)$ described below (see similar conditions in [7, Section IV]):

(sub$_j$) every t-node in $R(w)$ satisfies $\mathrm{type}(j-1)$;
(zero$_j$) there is a t-node $\tilde{w} \in R(w)$ such that $\mathfrak{n}(\tilde{w}) = 0$;
(uniq$_j$) distinct t-nodes in $R(w)$ encode different numbers;
(compl$_j$) for every t-node $w_1 \in R(w)$, if $\mathfrak{n}(w_1) < \mathrm{t}(j, n) - 1$
    then $\mathfrak{n}(w_2) = \mathfrak{n}(w_1) + 1$ for some t-node $w_2 \in R(w)$;
(aux) w is a t-node, every t-node in $R(w)$ has one x-child
    and one y-child, and every t-node in $R^2(w)$ has three
    children satisfying l, r and s, respectively.

We define $\mathrm{type}(0) \overset{\text{def}}{=} \top$, and for $j \geq 1$, $\mathrm{type}(j)$ is defined as

$$\mathrm{type}(j) \overset{\text{def}}{=} \mathrm{sub}(j) \land \mathrm{zero}(j) \land \mathrm{uniq}(j) \land \mathrm{compl}(j) \land \mathrm{aux},$$

where each conjunct expresses its homonymous property. The formulae for $\mathrm{sub}(j)$, aux and $\mathrm{zero}(j)$ can be defined as

$$\mathrm{sub}(j) \overset{\text{def}}{=} [t]\mathrm{type}(j-1);$$
$$\mathrm{aux} \overset{\text{def}}{=} t \land [t](\Diamond x * \Diamond y) \land [t]^2(\Diamond l * \Diamond s * \Diamond r);$$
$$\mathrm{zero}(1) \overset{\text{def}}{=} \langle t \rangle \bigwedge_{b \in [1,n]} \neg p_b;$$
$$\mathrm{zero}(j+1) \overset{\text{def}}{=} \langle t \rangle [t] \neg \mathrm{val}.$$

The challenge is therefore how to express $\mathrm{uniq}(j)$ and $\mathrm{compl}(j)$, to guarantee that the numbers of children of w span all over $[0, \mathrm{t}(j, n) - 1]$. The structural properties expressed by $\mathrm{type}(j)$ lead to strong constraints, which permits to control the effects of $*$ when submodels are constructed. This is a key point in designing $\mathrm{type}(j)$ as it helps us to control which edges are lost when considering a submodel.

Nominals, forks and number comparisons. In order to define $\mathrm{uniq}(j)$ and $\mathrm{compl}(j)$ (completing the definition of $\mathrm{type}(j)$), we introduce auxiliary formulae, characterising classes of models that emerge naturally when trying to capture the semantics of (uniq$_j$) and (compl$_j$).

Let us consider a finite forest $\mathfrak{M} = (W, R, V)$ and $w \in W$. A first ingredient is given by the concept of *local nominals*, borrowed from [7]. We say that $\mathrm{ax} \in \mathrm{Aux}$ is a (local) nominal for the depth $i \geq 1$ if there is exactly one t-node $w' \in R^i(w)$ having an ax-child. In this case, w' is said to be the world that corresponds to the local nominal ax. The following formula states that ax is a local nominal for the depth i:

$$\mathrm{nom}_i(\mathrm{ax}) \overset{\text{def}}{=} \langle t \rangle^i \Diamond \mathrm{ax} \land \bigwedge_{k \in [0, i-1]} [t]^k \neg (\langle t \rangle^{i-k} \Diamond \mathrm{ax} * \langle t \rangle^{i-k} \Diamond \mathrm{ax}).$$

We define the formula $@^i_{\mathrm{ax}}\varphi \overset{\text{def}}{=} \langle t \rangle^i (\Diamond \mathrm{ax} \land \varphi)$ which, under the hypothesis that ax is a local nominal for the depth i, states that φ holds on the t-node that corresponds to ax. Moreover, we define $\mathrm{nom}_i(\mathrm{ax} \neq \mathrm{bx}) \overset{\text{def}}{=} \mathrm{nom}_i(\mathrm{ax}) \land \mathrm{nom}_i(\mathrm{bx}) \land \neg @^i_{\mathrm{ax}} \Diamond \mathrm{bx}$, which states that ax and bx are two nominals for the depth i with respect to two distinct t-nodes.

As a second ingredient, we introduce the notion of *fork* that is a specific type of models naturally emerging when trying to compare the numbers $\mathfrak{n}(w_1)$ and $\mathfrak{n}(w_2)$ of two worlds

$w_1, w_2 \in R^i(w)$ (e.g. when checking whether $\mathfrak{n}(w_1) = \mathfrak{n}(w_2)$ or $\mathfrak{n}(w_2) = \mathfrak{n}(w_1) + 1$ holds). Given $j \geq i \geq 1$ we introduce the formula $\mathrm{fork}^i_j(\mathrm{ax}, \mathrm{bx})$ that is satisfied by $(\mathfrak{M}, w)$ iff:

- ax and bx are nominals for the depth i.
- w has exactly two t-children, say w_U and w_D.
- For every $k \in [1, i-1]$, both $R^k(w_U)$ and $R^k(w_D)$ contain exactly one t-child.
- The only t-node in $R^{i-1}(w_U)$, say w_{ax}, corresponds to the nominal ax. The only t-node in $R^{i-1}(w_D)$, say w_{bx}, corresponds to the nominal bx.
- If $i < j$, then $(\mathfrak{M}, w_{\mathrm{ax}})$ and $(\mathfrak{M}, w_{\mathrm{bx}})$ satisfy
$$\mathrm{type}_{\mathrm{lsr}}(j-i) \overset{\text{def}}{=} \mathrm{type}(j-i) \land [t](\Diamond l \land \Diamond s \land \Diamond r).$$

It should be noted that, whenever $(\mathfrak{M}, w)$ satisfies the formula $\mathrm{fork}^i_j(\mathrm{ax}, \mathrm{bx})$, we witness two paths of length i, both starting at w and leading to w_{ax} and w_{bx}, respectively. Worlds in this path may have Aux-children. Below, we schematise a model satisfying $\mathrm{fork}^i_j(\mathrm{ax}, \mathrm{bx})$:
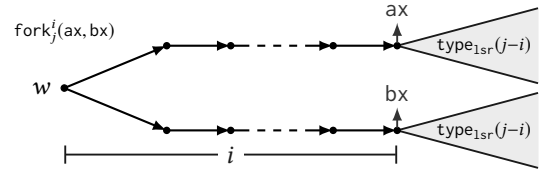
Since the definition of $\mathrm{fork}^i_j(\mathrm{ax}, \mathrm{bx})$ is recursive on i and j (due to $\mathrm{type}(j-i)$), we postpone its formal definition to the next two sections where we treat the base cases for $i = j$ and the inductive case for $j > i$ separately.

The last auxiliary formulae are $[\mathrm{ax} < \mathrm{bx}]^i_j$ and $[\mathrm{bx} = \mathrm{ax}+1]_j$. Under the hypothesis that $(\mathfrak{M}, w)$ satisfies $\mathrm{fork}^i_j(\mathrm{ax}, \mathrm{bx})$, the formula $[\mathrm{ax} < \mathrm{bx}]^i_j$ is satisfied whenever the two (distinct) worlds $w_{\mathrm{ax}}, w_{\mathrm{bx}} \in R^i(w)$ corresponding to the nominals ax and bx are such that $\mathfrak{n}(w_{\mathrm{ax}}) < \mathfrak{n}(w_{\mathrm{bx}})$. Similarly, under the hypothesis that $(\mathfrak{M}, w)$ satisfies $\mathrm{fork}^1_j(\mathrm{ax}, \mathrm{bx})$, the formula $[\mathrm{bx} = \mathrm{ax}+1]_j$ is satisfied whenever $\mathfrak{n}(w_{\mathrm{bx}}) = \mathfrak{n}(w_{\mathrm{ax}})+1$ holds. Both formulae are recursively defined, with base cases for $i = j$ and $j = 1$, respectively.

