The Complexity of Satisfiability for Fragments of Hybrid Logic — Part II

Arne Meier[a] Martin Mundhenk[b] Thomas Schneider[c]
Michael Thomas[a] Felix Weiss[b]

[a] *Universität Hannover, Germany*
{meier,thomas}@thi.uni-hannover.de

[b] *Universität Jena, Germany*
{martin.mundhenk,felix.weiss}@uni-jena.de

[c] *University of Manchester, UK*
schneider@cs.man.ac.uk

Abstract

Hybrid logic is an expressive specification language, but has an undecidable satisfiability problem in general. In this paper, we restrict the set of Boolean operators to monotone operators (for instance conjunction and disjunction) and the underlying frames to commonly used acyclic frames, namely transitive trees, total transitive trees, linear orders, and the natural numbers. We show that, under these restrictions, satisfiability is decidable for 16 fragments arising from different combinations of modal and hybrid operators. More precisely, we categorise these fragments to be PSPACE-complete, NP-complete or tractable, where the latter cases are contained in LOGDCFL or complete for NC^1.

Keywords: satisfiability, modal logic, complexity, hybrid logic

1 Introduction

Hybrid logic is an extension of modal logic with nominals, satisfaction operators and binders. The downarrow binder $\downarrow$, which is related to the freeze operator in temporal logic [12], provides high expressivity. The price paid is the undecidability of the satisfiability problem for the hybrid language with the downarrow binder $\downarrow$ [6,11,1]. In contrast, modal logic, and its extension with nominals and the satisfaction operator, is PSPACE-complete [13,1].

[1] Supported in part by the grants DFG VO 630/6-2, BC-ARC 1323, DAAD-ARC D/08/08881.

In order to regain decidability, syntactic and semantic restrictions have been considered. It has been shown in [24] that the absence of certain combinations of universal operators ($\Box$, $\wedge$) with $\downarrow$ brings back decidability, and that the hybrid language with $\downarrow$ is decidable over frames of bounded width. Furthermore, this language is decidable over transitive and complete frames [19], and over frames with an equivalence relation (ER frames) [18]. Adding @ or the global modality leads to undecidability over transitive frames [19], but not over ER frames [18]. Over linear frames and transitive trees, $\downarrow$ on its own does not add expressivity, but combinations with @ or the global modality do. These languages are decidable and of non-elementary complexity [10,19]; if the number of state variables is bounded, then they are of elementary complexity [23,26,7].

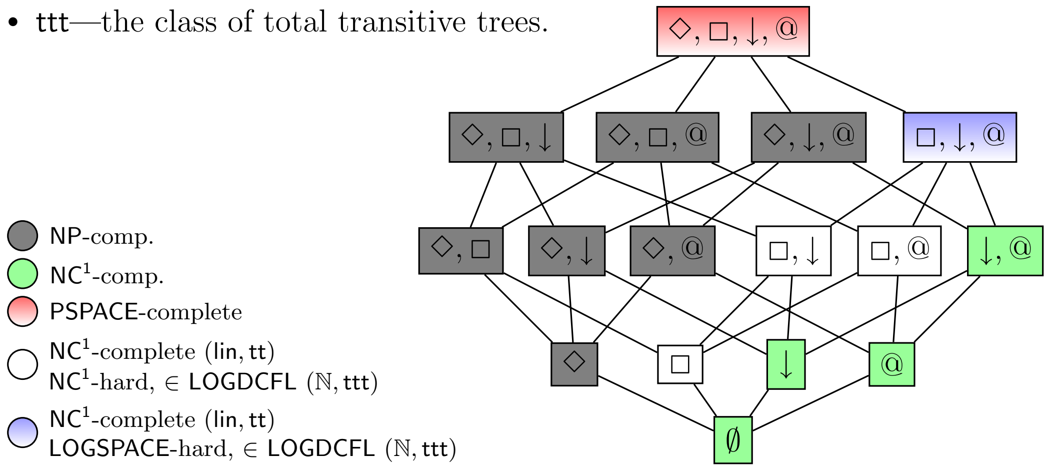
We aim for a more fine-grained distinction between decidable and undecidable fragments of different complexities by systematically restricting the set of Boolean operators and combining this with restrictions to the modal/hybrid operators and to the underlying frames. In [16], we have focussed on four frame classes that allow cycles, and nearly completely classified the decidability and complexity of satisfiability for fragments obtained by arbitrary combinations of Boolean operators and four modal/hybrid operators.

In this paper, we are focussing on acyclic frame classes such as transitive trees, total transitive trees, linear orders and the natural numbers. These are important for representing the flow of time. Over acyclic frames, the $\downarrow$ operator only adds expressivity to the modal language if it appears in combination with @, past operators, or the global modality because otherwise named states cannot be re-visited. Over the named frame classes, even the language with $\downarrow$ and the global modality is decidable, although of non-elementary complexity [10,19]. Hence, we are interested in differentiating between fragments with high and low complexity. We have observed that the complexity varies sharply among the many fragments, i.e., a single lower or upper bound carries over to another set of operators or another frame classes less often than it usually does over cyclic frame classes according to [16]. Therefore, the fragments over acyclic frame classes require a more separate treatment, and we are restricting this study to one prominent class of Boolean operators, namely monotone operators. A Boolean function is monotone if changing one of its arguments from 0 to 1 does not decrease its value. Every Boolean operator corresponding to a monotone Boolean function, for instance n-ary conjunction and disjunction, is called monotone.

It has been shown in the context of the temporal logics CTL and LTL that the complexity of the model-checking problem does not decrease if Boolean operators are restricted to monotone ones [3,5], but satisfiability becomes tractable [4,17]. In the presence of expressive hybrid operators, which give the satisfiability problem non-elementary complexity, it is not clear, and therefore interesting to find out, whether the restriction of Boolean operators leads to a significant decrease in complexity.

In this study, we will completely classify the computational complexity of satisfiability for fragments of hybrid logic with monotone Boolean operators and arbitrary combinations of the operators $\Diamond$, $\Box$, $\downarrow$ and @ over the above listed frame classes. We will prove all these satisfiability problems to be decidable and classify them into PSPACE-complete, NP-complete and tractable. We will further show that the tractable problems can be solved in LOGDCFL—the set of problems logarithmically reducible to deciding membership in a deterministic context free language—or are NC^1-complete. Our results are shown in Figure 1, which covers the following frame classes.

- $\mathbb{N}$—the class that consists of the frame $(\mathbb{N}, <)$,
- lin—the class of linear frames,
- tt—the class of transitive trees, and
- ttt—the class of total transitive trees.



$\Diamond, \Box, \downarrow, @$
$\Diamond, \Box, \downarrow$
$\Diamond, \Box, @$
$\Diamond, \downarrow, @$
$\Box, \downarrow, @$
$\Diamond, \Box$
$\Diamond, \downarrow$
$\Diamond, @$
$\Box, \downarrow$
$\Box, @$
$\downarrow, @$
$\Diamond$
$\Box$
$\downarrow$
$@$
$\emptyset$
NP-comp.
NC^1-comp.
PSPACE-complete
NC^1-complete (lin, tt)
NC^1-hard, $\in$ LOGDCFL ($\mathbb{N}$, ttt)
NC^1-complete (lin, tt)
LOGSPACE-hard, $\in$ LOGDCFL ($\mathbb{N}$, ttt)

Fig. 1. Our complexity results for satisfiability over $\mathbb{N}$, lin, tt and ttt with monotone Boolean operators and different combinations of modal/hybrid operators

2 Preliminaries

Hybrid Logic. In the following, we will introduce the notions and definitions of hybrid logic. The terminology is largely taken from [2].

Let PROP be a countable set of *atomic propositions*, NOM be a countable set of *nominals*, SVAR be a countable set of *variables* and ATOM = PROP $\cup$ NOM $\cup$ SVAR. We will stick with the common practice to denote atomic propositions by $p, q, \ldots$, nominals by $i, j, \ldots$, and variables by $x, y, \ldots$. We define the language of *hybrid (modal) logic* $\mathcal{HL}$ as the set of well-formed formulae of the form

$$\varphi ::= a \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \downarrow x.\varphi \mid @_t\varphi$$

where $a \in \mathsf{ATOM}$, $x \in \mathsf{SVAR}$ and $t \in \mathsf{NOM} \cup \mathsf{SVAR}$.

