

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

Arne Meier¹, Martin Mundhenk², Thomas Schneider³,
Michael Thomas¹, Volker Weber⁴, Felix Weiss²

¹*Theoretical Computer Science, University of Hannover, Germany*
[{meier,thomas}@thi.uni-hannover.de}](mailto:{meier,thomas}@thi.uni-hannover.de)

²*Institut für Informatik, Universität Jena, Germany*
[{martin.mundhenk,felix.weiss}@uni-jena.de}](mailto:{martin.mundhenk,felix.weiss}@uni-jena.de)

³*Computer Science, University of Manchester, UK*
schneider@cs.man.ac.uk

⁴*Fakultät für Informatik, Technische Universität Dortmund, Germany*
volker.weber@tu-dortmund.de

Abstract

The satisfiability problem of hybrid logics with the downarrow binder is known to be undecidable. This initiated a research program on decidable and tractable fragments. In this paper, we investigate the effect of restricting the propositional part of the language on decidability and on the complexity of the satisfiability problem over arbitrary, transitive, total frames, and frames based on equivalence relations. We also consider different sets of modal and hybrid operators. We trace the border of decidability and give the precise complexity of most fragments, in particular for all fragments including negation. For the monotone fragments, we are able to distinguish the easy from the hard cases, depending on the allowed set of operators.

Keywords: hybrid logic, satisfiability, decidability, complexity, Post's lattice

1 Introduction

Hybrid logics are well-behaved extensions of modal logic. However, their expressive power often has adverse effects on their computational properties: as a prominent example, the satisfiability problem for basic modal logic extended with the \downarrow binder is undecidable [7,13,1], as opposed to PSPACE-complete for basic modal logic [15] and modal logic extended with nominals and the satisfaction operator $@$ [1].

In order to regain decidability, many restrictions of the hybrid binder language have been considered. On the syntax side, it has been shown in [28] that restricting

* Supported in part by the grants DFG VO 630/6-1, DFG SCHW 678/4-1, DAAD-ARC D/08/08881, and BC-ARC 1323.

**A preliminary version of this paper has been accepted at MFCS 2009 and will appear without many of the proofs.

the interactions between \downarrow and universal operators (such as \wedge , \square) makes satisfiability decidable again. On the semantics side, the satisfiability problem for the \downarrow language has been investigated over different frame classes. It becomes decidable over frames with bounded width [28], over transitive and complete frames [19], and over frames with equivalence relations [18]. In the latter case, decidability is not lost if $@$ or the global modality is added to the language [18], which is not the case over transitive frames [19]. Furthermore, over linear frames and transitive trees, where \downarrow on its own is useless, extensions of the \downarrow language have been shown to be decidable, albeit nonelementarily, in [12,19]. But elementarily decidable fragments over these frame classes have been obtained by bounding the number of state variables [27,30,9]. An overview of complexity results for hybrid logics can be found in [25].

We are proposing to restrict the set of Boolean operators allowed in formulae, i. e., we will systematically replace the usual \wedge , \neg with arbitrary sets of Boolean operators that are not necessarily complete. All such possible sets are captured in Post’s lattice [22,8], which consists of all clones, i. e., all closed sets of Boolean functions. Each clone corresponds to a set of Boolean operators closed under nesting, and vice versa. The lattice can be used for carrying over upper and lower complexity bounds between clones. It will thus be possible to prove finitely many results that will be valid for an infinite number of sets of operators—and hence for infinitely many satisfiability problems.

This technique has been successfully applied to the complexity analysis of satisfiability for propositional logic [16] and modal logic [3], satisfiability and model checking for linear temporal logic [5,4], and satisfiability of constraint satisfaction problems [24,26].

Using Post’s lattice, we will investigate the complexity of the satisfiability problem for hybrid logics containing the modal operators \diamond and \square , and the following hybrid features: nominals, the satisfaction operator $@$, and the hybrid binder \downarrow . We will consider restrictions of this collection of operators, as well as the above described systematic restrictions to the Boolean operators allowed. We will carry out this analysis over four different frame classes: all frames, transitive frames, total frames (where every state has at least one successor), and frames with equivalence relations (ER frames). The work presented here is part of ongoing work that also includes acyclic frame classes such as transitive trees and linear structures.

This paper contains the most complete subset of our results obtained so far (see Figure 1), namely the following. We will show that, over each of the four frame classes mentioned above, satisfiability is as hard as in the full Boolean case whenever the negation of the implication or self-dual Boolean operators are allowed. (A Boolean function is self-dual if negating all of its arguments will always negate its value.) This means that, in these cases, satisfiability remains undecidable over arbitrary frames, total frames and, if the $@$ -operator is present, over transitive frames;4 and NEXPTIME-complete over transitive frames without $@$ and over ER frames. These results can be found in Section 3.3.2.

In Section 3.3.1, we also completely classify the complexity of fragments including only negation and the Boolean constants. We obtain completeness for LOGSPACE if the $@$ -operator is included and completeness for $AC^0[2]$ otherwise.

For all monotone fragments including the Boolean constant 0, we obtain a

for all $1 \leq i \leq n$. The Boolean values *false* and *true* correspond to constants, i. e., nullary functions, and will be denoted by 0 and 1.

A set of Boolean functions is called a *clone* if it contains all projections and is closed under arbitrary composition [21, Chapter 1]. The set of all Boolean clones forms a lattice, which has been completely classified by Post [22]. For a set B of Boolean functions, we denote by $[B]$ the smallest clone containing B and call B a *base* for $[B]$. Whenever we use B for a set in this paper, we assume that B is finite.

In order to introduce the clones relevant to this paper, we define the following notions for n -ary Boolean functions f :

- f is *t-reproducing* if $f(t, \dots, t) = t$, $t \in \{0, 1\}$.
- f is *monotone* if $a_1 \leq b_1, \dots, a_n \leq b_n$ implies $f(a_1, \dots, a_n) \leq f(b_1, \dots, b_n)$.
- f is *t-separating* if there exists an $i \in \{1, \dots, n\}$ such that $f(a_1, \dots, a_n) = t$ implies $a_i = t$, $t \in \{0, 1\}$.
- f is *self-dual* if $f \equiv \text{dual}(f)$, where $\text{dual}(f)(x_1, \dots, x_n) = \neg f(\neg x_1, \dots, \neg x_n)$.