For the base case, we define the formulae $\mathrm{fork}^j_j(\mathrm{ax}, \mathrm{bx})$ and $[\mathrm{ax} < \mathrm{bx}]^j_j$ (for arbitrary j), as well as $[\mathrm{bx} = \mathrm{ax}+1]_1$. From these formulae, we are then able to define $\mathrm{uniq}(1)$ and $\mathrm{compl}(1)$, which completes the characterisation of $\mathrm{type}(1)$ and $\mathrm{type}_{\mathrm{lsr}}(1)$. Afterwards, we consider the case $1 \leq i < j$ and $j \geq 2$, and define $\mathrm{fork}^i_j(\mathrm{ax}, \mathrm{bx})$, $[\mathrm{ax} < \mathrm{bx}]^i_j$, $[\mathrm{bx} = \mathrm{ax}+1]_j$, as well as $\mathrm{uniq}(j)$ and $\mathrm{compl}(j)$, by only relying on formulae that are already defined (by inductive reasoning).

Base cases: $i = j$ or $j = 1$. In what follows, we consider a finite forest $\mathfrak{M} = (W, R, V)$ and a world w. Following its informal description, we have

$$\mathrm{fork}^j_j(\mathrm{ax}, \mathrm{bx}) \overset{\text{def}}{=} \Diamond_{=2}t \land [t] \boxplus^{j-2}(t \Rightarrow \Diamond_{=1}t) \land \mathrm{nom}_j(\mathrm{ax} \neq \mathrm{bx}),$$

where $\boxplus^j \varphi \overset{\text{def}}{=} \top$ for $j < 0$. As previously explained, in the base case, the number $\mathfrak{n}(w')$ encoded by a t-node $w' \in R^j(w)$

is represented by the truth values of $p_1, \ldots, p_n$. Then, the formula $[\mathsf{ax} < \mathsf{bx}]_j^j$ is defined as

$$[\mathsf{ax} < \mathsf{bx}]_j^j \stackrel{\text{def}}{=} \bigvee_{u \in [1,n]} \Big(@_{\mathsf{ax}}^j \neg p_u \wedge @_{\mathsf{bx}}^j p_u \wedge \bigwedge_{v \in [u+1,n]} (@_{\mathsf{ax}}^j p_v \Leftrightarrow @_{\mathsf{bx}}^j p_v) \Big).$$

The satisfaction of $(\mathfrak{M}, w) \models \mathsf{fork}_j^j(\mathsf{ax}, \mathsf{bx})$ enforces that the distinct t-nodes $w_{\mathsf{ax}}, w_{\mathsf{bx}} \in R^j(w)$ corresponding to ax and bx satisfy $\mathfrak{n}(w_{\mathsf{ax}}) < \mathfrak{n}(w_{\mathsf{bx}})$, which can be shown by using standard properties about bit vectors.

The formula $[\mathsf{bx} = \mathsf{ax+1}]_1$ is similarly defined:

$$\bigvee_{u \in [1,n]} \Big(@_{\mathsf{ax}}^1 (\neg p_u \wedge \bigwedge_{v \in [1,u-1]} p_v) \wedge @_{\mathsf{bx}}^1 (p_u \wedge \bigwedge_{v \in [1,u-1]} \neg p_v) \wedge \bigwedge_{v \in [u+1,n]} (@_{\mathsf{ax}}^1 p_v \Leftrightarrow @_{\mathsf{bx}}^1 p_v) \Big).$$

Assuming $(\mathfrak{M}, w) \models \mathsf{fork}_1^1(\mathsf{ax}, \mathsf{bx})$, this formula states that the two distinct t-nodes $w_{\mathsf{ax}}, w_{\mathsf{bx}} \in R(w)$ corresponding to ax and bx are such that $\mathfrak{n}(w_{\mathsf{bx}}) = \mathfrak{n}(w_{\mathsf{ax}}) + 1$. Again, correctness is guaranteed by standard analysis on bit vectors.

To define $\mathsf{uniq}(1)$, we recall that a model satisfying $\mathsf{type}(1)$ satisfies the formula aux and hence every t-node in $R(w)$ has two auxiliary children, one x-node and one y-node. The idea is to use these two Aux-children and rely on $*$ to state that it is not possible to find a submodel of $\mathfrak{M}$ such that w has only two distinct children w_{x} and w_{y} corresponding to the nominals x and y, respectively, and such that $\mathfrak{n}(w_{\mathsf{x}}) = \mathfrak{n}(w_{\mathsf{y}})$. In a sense, the operator $*$ simulates a second-order quantification on x and y. Let $[\mathsf{x} = \mathsf{y}]_1^1 \stackrel{\text{def}}{=} \neg([\mathsf{x} < \mathsf{y}]_1^1 \vee [\mathsf{y} < \mathsf{x}]_1^1)$. We define $\mathsf{uniq}(1) \stackrel{\text{def}}{=} \neg(\top * (\mathsf{fork}_1^1(\mathsf{x}, \mathsf{y}) \wedge [\mathsf{x} = \mathsf{y}]_1^1))$.

To capture $\mathsf{compl}(1)$ we state that it is not possible to find a submodel of $\mathfrak{M}$ that looses x-nodes from $R^2(w)$, keeps all y-nodes, and is such that **(i)** x is a local nominal for the depth 1, corresponding to a world w_{x} encoding $\mathfrak{n}(w_{\mathsf{x}}) < 2^n - 1$; **(ii)** there is no submodel where w has two t-children, w_{x} and a second world w_{y}, such that w_{y} corresponds to the nominal y and $\mathfrak{n}(w_{\mathsf{y}}) = \mathfrak{n}(w_{\mathsf{x}}) + 1$. Thus, $\mathsf{compl}(1)$ is defined as:

$$\neg \big(\Box \bot * ([t] \Diamond \mathsf{y} \wedge @_{\mathsf{x}}^1 \neg 1_1 \wedge \neg(\top * (\mathsf{fork}_1^1(\mathsf{x}, \mathsf{y}) \wedge [\mathsf{y} = \mathsf{x+1}]_1))) \big).$$

The subscript "1" in the formula 1_1 refers to the fact that we are treating the base case of $\mathsf{compl}(j)$ with $j = 1$. We have $1_1 \stackrel{\text{def}}{=} \bigwedge_{i \in [1,n]} p_i$, reflecting the encoding of $2^n - 1$.

This concludes the definition of $\mathsf{type}(1)$ (and $\mathsf{type}_{\mathsf{lsr}}(1)$), which is established correct with respect to its specification.

Lemma 4.1. *Let* $\mathfrak{M}, w \models \mathsf{init}(1)$. *We have* $\mathfrak{M}, w \models \mathsf{type}(1)$ *iff* $(\mathfrak{M}, w)$ *satisfies* (sub₁), (zero₁), (uniq₁), (compl₁) *and* (aux).

Inductive case: $1 \leq i < j$. As an implicit inductive hypothesis used to prove that the formulae are well-defined, we assume that $[\mathsf{bx} = \mathsf{ax+1}]_{j'}$ and $\mathsf{type}(j')$ are already defined for every $j' < j$, whereas $\mathsf{fork}_{j'}^{i'}(\mathsf{ax}, \mathsf{bx})$, and $[\mathsf{ax} < \mathsf{bx}]_{j'}^{i'}$ are already defined for every $1 \leq i' \leq j'$ such that $j' - i' < j - i$. Therefore, we define:

$$\mathsf{fork}_j^i(\mathsf{ax}, \mathsf{bx}) \stackrel{\text{def}}{=} \mathsf{fork}_i^i(\mathsf{ax}, \mathsf{bx}) \wedge [t]^i \mathsf{type}_{\mathsf{lsr}}(j - i).$$

It is easy to see that this formula is well-defined: $\mathsf{fork}_i^i(\mathsf{ax}, \mathsf{bx})$ is from the base case, whereas $\mathsf{type}_{\mathsf{lsr}}(j-i)$ is defined by inductive hypothesis, since we have $j - i < j$.