Formulae of $\mathcal{HL}$ are interpreted on *(hybrid) Kripke structures* $K = (W, R, \eta)$, consisting of a set of *states* W, a *transition relation* $R\colon W \times W$, and a *labeling function* $\eta\colon \mathsf{PROP} \cup \mathsf{NOM} \to \wp(W)$ that maps PROP and NOM to subsets of W with $|\eta(i)| = 1$ for all $i \in \mathsf{NOM}$. In order to evaluate $\downarrow$-formulae, an assignment $g\colon \mathsf{SVAR} \to W$ is necessary. Given an assignment g, a state variable x and a state w, an *x-variant g^x_w of g* is defined by $g^x_w(x) = w$ and $g^x_w(x') = g(x')$ for all $x \neq x'$. For any $a \in \mathsf{ATOM}$, let $[\eta, g](a) = \{g(a)\}$ if $a \in \mathsf{SVAR}$ and $[\eta, g](a) = \eta(a)$, otherwise. The satisfaction relation of hybrid formulae is defined as follows.

$$\begin{array}{lll}
K, g, w \models a & \text{iff} & w \in [\eta, g](a),\ a \in \mathsf{ATOM}, \\
K, g, w \models \top, & & \\
K, g, w \not\models \bot, & & \\
K, g, w \models \neg\varphi & \text{iff} & K, g, w \not\models \varphi, \\
K, g, w \models \varphi \wedge \psi & \text{iff} & K, g, w \models \varphi \text{ and } K, g, w \models \psi, \\
K, g, w \models \varphi \vee \psi & \text{iff} & K, g, w \models \varphi \text{ or } K, g, w \models \psi, \\
K, g, w \models \Diamond\varphi & \text{iff} & \exists w' \in W(wRw' \,\&\, K, g, w' \models \varphi), \\
K, g, w \models \Box\varphi & \text{iff} & \forall w' \in W(wRw' \Rightarrow K, g, w' \models \varphi), \\
K, g, w \models @_t\varphi & \text{iff} & K, g, [\eta, g](t) \models \varphi, \\
K, g, w \models \downarrow x.\varphi & \text{iff} & K, g^x_w, w \models \varphi.
\end{array}$$

A hybrid formula φ is said to be *satisfiable* if there exists a Kripke structure $K = (W, R, \eta)$, a $w \in W$ and an assignment $g\colon \mathsf{SVAR} \to W$ with $K, g, w \models \varphi$.

The *at* operator $@_t$ shifts evaluation to the state named by $t \in \mathsf{NOM} \cup \mathsf{SVAR}$. The *downarrow binder* $\downarrow x.$ binds the state variable x to the current state. The symbols $@_x$, $\downarrow x.$ are called *hybrid operators* whereas the symbols $\Diamond$ and $\Box$ are called *modal operators*.

The scope of an occurrence of the binder $\downarrow$ is defined as usual. For a state variable x, an occurrence of x or $@_x$ in a formula φ is called *bound* if this occurrence is in the scope of some $\downarrow$ in φ, *free* otherwise. φ is said to contain a free state variable if some x or $@_x$ occurs free in φ.

Given two formulae φ, α and a subformula ψ of φ, we use $\varphi[\psi/\alpha]$ to denote the result of replacing each occurrence of ψ in φ with α. If we want to replace only one previously specified occurrence of ψ, we write $\varphi[\psi//\alpha]$.

For considering fragments of hybrid logics, we define subsets of the language $\mathcal{HL}$ as follows. Let O be a set of hybrid and modal operators, i.e., a subset of $\{\Diamond, \Box, \downarrow, @\}$. We define $\mathcal{HL}(O)$ to denote the set of well-formed hybrid formulae using only the operators in O, and $\mathcal{MHL}(O)$ to be the set of all formulae in $\mathcal{HL}(O)$ that do not use $\neg$.

Properties of Frames. A *frame* F is a pair (W, R), where W is a set of states and $R \subseteq W \times W$ a transition relation. A frame $F = (W, R)$ is called

- transitive if R is transitive (for all $u, v, w \in W$: $uRv \land vRw \to uRw$),
- linear if R is transitive and trichotomous ($\forall u, v \in W$: uRv or $u = v$ or vRu),
- a tree if (W, R) is acyclic and connected, and every state in W has at most one R-predecessor,
- a transitive tree if R is the transitive closure of S such that (W, S) is a tree,
- a total transitive tree if F is a tree and total (each $u \in W$ has an R-successor).

In this paper we will consider the frame classes listed on Page 3. Furthermore, all stands for the class of all frames.

The Satisfiability Problem. Let $K = (W, R, \eta)$ be a Kripke structure. Say that K *is based on a frame* F iff F is the frame underlying K, i. e., $F = (W, R)$. The *satisfiability problem* $\mathfrak{F}\text{-}\mathsf{SAT}(O)$ is defined as follows: given an $\mathcal{HL}(O)$-formula φ, is there a Kripke structure $K = (W, R, \eta)$ based on a frame from $\mathfrak{F}$, an assignment $g\colon \mathsf{SVAR} \to W$ and a $w \in W$ such that $K, g, w \models \varphi$? The monotone satisfiability problem $\mathfrak{F}\text{-}\mathsf{MSAT}(O)$ is defined analogously, with $\mathcal{HL}(O)$ replaced by $\mathcal{MHL}(O)$. In case $\mathfrak{F} = \mathsf{all}$, we will omit the prefix and simply write $\mathsf{SAT}(O)$ or $\mathsf{MSAT}(O)$. Furthermore, we will often omit the set parentheses when giving O explicitly, e.g., $\mathsf{SAT}(\Diamond, \Box, \downarrow, @)$.

Technical assumptions. In order to make proofs of our results easier, we assume that any instance of a satisfiability problem contains no free state variables. This does not restrict generality because free state variables can be simulated by nominals. Furthermore, we assume that, for each state variable x such that x or $@_x$ occurs in φ, x is bound exactly once by $\downarrow$ in φ. This is no significant restriction as well because multiply bound variables can be named apart, which is a well-established and computationally easy procedure.

Complexity Theory. We assume familiarity with the standard notions of complexity theory as, e. g., defined in [20]. In particular, we will make use of the classes $\mathsf{LOGSPACE}$, $\mathsf{NLOGSPACE}$, NP, PSPACE, and coRE. The complexity class $\mathsf{NONELEMENTARY}$ is the set of all languages A that are decidable and for which there exists no $k \in \mathbb{N}$ such that A can be decided using an algorithm whose running time is bounded by $\exp_k(n)$, where $\exp_k(n)$ is the k-th iteration of the exponential function (e.g., $\exp_3(n) = 2^{2^{2^n}}$).

Furthermore, we will need two non-standard complexity classes whose definition relies on circuit complexity and formal languages, see for instance [25,14]. The class NC^1 is defined as the set of languages recognizable by a logtime-uniform Boolean circuits of logarithmic depth and polynomial size over $\{\land, \lor, \neg\}$, where the fan-in of $\land$ and $\lor$ gates is fixed to 2. The class $\mathsf{LOGDCFL}$ is defined as the set of languages reducible in logarithmic space to some deterministic context-free language.

The following relations between the considered complexity classes are known. $\mathsf{NC}^1 \subseteq \mathsf{LOGSPACE} \subseteq \mathsf{LOGDCFL} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subset \mathsf{coRE}$. It is unknown whether LOGDCFL contains NLOGSPACE or vice versa.

A language A is *constant-depth reducible* to D, $A \leqslant_{\mathrm{cd}} D$, if there is a logtime-uniform AC^0-circuit family with oracle gates for D that decides membership in A. Unless otherwise stated, all reductions in this paper are $\leqslant_{\mathrm{cd}}$-reductions.

Known results. The following theorem summarizes results for hybrid languages with Boolean operators $\wedge, \vee, \neg$ that are known from the literature. Since $\Box\varphi = \neg\Diamond\neg\varphi$, the $\Box$-operator is implicitly present in all fragments containing $\Diamond$.