The clones relevant to this paper are listed in Table 1. The definition of all Boolean clones can be found, e. g., in [8]. Notice that $[B \cup \{1\}] = \text{BF}$ if and only if $[B] \supseteq S_1$ or $[B] \supseteq D$.

Name	Definition	Base
BF	All Boolean functions	$\{\wedge, \neg\}$
R_1	1-reproducing functions	$\{\vee, \rightarrow\}$
M	monotone functions	$\{\vee, \wedge, 0, 1\}$
S_1	1-separating functions	$\{\neq\}$
S_{11}	$S_1 \cap M$	$\{x \wedge (y \vee z), 0\}$
D	self-dual functions	$\{(x \wedge \bar{y}) \vee (x \wedge \bar{z}) \vee (\bar{y} \wedge \bar{z})\}$
V	constant or n -ary OR functions	$\{\vee, 0, 1\}$
E	constant or n -ary AND functions	$\{\wedge, 0, 1\}$
E_0		$\{\wedge, 0\}$
N	functions depending on at most one variable	$\{\neg, 0, 1\}$
N_2		$\{\neg\}$
I	constant or identity functions	$\{\text{id}, 0, 1\}$
I_0		$\{\text{id}, 0\}$
I_1		$\{\text{id}, 1\}$
I_2		$\{\text{id}\}$

Table 1
Boolean clones relevant to this paper, with definitions and bases.

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

$$\varphi ::= a \mid c(\varphi, \dots, \varphi) \mid \diamond\varphi \mid \square\varphi \mid \downarrow x.\varphi \mid @_t\varphi$$

where $a \in \text{ATOM}$, c is a Boolean operator, $x \in \text{SVAR}$ and $t \in \text{NOM} \cup \text{SVAR}$. Note that the usual cases \top and \perp are covered by the Boolean constants 1 and 0.

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

$$\begin{aligned}
 K, g, w \models a & \quad \text{iff } w \in [\eta, g](a), a \in \text{ATOM}, \\
 K, g, w \models c(\varphi_1, \dots, \varphi_n) & \quad \text{iff } f_c(t_1, \dots, t_n) = 1, \text{ where } t_i \text{ is the truth value of} \\
 & \quad K, g, w \models \varphi_i, 1 \leq i \leq n, \\
 K, g, w \models \diamond\varphi & \quad \text{iff } K, g, w' \models \varphi \text{ for some } w' \in W \text{ with } wRw', \\
 K, g, w \models \square\varphi & \quad \text{iff } K, g, w' \models \varphi \text{ for all } w' \in W \text{ with } wRw', \\
 K, g, w \models @_t\varphi & \quad \text{iff } K, g, w \models \varphi \text{ for } w \in W \text{ such that } w \in \eta(t), \\
 K, g, w \models \downarrow x.\varphi & \quad \text{iff } K, g_w^x, w \models \varphi.
 \end{aligned}$$

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 $g: \text{SVAR} \rightarrow W$ such that $K, g, w \models \varphi$.

The *at* operator $@_t$ shifts evaluation to the state named by $t \in \text{NOM} \cup \text{SVAR}$. 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 \square are called *modal operators*.

In order to consider fragments of hybrid logics, we define subsets of the language \mathcal{HL} as follows. Let B be a finite set of Boolean functions and let O be a set of hybrid and modal operators. We define $\mathcal{HL}(O, B)$ to denote the set of well-formed hybrid formulae using the operators in O and the Boolean connectives in B only.

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. We will refer to a frame as being *transitive*, *total* or *ER* whenever its transition relation R is transitive ($uRv \wedge vRw \rightarrow uRw$), total ($\forall u \exists v (uRv)$), or an equivalence relation, i. e., reflexive (uRu), transitive and symmetric ($uRv \rightarrow vRu$). In this paper we will consider the class *all* of all frames, the class *trans* of all transitive frames, the class *total* of all total frames, and the class *ER* of all ER 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., $F = (W, R)$. We define the *satisfiability problems* for the fragments of \mathcal{HL} over frame classes defined above as:

$$\begin{aligned}
 \text{Problem:} & \quad \mathfrak{F}\text{-SAT}(O, B) \\
 \text{Input:} & \quad \text{an } \mathcal{HL}(O, B)\text{-formula } \varphi \\
 \text{Output:} & \quad \text{is there a Kripke structure } K = (W, R, \eta) \text{ based on a frame} \\
 & \quad \text{from } \mathfrak{F}, \text{ an assignment } g: \text{SVAR} \rightarrow W \text{ and a } w \in W \text{ such that} \\
 & \quad K, g, w \models \varphi?
 \end{aligned}$$

Please note that, for ease of notation, we do not explicitly mention \mathcal{HL} in the above problem definition. In case $\mathfrak{F} = \text{all}$, we will omit the prefix and simply write

$\text{SAT}(O, B)$.

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 LOGSPACE, NL, P, coNP, PSPACE, NEXPTIME, and coRE.

We will now introduce the notions of circuit complexity required for this paper, for more information on circuit complexity the reader is referred to [29]. The class NC^1 is defined as the set of languages recognizable by a logtime-uniform Boolean circuit of logarithmic depth and polynomial size over $\{\wedge, \vee, \neg\}$, where the fan-in of \wedge and \vee gates is fixed to 2.

The class AC^0 is defined as the set of languages recognizable by a logtime-uniform Boolean circuit of constant depth and polynomial size over $\{\wedge, \vee, \neg\}$, where the fan-in of gates of the type \wedge and \vee is not bounded. If, in addition, modulo-2 gates are allowed, then the corresponding class is $\text{AC}^0[2]$. Both AC^0 and $\text{AC}^0[2]$ are strictly contained in NC^1 . Altogether, the following inclusions are known.

$$\begin{aligned} \text{AC}^0 \subseteq \text{AC}^0[2] \subset \text{NC}^1 \subseteq \text{LOGSPACE} \subseteq \text{NL} \subseteq \text{P} \subseteq \\ \text{coNP} \subseteq \text{PSPACE} \subset \text{NEXPTIME} \subset \text{coRE} \end{aligned}$$

A language A is *constant-depth reducible* to D , $A \leq_{\text{cd}} D$, 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 \leq_{cd} -reductions.