Consider now $[\mathsf{ax} < \mathsf{bx}]_j^i$. Assuming $\mathfrak{M}, w \models \mathsf{fork}_j^i(\mathsf{ax}, \mathsf{bx})$, we wish to express $\mathfrak{n}(w_{\mathsf{ax}}) < \mathfrak{n}(w_{\mathsf{bx}})$ for the two distinct worlds $w_{\mathsf{ax}}, w_{\mathsf{bx}} \in R^i(w)$ corresponding to the nominals ax and bx, respectively. As $i < j$, $\mathfrak{n}(w_{\mathsf{ax}})$ (resp. $\mathfrak{n}(w_{\mathsf{bx}})$) is encoded using the truth value of val on the t-children of w_{ax} (resp. w_{bx}). To rely on arithmetical properties of binary numbers used to define $[\mathsf{ax} < \mathsf{bx}]_j^j$, we need to find two partitions $P_{\mathsf{ax}} = \{L_{\mathsf{ax}}, S_{\mathsf{ax}}, R_{\mathsf{ax}}\}$ and $P_{\mathsf{bx}} = \{L_{\mathsf{bx}}, S_{\mathsf{bx}}, R_{\mathsf{bx}}\}$, one for the t-children of w_{ax} and another one for those of w_{bx} s.t.:

(LSR): Given $b \in \{\mathsf{ax}, \mathsf{bx}\}$, P_b splits the t-children as follows:
- there is a t-child s_b of w_b such that $S_b = \{s_b\}$;
- $\mathfrak{n}(r) < \mathfrak{n}(s_b) < \mathfrak{n}(l)$, for every $r \in R_b$ and $l \in L_b$.

(LESS): P_{ax} and P_{bx} have constraints to satisfy $<$:
- $\mathfrak{n}(s_{\mathsf{ax}}) = \mathfrak{n}(s_{\mathsf{bx}})$, $\mathfrak{M}, s_{\mathsf{ax}} \models \neg \mathsf{val}$ and $\mathfrak{M}, s_{\mathsf{bx}} \models \mathsf{val}$;
- for every $l_{\mathsf{ax}} \in L_{\mathsf{ax}}$ and $l_{\mathsf{bx}} \in L_{\mathsf{bx}}$, if $\mathfrak{n}(l_{\mathsf{ax}}) = \mathfrak{n}(l_{\mathsf{bx}})$ then $\mathfrak{M}, l_{\mathsf{ax}} \models \mathsf{val}$ iff $\mathfrak{M}, l_{\mathsf{bx}} \models \mathsf{val}$.

It is important to notice that these conditions essentially revolve around the numbers encoded by t-children, which will be compared using the already defined (by inductive reasoning) formulae $[\mathsf{ax} < \mathsf{bx}]_{j'}^{i'}$, where $j' - i' < j - i$. Since the semantics of $[\mathsf{ax} < \mathsf{bx}]_j^i$ is given under the hypothesis that $\mathfrak{M}, w \models \mathsf{fork}_j^i(\mathsf{ax}, \mathsf{bx})$, we can assume that every child of w_{ax} and w_{bx} has all the possible Aux-children. Then, we rely on the auxiliary propositions in $\{l, s, r\}$ in order to mimic the reasoning done in (LSR) and (LESS).

We start by considering the constraints involved in (LSR) and express them with the formula $\mathsf{lsr}(j)$, which is satisfied by a pointed forest $(\mathfrak{M} = (W, R, V), w)$ whenever:

- $(\mathfrak{M}, w)$ satisfies $\mathsf{type}(j)$.
- Every t-child of w has exactly one $\{l, s, r\}$-child, and only one of these t-children (say w') has an s-child.
- Every t-child of w that has an l-child (resp. r-child) encodes a number greater (resp. smaller) than $\mathfrak{n}(w')$.

Despite this formula being defined in terms of $\mathsf{type}(j)$, we only rely on $\mathsf{lsr}(j - i)$ (which is defined by inductive reasoning) in order to define $[\mathsf{ax} < \mathsf{bx}]_j^i$. The picture below schematises a model satisfying $\mathsf{lsr}(j)$.

w $\mathsf{lsr}(j)$, implies $\mathsf{type}(j)$
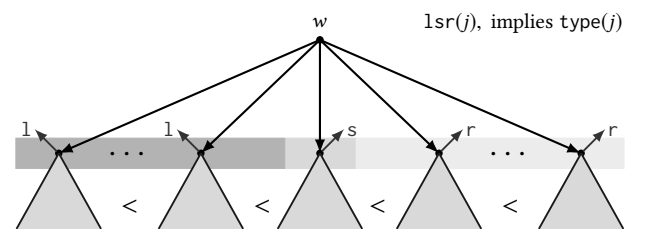
The definition of $\mathsf{lsr}(j)$ follows closely its specification:

$$\mathsf{lsr}(j) \stackrel{\text{def}}{=} \mathsf{type}(j) \wedge \mathsf{nom}_1(s) \wedge \neg(\top * (\mathsf{fork}_j^1(s, l) \wedge \neg[s < l]_j^1))$$
$$\wedge \neg(\top * (\mathsf{fork}_j^1(s, r) \wedge \neg[r < s]_j^1)) \wedge [t] \Diamond_{=1}(l \vee s \vee r).$$

We define the formula $[\mathsf{ax} < \mathsf{bx}]_j^i$ as follows:

$$\top * \big(\mathsf{nom}_i(\mathsf{ax} \neq \mathsf{bx}) \wedge [t]^i \mathsf{lsr}(j - i) \wedge S_j^i(\mathsf{ax}, \mathsf{bx}) \wedge L_j^i(\mathsf{ax}, \mathsf{bx}) \big),$$

where $S_j^i(ax, bx)$ and $L_j^i(ax, bx)$ check the first and second condition in (LESS), respectively. In particular, by defining $[ax = bx]_j^i \stackrel{\text{def}}{=} \neg([ax < bx]_j^i \vee [bx < ax]_j^i)$, we have

$$S_j^i(ax, bx) \stackrel{\text{def}}{=} \top * \big(\mathsf{fork}_j^{i+1}(x, y) \wedge @_{ax}^i \langle t \rangle (\Diamond s \wedge \Diamond x) \wedge$$
$$@_{bx}^i \langle t \rangle (\Diamond s \wedge \Diamond y) \wedge [x = y]_j^{i+1} \wedge @_x^{i+1} \neg val \wedge @_y^{i+1} val\big)$$

$$L_j^i(ax, bx) \stackrel{\text{def}}{=} \neg \big(\top * (\mathsf{fork}_j^{i+1}(x, y) \wedge @_{ax}^i \langle t \rangle (\Diamond l \wedge \Diamond x) \wedge$$
$$@_{bx}^i \langle t \rangle (\Diamond l \wedge \Diamond y) \wedge [x = y]_j^{i+1} \wedge \neg (@_x^{i+1} val \Leftrightarrow @_y^{i+1} val))\big).$$

Both $\mathsf{fork}_j^{i+1}(x, y)$ and $[x = y]_j^{i+1}$ used in these formulae are defined recursively. The formula $S_j^i(ax, bx)$ states that there is a submodel $\mathfrak{M}' \sqsubseteq \mathfrak{M}$ such that

I. $\mathfrak{M}', w \models \mathsf{fork}_j^{i+1}(x, y)$;
II. s_{ax} corresponds to the nominal x at depth $i + 1$;
III. s_{bx} corresponds to the nominal y at depth $i + 1$;
IV-VI. $\mathfrak{n}(s_{ax}) = \mathfrak{n}(s_{bx})$, $\mathfrak{M}, s_{ax} \not\models val$ and $\mathfrak{M}, s_{bx} \models val$.

(The enumeration I-VI refers to the conjuncts in the formula)

$S_j^i(ax, bx)$ correctly models the first condition of (LESS). Regarding $L_j^i(ax, bx)$ and (LESS), a similar analysis can be performed. We define $\mathsf{LS}_j^i(ax, bx) \stackrel{\text{def}}{=} L_j^i(ax, bx) \wedge S_j^i(ax, bx)$.

Let us consider $[bx = ax{+}1]_j$. Under the hypothesis that $\mathfrak{M}, w \models \mathsf{fork}_j^i(ax, bx)$, this formula must express $\mathfrak{n}(w_{bx}) = \mathfrak{n}(w_{ax}) + 1$ for the two (distinct) worlds $w_{ax}, w_{bx} \in R^i(w)$. Then, as done for defining $[ax < bx]_j^i$, we take advantage of arithmetical properties on binary numbers and we search for two partitions $P_{ax} = \{L_{ax}, S_{ax}, R_{ax}\}$ and $P_{bx} = \{L_{bx}, S_{bx}, R_{bx}\}$ of the t-children of w_{ax} and w_{bx}, respectively, such that P_{ax} and P_{bx} satisfy (LSR) as well as the condition below:

(PLUS): P_{ax} and P_{bx} have the arithmetical properties of $+1$:
- P_{ax} and P_{bx} satisfy (LESS);
- for every $r_{ax} \in R_{ax}$, we have $\mathfrak{M}, r_{ax} \models val$;
- for every $r_{bx} \in R_{bx}$, we have $\mathfrak{M}, r_{ax} \not\models val$,

where $S_{ax} = \{s_{ax}\}$ and $S_{bx} = \{s_{bx}\}$, as required by (LSR).