Theorem 2.1 ([1,2,15,19,21])

(1) $\mathsf{SAT}(\Diamond, \downarrow)$ *and* $\mathsf{SAT}(\Diamond, \downarrow, @)$ *are* coRE*-complete.*

(2) $\mathfrak{F}$-$\mathsf{SAT}(\Diamond, \downarrow)$ *and* $\mathfrak{F}$-$\mathsf{SAT}(\Diamond, \downarrow, @)$*, for* $\mathfrak{F} \in \{\mathsf{tt}, \mathsf{lin}, \mathbb{N}\}$*, are in* NONELEMENTARY*.*

(3) tt-$\mathsf{SAT}(\Diamond)$ *and* tt-$\mathsf{SAT}(\Diamond, @)$ *are* PSPACE*-complete.*

(4) $\mathfrak{F}$-$\mathsf{SAT}(\Diamond)$ *and* $\mathsf{SAT}(\Diamond, @)$*, with* $\mathfrak{F} \in \{\mathsf{lin}, \mathbb{N}\}$*, are* NP*-complete.*

3 Results for Monotone Fragments

3.1 PSPACE *results*

We start by analyzing only the frame class lin. It will be easy to carry over the results to the other acyclic classes. For an arbitrary $\varphi \in \mathsf{lin}$-$\mathsf{MSAT}(\Diamond, \Box, \downarrow, @)$ with $|\varphi| = n$, we will show the existence of a model K which either is of size linear in n or has a "prefix" of at most n states after which all states agree in η. This result will enable us to show PSPACE-membership of lin-$\mathsf{MSAT}(\Diamond, \Box, \downarrow, @)$. PSPACE-hardness follows from a reduction from QBFSAT.

We first define the set $\mathfrak{K}_\varphi$ which contains all linear models that match the above description.

Definition 3.1 Let $\varphi \in \mathcal{MHL}(\Diamond, \Box, \downarrow, @)$ and $|\varphi| = n$. We define the linear Kripke structures L^m ($m \in \mathbb{N}$), L^∞ and the sets $\mathfrak{K}_\varphi^{\leqslant n}$, $\mathfrak{K}_\varphi^\infty$, $\mathfrak{K}_\varphi$ of Kripke structures as follows:

$$\begin{aligned}
L^m &= (\{1, \dots, m\}, <) \qquad L^\infty = (\mathbb{N}, <) \\
\mathfrak{K}_\varphi^{\leqslant n} &= \{(W, R, \eta) \mid (W, R) = L^m \text{ for some } m \leqslant n\} \\
\mathfrak{K}_\varphi^\infty &= \{(W, R, \eta) \mid (W, R) = L^\infty \text{ and} \\
&\qquad \forall a \in \mathsf{PROP} \cup \mathsf{NOM}\, \forall i, j \geq n\, (i \in \eta(a) \Leftrightarrow j \in \eta(a)\} \\
\mathfrak{K}_\varphi &= \mathfrak{K}_\varphi^{\leqslant n} \cup \mathfrak{K}_\varphi^\infty
\end{aligned}$$

Now let $\varphi \in \mathsf{lin}$-$\mathsf{MSAT}(\Diamond, \Box, \downarrow, @)$, $K = (W, R, \eta)$ be a linear model, $w \in W$ a state and g an assignment with $K, g, w \models \varphi$. We give a construction that leads to $K' = (W', R', \eta')$ with $K', g, w \models \varphi$. Algorithm 1 returns on input (φ, g, w) a subset V of W such that $|V| \leqslant n$. Informally speaking, V contains all states that are relevant for the evaluation of φ over K. We set $W' = V \cup \{w\} \cup \{w \mid w \in \eta(i)$ *and* $i \in \mathsf{NOM}$ *that occurs in* $\varphi\}$. The transition relation R' is the restriction of R to W', and the labeling function η' is the restriction of η to W'. In case W is infinite, let v be the last state in the order imposed by R'. We add an infinite chain $vR'u_1R'u_2R' \dots$ to (W', R') and add u_i to $\eta(p)$ for each $p \in \mathsf{PROP}$. Now the following holds.

Algorithm 1 Function *relev-states*

```
Require: a formula $\psi$, an assignment $f$, a state $v$
 1: if $\psi \in \mathsf{ATOM} \cup \{\bot, \top\}$ then
 2:     $w_\psi := v$, $g_\psi := f$
 3:     return $\emptyset$
 4: else if $\psi = \Box\alpha$ then
 5:     $w_\psi := v$, $g_\psi := f$
 6:     return $\emptyset$
 7: else if $\psi = \alpha \wedge \beta$ then
 8:     $w_\psi := v$, $g_\psi := f$
 9:     return relev-states($\alpha$,$f$,$v$) $\cup$ relev-states($\beta$,$f$,$v$)
10: else if $\psi = \alpha \vee \beta$ then
11:     $w_\psi := v$, $g_\psi := f$
12:     if $K, v, f \models \alpha$ then return relev-states($\alpha$,$f$,$v$)
13:     else if $K, v, f \models \beta$ then return relev-states($\beta$,$f$,$v$)
14:     end if
15: else if $\psi = \Diamond\alpha$ then
16:     $w_\psi := v$, $g_\psi := f$
17:     choose a state $v' \in W$ with $vRv'$ and $K, v', f \models \alpha$
18:     return relev-states($\alpha$,$f$,$v'$) $\cup \{v'\}$
19: else if $\psi = @_t\alpha$ then
20:     $w_\psi := v$, $g_\psi := f$
21:     return relev-states($\alpha$,$f$,$[\eta, f](t)$)
22: else if $\psi = \downarrow x.\alpha$ then
23:     $w_\psi := v$, $g_\psi := f$
24:     return relev-states($\alpha$,$f^x_v$,$v$)
25: end if
```

Lemma 3.2 *Let K' be defined as above. Then*

(i) *$K' \in \mathfrak{K}_\varphi$, and*

(ii) *$K', g, w \models \varphi$.*

Proof. For (1), observe that, due to its construction, K' satisfies the following. If W is finite, then W' consists of at most $\#\Diamond(\varphi) + \#\mathrm{nom}(\varphi) \leqslant n$ states, where $\#\Diamond(\varphi)$ (resp. $\#\mathrm{nom}(\varphi)$) is the number of $\Diamond$-operators (resp. nominals) occurring in φ. Therefore, $K' \in \mathfrak{K}_\varphi^{\leqslant n}$ in this case. If W is infinite, then only states that agree in $\eta(a)$ for $a \in \mathsf{PROP} \cup \mathsf{NOM}$ are attached, hence $K' \in \mathfrak{K}_\varphi^\infty$. We show (2) with the following claim.

Claim 3.3 *Let* $\mathrm{Sub}(\varphi)$ *be the set of all subformulae of* φ*. For every* $\psi \in \mathrm{Sub}(\varphi)$*, it holds that if* $K, g_\psi, w_\psi \models \psi$*, then* $K', g_\psi, w_\psi \models \psi$*, where* w_ψ *and* g_ψ *are calculated in Alg. 1.*

Proof of Claim. We prove this claim by induction on φ. For the initial step we assume that $\psi \in \mathsf{ATOM} \cup \{\bot, \top\}$. In this case the claim is true because η' is the restriction of η on W'. For the induction step we distinguish several cases.

- The cases $\psi = \alpha \wedge \beta$ or $\psi = \alpha \vee \beta$ are trivial.
- Case $\psi = \Diamond\alpha$:
  $K, g_\psi, w_\psi \models \Diamond\alpha$ if and only if there is a v' with $w_\psi R v'$ and $K, g_\psi, v' \models \alpha$. For the chosen v', it holds that $w_\alpha = v'$ as in Algorithm 1 (The state v' will be added in Line 18). In order to satisfy α we can also use g_ψ as assignment, so $g_\alpha = g_\psi$. The induction hypothesis yields $K', g_\alpha, w_\alpha \models \alpha$. So $K', g_\psi, v' \models \alpha$ and $K', g_\psi, w_\psi \models \Diamond\alpha$.
- Case $\psi = {\downarrow}x.\alpha$:
  $K, g_\psi, w_\psi \models {\downarrow}x.\alpha$ iff $K, (g_\psi)^x_{w_\psi}, w_\psi \models \alpha$.
  Clearly $(g_\psi)^x_{w_\psi} = g_\alpha$ and $w_\psi = w_\alpha$. The induction hypothesis yields $K', (g_\psi)^x_{w_\psi}, w_\psi \models \alpha$. This leads to $K', g_\psi, w_\psi \models {\downarrow}x.\alpha$.
- Case $\psi = \Box\alpha$:

$$\begin{aligned}
&K, g_\psi, w_\psi \models \Box\alpha \\
&\quad\Leftrightarrow\ \forall w' \in W\ (w_\psi R w \Rightarrow K, g_\psi, w' \models \alpha) \\
&\quad\Rightarrow\ \forall w' \in W'\ (w_\psi R w \Rightarrow K, g_\psi, w' \models \alpha) \\
&\quad\Rightarrow\ \forall w' \in W'\ (w_\psi R w \Rightarrow K', g_\psi, w' \models \alpha)
\end{aligned}$$

  Correctness of the last implication follows from the monotonicity of ψ and the fact that for every $v \in W \setminus W'$ and every $u \in W'$, it holds that $\eta^{-1}(v) \subseteq \eta^{-1}(u)$. This eventually leads to $K', g_\psi, w_\psi \models \Box\alpha$. $\Diamond\!\!\!\Box$

We thus know that for all $\varphi \in \mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow, @)$, there exists a model K, a state w in K, and an assignment g such that $K, g, w \models \varphi$, and K either is of linear size in $|\varphi|$ or K's states are indistinguishable except for a prefix of size linear in $|\varphi|$. Using this result, we obtain the following proposition.