Known results. The following theorem summarizes results for hybrid binder languages with Boolean operators \wedge, \vee, \neg that have been known from the literature.

Theorem 2.1 ([1,19,18])

- (1) $\text{SAT}(\{\diamond, \downarrow\}, \{\wedge, \vee, \neg\})$ and $\text{SAT}(\{\diamond, \square, \downarrow, @\}, \{\wedge, \vee, \neg\})$ are coRE-complete.
- (2) $\text{trans-SAT}(\{\diamond, \downarrow\}, \{\wedge, \vee, \neg\})$ is NEXPTIME-complete.
- (3) $\text{trans-SAT}(\{\diamond, \downarrow, @\}, \{\wedge, \vee, \neg\})$ is coRE-complete.
- (4) $\text{ER-SAT}(\{\diamond, \downarrow\}, \{\wedge, \vee, \neg\})$ is NEXPTIME-complete.
- (5) $\text{ER-SAT}(\{\diamond, \downarrow, @\}, \{\wedge, \vee, \neg\})$ is NEXPTIME-complete.

3 Results

In this section, we present our results ordered by clones. Section 3.1 considers clones containing only 1-reproducing functions. Clones containing the Boolean constant 0 but not negation are considered in Section 3.2. Finally, in Section 3.3, we study satisfiability problems based on clones with negation.

This arrangement is motivated by the observation that the availability of the Boolean constant 0 and/or negation has a very strong impact on our results. Although we obtain different complexities for the clones including 0 but not negation (namely, I, V, E, and M), the results for these clones follow a certain pattern. But if we add negation, this picture changes completely.

Please note that in contrast with the importance of the presence of 0, which makes the difference between trivial and nontrivial problems, hybrid languages can

always express the constant 1 as $\downarrow x.x$ or $@_x x$. Therefore, we only have to consider clones including 1.

3.1 Why we cannot say anything without saying “false”

We start our investigation at the clone \mathbf{l}_2 , which contains only the identity function.¹ Obviously, every hybrid \mathbf{l}_2 -formula is satisfied by the model consisting of a singleton reflexive state to which all propositions, nominals, and state variables are labeled.

But this observation takes us much further, as we can add conjunction, disjunction, and implication for example, and still satisfy every formula by the same model. In fact, we can add every 1-reproducing function, i. e., every function that produces 1 if all parameters are 1, obtaining the following result.

Theorem 3.1 \mathfrak{F} -SAT($\{\diamond, \square, \downarrow, @\}, B$) for $[B] \subseteq \mathbf{R}_1$ and all considered frame classes $\mathfrak{F} \in \{\text{all, trans, total, ER}\}$ is trivial.

Proof. All Boolean functions in the clone \mathbf{R}_1 are 1-reproducing, hence every propositional \mathbf{R}_1 -formula φ is satisfiable. It is easily seen that every modal \mathbf{R}_1 -formula is satisfiable by the singleton reflexive Kripke structure $K = (\{w\}, \{(w, w)\}, \eta)$ with $\eta(p) = \{w\}$ for all $p \in \text{PROP} \cup \text{NOM}$, which is included in all frame classes we consider here. As K includes only one state and every $\mathcal{HL}(\{\diamond, \square\}, \mathbf{R}_1)$ -formula is satisfied in w , bindings and jumps do not change satisfiability. Therefore, all hybrid \mathbf{R}_1 -formulae are satisfiable. \square

It is interesting to note which Boolean operations are not contained in \mathbf{R}_1 . The most basic ones are the Boolean constant 0 and negation, as every clone in Post’s lattice that is not below \mathbf{R}_1 contains one of these.

As hybrid languages can always express the Boolean constant 1, the presence of negation implies the availability of 0. Therefore, there are two kinds of clones remaining: those containing 0 but not negation, and those containing negation. In the following subsection, we will consider the first kind, i. e., the monotone clones below \mathbf{M} . Clones with negation will be considered in Section 3.3.

3.2 Everything but negation – The monotone clones

In this section, we consider the clones below \mathbf{M} that contain the Boolean constant 0; satisfiability for the clones without 0 is trivial by Theorem 3.1. Roughly speaking, we consider the clones \mathbf{I} , \mathbf{V} , \mathbf{E} , and \mathbf{M} . We start with \mathbf{I} and then jump to \mathbf{M} . Clones containing either disjunction or conjunction are considered last, as some results will easily follow from the preceding cases.

3.2.1 The clone \mathbf{I} The clone \mathbf{I} is of particular interest, as it allows us to study the effect of having the Boolean constant 0 at our disposal, yielding the following two observations.

First, the Boolean constant 0 distinguishes trivial from nontrivial satisfiability problems. While all satisfiability problems for clones without 0 are trivial (Theorem 3.1), all problems for clones with 0 are not. The precise complexity of the

¹ Please remember that we can always express the Boolean constant 1 by $\downarrow x.x$. Hence, there is no difference between the satisfiability problems for \mathbf{l}_2 and \mathbf{l}_1 .

latter problems will vary from almost trivial cases (Theorem 3.2) to LOGSPACE-completeness (Theorem 3.3), depending on the modal and hybrid operators allowed. Higher complexities and even undecidability occur if we add further Boolean functions as we do in the following sections.

Second, Theorems 3.2 and 3.3 demonstrate a duality between easy and hard cases, which we will see in all results for clones below M . For the full set of modal and hybrid operators, satisfiability problems over the class of all frames and the class of transitive frames will be considerably harder than those over total frames and equivalence relations. Furthermore, if we drop the \Box -operator when considering arbitrary or transitive frames, complexity will drop to where it is for total frames and equivalence relations.