The definition of $[bx = ax{+}1]_j$ is similar to $[ax < bx]_j^i$:

$$\top * \big(\mathsf{nom}_1(ax \neq bx) \wedge [t]\mathsf{lsr}(j-1) \wedge \mathsf{LS}_x^1(ax, bx) \wedge R(ax, bx)\big),$$

where $R(ax, bx) \stackrel{\text{def}}{=} @_{ax}^1 [t](\Diamond r \Rightarrow val) \wedge @_{bx}^1 [t](\Diamond r \Rightarrow \neg val)$ captures the last two conditions of (PLUS).

To define $\mathsf{uniq}(j)$ and $\mathsf{compl}(j)$, we rely on $\mathsf{fork}_j^i(ax, bx)$, $[ax < bx]_j^i$ and $[bx = ax{+}1]_j$.

$$\mathsf{uniq}(j) \stackrel{\text{def}}{=} \neg \big(\top * (\mathsf{fork}_j^1(x, y) \wedge [x = y]_j^1)\big)$$

$$\mathsf{compl}(j) \stackrel{\text{def}}{=} \neg \Big(\Box \bot * \big([t](\mathsf{type}_{\mathsf{lsr}}(j-1) \wedge \Diamond y) \wedge \mathsf{nom}_1(x) \wedge$$
$$@_x^1 \neg 1_j \wedge \neg (\top * (\mathsf{fork}_j^1(x, y) \wedge [y = x{+}1]_j))\big)\Big),$$

where $1_j \stackrel{\text{def}}{=} [t]val$ reflects the encoding of $t(j, n) - 1$ for $j > 1$. The main difference between $\mathsf{compl}(1)$ and $\mathsf{compl}(j)$ ($j > 1$) is that the conjunct $[t]\Diamond y$ of $\mathsf{compl}(1)$ is replaced by $[t](\mathsf{type}_{\mathsf{lsr}}(j-1) \wedge \Diamond y)$ in $\mathsf{compl}(j)$, as needed to correctly evaluate $\mathsf{fork}_j^1(x, y)$. Indeed, the difference between $\mathsf{fork}_1^1(x, y)$ and $\mathsf{fork}_j^1(x, y)$ is precisely that the latter requires $[t]\mathsf{type}_{\mathsf{lsr}}(j-1)$. The definition of $\mathsf{type}(j)$ is now complete. We can state its correctness.

Lemma 4.2. *Let $\mathfrak{M}, w \models \mathsf{init}(j)$. We have $\mathfrak{M}, w \models \mathsf{type}(j)$ iff $(\mathfrak{M}, w)$ satisfies (sub$_j$), (zero$_j$), (uniq$_j$), (compl$_j$) and (aux).*

The size of $\mathsf{type}(j)$ is exponential in $j > 1$ and polynomial in $n \geq 1$. As its size is elementary, we can use this formula as a starting point to reduce Tile_k.

4.2 Tiling a grid $[0, t(k, n) - 1] \times [0, t(k, n) - 1]$

Below, we briefly explain how to use previous developments to define a uniform reduction from Tile_k, for every $k \geq 2$. Several adaptations are needed to encode smoothly the grid but the hardest part was the design of $\mathsf{type}(j)$. Let $k \geq 2$ and $(\mathcal{T}, c)$ be an instance of Tile_k. We can construct a formula $\mathsf{tiling}_{\mathcal{T}, c}(k)$ that is satisfiable if and only if $(\mathcal{T}, c)$ as a solution. To represent $[0, t(k, n) - 1]^2$ in some pointed forest $(\mathfrak{M}, w)$, where $\mathfrak{M} = (W, R, V)$, we recycle the ideas for defining $\mathsf{type}(k)$. From Lemma 4.2, we know that if $\mathfrak{M}, w \models \mathsf{init}(k) \wedge \mathsf{type}(k)$ then the t-children of w encode the interval $[0, t(k, n) - 1]$. A position in the grid is however a pair of numbers, hence the *crux of our encoding* rests on the fact that each $w' \in R(w)$ encodes *two* numbers $\mathfrak{n}_{\mathcal{H}}(w')$ and $\mathfrak{n}_{\mathcal{V}}(w')$. Similarly to $\mathsf{type}(k)$, these numbers are represented by the truth values on the t-children of w', with the help of *new propositions* $val_{\mathcal{H}}$ and $val_{\mathcal{V}}$. We are in luck: since both numbers are from $[0, t(k, n) - 1]$, w' just needs as many children as when encoding a single number, and therefore if $\mathfrak{M}, w \models \mathsf{tiling}_{\mathcal{T}, c}(k)$ then $\mathfrak{M}, w' \models \mathsf{type}(k{-}1)$. In fact, the portion of $\mathsf{tiling}_{\mathcal{T}, c}(k)$ that encodes the grid can be described quite naturally by slightly updating the characterisation of $\mathsf{type}(k)$. For example, (uniq$_j$) becomes

(uniq$_{\mathcal{T}, k}$)   for all distinct t-nodes $w_1, w_2 \in R(w)$
$\mathfrak{n}_{\mathcal{H}}(w_1) \neq \mathfrak{n}_{\mathcal{H}}(w_2)$ or $\mathfrak{n}_{\mathcal{V}}(w_1) \neq \mathfrak{n}_{\mathcal{V}}(w_2)$.

The formula $\mathsf{uniq}(k)$ has to be updated accordingly, but without major differences or complications. Of course, more is required as $\mathsf{tiling}_{\mathcal{T}, c}(k)$ must also encode the tiling conditions (first) and (hor&vert). Fortunately, the kit of formulae defined for $\mathsf{type}(k)$ allows us to have access to $\mathfrak{n}_{\mathcal{H}}(w')$ and $\mathfrak{n}_{\mathcal{V}}(w')$ in such a way that both conditions can be expressed rather easily. For example, to express vertical constraints, we design a formula stating that for all t-nodes $w_1, w_2 \in R(w)$, if $\mathfrak{n}_{\mathcal{V}}(w_2) = \mathfrak{n}_{\mathcal{V}}(w_1){+}1$ and $\mathfrak{n}_{\mathcal{H}}(w_2) = \mathfrak{n}_{\mathcal{H}}(w_1)$ then there is $(c_1, c_2) \in \mathcal{V}$ such that $w_1 \in V(c_1)$ and $w_2 \in V(c_2)$. Further details are omitted by lack of space.

Theorem 4.3. *Sat(ML($*$)) is* Tower*-complete.*

5 ML($*$) Strictly Less Expressive Than GML

Below, we focus on the expressivity of ML($*$). We first show ML($*$) $\leq$ GML and then we prove the strictness of the inclusion. The former result takes advantage of the notion of g-bisimulation, i.e. the underlying structural indistinguishability relation of GML, studied in [20]. To show ML($*$) $<$ GML, we define an ad hoc notion of Ehrenfeucht-Fraïssé games for ML($*$), see e.g. classical definitions in [31] and similar

approaches in [13, 18]. Then, we design a simple formula in GML that cannot be expressed in ML(∗).

5.1 ML(∗) is not more expressive than GML

To establish that ML(∗) ⪯ GML, we proceed as in Section 3.1. In fact, by Lemma 2.2, given φ_1, φ_2 in GML, the formula $\varphi_1 * \varphi_2$ is equivalent to $\blacklozenge(\varphi_1 | \varphi_2)$. Moreover, we know that given φ_1, φ_2 in GML, $\varphi_1 | \varphi_2$ is equivalent to some formula in GML, as shown in Section 3. So, to prove that ML(∗) ⪯ GML by applying the proof schema of Theorem 3.2, it is sufficient to show that given φ in GML, there is ψ in GML such that $\blacklozenge\varphi \equiv \psi$. To do so, we rely on the indistinguishability relation of GML, called g-bisimulation [20].