Proposition 3.4 $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow, @)$ *is in* PSPACE*.*

Proof. If φ is satisfiable in a linear model, then there exists a model $K' \in \mathfrak{K}_\varphi$ that satisfies φ. Conversely, if φ is not satisfiable in any linear model, then it cannot be satisfiable in any of the linear models in $\mathfrak{K}_\varphi$. Therefore, the following algorithm decides $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow, @)$.

> Given φ, nondeterministically guess a $K \in \mathfrak{K}_\varphi$, a state w from K and an assignment g and verify whether $K, g, w \models \varphi$.

This is an $\mathsf{NPSPACE} = \mathsf{PSPACE}$ algorithm for the following reasons. Let $|\varphi| = n$. If we represent all infinite models in $\mathfrak{K}_\varphi$ using their prefix of size n and an additional bit whether the last state occurs on an infinite chain, such a finite representation can be guessed in $\mathcal{O}(n)$ steps, the same for all finite models in $\mathfrak{K}_\varphi$. Remember that, for guessing η, it suffices to guess the assignment only for symbols occurring in φ. Now model checking on *finite* structures has been shown to be PSPACE-complete in [9]. In order to accommodate the finitely represented infinite structures, it suffices to make a simple adjustment to the techniques underlying the model checkers `MCLITE` and `MCFULL` in [10]. This adjustment consists in modifying the definition of $R^{-1}(w)$ for the repeatedly occurring state w in a way that $R^{-1}(w)$ contains w. With this modification, the original correctness and complexity proofs in [10] go through. $\Box$

In the next step we use a reduction from QBFSAT to show PSPACE-hardness of $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow, @)$.

Proposition 3.5 $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow, @)$ *is* PSPACE*-hard.*

Proof. We reduce from QBFSAT, the problem to decide whether a given quantified Boolean formula is valid. Let φ be an instance of QBFSAT and assume w.l.o.g. that negations occur only directly in front of atomic propositions. We define the transformation as $f\colon \varphi \mapsto {\downarrow} r.\Diamond{\downarrow} s.\Diamond h(\varphi)$ where h is given as follows: let ψ, χ be quantified Boolean formulae and let x_k be a variable in φ, then

$$\begin{aligned} h(\exists x_k \psi) &:= @_r\Diamond{\downarrow} x_k.h(\psi), & h(\forall x_k \psi) &:= @_r\Box{\downarrow} x_k.h(\psi),\\ h(\psi \wedge \chi) &:= h(\psi) \wedge h(\chi), & h(\psi \vee \chi) &:= h(\psi) \vee h(\chi),\\ h(\neg x_k) &:= @_s\Diamond x_k, & h(x_k) &:= @_s x_k. \end{aligned}$$

For example, the QBF $\psi = \forall x \exists y (x \wedge y) \vee (\neg x \wedge \neg y)$ is mapped to $f(\varphi) = {\downarrow} r.\Diamond{\downarrow} s.\Diamond @_r\Box{\downarrow} x_0.@_r\Diamond{\downarrow} x_1.(@_s x_0 \wedge @_s x_1) \vee (@_s\Diamond x_0 \wedge @_s\Diamond x_1)$. Intuitively, this construction requires the existence of an initial state named r, a successor state s that represents the truth value $\top$, and one or more successor states of r which together represent $\bot$. The quantifiers $\exists, \forall$ are replaced by the modal operators $\Diamond, \Box$ which range over s and its successor states. Finally, positive literals are enforced to be true at s, negative literals strictly after s.

For every model of $f(\varphi)$, it holds that r is situated at the first state of the model and that state has a successor labelled by s. By virtue of the function h, positive literals have to be mapped to s, whereas negative literals have to be mapped to some state other than s. An easy induction on the structure of formulae shows that $\varphi \in$ QBFSAT if and only if $f(\varphi) \in \mathsf{lin\text{-}MSAT}(\downarrow, @, \Diamond, \Box)$.$\Box$

We immediately obtain:

Theorem 3.6 $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow, @)$ *is* PSPACE*-complete.*

The result of Theorem 3.6 carries over to the frame class $\mathbb{N}$ directly. It will also carry over to tt and ttt with a few adaptations: we have to modify the definition of $\mathfrak{K}_\varphi$ in a way that it consists of (a) all finite trees of size at most $|\varphi|$ and (b) all such finite trees with each branch extended by an infinite path of indistinguishable states. Now the proof of Lemma 3.2 and the preceding construction of K' go through for trees as well, since the "tree prefix" of K' is still linear in the size of the underlying formula. The proofs of Propositions 3.5 and 3.4 go through immediately.

Corollary 3.7 $\mathfrak{F}$-$\mathsf{MSAT}(\Diamond, \Box, \downarrow, @)$ *is* PSPACE*-complete for* $\mathfrak{F} \in \{\mathbb{N}, \mathsf{tt}, \mathsf{ttt}\}$.

3.2 NP *results*

We start by proving NP-hardness of $\mathsf{lin\text{-}MSAT}(\Diamond)$ and will then generalize this result to the other three acyclic frame classes. Recall that, owing to the presence of nominals, $\mathcal{MHL}(\Diamond)$ is not modal logic with the $\Diamond$-operator. As we furthermore have no binder $\downarrow$, we will write $K, w \models \varphi$ instead of $K, g, w \models \varphi$ (the assignment g is irrelevant).

Lemma 3.8 $\mathsf{lin\text{-}MSAT}(\Diamond)$ *is* NP*-hard.*

Proof. We reduce from 3SAT. Let $\varphi = c_1 \wedge \ldots \wedge c_n$ be an instance of 3SAT with clauses $c_1, \ldots, c_n$ and variables $x_1 \ldots x_m$. We define the transformation as

$$f \colon \varphi \mapsto \psi^1 \wedge \left(\bigwedge_{\ell=0}^{n} \psi_\ell^2 \right) \wedge h(\varphi),$$

where each x_ℓ is a nominal, $\psi^1 := \Diamond(i_0 \wedge \Diamond i_1)$, $\psi_\ell^2 := \Diamond(i_0 \wedge x_\ell) \vee \Diamond(i_1 \wedge x_\ell)$, and the function h is defined as follows: let l_k^j be a literal in clause c_j, then

$$\begin{aligned} h(l_k^j) &:= \begin{cases} (i_1 \wedge x) \ \textit{if}\ l_k^j = x, \\ (i_0 \wedge x) \ \textit{if}\ l_k^j = \neg x; \end{cases} \\ h(c_j) &:= \Diamond(h(l_1^j) \vee h(l_2^j) \vee h(l_3^j)), \quad \text{where } c_j = (l_1^j \vee l_2^j \vee l_3^j); \\ h(c_1 \wedge \cdots \wedge c_n) &:= h(c_1) \wedge \cdots \wedge h(c_n). \end{aligned}$$

Using ψ^1 we enforce the existence of two successors w_1 and w_2 of the state satisfying φ. The subformulae ψ_ℓ^2 simulate the assignment of the variables in φ, enforcing that each x_ℓ is true in either w_1 or w_2. With the following claim NP-hardness of $\mathsf{lin\text{-}MSAT}(\Diamond)$ follows.