Intuitively speaking, we might say that the complexity gap we observe in the results for monotone clones is due to the ability to express that a state has no successor by $\Box 0$. On the one hand, if we cannot express this property because of the absence of \Box or if there are no such states because we only consider frames with a total accessibility relation, satisfiability for the clone $\mathbb{1}$ is *almost trivial*, i. e., we only need to look at one symbol of a formula to determine its satisfiability.

Theorem 3.2 *The following satisfiability problems are almost trivial.*²

- (i) total-SAT($\{\diamond, \Box, \downarrow, @\}, B$) and ER-SAT($\{\diamond, \Box, \downarrow, @\}, B$) for $[B] \subseteq \mathbb{1}$.
- (ii) SAT($\{\diamond, \downarrow, @\}, B$) and trans-SAT($\{\diamond, \downarrow, @\}, B$) for $[B] \subseteq \mathbb{1}$.

Proof. Every formula in $\mathcal{HL}(\{\diamond, \downarrow, @\}, \mathbb{1})$ consists of a sequence of operators followed eventually by one final symbol from $\text{ATOM} \cup \{0, 1\}$. For all considered frame classes, these formulae are satisfiable if and only if this final symbol is not 0. For the frame classes total and ER, this also holds if we add \Box to the operators allowed. \square

On the other hand, the proof of the following theorem shows how to use $\Box 0$ to obtain LOGSPACE-hardness, without using any further Boolean connectives. A matching upper bound will be presented in Theorem 3.17.

Theorem 3.3 SAT($\{\diamond, \Box, \downarrow, @\}, B$) and trans-SAT($\{\diamond, \Box, \downarrow, @\}, B$) are LOGSPACE-hard for $[B] \supseteq \mathbb{1}_0$.

Proof. We give a reduction from the problem *Order between Vertices* (ORD) which is known to be LOGSPACE-complete [11].

Problem: ORD
Input: a finite set of vertices V , a successor-relation S on V , and two vertices $s, t \in V$
Output: is $s \leq_S t$, where \leq_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 a $\mathcal{HL}(\{\diamond, \Box, \downarrow, @\}, \mathbb{1}_0)$ -formula φ that is satisfiable if and only if $s \leq_S t$.

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

² More precisely, they are in Δ_0^R , a class strictly below DLOGTIME [23].

successor state. The second part of φ binds a state variable v_l to the state labeled by s iff $s \leq_S v_l$. Let α denote the concatenation of all $@_{v_k} \downarrow v_l$ with $(v_k, v_l) \in S$ and $v_k \neq s$, and α^n denotes the n -fold concatenation of α . Essentially, α^n uses the assignment to collect all v_i with $s \leq_S v_i$ in the state labeled s .

Claim. $K, g, u \models \alpha^n @_x \psi$ iff $K, g', u \models @_x \psi$ for g' with $g'(v_i) = g(s)$ for all $v_i \geq_S s$ and $g'(v_i) = g(v_i)$ for all $v_i \not\geq_S s$.

It is not hard to prove the Claim. The last part of φ guarantees that s was initially bound to another state than the remaining variables, and checks whether s and t are bound to the same state after this procedure.

$$\varphi = \downarrow v_1. \downarrow v_2. \dots \downarrow v_n. \diamond \downarrow s. \alpha^n @_t \square 0$$

To prove the correctness of our reduction, we show that φ is satisfiable if and only if $s \leq_S t$. If $s \leq_S t$, then for $K = (\{u, w\}, \{(u, w)\}, \eta)$ with arbitrary η and g it holds that $K, g, u \models \varphi$. For $s \not\leq_S t$, consider any K with state u . We show that $K, g, u \not\models \varphi$. Let g_1 be the assignment obtained from g after the bindings of the first part $\downarrow v_1. \downarrow v_2. \dots \downarrow v_n. \diamond \downarrow s$ of φ , and let g'_1 be the assignment obtained from g_1 after the first part and the second part α^n . By the Claim it follows that $g'_1(t) = g_1(t) = \{u\}$. If u has no successor, then the first part $\downarrow v_1. \downarrow v_2. \dots \downarrow v_n. \diamond \downarrow s$ of φ is not satisfied. If u has a successor, then $K, g'_1, u \not\models \square 0$, and it follows that $K, g'_1, u \not\models @_t \square 0$ and therefore $K, g, u \not\models \varphi$. \square

3.2.2 The clone M Let us now consider the clone M of all monotone functions. Here, more precisely for all clones between S_{11} and M, we obtain the same duality as in the previous section, only at a higher level of complexity.

For the “hard cases”, i. e., those satisfiability problems where we consider non-total frame classes and all modal and hybrid operators, we obtain PSPACE-hardness. For the class of all frames, this follows immediately from the corresponding result for the modal satisfiability problem.

Theorem 3.4 ([3,14]) *SAT*($\{\diamond, \square, \downarrow, @\}, B$) is PSPACE-hard for $[B] \supseteq S_{11}$.

Unfortunately, the proof of this result does not generalize to transitive frames. We give an alternative proof by reducing from the satisfiability problem for quantified Boolean formulae.

Lemma 3.5 *trans-SAT*($\{\diamond, \square, \downarrow, @\}, \{\wedge, \vee, 0\}$) is PSPACE-hard.

Proof. QBFSAT is a standard PSPACE-complete set. Its instances are quantified Boolean formulae in conjunctive normal form, e. g.,

$$\varphi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4).$$

A quantified Boolean formula is in QBFSAT if and only if it evaluates to true. We give a reduction that reduces QBFSAT to *trans-SAT*($\{\diamond, \square, \downarrow, @\}, \{\wedge, \vee, 0\}$). An instance $\varphi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \psi$, where $Q_i \in \{\exists, \forall\}$ for $1 \leq i \leq n$ is transformed to

$$f(\varphi) = (@_s \diamond \diamond 1) \wedge (@_s \diamond \square 0) \wedge @_s h(Q_1) \downarrow x_1. @_s h(Q_2) \downarrow x_2. \dots @_s h(Q_n) \downarrow x_n. r(\psi),$$

where $h(Q) = \diamond (Q = \square)$ if $Q = \exists (Q = \forall)$, and $r(\psi)$ is obtained from ψ by replacing all appearances of *positive* literals x_i with $@_{x_i}\diamond 1$, respectively replacing all appearances of *negative* literals $\neg x_i$ with $@_{x_i}\square 0$. Constant symbols remain unchanged. As an example, the QBFSAT instance φ_0 from above is transformed to

$$\begin{aligned}
 & (@_s \diamond \diamond 1) \wedge (@_s \diamond \square 0) \wedge @_s \diamond \downarrow x_1 . @_s \square \downarrow x_2 . @_s \diamond \downarrow x_3 . @_s \square \downarrow x_4 . \\
 & \quad (@_{x_1} \diamond 1 \vee @_{x_2} \square 0) \wedge (@_{x_1} \square 0 \vee @_{x_2} \diamond 1 \vee @_{x_3} \diamond 1 \vee @_{x_4} \square 0).
 \end{aligned}$$

It is clear that f is a \leq_{cd} -reduction.