A g-bisimulation is a refinement of the classical back-and-forth conditions of a bisimulation (see e.g. [8]), tailored towards capturing graded modalities. It relates models with similar structural properties, but up to parameters $m, k \in \mathbb{N}$ responsible for the modal degree and the graded rank, respectively. The following invariance result holds: g-bisimilar models are modally equivalent in GML (up to formulae of modal degree m and graded rank at most k). For simplicity, we present the construction of the above-mentioned formula ψ by directly using the notion of model equivalence, without going explicitly through g-bisimulations.

Given $m, k \in \mathbb{N}$ and P $\subseteq_{\text{fin}}$ AP, we write GML$[m, k, \text{P}]$ to denote the set of GML formulae ψ having md$(\psi) \leq m$, gr$(\psi) \leq k$ and propositional variables from P. GML$[m, k, \text{P}]$ is finite up to logical equivalence [20]. Given pointed forests $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$, we write $(\mathfrak{M}, w) \equiv^{\text{P}}_{m,k} (\mathfrak{M}', w')$ whenever $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ are GML$[m, k, \text{P}]$-indistinguishable, i.e. for every ψ in GML$[m, k, \text{P}]$, $\mathfrak{M}, w \models \psi$ iff $\mathfrak{M}', w' \models \psi$. We write $\mathcal{T}^{\text{P}}(m, k)$ to denote the quotient set induced by the equivalence relation $\equiv^{\text{P}}_{m,k}$. As GML$[m, k, \text{P}]$ is finite up to logical equivalence, we get that $\mathcal{T}^{\text{P}}(m, k)$ is finite.

To establish that GML is closed under $\blacklozenge$, we show that there is a function $\mathfrak{f} : \mathbb{N}^2 \to \mathbb{N}$ such that for all $m, k \in \mathbb{N}$ and P $\subseteq_{\text{fin}}$ AP, if two models are in the same equivalence class of $\equiv^{\text{P}}_{m,\mathfrak{f}(m,k)}$, then they satisfy the same formulae of the form $\blacklozenge\varphi$, where φ is in GML$[m, k, \text{P}]$. By standard arguments and using the fact that GML$[m, \mathfrak{f}(m, k), \text{P}]$ is finite up to logical equivalence, we then conclude that $\blacklozenge\varphi$ is equivalent to a formula in GML$[m, \mathfrak{f}(m, k), \text{P}]$. Similar approaches are followed in [22, 23, 34]. As we are not interested in the size of the equivalent formula, we can simply use the cardinality of $\mathcal{T}^{\text{P}}(m, k)$ in order to inductively define a suitable function:

$$\mathfrak{f}(0, k) \stackrel{\text{def}}{=} k, \quad \mathfrak{f}(m + 1, k) \stackrel{\text{def}}{=} k \times (|\mathcal{T}^{\text{P}}(m, \mathfrak{f}(m, k))| + 1).$$

In conformity with the results in Section 4, the map $\mathfrak{f}$ can be shown to be a non-elementary function. To prove that $\mathfrak{f}$ satisfies the required properties, we start by showing a technical lemma which essentially formalises a simulation argument on the relation $\equiv^{\text{P}}_{m,\mathfrak{f}(m,k)}$ with respect to the submodel relation. By taking submodels as with the $\blacklozenge$ operator, equivalence in GML is preserved.

Lemma 5.1. *Let* $(\mathfrak{M}, w) \equiv^{\text{P}}_{m,\mathfrak{f}(m,k)} (\mathfrak{M}', w')$ *where* $m, k \in \mathbb{N}$, P $\subseteq_{\text{fin}}$ AP, $\mathfrak{M} = (W, R, V)$ *and* $\mathfrak{M}' = (W', R', V')$. *Let* $R_1 \subseteq R$. *There is* $R'_1 \subseteq R'$ *s.t.* $((W, R_1, V), w) \equiv^{\text{P}}_{m,k} ((W', R'_1, V'), w')$ *and if* $R_1(w) = R(w)$, *then* $R'_1(w') = R'(w')$.

The proof of Lemma 5.1 is by induction on m. The last condition about $R_1(w) = R(w)$ will serve in the proof of Lemma 5.2, as it allows us to capture the semantics of $\blacklozenge$, by preserving the children of the world w'. In the proof, we rely on the properties of g-bisimulations [20] to define a binary relation $\leftrightarrow$ between worlds of $R(w)$ and $R'(w')$. Every $w_1 \leftrightarrow w'_1$ is such that $(\mathfrak{M}, w_1) \equiv^{\text{P}}_{m-1,\mathfrak{f}(m-1,k)} (\mathfrak{M}', w'_1)$. The operator $\blacklozenge$ does not necessarily preserve the children of w_1 and w'_1, so that the induction hypothesis, naturally defined from the statement of Lemma 5.1, is applied on models where the condition $R_1(w_1) = R(w_1)$ may not hold. We show that for all $R_1 \subseteq R$, it is possible to construct $R'_1 \subseteq R'$ such that, for all $w_1 \leftrightarrow w'_1$, $((W, R_1, V), w_1) \equiv^{\text{P}}_{m-1,k} ((W', R'_1, V'), w'_1)$. The result is then lifted to $((W, R_1, V), w) \equiv^{\text{P}}_{m,k} ((W', R'_1, V'), w')$, again thanks to the properties of the g-bisimulation.

Intuitively, Lemma 5.1 states that given two models satisfying the same formulae up to the parameters m and $\mathfrak{f}(m, k)$, we can extract submodels satisfying the same formulae up to m and k (reduced graded rank). This allows us to conclude that if φ is in GML, there is some GML formula equivalent to $\blacklozenge\varphi$ (Lemma 5.2). In other words, the operator $\blacklozenge$ can be eliminated to obtain a GML formula. This, together with Lemma 2.2 and Theorem 3.2 entail ML(∗) ⪯ GML.

Lemma 5.2. *For every* $\varphi \in$ GML$[m, k, \text{P}]$ *there is a formula* $\psi \in$ GML$[m, \mathfrak{f}(m, k), \text{P}]$ *such that* $\blacklozenge\varphi \equiv \psi$.

5.2 Showing ML(∗) < GML with EF games for ML(∗)

We tackle the problem of showing that ML(∗) is strictly less expressive than GML. To do so, we adapt the notion of Ehrenfeucht-Fraïssé games (EF games, in short) [31] to ML(∗), and use it to design a GML formula that is not expressible in ML(∗). We write ML(∗)$[m, s, \text{P}]$ for the set of formulae φ of ML(∗) having md$(\varphi) \leq m$, at most s nested ∗, and atomic propositions from P $\subseteq_{\text{fin}}$ AP. It is easy to see that ML(∗)$[m, s, \text{P}]$ is finite up to logical equivalence.

We introduce the EF games for ML(∗). A game is played between two players: the *spoiler* and the *duplicator*. A game state is a triple made of two pointed forests $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ and a rank (m, s, P), where $m, s \in \mathbb{N}$ and P $\subseteq_{\text{fin}}$ AP. The goal of the spoiler is to show that the two models are different. The goal of the duplicator is to counter the spoiler and to show that the two models are similar. Two models are different whenever there is $\varphi \in$ ML(∗)$[m, s, \text{P}]$ that is satisfied by only one of the two models. The EF games for ML(∗) are formally defined in Figure 1. The exact correspondence between the game and the logic is formalised in Lemma 5.3.

Using the standard definitions in [31], the duplicator has a *winning strategy* for the game $((\mathfrak{M}, w), (\mathfrak{M}', w'), (m, s, \text{P}))$

Game on $[(\mathfrak{M}_1=(W_1, R_1, V_1), w_1), (\mathfrak{M}_2=(W_2, R_2, V_2), w_2), (m, s, \mathsf{P})].$

if there is $p \in \mathsf{P}$ s.t. $w_1 \in V_1(p)$ iff $w_2 \notin V_2(p)$ then the spoiler wins.
else the spoiler chooses $i \in \{1, 2\}$ and plays on $\mathfrak{M}_i$. The duplicator replies on $\mathfrak{M}_j$ where $j \neq i$. The spoiler must choose one of the following moves, otherwise the duplicator wins:
modal move: if $m \geq 1$ and $R_i(w_i) \neq \emptyset$ then the spoiler **can** choose to play a modal move by selecting an element $w'_i \in R_i(w_i)$. Then,
 • the duplicator must reply with a $w'_j \in R_j(w_j)$ (else, the spoiler wins);
 • the game continues on $[(\mathfrak{M}_1, w'_1), (\mathfrak{M}_2, w'_2), (m - 1, s, \mathsf{P})].$
spatial move: if $s \geq 1$ then the spoiler **can** choose to play a spatial move by selecting two finite forests $\mathfrak{M}_i^1$ and $\mathfrak{M}_i^2$ s.t. $\mathfrak{M}_i^1 + \mathfrak{M}_i^2 = \mathfrak{M}_i$. Then,
 • the duplicator replies with two forests $\mathfrak{M}_j^1$ and $\mathfrak{M}_j^2$ s.t. $\mathfrak{M}_j^1 + \mathfrak{M}_j^2 = \mathfrak{M}_j$;
 • The game continues on $[(\mathfrak{M}_1^k, w_1), (\mathfrak{M}_2^k, w_2), (m, s - 1, \mathsf{P})]$, where $k \in \{1, 2\}$ is chosen by the spoiler.