Claim 3.9 *$\varphi \in$ 3SAT iff $h(\varphi) \in \mathsf{lin\text{-}MSAT}(\Diamond)$.*

Proof of Claim. We will first show that $h(\varphi) \in \mathsf{lin\text{-}MSAT}(\Diamond)$ implies $\varphi \in$ 3SAT. If $K, w_0 \models h(\varphi)$ with $K = (W, R, \eta)$ based on a linear frame, then the following applies to K. Let $\{w_1\} = \eta(i_0)$, $\{w_2\} = \eta(i_1)$, and

- $\{w_0, w_1, w_2\} \subseteq W$ with w_0, w_1, w_2 pairwise different;
- $\{(w_0, w_1), (w_1, w_2)\} \subseteq R$; and
- for all x_j with $1 \leqslant j \leqslant m : \eta(x_j) \subseteq \{w_1, w_2\}$

We build a propositional logic assignment $\beta = (\beta_1 \dots \beta_m)$ that satisfies φ, where $\beta_i \in \{\bot, \top\}$ is the truth value for x_i, as follows. $\beta_j = \bot$ if $\eta(i_0) = \eta(x_j)$, and $\beta_j = \top$ if $\eta(i_1) = \eta(x_j)$. From the construction of $h(\varphi)$, it clearly follows that β satisfies φ. For the other direction, suppose that φ is satisfied by the propositional logic assignment $\beta = (\beta_1 \dots \beta_m)$. We construct a linear model $K := (W, R, \eta)$ containing a state w such that $K, w \models h(\varphi)$.

$$W := \{w, w_0, w_1\} \qquad \eta(i_j) := \{w_j\} \textit{ for } j \in \{0, 1\}$$

$$R := \{(w, w_0), (w_0, w_1), (w, w_1)\} \qquad \eta(x_j) := \begin{cases} \{w_0\} & \textit{if } \beta_j = \bot \\ \{w_1\} & \textit{if } \beta_j = \top \end{cases}$$

It follows from the construction of K that $K, w \models h(\varphi)$. The conjunct $h(\varphi)$ is of the form $(h(l_1^1) \vee h(l_1^2) \vee h(l_1^3)) \wedge \cdots \wedge (h(l_n^1) \vee h(l_n^2) \vee h(l_n^3))$. Hence, under assignment β, at least one literal in every clause evaluates to true. The variable in this literal satisfies the same clause in $h(\varphi)$. Hence every clause in $h(\varphi)$ is satisfied in w in K. So it holds that $K, w \models h(\varphi)$. $\Diamond\!\!\!\Box$

We will now show that the problems $\mathsf{lin\text{-}MSAT}(\Diamond, \downarrow, @)$, $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, @)$, and $\mathsf{lin\text{-}MSAT}(\Diamond, \Box, \downarrow)$ are in NP.

Lemma 3.10 *$\mathsf{lin\text{-}MSAT}(\Diamond, \downarrow, @)$ is in NP.*

Proof. We will first show that every instance of $\mathsf{lin\text{-}MSAT}(\Diamond, \downarrow, @)$ can be easily transformed into an equisatisfiable binder-free formula. We will then show that the $\vee$-connective can be eliminated, too. This will result in a nondeterministic polynomial-time decision procedure.

Let φ be an instance of $\mathsf{lin\text{-}MSAT}(\Diamond, \downarrow, @)$. Observe that if there exists $\downarrow x.$ in φ, then we may delete $\downarrow x.$ from φ and replace all occurrences of x in the scope of $\downarrow x.$ with a fresh nominal x'. This is possible because we have no $\Box$-operator and x is hence bound to a unique state. Moreover, if $K, g, w \models \varphi$ and g' is the assignment obtained from g after binding x to w, set $\eta(x') = g'(x)$. Hence we may w.l.o.g. assume that φ is $\downarrow$-free and omit the assignment g, *i.e.*, we will write $K, w \models \varphi$ instead of $K, g, w \models \varphi$.

We will now show that φ is satisfiable if and only if there exists a way of choosing one disjunct for each disjunction in φ, such that the $\vee$-free formula resulting from φ by omitting all non-chosen disjuncts is satisfiable. The intuitive reason for the correctness of this procedure is the absence of $\Box$: with $\Box$, there are formulae $\Box(\alpha \vee \beta)$ whose satisfaction in a state w depends on each of α, β being true in some, but not all, of the states behind w. Deleting one of the disjuncts would force α or β to be true in all states behind w. Our construction works because this problem does not occur without $\Box$.

We construct a set M_φ which consists of all $\vee$-free formulae obtained from φ as described above. We generate this set inductively and start with $M_\varphi = \{\varphi\}$. For every formula ψ that contains a subformula $\alpha \vee \beta$, remove ψ from M_φ and add ψ_α and ψ_β where $\psi_\alpha = \psi[(\alpha \vee \beta)//\alpha]$ and $\psi_\beta = \psi[(\alpha \vee \beta)//\beta]$. Repeat this as long as M_φ still contains a $\vee$-formula. Now φ is satisfiable if some formula in M_φ is satisfiable.

Claim 3.11 $\varphi \in \textsf{lin-MSAT}(\Diamond, \downarrow, @)$ *if and only if there is a satisfiable* $\gamma \in M_\varphi$.

Proof of Claim. It suffices to consider a single contruction step of M_φ. We prove that for an arbitrary $\psi \in M_\varphi$ containing $\alpha \vee \beta$, ψ is satisfiable if and only if ψ_α or ψ_β is satisfiable. (Note: M_φ contains $\vee$-subformulae as long as the construction not ended.)

If ψ_α is satisfiable in some model K and state w, then K, w also satisfies ψ_α with α replaced by $\alpha \vee \beta$. Hence ψ is satisfiable. The case that ψ_β is satisfiable is analogous.

For the other direction, let $K = (W, R, \eta)$ be a model and w a state with $K, w \models \varphi$. For every subformula ψ that needs to evaluate to true in order for $K, w \models \varphi$ to hold, we define w_ψ as follows:

- $w_\varphi := w$;
- if $\psi = \alpha \vee \beta$, then $w_\alpha := w_\psi$ if $K, w_\psi \models \alpha$, and $w_\beta := w_\psi$ if $K, w_\psi \models \beta$;
- if $\psi = \alpha \wedge \beta$, then $w_\alpha := w_\psi$ and $w_\beta := w_\psi$;
- if $\psi = \Diamond\alpha$, then $w_\alpha := v$ for some v with $w_\psi R v$ and $K, v \models \alpha$;
- if $\psi = @_x\alpha$, then $w_\alpha := v$ with $\{v\} = \eta(x)$.

If we evaluate φ recursively in w, then w_ψ is the state for the evaluation of the subformula ψ of φ and it holds that $K, w_\psi \models \psi$. Let $\psi = \alpha \vee \beta$ be a subformula of φ that has to evaluate to true for $K, w \models \varphi$. Then $K, w_\psi \models \alpha \vee \beta$. This leads to $K, w_\psi \models \alpha$ or $K, w_\psi \models \beta$. So clearly $K, w \models \psi_\alpha$ or $K, w \models \psi_\beta$. (For subformulae that do not necessarily have to evaluate to true for $K, w \models \varphi$, this holds obviously as well.)

Now if φ is satisfiable, the algorithm generates at least one satisfiable formula in every cycle of the while loop, and therefore M_φ contains a satisfiable formula when the algorithm is finished. $\Diamond$

Now let $\psi \in M_\varphi$; ψ does not contain any $\vee$. Due to the proof of Proposition 3.4, ψ is satisfiable in a linear model if and only if there exists a model $K' \in \mathfrak{K}_\psi$.

Furthermore, in the absence of $\Box$, we can even assume w.l.o.g. that $K' \in \mathfrak{K}_{\psi}^{\leqslant n}$. In particular, K' is finite.

To decide $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond, \downarrow, @)$, we can nondeterministically guess $\psi \in M_\varphi$, a model K and a state w, and verify whether $K, g, w \models \psi$ for an arbitrary assigment g. Now the verification step can be done with a straightforward nondeterministic polynomial-time algorithm because ψ contains no $\Box$ and every state variable that occurs in ψ is bound to exactly one state. Every time a subformula $\Diamond\alpha$ is reached during the evaluation of ψ, a state is guessed and α evaluated there. $\Box$

Lemma 3.12 $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond, \Box, @)$ *is in* NP.

Proof. We use the same algorithm as in the proof of Proposition 3.4. The only difference is that we have no binder here and therefore the verification step can be done in polynomial time due to [9]. $\Box$

Furthermore, the following is known:

Proposition 3.13 *[10]:* $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond, \Box, \downarrow)$ *is in* NP.

We now have all NP results for lin and can state the following theorem.