The following claim implies $\text{QBFSAT} \leq_{\text{cd}} \text{trans-SAT}(\{\diamond, \square, \downarrow, @\}, \{\wedge, \vee, 0\})$.

Claim. φ evaluates to true if and only if $K_2, g, s \models f(\varphi)$ for $K_2 = (W, R, \eta)$ with the transitive frame $(W, R) = (\{s, t\}, \{(s, s), (s, t)\})$ and $\eta(x_i) = W$.

The proof of the Claim is straightforward. \square

PSPACE-hardness also follows for all clones containing S_{11} .

Theorem 3.6 $\text{trans-SAT}(\{\diamond, \square, \downarrow, @\}, B)$ is PSPACE-hard for $[B] \supseteq S_{11}$.

Proof. It suffices to show $\text{trans-SAT}(O, \{\wedge, \vee, 0\}) \leq_{\text{cd}} \text{trans-SAT}(O, B)$. The result then follows from Lemma 3.5.

Take $\varphi \in \mathcal{HL}(O, \{\wedge, \vee, 0\})$. Since $[B \cup \{1\}] = M$, we can rewrite φ as an $\mathcal{HL}(O, B \cup \{1\})$ -formula φ' , leaving modal and hybrid operators untouched. Due to [16,26], this can be computed in polynomial time. Now we can easily transform φ' into an $\mathcal{HL}(O, B)$ -formula φ'' , replacing all occurrences of 1 with $@_z z$. Clearly, φ and φ'' are equisatisfiable over transitive (and even over arbitrary) frames. \square

The proof of Theorem 3.6 crucially depends on the existence of states without successor, and the ability to express this property: the truth values \perp (resp. \top) are encoded as states having no (resp. at least one) successor. If there are no such states (Theorem 3.7) or if we cannot express this property (Corollary 3.8), complexity drops to NC^1 .

Theorem 3.7 $\text{total-SAT}(O, B)$ and $\text{ER-SAT}(O, B)$ are NC^1 -complete under \leq_{cd} -reductions for $O \subseteq \{\diamond, \square, \downarrow, @\}$ and $S_{11} \subseteq [B] \subseteq M$.

Proof. Any propositional M -formula φ is satisfiable if and only if it is satisfied by the assignment that sets all atoms to true. We generalize this result to hybrid logic for frame classes **total** and **ER**.

Let $\varphi \in \mathcal{HL}(\{\diamond, \square, \downarrow, @\}, M)$ and let $K_1 := (\{w_1\}, \{(w_1, w_1)\}, \eta_1)$ be the reflexive singleton Kripke structure with $\eta_1(p) = \{w_1\}$ for all $p \in \text{PROP} \cup \text{NOM}$. The following claim is easy to verify.

Claim. $K_1, g_1, w_1 \models \psi$ is equivalent to $K_1, g_1, w_1 \models \lambda\psi$ for any operator λ .

We show that $\varphi \in \text{total-SAT}(\{\diamond, \square, \downarrow, @\}, M)$ if and only if $K_1, g_1, w_1 \models \varphi$. We proceed by induction on the structure of φ . If $\varphi \in \text{PROP} \cup \text{NOM} \cup \text{SVAR}$, then $\varphi \in \text{total-SAT}(\{\diamond, \square, \downarrow, @\}, M)$ and $K_1, g_1, w_1 \models \varphi$.

For the inductive step, assume that the claim holds for all subformulae of φ .

- $\varphi = \square\psi$ ($\varphi = \diamond\psi$). For total frames, $\square\psi$ ($\diamond\psi$) is satisfiable if and only if ψ is satisfiable. By induction hypothesis, this is equivalent to $K_1, g_1, w_1 \models \psi$, and by the Claim this is equivalent to $K_1, g_1, w_1 \models \square\psi$ ($K_1, g_1, w_1 \models \diamond\psi$).

- $\varphi = \downarrow x.\psi$. Then $\varphi \in \text{total-SAT}(\{\diamond, \square, \downarrow, @\}, \mathbf{M})$ if and only if there exists K, g, w such that $K, g_w^x, w \models \psi$. By induction hypothesis and by the Claim this is equivalent to $K_1, g_1, w_1 \models \downarrow x.\psi$.
- The remaining cases for hybrid operators follow similarly.
- $\varphi = c(\psi_1, \dots, \psi_n)$ with $c \in \mathbf{M}$. Assume that φ is satisfied by some Kripke structure K under assignment g in state w . By induction hypothesis we obtain: if $K, g, w \models \psi_i$ then $K_1, g_1, w_1 \models \psi_i$. Since $c \in \mathbf{M}$, it follows that $K, g, w \models c(\psi_1, \dots, \psi_n)$ if and only if $K_1, g_1, w_1 \models c(\psi_1, \dots, \psi_n)$.

This shows that deciding $\varphi \in \text{total-SAT}(\{\diamond, \square, \downarrow, @\}, \mathbf{M})$ is equivalent to deciding whether $K_1, g_1, w_1 \models \varphi$. In order to decide the latter, all hybrid and modal operators of φ can be ignored, as K_1 is a singleton model. Thus deciding $K_1, g_1, w_1 \models \varphi$ is equivalent to the evaluation problem for propositional M-formula, which is NC^1 -complete under \leq_{cd} -reductions [26].