Figure 1. Ehrenfeucht-Fraïssé games for $\mathsf{ML}(*)$

if she can play in a way that guarantees her to win regardless how the spoiler plays. When this is the case, we write $(\mathfrak{M}, w) \approx_{m,s}^{\mathsf{P}} (\mathfrak{M}', w')$. Similarly, the spoiler has a *winning strategy*, written $(\mathfrak{M}, w) \not\approx_{m,s}^{\mathsf{P}} (\mathfrak{M}', w')$, if he can play in a way that guarantees him to win, regardless how the duplicator plays. Lemma 5.3 guarantees that the games are well-defined.

Lemma 5.3. $(\mathfrak{M}, w) \not\approx_{m,s}^{\mathsf{P}} (\mathfrak{M}', w')$ *iff there is a formula* φ *in* $\mathsf{ML}(*)[m, s, \mathsf{P}]$ *such that* $\mathfrak{M}, w \models \varphi$ *and* $\mathfrak{M}', w' \not\models \varphi$.

Lemma 5.3 is proven with standard arguments from [31], for instance the left-to-right direction, i.e. the *completeness of the game*, is by induction on the rank (m, s, P). Thanks to the EF games, we are able to find a GML formula φ that is not expressible in $\mathsf{ML}(*)$. By Lemma 2.1 and as $\mathsf{ML}(\mathbin{|}) \approx \mathsf{GML}$, such a formula is necessarily of modal degree at least 2. Happily, $\varphi = \Diamond_{=2} \Diamond_{=1} \top$ does the job and cannot be expressed in $\mathsf{ML}(*)$. For the proof, we show that for every rank (m, s, P), there are two structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ such that $(\mathfrak{M}, w) \approx_{m,s}^{\mathsf{P}} (\mathfrak{M}', w')$, $\mathfrak{M}, w \models \varphi$ and $\mathfrak{M}', w' \not\models \varphi$. The inexpressibility of φ then stems from Lemma 5.3. The two structures are represented below $((\mathfrak{M}, w)$ on the left).
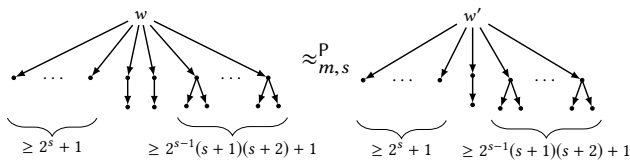
In the following, we say that a world has *type i* if it has i children. As one can see in the figure above, children of the current worlds w and w' are of three types: 0, 1 or 2. When the spoiler performs a spatial move in the game, a world of type i can take, in the submodels, a type between 0 and i. That is, the number of children of a world weakly monotonically decreases when taking submodels. This monotonicity, together with the finiteness of the game, lead to bounds on the number of children of each type, over which the duplicator is guaranteed to win. For instance, the bound for worlds of type 2 is given by the value $2^s(s + 1)(s + 2)$, where s is the number of spatial moves in the game. In the two presented

pointed forests, one child of type 0 and one of type 2 are added with respect to these bounds, so that the duplicator can make up for the different numbers of children of type 1.

Lemma 5.4. $\mathsf{ML}(*)$ *cannot characterise the class of models satisfying the* GML *formula* $\Diamond_{=2} \Diamond_{=1} \top$.

Notice that $\mathsf{ML}(*)$ is more expressive than ML. Indeed, the formula $\Diamond\top * \Diamond\top$ distinguishes the two models on the right, which are bisim-ilar and hence indistinguishable in ML [46].

By $\mathsf{ML}(*) \preceq \mathsf{GML}$, Lemma 5.4 and Theorem 3.2, we conclude.

Theorem 5.5. $\mathsf{ML} \prec \mathsf{ML}(*) \prec \mathsf{GML} \approx \mathsf{ML}(\mathbin{|})$.

6 $\mathsf{ML}(\mathbin{|})$, $\mathsf{ML}(*)$ and Sister Logics

Below, we show how our new results on $\mathsf{ML}(\mathbin{|})$ and $\mathsf{ML}(*)$ allow us to make substantial contributions for sister logics.

6.1 Static ambient logic

Static ambient logic (SAL) is a formalism proposed to reason about spatial properties of concurrent processes specified in the ambient calculus [15]. In [12], the satisfiability and validity problems for a very expressive fragment of SAL are shown to be decidable and conjectured to be in PSPACE (see [12, Section 6]). We invalidate this conjecture by showing that the intensional fragment of SAL (see [32]), herein denoted $\mathsf{SAL}(\mathbin{|})$, is already $\mathsf{AExp_{POL}}$-complete. More precisely, we design semantically faithful reductions between $\mathsf{Sat}(\mathsf{ML}(\mathbin{|}))$ and $\mathsf{Sat}(\mathsf{SAL}(\mathbin{|}))$ (in both directions), leading to the abovementioned result by Corollary 3.8. $\mathsf{SAL}(\mathbin{|})$ formulae are from

$$\varphi := \top \mid \mathbb{0} \mid \mathsf{n}[\varphi] \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi \mathbin{|} \varphi,$$

where $\mathsf{n} \in \mathsf{AP}$ is an *ambient name*. Historically, the semantics of SAL is given on a class of syntactically defined finite trees. However, this class of models is isomorphic to the class of finite trees $\mathfrak{M} = (W, R, V)$, such that each world in W satisfies exactly one atomic proposition (its ambient name). Then, the satisfaction relation $\models$ for $\mathsf{SAL}(\mathbin{|})$ is standard for $\top$ and Boolean connectives, $\varphi_1 \mathbin{|} \varphi_2$ is as in $\mathsf{ML}(\mathbin{|})$, and otherwise

$$\mathfrak{M}, w \models \mathbb{0} \quad \Leftrightarrow R(w) = \emptyset;$$
$$\mathfrak{M}, w \models \mathsf{n}[\varphi] \Leftrightarrow \text{there is } w' \in W \text{ such that } R(w) = \{w'\},$$
$$w' \in V(\mathsf{n}) \text{ and } \mathfrak{M}, w' \models \varphi.$$

With such a presentation, $\mathsf{SAL}(\mathbin{|})$ is a fragment of $\mathsf{ML}(\mathbin{|})$, where $\mathbb{0}$ and $\mathsf{n}[\varphi]$ correspond to $\Box\bot$ and $\Diamond_{=1}\top \wedge \Diamond(\mathsf{n} \wedge \varphi)$, respectively. However, to reduce $\mathsf{Sat}(\mathsf{SAL}(\mathbin{|}))$ to $\mathsf{Sat}(\mathsf{ML}(\mathbin{|}))$, we must deal with the constraint on V (uniqueness of the ambient name). Let φ be in $\mathsf{SAL}(\mathbin{|})$ written with the ambient names in $\mathsf{N} = \{\mathsf{n}_1, \dots, \mathsf{n}_m\}$. It is known (see [12, Lemma 8]) that if φ is satisfiable, then it can be satisfied by a tree having ambient names from $\mathsf{N} \cup \{\overline{\mathsf{n}}\}$, where $\overline{\mathsf{n}}$ is a fresh name. Thus, we can show that φ is satisfiable iff so is the $\mathsf{ML}(\mathbin{|})$ formula

$$\varphi \wedge \boxplus^{\mathsf{md}(\varphi)}(\bigvee_{\mathsf{n}\in\mathsf{N}\cup\{\overline{\mathsf{n}}\}}(\mathsf{n} \wedge \bigwedge_{\mathsf{n}'\in(\mathsf{N}\cup\{\overline{\mathsf{n}}\})\setminus\{\mathsf{n}\}} \neg\mathsf{n}')),$$

where the right conjunct states that V, restricted to the propositions in $\mathsf{N} \cup \{\overline{\mathsf{n}}\}$, forms a partition of the worlds reachable from the current one in at most $\mathsf{md}(\varphi)$ steps.