Theorem 3.14 *Let* $\{\Diamond\} \subseteq O$, *and* $O \subseteq \{\Diamond, \downarrow, @\}$ *or* $O \subseteq \{\Diamond, \Box, \downarrow\}$ *or* $O \subseteq \{\Diamond, \Box, @\}$. *Then* $\mathsf{lin}\text{-}\mathsf{MSAT}(O)$ *is* NP*-complete.*

Proof. Lemma 3.8 states NP-hardness of $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond)$. We have proven NP-membership for $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond, \downarrow, @)$ in Lemma 3.10, for $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond, \Box, \downarrow)$ in Proposition 3.13 and for $\mathsf{lin}\text{-}\mathsf{MSAT}(\Diamond, \Box, @)$ in Lemma 3.12. $\Box$

The result of Theorem 3.14 carries over to the frame classes $\mathbb{N}$, tt and ttt. The small model property we use in Lemma 3.12 also holds for the other acyclic frame classes. Since, on *any* acyclic frame class $\mathfrak{F}$, it holds that $\mathfrak{F}\text{-}\mathsf{SAT}(\Diamond, \Box, \downarrow) \leqslant_{\mathrm{cd}} \mathfrak{F}\text{-}\mathsf{SAT}(\Diamond, \Box)$, Proposition 3.13 carries over to the other frame classes. Furthermore, the reduction from 3SAT in Lemma 3.8 holds in the same way for tt. For frame classes $\mathbb{N}$ and ttt, we use an infinite model instead of the three state model, but the same argument goes through.

Corollary 3.15 *Let* $\mathfrak{F} \in \{\mathbb{N}, \mathsf{tt}, \mathsf{ttt}\}$ *and* $\{\Diamond\} \subseteq O$, *and* $O \subseteq \{\Diamond, \downarrow, @\}$ *or* $O \subseteq \{\Diamond, \Box, \downarrow\}$ *or* $O \subseteq \{\Diamond, \Box, @\}$. *Then* $\mathfrak{F}\text{-}\mathsf{MSAT}(O)$ *is* NP*-complete.*

3.3 NC^1 *and* $\mathsf{LOGSPACE}$ *results*

In this part, we show NC^1-completeness of $\mathfrak{F}\text{-}\mathsf{MSAT}(O)$ for $\mathfrak{F} \in \{\mathsf{lin}, \mathbb{N}, \mathsf{tt}, \mathsf{ttt}\}$ and $O \subseteq \{\downarrow, @\}$. For the cases $\{\Box\} \subseteq O \subseteq \{\Box, \downarrow, @\}$ we have different results for different frame classes.

In [22, Lemma 2.4.7], NC^1-completeness of the Boolean Formula Value Problem for monotone propositional formulae is shown, so the following holds.

Lemma 3.16 *$\mathfrak{F}$-$\mathsf{MSAT}(\emptyset)$ is NC^1-hard if $\mathfrak{F} \in \{\mathsf{lin}, \mathbb{N}, \mathsf{tt}, \mathsf{ttt}\}$.*

Lemma 3.17 *lin-$\mathsf{MSAT}(\downarrow, @)$ is in NC^1.*

Proof. Let φ be an instance of lin-$\mathsf{MSAT}(\downarrow, @)$. Let φ' be a transformation of φ where every $\downarrow x.$ and $@_x$ in φ is deleted and every atom is replaced by the constant $\top$. Then the following claim holds.

Claim 3.18 *$\varphi \in \mathsf{lin}$-$\mathsf{MSAT}(\downarrow, @)$ if and only if φ' is a propositional tautology.*

Proof of Claim. For the direction from left to right, let $K = (W, R, \eta)$ be a model, w the smallest state in W with respect to R, and $w \in \eta(x)$ for every $x \in \mathsf{PROP} \cup \mathsf{NOM}$. Let g be an assignment with $g(y) = w$ for every $y \in \mathsf{SVAR}$. If a formula φ that is free of modal operators is satisfiable, then we can w.l.o.g. assume that it is satisfiable in the model $K = (W, R, \eta)$. As $w \in \eta(x)$ for every $x \in \mathsf{PROP} \cup \mathsf{NOM}$ and $g(y) = w$ for every $y \in \mathsf{SVAR}$, every atom is true in the state w. If we evaluate φ in this model, we can replace every atom with $\top$. After replacing the atoms, we can ignore every $\downarrow$ and $@$. This construction yields to $\varphi' \equiv c$, $c \in \{\bot, \top\}$. If $K, g, w \models \varphi$, then $K, g, w \models \varphi'$, and for this reason φ' is a valid propositional logic formula.

Conversely, if φ' is a tautology, then we obtain a model $K = (W, R, \eta)$ for φ as follows: (W, R) is a linear frame and $w \in \eta(x)$ for every $x \in \mathsf{PROP} \cup \mathsf{NOM}$ for the first state $w \in W$ with respect to R. With $g(y) = w$ for every $y \in \mathsf{SVAR}$ follows $K, g, w \models \varphi$. $\Diamond$

For testing whether φ' is a valid propositional logic formula, we use [22, Theorem 2.4.2]. Hence we can check the satisfiablity of φ by ignoring all occurrences of $\downarrow x.$ and $@_x$, treating all atoms as the constant $\top$, and applying [22, Theorem 2.4.2] to the generated formula. $\square$

Lemmas 3.16 and 3.17 also hold for the frame classes $\{\mathbb{N}, \mathsf{tt}, \mathsf{ttt}\}$ because we only need to look at the first state of a model, so we can state the following.

Theorem 3.19 *Let $\mathfrak{F} \in \{\mathsf{lin}, \mathbb{N}, \mathsf{tt}, \mathsf{ttt}\}$ and O be such that $\emptyset \subseteq O \subseteq \{\downarrow, @\}$. Then $\mathfrak{F}$-$\mathsf{MSAT}(O)$ is NC^1-complete.*

We will now discuss the same fragments with an additional $\Box$-operator.

Lemma 3.20 *lin-$\mathsf{MSAT}(\Box, \downarrow, @)$ is in NC^1.*

Proof. Let φ be an instance of lin-$\mathsf{MSAT}(\Box, \downarrow, @)$. We show that we can replace every subformula $\Box\psi$ of φ with the constant $\top$, which yields an equisatisfiable formula φ'.

Claim 3.21 *$\varphi \in \mathsf{lin}$-$\mathsf{MSAT}(\Box, \downarrow, @)$ if and only if $\varphi' \in \mathsf{lin}$-$\mathsf{MSAT}(\downarrow, @)$.*

Proof of Claim. If φ is satisfiable, then φ' is clearly satisfiable as well, due to the monotonicity of φ. On the other hand, if φ' is satisfiable, then it is satisfied in K, g, w defined as follows: $K = (\{w\}, \emptyset, \eta)$ with $\eta(x) = \{w\}$ for every $x \in \mathsf{NOM} \cup \mathsf{PROP}$; $g(y) = w$ for every $x \in \mathsf{SVAR}$. This is because φ' contains no modal operator and is a monotone formula. Now every $\Box$ formula is satisfied in this model because w has no successor, hence $K, g, w \models \varphi$. $\Diamond$

Note that in an NC^1-algorithm, it is not possible to replace all occurrences of $\Box$-subformula. Yet, we can treat them as the constant 1. Therefore, ignoring all $\Box$-subformulae in φ, we can use the NC^1 algorithm for $\mathsf{lin}\text{-}\mathsf{MSAT}(\downarrow, @)$. $\Box$

Lemma 3.20 carries over to tt, as K is also a transitive tree. From Lemmas 3.16 and 3.20, we obtain the following set of results.

Corollary 3.22 *Let O be such that $\{\Box\} \subseteq O \subseteq \{\Box, \downarrow, @\}$. Then* $\mathsf{lin}\text{-}\mathsf{MSAT}(O)$ *and* $\mathsf{tt}\text{-}\mathsf{MSAT}(O)$ *are* NC^1*-complete.*

Since the frame underlying the one-state model K constructed in the previous proof is neither $(\mathbb{N}, <)$ nor in ttt, we will now treat the fragments with $\Box$ over these two frame classes separately.

Lemma 3.23 $\mathbb{N}\text{-}\mathsf{MSAT}(\Box, \downarrow, @)$ *is* $\mathsf{LOGSPACE}$*-hard.*

Proof. This proof is very similar to the proof of Theorem 3.3. in [16]. We give a reduction from the problem *Order between Vertices* (ORD) which is known to be $\mathsf{LOGSPACE}$-complete [8] and defined as follows.