The same arguments apply for $\text{ER-SAT}(\{\diamond, \square, \downarrow, @\}, \mathbf{M})$. \square

The proof of Theorem 3.7, shows that deciding $\varphi \in \text{total-SAT}(\{\diamond, \square, \downarrow, @\}, \mathbf{M})$ is equivalent to deciding whether $K_1, g_1, w_1 \models \varphi$, for the singleton reflexive model K_1 mapping all atomic propositions into state w_1 . In order to decide the latter, all hybrid and modal operators of φ can be ignored, as K_1 is a singleton model. There, only the treatment of the \square -operator depends on the transition relation being total or an equivalence relation. If this operator is not allowed, the same argumentation goes through for our other frame classes, too.

Corollary 3.8 *SAT(O, B) and trans-SAT(O, B) are NC^1 -complete for $O \subseteq \{\diamond, \downarrow, @\}$ and $S_{11} \subseteq [B] \subseteq \mathbf{M}$.*

3.2.3 The clones \mathbf{V} and \mathbf{E} If we do not consider conjunction and disjunction but only one of these, the complexity of formula evaluation decreases from NC^1 -complete to below AC^0 . As the complexity for the “easy cases” for \mathbf{M} was determined by this complexity (Theorem 3.7 and Corollary 3.8), the following results are not too surprising.

Theorem 3.9 *total-SAT($\{\diamond, \square, \downarrow, @\}, B$) and ER-SAT($\{\diamond, \square, \downarrow, @\}, B$) are in AC^0 for $[B] \subseteq \mathbf{E}$ or $[B] \subseteq \mathbf{V}$.*

Proof. As $\mathbf{V} \subseteq \mathbf{M}$ and $\mathbf{E} \subseteq \mathbf{M}$, the hybrid operators \downarrow and $@_x$ may be ignored as in the proof of Theorem 3.7. It has been shown by Schnoor [26], that evaluation of \mathbf{V} -formulae (\mathbf{E} -formulae) is in coNLOGTIME (NLOGTIME , resp.). Since $\text{NLOGTIME} \cup \text{coNLOGTIME} \subseteq \text{AC}^0$, the theorem applies. \square

Just as we obtained Corollary 3.8 from Theorem 3.7, we obtain the following result from Theorem 3.9.

Corollary 3.10 *SAT(O, B) and trans-SAT(O, B) are in AC^0 for $O \subseteq \{\diamond, \downarrow, @\}$ and $[B] \subseteq \mathbf{E}$ or $[B] \subseteq \mathbf{V}$.*

This result is optimal in the sense that including all modal and hybrid operators we immediately get LOGSPACE lower bounds from Theorem 3.3.

For the case of conjunctions, this result can be improved. Considering arbitrary frames, a coNP lower bound is already known for the modal satisfiability problem.

Theorem 3.11 ([3,10]) *SAT*($\{\diamond, \square\}, B$) is coNP-hard for $[B] \supseteq E_0$.

As before, the proof of this result does not generalize to transitive frames. Here, we are able to show NL-hardness.

Theorem 3.12 *trans-SAT*($\{\diamond, \square, \downarrow, @\}, B$) is NL-hard for $[B] \supseteq E_0$.

Proof. We give a reduction from the unreachability problem for acyclic graphs. For acyclic graphs, there is a path from s to t if and only if s appears before t in every topological sorting of the nodes. We make use of the negation of this statement.

Let $G = (V, E, s, t)$ be an instance of unreachability for acyclic digraphs with $V = \{1, \dots, n\}$. We construct a $\mathcal{HL}(\{\diamond, \square, \downarrow, @\}, \{\wedge, 0\})$ -formula $\varphi = \varphi_1 \wedge \varphi_2$ that is satisfiable if and only if G has no path from s to t .

We will use a state variable x_i for each node i and an additional state variable r that will be used for the “root state”. The first part of φ is used to guarantee that every Kripke structure K that satisfies φ consists of at least $n + 1$ states with an acyclic transition relation.

$$\varphi_1 = \diamond^n 1 \wedge \square^{n+1} 0$$

The second part of φ gives names to the states, such that the order of the states reflects a topological order of the corresponding nodes of the graph, and checks whether s is behind t .

$$\varphi_2 = \downarrow r. @_r \diamond \downarrow x_1. \dots @_r \diamond \downarrow x_n. \bigwedge_{(i,j) \in E} (@_{x_i} \diamond x_j) \wedge @_{x_t} \diamond x_s$$

Note that it does not matter here if more than one variable is assigned to one state as long as the edge-relation is respected.

It is not too hard to see that φ is satisfiable (over a transitive frame) if and only if there exists a topological ordering of V in which t appears before s . Consequently, *trans-SAT*($\{\diamond, \square, \downarrow, @\}, E_0$) is NL-hard. \square

We conjecture that all lower bounds provided in this section (except for probably the last one) are optimal. Nevertheless, matching upper bounds are missing.

Theorem 3.13 *SAT*($\{\diamond, \square, \downarrow, @\}, V$) is in LOGSPACE.

Proof. Let $\varphi \in \mathcal{HL}(\{\diamond, \square, \downarrow, @\}, V)$, and let a_1, \dots, a_n be all atoms (i.e., occurrences of 0, 1, atomic propositions, nominals or state variables) in φ . Given a subformula ψ of φ and i such that a_i occurs in ψ , we define $\psi[a_i]$ to be the following formula.

- If $\psi = a_i$, then $\psi[a_i] = a_i$.
- If $\psi = \vartheta_1 \vee \vartheta_2$, then $\psi[a_i] = \vartheta_j[a_i]$, where j is such that ϑ_j contains a_i .
- If $\psi = O\vartheta$, for $O \in \{\diamond, \square, \downarrow x, @_x\}$, then $\psi[a_i] = O\vartheta[a_i]$.

Clearly, each $\psi[a_i]$ is in $\mathcal{HL}(\{\diamond, \square, \downarrow, @\}, 1)$.

The following claim yields a decision procedure for *SAT*($\{\diamond, \square, \downarrow, @\}, V$) that is based on a procedure for *SAT*($\{\diamond, \square, \downarrow, @\}, 1$), which can be found in Theorem 3.17.