Reducing Sat(ML($|$)) to Sat(SAL($|$)) requires a bit more work. Let $\mathfrak{M} = (W, R, V)$ be a finite forest and $w \in W$. Assume we want to check the satisfiability status of φ in ML($|$) having atomic propositions from P $= \{p_1, \ldots, p_m\}$ and with n occurrences of $|$. We encode $(\mathfrak{M}, w)$ into a model $(\mathfrak{M}' = (W', R', V'), w)$ of SAL($|$) as follows. Let rel and ap be two ambient names not in P. The ambient name rel encodes the relation R whereas ap can be seen as a *container* for propositional variables holding on the current world. *(i)* We require $W \subseteq W'$, $R \subseteq R'$ and $\bigcup_{i \in [0, \mathrm{md}(\varphi)]} R^i(w) \subseteq V'(\mathrm{rel})$, i.e., every world reachable from w in at most $\mathrm{md}(\varphi)$ steps has the ambient name rel. Let w' be one of these worlds and suppose that $\{p \mid w' \in V(p)\} \cap P = \{q_1, \ldots, q_l\}$. *(ii)* We require W' to contain $n + 1$ worlds $w'_1, \ldots, w'_{n+1} \in R'(w') \setminus R(w')$, all having ambient name ap. These worlds encode copies of w''s valuation, similarly to what is done in Section 3.2 to encode teams from PL[~]. *(iii)* For all $j \in [1, n + 1]$, $R'(w'_j)$ contains l worlds, all satisfying 0 and a distinct ambient name from $\{q_1, \ldots, q_l\}$. Below we schematise the encoding (w.r.t. w').
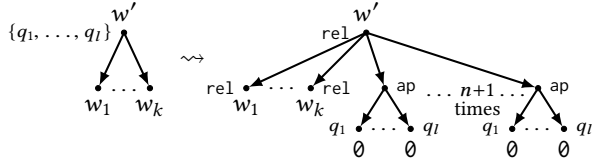
Let $\mathsf{n} \in$ AP. We define the modality $\langle \mathsf{n} \rangle \varphi \stackrel{\mathrm{def}}{=} \mathsf{n}[\varphi] | \top$ and its dual $[\mathsf{n}]\varphi \stackrel{\mathrm{def}}{=} \neg \langle \mathsf{n} \rangle \neg \varphi$. We write $\forall[\mathsf{n}]$ for $\neg((\neg 0 \wedge \neg \mathsf{n}[\top]) | \top)$, so that $(\mathfrak{M}, w) \models \forall[\mathsf{n}]$ whenever every child of w has the ambient name n. Moreover, $[\# \geq 0] \stackrel{\mathrm{def}}{=} \top$ and $[\# \geq \beta + 1] \stackrel{\mathrm{def}}{=} \neg 0 | [\# \geq \beta]$, so that $(\mathfrak{M}, w) \models [\# \geq \beta]$ whenever w has at least β children. Lastly, $[\# = \beta] \stackrel{\mathrm{def}}{=} [\# \geq \beta] \wedge \neg[\# \geq \beta + 1]$. The models of SAL($|$) encoding models of ML($|$) are characterised by

$$C_\varphi \stackrel{\mathrm{def}}{=} \bigwedge_{j \in [0, \mathrm{md}(\varphi)]} [\mathrm{rel}]^j \Big(\forall[\mathrm{rel}] | (\forall[\mathrm{ap}] \wedge [\# = n+1] \wedge [\mathrm{ap}]((p_1[0] \vee 0) |$$
$$\ldots | (p_m[0] \vee 0)) \wedge \bigwedge_{i \in [1, m]} (\langle \mathrm{ap} \rangle \langle p_i \rangle \top \Rightarrow [\mathrm{ap}] \langle p_i \rangle \top)) \Big).$$

Lastly, we define the translation of φ, written $\tau(\varphi)$, into SAL($|$). It is homomorphic for Boolean connectives and $\top$, $\tau(p) \stackrel{\mathrm{def}}{=} \langle \mathrm{ap} \rangle \langle p \rangle \top$ and otherwise it is inductively defined:

$$\tau(\Diamond \psi) \stackrel{\mathrm{def}}{=} \langle \mathrm{rel} \rangle \tau(\psi);$$

$$\tau(\psi_1 | \psi_2) \stackrel{\mathrm{def}}{=} (\tau(\psi_1) \wedge \langle \mathrm{ap} \rangle_{\geq j} \top) | (\tau(\psi_2) \wedge \langle \mathrm{ap} \rangle_{\geq k} \top),$$

where in $\tau(\psi_1 | \psi_2)$, j (resp. k) is the number of occurrences of $|$ in ψ_1 (resp. ψ_2) plus one and $\langle \mathrm{ap} \rangle_{\geq \alpha} \top \stackrel{\mathrm{def}}{=} (\forall[\mathrm{ap}] \wedge [\# = \alpha]) | \top$.

We show that φ is satisfiable in ML($|$) iff $C_\varphi \wedge \tau(\varphi)$ is satisfiable in SAL($|$), leading to the following results about the complexity of static ambient logics.

Corollary 6.1. Sat(SAL($|$)) *is* $\mathrm{AExp}_{\mathrm{POL}}$-*complete.* Sat(SAL) *with* SAL *from [12] is* $\mathrm{AExp}_{\mathrm{POL}}$-*hard.*

6.2 Modal separation logic

The family of *modal separation logics* (MSL), combining separating and modal connectives, has been recently introduced in [21]. Its models, inspired from the memory states used in

separation logic (see also [17]), are Kripke-style structures $\mathfrak{M} = (W, R, V)$, where $W = \mathbb{N}$ and $R \subseteq W \times W$ is finite and functional. Hence, unlike finite forests, $\mathfrak{M}$ may have loops.

Among the fragments studied in [21], the modal separation logic MSL($*, \Diamond^{-1}$) was left with a huge complexity gap (between PSPACE and TOWER). Its formulae are defined from

$$\varphi := p \mid \Diamond^{-1}\varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi * \varphi.$$

The satisfaction relation is as in ML($*$) for $p \in$ AP, Boolean connectives and $\varphi_1 * \varphi_2$, otherwise

$$\mathfrak{M}, w \models \Diamond^{-1}\varphi \Leftrightarrow \exists w' \text{ s.t. } (w', w) \in R \text{ and } \mathfrak{M}, w' \models \varphi.$$

Since MSL($*, \Diamond^{-1}$) is interpreted over a finite and functional relation, $\Diamond^{-1}$ effectively works as the $\Diamond$ modality of ML($*$). Then, assume we want to check the satisfiability of φ in ML($*$) by relying on an algorithm for Sat(MSL($*, \Diamond^{-1}$)). We simply need to consider the formula $\varphi[\Diamond \leftarrow \Diamond^{-1}]$ obtained from φ by replacing every occurrence of $\Diamond$ by $\Diamond^{-1}$, and check if it can be satisfied by a *locally acyclic* model $(\mathfrak{M}, w)$ of MSL, i.e. one where w does not belong to a loop of length $\leq \mathrm{md}(\varphi)$. Local acyclicity can be enforced by the formula

$$\mathtt{locacycl} \stackrel{\mathrm{def}}{=} r \wedge \bigwedge_{i \in [1, \mathrm{md}(\varphi)]} (\Box^{-1})^i \neg r,$$

where $r \in$ AP is fresh. Then, φ in ML($*$) is satisfiable iff $\varphi[\Diamond \leftarrow \Diamond^{-1}] \wedge \mathtt{locacycl}$ in MSL($*, \Diamond^{-1}$) is satisfiable. Hence, the results in Section 4 allow us to close the complexity gap.

Corollary 6.2. Sat(MSL($*, \Diamond^{-1}$)) *is* TOWER-*complete.*

7 Conclusion

We have studied and compared ML($|$) and ML($*$), two modal logics interpreted on finite forests and featuring composition operators. We have not only characterised the expressive power and the complexity for both logics, but also identified remarkable differences and export our results to other logics. ML($|$) is shown as expressive as GML, and its satisfiability problem is found to be $\mathrm{AExp}_{\mathrm{POL}}$-complete. Besides the obvious similarities between ML($|$) and ML($*$), these results are counter-intuitive: though the logic ML($*$) is strictly less expressive than GML (and consequently, than ML($|$)), Sat(ML($*$)) is TOWER-complete. We also recalled that there are logspace reductions from ML($*$) and ML($|$) to the second-order modal logic QK^t from [7].