Problem:	ORD
Input:	A finite set of vertices V, a successor-relation S on V, and two vertices $s, t \in V$.
Output:	Is $s \leqslant_S t$, where $\leqslant_S$ denotes the unique total order induced by S on V?

Notice that (V, S) is a directed line-graph. Let (V, S, s, t) be an instance of ORD. We construct an $\mathcal{MHL}(\Box, \downarrow, @)$-formula φ that is satisfiable if and only if $s \leqslant_S t$. We use $V = \{v_1, \dots, v_n\}$ as state variables. The formula φ consists of three parts. The first part binds all variables except s to one state and the variable s to a successor of this state. The second part of φ binds a state variable v_l to the state labeled by s iff $s \leqslant_S v_l$. Let α denote the concatenation of all $@_{v_k}{\downarrow}v_l$ with $(v_k, v_l) \in S$ and $v_l \neq s$, and α^n denotes the n-fold concatenation of α. Essentially, α^n uses the assignment to collect all v_i with $s \leqslant_S v_i$ in the state labeled s. The last part of φ checks whether s and t are bound to the same state after this procedure. That is, $\varphi = {\downarrow}v_1.{\downarrow}v_2. \cdots {\downarrow}v_n.\Box{\downarrow}s.\ \alpha^n\ @_s t$. To prove the correctness of our reduction, we show that φ is satisfiable iff $s \leqslant_S t$.

If $s \leqslant_S t$, then for $K = (\{0, 1, 2, \dots\}, <, \eta)$ with arbitrary η and g it holds that $K, g, 0 \models \varphi$. For $s \nleqslant_S t$ we show that $K, g, w \not\models \varphi$ for any K based on the frame $(\mathbb{N}, <)$, any assignment g and any state w. Let g_1 be the assignment obtained from g after the bindings in the prefix ${\downarrow}v_1.{\downarrow}v_2. \cdots {\downarrow}v_n.\Box{\downarrow}s$ of φ, and let g_1' be the assignment obtained from g_1 after evaluating the prefix of φ upto and including α^n. It holds that $g_1'(t) = g_1(t) = \{0\}$ and s is bound to a successor of 0 because we only have acyclic frames. This leads to $K, g_1', 0 \not\models @_t s$ and therefore $K, g, 0 \not\models \varphi$. □

Now we give an upper bound for $\mathbb{N}$-$\mathsf{MSAT}(\Box, \downarrow, @)$.

Lemma 3.24 $\mathbb{N}$-$\mathsf{MSAT}(\Box, \downarrow, @)$ *is in* $\mathsf{LOGDCFL}$.

Proof. We will give a universal model and define two sets of formulae that are true in different states of this model. The set F (A) consists of all formulae that are satisfied in the first state (in all states) of the universal model. For a given formula we show that it is satisfiable iff it is contained in one of these sets. Algorithm 2 then decides for a formula membership in these sets.

Let $K = (\mathbb{N}, <, \eta)$ with $\eta(x) = \{0\}$ for $x \in \mathsf{NOM}$ and $\eta(x) = \mathbb{N}$ for $x \in \mathsf{PROP}$, let g be an assignment with $g(x) = 0$ for every state variable x and let φ be an instance of $\mathbb{N}$-$\mathsf{MSAT}(\Box, \downarrow, @)$.

Claim 3.25 $\varphi \in \mathbb{N}$-$\mathsf{MSAT}(\Box, \downarrow, @)$ *iff* $K, g, 0 \models \varphi$.

Proof of Claim. It is clear that if $K, g, 0 \models \varphi$ then $\varphi \in \mathbb{N}$-$\mathsf{MSAT}(\Box, \downarrow, @)$. For the converse, let K' be an arbitrary $\mathbb{N}$-model satisfying φ in some state $w \in \mathbb{N}$. Even though w and states named by nominals do not need to be 0, we cannot distinguish between all those states without $\Diamond$ and with only monotone operators. Therefore we can merge them into one state where all nominals and atomic propositions are true, and discard all predecessor and successor states, yielding K. This construction preserves satisfiability due to the absence of $\Diamond$ and non-monotone Boolean operators. $\Diamond$

Now we define two sets F and A of instances of $\mathbb{N}$-$\mathsf{MSAT}(\Box, \downarrow, @)$: they are the smallest sets satisfying the following conditions. If $\varphi \in F$, then $K, g, 0 \models \varphi$, and if $\varphi \in A$, then $K, g, n \models \varphi$ for all $n \in \mathbb{N}$.

- $\mathsf{ATOM} \subseteq F$;
- if $\alpha \in F$ or $\beta \in F$, then $\alpha \vee \beta \in F$;
- if $\alpha \in F$ and $\beta \in F$, then $\alpha \wedge \beta \in F$;
- if $\alpha \in A$, then $\Box\alpha \in F$;
- if $\alpha \in F$, then $@_x\alpha \in F$ and $\downarrow x.\alpha \in F$ for every $x \in \mathsf{NOM} \cup \mathsf{SVAR}$;

- if $x \in \mathsf{PROP}$, then $x \in A$;
- if $\alpha \in A$ or $\beta \in A$, then $\alpha \vee \beta \in A$;
- if $\alpha \in A$ and $\beta \in A$, then $\alpha \wedge \beta \in A$;
- if $\alpha \in A$, then $\Box\alpha \in A$;
- if $\alpha \in F$, then $@_x\alpha \in A$ for every $x \in \mathsf{NOM} \cup \mathsf{SVAR}$;
- if $\alpha \in F$ and $\alpha[x/\top] \in A$, then $\downarrow x.\alpha \in A$ for every $x \in \mathsf{SVAR}$;

where $\alpha[x/\top]$ is the formula obtained from α by replacing every occurrence of

x in α by $\top$.

Claim 3.26 *It holds that*

(i) $\varphi \in F \Leftrightarrow K, g, 0 \models \varphi$, *and*
(ii) $\varphi \in A \Leftrightarrow K, g, n \models \varphi$ *for all* $n \in \mathbb{N}$.

Proof of Claim. We prove this by induction on the construction of φ. The initial step is clear. For the induction step we have to distinguish several cases.

$\varphi = \alpha \vee \beta$:
$$\begin{aligned}\varphi \in F &\Leftrightarrow \alpha \in F \text{ or } \beta \in F\\ &\Leftrightarrow K, g, 0 \models \alpha \text{ or } K, g, 0 \models \beta\\ &\Leftrightarrow K, g, 0 \models \alpha \vee \beta\end{aligned}$$
The case $\varphi \in A$ is analogous.

$\varphi = \alpha \wedge \beta$: This case is analogous to $\varphi = \alpha \vee \beta$.

$\varphi = \Box\alpha$:
$$\begin{aligned}\varphi \in F &\Leftrightarrow \alpha \in A\\ &\Leftrightarrow K, g, n \models \alpha \text{ for all } n \in \mathbb{N}\\ &\Leftrightarrow K, g, 0 \models \alpha\end{aligned}$$
The case $\varphi \in A$ is analogous.

$\varphi = @_x\alpha$:
$$\begin{aligned}\varphi \in F &\Leftrightarrow \alpha \in F\\ &\Leftrightarrow K, g, 0 \models \alpha\\ &\Leftrightarrow K, g, 0 \models @_x\alpha \ ([\eta, g](x) = 0)\end{aligned}$$
The case $\varphi \in A$ is analogous.