Claim. φ is satisfiable if and only if there is some $i = 1, \dots, n$ such that $\varphi[a_i]$ is satisfiable.

Since $\text{SAT}(\{\diamond, \square, \downarrow, @\}, \mathbf{1})$ is in LOGSPACE, the claim enables us to decide $\text{SAT}(\{\diamond, \square, \downarrow, @\}, \mathbf{V})$ in LOGSPACE as follows: for each a_i , test whether $\varphi[a_i]$ is satisfiable. If one of these tests is positive, accept; otherwise reject. The administrative effort of traversing through all atoms and determining the respective $\varphi[a_i]$ involves only constantly many counters of logarithmic size and determining scopes of operators. Therefore, the whole decision procedure can be performed in logarithmic space.

It remains to prove the claim. The “ \Leftarrow ” direction is obvious: if $\varphi[a_i]$ is satisfiable in a state s of a Kripke structure K under an assignment g , then K, g, s also satisfies φ . For the “ \Rightarrow ” direction, we recall hybrid tableau techniques as described in [6]. If φ is satisfiable, then there exists a tableau with $@_i\varphi$ at its root, for a fresh nominal i , and an open and complete branch in this tableau. This branch by itself is a tableau for $\varphi[a_i]$, and therefore $\varphi[a_i]$ is satisfiable. \square

3.3 Clones including negation

We now turn to the clones including negation. Negation immediately limits the number of relevant satisfiability problems in two ways. First, as $\square\varphi \equiv \neg\diamond\neg\varphi$, we cannot exclude the \square -operator and keep \diamond as we did for monotone clones. Therefore, we have to consider only two hybrid languages: with and without $@$.

Second, as we can obtain one Boolean constant from the other and as $\mathbf{1}$ is always expressible in hybrid languages, we only need to consider clones with both constants. These are **N** (only negation), **L** (exclusive or), and **BF** (all Boolean functions).

While we will completely classify all satisfiability problems based on **N** and **BF**, we will not provide any specific results for **L**.

3.3.1 Negation only The results for the satisfiability problems based on **N** stick out from our other results, as **N** is the only clone (besides those for which satisfiability is trivial) where all complexity results are the same for all frame classes we consider. We show that satisfiability for the hybrid language including $@$ is LOGSPACE-complete, while it is $\text{AC}^0[2]$ -complete for the language without $@$. We start with the lower bound for the latter result, which holds even in the absence of modal operators.

Theorem 3.14 $\mathfrak{F}\text{-SAT}(\{\downarrow\}, B)$ is $\text{AC}^0[2]$ -hard for $[B] \supseteq \mathbf{N}_2$ and all considered frame classes $\mathfrak{F} \in \{\text{all}, \text{trans}, \text{total}, \text{ER}\}$.

Proof. We give a reduction from the $\text{AC}^0[2]$ -complete problem PARITY.

Let $a := a_1 \cdots a_n \in \{0, 1\}^n$ be an instance for PARITY. Then the following $\mathcal{HL}(\{\downarrow\}, \mathbf{N}_2)$ -formula is satisfiable if and only if a contains an even number of ones.

$$\varphi_a := \downarrow x.b_1 \dots b_n x,$$

where $b_i = \neg$ if $a_i = 1$ and the empty string otherwise. That is, the number of negations in φ_a is exactly the number of ones in a . As the satisfiability of φ_a does not depend on the transition relation, the theorem follows. \square

A matching upper bound is provided by the following theorem.

Theorem 3.15 \mathfrak{F} -SAT($\{\diamond, \square, \downarrow\}, B$) is $\text{AC}^0[2]$ -complete for $[B] \subseteq \mathbb{N}$ and all considered frame classes $\mathfrak{F} \in \{\text{all}, \text{trans}, \text{total}, \text{ER}\}$.

The key to the proof of this theorem is that a given formula can be transformed into negation normal form by an $\text{AC}^0[2]$ -circuit. Determining satisfiability afterwards is easy.

Proof. Let $\varphi \in \mathcal{HL}(\{\diamond, \square, \downarrow\}, B)$. Before we consider the different frame classes, we observe that we can assume that φ is of the form

$$(\{\diamond, \square\} \cup \{\downarrow x. \mid x \in \text{SVAR}\})^* \lambda,$$

where $\lambda \in \{\neg x, x, \neg p, p, \neg n, n, 1, 0\}$ for a variable $x \in \text{SVAR}$, a proposition $p \in \text{PROP}$, and a nominal $n \in \text{NOM}$. In particular, there is at most one negation and if there is a negation, it is the second to last symbol of φ . If φ is not given in this form, an equivalent formula can be computed by simply drawing negations inward, replacing operators by their duals and eliminating double negations. More precisely, we have to compute for every operator in φ whether it is in the scope of an odd or even number of negations. This can be done by an $\text{AC}^0[2]$ -circuit.

Now, assume that φ is of the form described above, and let us distinguish two cases. First, assume that λ is not the negation of a bound state variable. In this case, it is rather trivial to determine satisfiability of φ : over total or ER frames, φ is unsatisfiable if and only if $\lambda = 0$; over arbitrary or transitive frames, φ is unsatisfiable if and only if $\lambda = 0$ and φ contains no \square -operator.

Second, if $\lambda = \neg x$ for some state variable x , satisfiability of φ depends only on the modal operators between the last \downarrow -operator binding x and λ . If there are no modal operators in between, then φ is clearly unsatisfiable; otherwise, we need to distinguish between frame classes.

Over arbitrary, transitive and total frames, φ is satisfied in the model obtained from the natural numbers with the usual order $<$ as transition relation if and only if there is a modal operator in between. Over ER frames, φ is satisfiable if and only if φ is satisfiable in a complete frame, i.e., one where the transition relation contains every possible edge ($R = W \times W$). In such a model, satisfiability of φ depends only on the last modal operator. If this operator is a \diamond , then φ is satisfiable in model containing at least two states. If the last modal operator is a \square , then φ is unsatisfiable.

Therefore, the complexity of satisfiability is in all cases dominated by the translation of the given formula into the form described above, and hence in $\text{AC}^0[2]$. \square

Now, we turn to the hybrid language including the $@$ -operator.