Our proof techniques go beyond what is known in the literature. For instance, to design the TOWER-hardness proof we needed substantial modifications from the proof introduced in [7] for QK^t. On the other hand, to show the expressivity inclusion of ML($*$) within GML, we provided a novel definition of Ehrenfeucht-Fraïssé games for ML($*$).

Lastly, our framework led to the characterisation of the satisfiability problems for two sister logics . We proved that the satisfiability problem for the modal separation logic MSL($*, \Diamond^{-1}$) is TOWER-complete [21]. Moreover, the satisfiability problem for the static ambient logic SAL($|$) is $\mathrm{AExp}_{\mathrm{POL}}$-complete, solving open problems from [12, 21] and paving the way to study the complexity of the full SAL.

Acknowledgements

References

[1] C. Areces, R. Fervari, and G. Hoffmann. 2012. Moving Arrows and Four Model Checking Results. In *WoLLIC'12 (LNCS)*, Vol. 7456. Springer, 142–153.

[2] C. Areces, R. Fervari, and G. Hoffmann. 2015. Relation-changing modal operators. *Logic Journal of the IGPL* 23, 4 (2015), 601–627.

[3] G. Aucher, Ph. Balbiani, L. Fariñas del Cerro, and A. Herzig. 2009. Global and Local Graph Modifiers. *Electronic Notes in Theoretical Computer Science* 231 (2009), 293–307.

[4] G. Aucher, J. van Benthem, and D. Grossi. 2018. Modal logics of sabotage revisited. *JLC* 28, 2 (2018), 269–303.

[5] Ph. Balbiani, A. Baltag, H. van Ditmarsch, A. Herzig, T. Hoshi, and T. De Lima. 2008. 'Knowable' as 'known after an announcement'. *Review on Symbolic Logic* 1, 3 (2008), 305–334.

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

[7] B. Bednarczyk and S. Demri. 2019. Why Propositional Quantification Makes Modal Logics on Trees Robustly Hard?. In *LiCS'19*. IEEE, 1–13.

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

[9] I. Boneva, J.-M. Talbot, and S. Tison. 2005. Expressiveness of a Spatial Logic for Trees. In *LiCS'05*. IEEE Computer Society, 280–289.

[10] L. Bozzelli, A. Molinari, A. Montanari, and A. Peron. 2017. On the Complexity of Model Checking for Syntactically Maximal Fragments of the Interval Temporal Logic HS with Regular Expressions. In *GandALF'17 (EPTCS)*, Vol. 256. 31–45.

[11] L. Bozzelli, H. van Ditmarsch, and S. Pinchinat. 2015. The complexity of one-agent refinement modal logic. *TCS* 603 (2015), 58–83.

[12] C. Calcagno, L. Cardelli, and A.D. Gordon. 2003. Deciding validity in a spatial logic for trees. In *TLDI'03*. ACM, 62–73.

[13] C. Calcagno, T. Dinsdale-Young, and Ph. Gardner. 2010. Adjunct elimination in Context Logic for trees. *Inf. Comput.* 208, 5 (2010), 474–499.

[14] C. Calcagno, Ph. Gardner, and U. Zarfaty. 2005. Context logic and tree update. In *POPL'05*. 271–282.

[15] L. Cardelli and A.D. Gordon. 2001. Formal Methods for Distributed Processing. Cambridge University Press, New York, NY, USA, Chapter Mobile Ambients, 198–229.

[16] D. Cooper. 1972. Theorem proving in arithmetic without multiplication. *Machine Learning* 7 (1972), 91–99.

[17] J.-R. Courtault and D. Galmiche. 2018. A modal separation logic for resource dynamics. *JLC* 28, 4 (2018), 733–778.

[18] A. Dawar, Ph. Gardner, and G. Ghelli. 2004. Adjunct Elimination Through Games in Static Ambient Logic. In *FST&TCS'04 (LNCS)*, Vol. 3328. Springer, 211–223.

[19] A. Dawar, Ph. Gardner, and G. Ghelli. 2007. Expressiveness and complexity of graph logic. *I&C* 205, 3 (2007), 263–310.

[20] M. de Rijke. 2000. A Note on Graded Modal Logic. *Studia Logica* 64, 2 (2000), 271–283.

[21] S. Demri and R. Fervari. 2019. The power of modal separation logics. *JLC* 29, 8 (2019), 1139–1184.

[22] S. Demri, R. Fervari, and A. Mansutti. 2019. Axiomatising Logics with Separating Conjunction and Modalities. In *JELIA'19 (LNAI)*, Vol. 11468. Springer, 692–708.

[23] M. Echenim, R. Iosif, and N. Peltier. 2019. The Bernays-Schönfinkel-Ramsey Class of Separation Logic on Arbitrary Domains. In *FoSSaCS'19 (LNCS)*, Vol. 11425. Springer, 242–259.

[24] K. Fine. 1970. Propositional quantifiers in modal logic. *Theoria* 36 (1970), 336–346.

[25] D. Galmiche, D. Méry, and D. Pym. 2005. The Semantics of BI and Resource Tableaux. *Mathematical. Structures in Comp. Sci.* 15, 6 (Dec. 2005), 1033–1088.

[26] Ch. Haase. 2018. A survival guide to Presburger arithmetic. *SIGLOG News* 5, 3 (2018), 67–82.

[27] M. Hannula, J. Kontinen, J. Virtema, and H. Vollmer. 2018. Complexity of Propositional Logics in Team Semantic. *ACM ToCL* 19, 1 (2018), 2:1–2:14.

[28] S. Ishtiaq and P. O'Hearn. 2001. BI as an assertion language for mutable data structures. In *POPL'01*. 14–26.

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

[30] F. Laroussinie and N. Markey. 2014. Quantified CTL: Expressiveness and Complexity. *LMCS* 10, 4:17 (2014).

[31] L. Libkin. 2004. *Elements of Finite Model Theory*. Springer.

[32] É. Lozes. 2004. Adjuncts elimination in the static ambient logic. *Electronic Notes in Theoretical Computer Science* 96 (2004), 51–72.

[33] C. Lutz. 2006. Complexity and succinctness of public announcement logic. In *AAMAS'06*. ACM, 137–143.

[34] A. Mansutti. 2018. Extending Propositional Separation Logic for Robustness Properties. In *FSTTCS (LIPIcs)*, Vol. 122. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 42:1–42:23.

[35] P.W. O'Hearn, J.C. Reynolds, and H. Yang. 2001. Local Reasoning about Programs that Alter Data Structures. In *CSL'01 (LNCS)*, Vol. 2142. Springer, 1–19.

[36] C. H. Papadimitriou. 1994. *Computational complexity*. Addison-Wesley. I–XV, 1–523 pages.

[37] J. Plaza. 1989. Logics of public communication. In *ISMIS'89, Charlotte, North Carolina, USA*.

[38] M. Presburger. 1929. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du premier congrès de mathématiciens des Pays Slaves, Warszawa*. 92–101.

[39] D. Pym. 2002. *The semantics and proof theory of the logic of bunched implications*. Applied Logic, Vol. 26. Kluwer Academic Publishers.

[40] M. Rabin. 1969. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* 41 (1969), 1–35.

[41] C. Reddy and W. Loveland. 1978. Presburger arithmetic with bounded quantifier alternation. In *STOC'78*. ACM press, 320–325.

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

[43] S. Schmitz. 2016. Complexity Hierarchies beyond Elementary. *TOCT* 8, 1 (2016), 3:1–3:36.

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

[45] S. Tobies. 2001. PSPACE Reasoning for Graded Modal Logics. *JLC* 11 (2001), 85–106.

[46] J. van Benthem. 1976. *Modal Correspondence Theory*. Ph.D. Dissertation. University of Amsterdam.

[47] M. Vardi and P. Wolper. 1986. Automata-theoretic techniques for modal logics of programs. *JCSS* 32 (1986), 183–221.

[48] M. Vardi and P. Wolper. 1994. Reasoning about Infinite Computations. *I&C* 115 (1994), 1–37.