$\varphi = \downarrow x.\alpha$:
$$\begin{aligned}& K, g, 0 \models \downarrow x.\alpha\\ \Leftrightarrow\ & K, g^0_x, 0 \models \alpha\\ \Leftrightarrow\ & K, g, 0 \models \alpha \ (g = g^0_x \text{ by construction of } g)\\ \Leftrightarrow\ & \alpha \in F\\ \Leftrightarrow\ & \downarrow x.\alpha \in F\end{aligned}$$
and
$$\begin{aligned}& K, g, n \models \downarrow x.\alpha \text{ for all } n \in \mathbb{N}\\ \Leftrightarrow\ & K, g^n_x, n \models \alpha \text{ for all } n \in \mathbb{N}\\ \Leftrightarrow\ & K, g^0_x, 0 \models \alpha \text{ and } K, g^n_x, n \models \alpha \text{ for all } n \in \mathbb{N}\\ \Leftrightarrow\ & K, g, 0 \models \alpha \text{ and } K, g, n \models \alpha[x/\top] \text{ for all } n \in \mathbb{N}\\ \Leftrightarrow\ & \alpha \in F \text{ and } \alpha[x/\top] \in A\\ \Leftrightarrow\ & \downarrow x.\alpha \in A\end{aligned}$$

$\Diamond$

Claims 3.25 and 3.26 imply that $\varphi \in \mathbb{N}\text{-}\mathsf{MSAT}(\Box, \downarrow, @)$ if and only if $\varphi \in F$, which is tested by the function $\mathrm{F}(\varphi)$ in Algorithm 2. Its correctness is straightforward from the definitions of the sets F and A (the function $\mathrm{A}(\varphi)$

Algorithm 2 $\mathbb{N}$-MSAT$(\Box, \downarrow, @)$ algorithm.

```
Require: a formula $\varphi$
 1: return  $\mathrm{F}(\varphi)$

 2: function $\mathrm{F}(\psi)$ // returns a truth value
 3: if $\psi \in \mathsf{ATOM}$ then return *true*
 4: else if $\psi = \alpha \vee \beta$ then return $\max\{\mathrm{F}(\alpha), \mathrm{F}(\beta)\}$
 5: else if $\psi = \alpha \wedge \beta$ then return $\min\{\mathrm{F}(\alpha), \mathrm{F}(\beta)\}$
 6: else if $\psi = \Box\alpha$ then return $\mathrm{A}(\alpha)$
 7: else if $\psi = @_x\alpha$ then return $\mathrm{F}(\alpha)$
 8: else if $\psi = \downarrow x.\alpha$ then return $\mathrm{F}(\alpha)$
 9: end if
10: return  *false*

11: function $\mathrm{A}(\psi)$ // returns a truth value
12: if $\psi \in \mathsf{PROP}$ then return *true*
13: else if $\psi \in \mathsf{NOM} \cup \mathsf{SVAR}$ then return *false*
14: else if $\psi = \alpha \vee \beta$ then return $\max\{\mathrm{A}(\alpha), \mathrm{A}(\beta)\}$
15: else if $\psi = \alpha \wedge \beta$ then return $\min\{\mathrm{A}(\alpha), \mathrm{A}(\beta)\}$
16: else if $\psi = \Box\alpha$ then return $\mathrm{A}(\alpha)$
17: else if $\psi = @_x\alpha$ then return $\mathrm{F}(\alpha)$
18: else if $\psi = \downarrow x.\alpha$ then return $\min\{\mathrm{F}(\alpha), \mathrm{A}(\alpha[x/\top])\}$
19: end if
20: return  *false*
```

decides membership of φ in A).

Furthermore, Algorithm 2 can be implemented on a LOGDCFL-machine, as the recursion depth is bounded by the input length and can be stored on the stack while all remaining variables can be stored in logarithmic space. □

Lemmas 3.23 and 3.24 also hold for total transitive trees, as formulae without $\Diamond$ are, informally speaking, not able to distinguish between the different branches of a tree. Altogether, this leads to:

Corollary 3.27 *The fragments* $\mathbf{N}$-MSAT$(\{\Box, \downarrow, @\})$ *and* ttt-MSAT$(\{\Box, \downarrow, @\})$ *are* LOGSPACE*-hard and contained in* LOGDCFL*.*

4 Conclusion

We have completely classified the complexity of all fragments of hybrid logic with monotone Boolean operators and arbitrary combinations of four modal and hybrid operators over four acyclic frame classes. In contrast to the case with arbitrary Boolean operators, all fragments are of elementary complexity. We have classified their complexity into PSPACE-complete, NP-complete and tractable and shown that the tractable cases are either NC^1-complete or in LOGDCFL.

The result of LOGDCFL-containment is currently complemented by a LOGSPACE lower bound, and it remains to find matching bounds. Furthermore, we plan to extend our study to arbitrary sets of Boolean operators in the same spirit as in [16].

References

[1] Areces, C., P. Blackburn and M. Marx, *A road-map on complexity for hybrid logics*, in: *Proc. CSL-99*, LNCS **1683**, 1999, pp. 307–321.

[2] Areces, C., P. Blackburn and M. Marx, *The computational complexity of hybrid temporal logics*, Logic Journal of the IGPL **8** (2000), pp. 653–679.

[3] Bauland, M., M. Mundhenk, T. Schneider, H. Schnoor, I. Schnoor and H. Vollmer, *The tractability of model checking for LTL: the good, the bad, and the ugly fragments*, in: *Proc. M4M-5*, ENTCS **231**, 2009, pp. 277–292.

[4] Bauland, M., T. Schneider, H. Schnoor, I. Schnoor and H. Vollmer, *The complexity of generalized satisfiability for Linear Temporal Logic*, Log. Meth. in Comp. Sci. **5** (2009).

[5] Beyersdorff, O., A. Meier, M. Mundhenk, T. Schneider, M. Thomas and H. Vollmer, *Model checking CTL is almost always inherently sequential*, in: *Proc. TIME*, 2009.

[6] Blackburn, P. and J. Seligman, *Hybrid languages*, JoLLI **4** (1995), pp. 41–62.

[7] Bozzelli, L. and R. Lanotte, *Complexity and succinctness issues for linear-time hybrid logics*, in: *Proc. of 11th JELIA*, LNCS **5293**, 2008, pp. 48–61.

[8] Etessami, K., *Counting quantifiers, successor relations, and logarithmic space*, J. of Comp. and Sys. Sci. **54** (1997), pp. 400–411.

[9] Franceschet, M. and M. de Rijke, *Model checking for hybrid logics (with an application to semistructured data)*, Journal of Applied Logic **4** (2006), pp. 279–304.

[10] Franceschet, M., M. de Rijke and B. Schlingloff, *Hybrid logics on linear structures: Expressivity and complexity*, in: *Proc. 10th TIME*, 2003, pp. 166–173.

[11] Goranko, V., *Hierarchies of modal and temporal logics with reference pointers*, Journal of Logic, Language and Information **5** (1996), pp. 1–24.

[12] Henzinger, T., *Half-order modal logic: How to prove real-time properties*, in: *Proc. PODC*, 1990, pp. 281–296.

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

[14] Mahajan, M., *Polynomial size log depth circuits: between NC^1 and AC^1*, Bulletin of the EATCS **91** (2007).

[15] Markey, N., *Past is for free: on the complexity of verifying linear temporal properties with past*, Acta Informatica **40** (2004), pp. 431–458.

[16] Meier, A., M. Mundhenk, T. Schneider, M. Thomas, V. Weber and F. Weiss, *The complexity of satisfiability for fragments of hybrid logic — Part I*, in: *Proc. MFCS*, LNCS **5734**, 2009, pp. 587–599.

[17] Meier, A., M. Mundhenk, M. Thomas and H. Vollmer, *The Complexity of Satisfiability for Fragments of CTL and CTL**, International Journal of Foundations of Computer Science (IJFCS) **20** (2009), pp. 901–918.

[18] Mundhenk, M. and T. Schneider, *The complexity of hybrid logics over equivalence relations*, JoLLI **18** (2009), pp. 433–624.

[19] Mundhenk, M., T. Schneider, T. Schwentick and V. Weber, *Complexity of hybrid logics over transitive frames*, in: *Proc. M4M-4*, 2005, see http://arxiv.org/abs/0806.4130.

[20] Papadimitriou, C. H., "Computational Complexity," Addison-Wesley, 1994.

[21] Schneider, T., "The Complexity of Hybrid Logics over Restricted Classes of Frames," Ph.D. thesis, Univ. of Jena (2007).

[22] Schnoor, H., "Algebraic Techniques for Satisfiability Problems," Ph.D. thesis, Univ. of Hannover (2007), see http://www.thi.uni-hannover.de/fileadmin/forschung/arbeiten/hschnoor-diss.pdf.

[23] Schwentick, T. and V. Weber, *Bounded-variable fragments of hybrid logics*, in: *Proc. 24th STACS*, LNCS **4393** (2007), pp. 561–572.

[24] ten Cate, B. and M. Franceschet, *On the complexity of hybrid logics with binders*, in: *Proc. 19th CSL, 2005*, LNCS **3634** (2005), pp. 339–354.

[25] Vollmer, H., "Introduction to Circuit Complexity," Springer, 1999.

[26] Weber, V., *Branching-time logics repeatedly referring to states*, J. of Logic, Language and Information **18** (2009), pp. 593–624.