Theorem 3.16 \mathfrak{F} -SAT($\{\diamond, \downarrow, @\}, B$) is LOGSPACE-hard for $[B] \supseteq \mathbb{N}_2$ and all considered frame classes $\mathfrak{F} \in \{\text{all}, \text{trans}, \text{total}, \text{ER}\}$.

Proof. For $\mathfrak{F} \in \{\text{all}, \text{trans}\}$ this follows from Theorem 3.3. To capture the other frame classes as well, we give a reduction from the problem *Order between Vertices* (ORD) similar to the one in the proof of Theorem 3.3. Let α be the string obtained

from (V, S, s, t) as in the proof of Theorem 3.3. Then

$$\varphi = \downarrow x_1. \diamond \downarrow x_2. \diamond \downarrow x_3. \cdots \diamond \downarrow x_n. \alpha^n @_s \neg t.$$

From $K, w, g \models \varphi$ it follows that $s \not\leq_S t$ using the claim in the proof of Theorem 3.3. If $s \leq_S t$ then every Kripke structure in which s and t can be bound to different states satisfies φ . This proves the correctness of our reduction. \square

Again, we can provide a matching upper bound for all considered frame classes, thereby establishing LOGSPACE-completeness of the corresponding satisfiability problems.

Theorem 3.17 \mathfrak{F} -SAT $(\{\diamond, \square, \downarrow, @\}, B)$ is in LOGSPACE for $[B] \subseteq \mathbb{N}$ and all considered frame classes $\mathfrak{F} \in \{\text{all, trans, total, ER}\}$.

Proof. As in the proof of Theorem 3.14, we can assume that the given formula $\varphi \in \mathcal{HL}(\{\diamond, \square, \downarrow, @\}, \mathbb{N})$ is of the form

$$\varphi = (\{\diamond, \square\} \cup \{@_t \mid t \in \text{NOM} \cup \text{SVAR}\} \cup \{\downarrow x. \mid x \in \text{SVAR}\})^* \lambda,$$

where $\lambda \in \{\neg x, x, \neg p, p, \neg i, i, 1, 0\}$ for a variable $x \in \text{SVAR}$, a proposition $p \in \text{PROP}$, and a nominal $i \in \text{NOM}$. We consider each frame class separately.

Over total frames, to decide whether $\varphi \in \text{total-SAT}(\{\diamond, \square, \downarrow, @\}, B)$, we distinguish the following three cases.

$\lambda \in \{0, 1\}$: As the transition relation is total, we obtain that $\varphi \in \text{total-SAT}$ iff $\lambda = 1$.

$\lambda \in \{p, \neg p, x, i\}$: In this case, φ is satisfied by the singleton reflexive model.

$\lambda \in \{\neg x, \neg i\}$: For the last case that λ is a negated nominal or a negated state variable. First consider the case that φ is free of unbound state variables and nominals, i.e., $\lambda = \neg x$ for some $x \in \text{SVAR}$. We can assume w.l.o.g. that each state variable is bound at most once. This can be achieved by bound renaming, which is computable in logarithmic space because it only involves computing scopes of binders and counting occurrences of \downarrow and $@$ operators.

Let $\overline{\text{ATOM}} := \{\neg a \mid a \in \text{ATOM}\}$. We define a transformation $(\cdot)_\Delta$ as follows: For a formula $\psi = P_1 \cdots P_n a$ with $a \in \text{ATOM} \cup \overline{\text{ATOM}}$ and $P_i \in \{\downarrow x., @_t, \diamond, \square \mid x \in \text{SVAR}, t \in \text{NOM} \cup \text{SVAR}\}$, let $(\psi_j)_{j \in \mathbb{N}}$ be the family of strings obtained as follows: $\psi_0 := \psi$, and ψ_{j+1} is derived from ψ_j by deleting the substring $P_\ell \cdots P_k$, $1 \leq \ell < k \leq n$, from ψ_j where $P_\ell = \downarrow x.$, $P_k = @_x$ and such that the suffix $P_{k+1} \cdots P_n$ is free of $@$ -operators. Note that, in the first step, such an ℓ exists by assumption. Obviously, there is some $\tilde{j} \leq n$ such that $\psi_{\tilde{j}} = \psi_{\tilde{j}+1}$. Intuitively, $\psi_{\tilde{j}}$ is obtained by reading ψ from right to left, discarding all symbols between, and including, any $@_x$ -operator and the corresponding $\downarrow x.$ We define $(\psi)_\Delta$ as $\psi_{\tilde{j}}$ with all remaining $\downarrow x.$ -operators removed. Obviously, $(\psi)_\Delta \in \{\diamond, \square\}^* \cdot (\text{ATOM} \cup \overline{\text{ATOM}})$. For example, if $\psi := \diamond \square \downarrow x. \diamond \diamond \downarrow y. \square @_x \diamond \downarrow z. \square @_z y$, then $(\psi)_\Delta = \diamond \square \diamond y = P_1 P_2 P_3 y$.

Let $\varphi = P_1 \cdots P_n \neg x$ denote the given formula and let $1 \leq m \leq n$ be such that $P_m = \downarrow x.$ We define φ' as $P_1 \cdots P_m 1$. The formulae $(\varphi)_\Delta$ and $(\varphi')_\Delta$ describe the sequence of modalities relevant to reach the state(s) that must not be labeled with x , resp. the sequence of modalities relevant to reach the state that is actually

labeled by x . It remains to check that the sequences described by $(\varphi)_\Delta$ and $(\varphi')_\Delta$ are “compatible”. Denote by κ the maximal index such that $i_\kappa = j_\kappa$. Then $P_1 \cdots P_{i_\kappa}$ describes the common prefix of both formulae that may be ignored w. r. t. the satisfiability of φ . We therefore omit $P_1 \cdots P_{i_\kappa}$ from both sequences $(\varphi')_\Delta$ and $(\varphi)_\Delta$, i.e., we set $(\varphi')_\Delta = P_{j_{\kappa+1}} \cdots P_{j_\nu} \mathbf{1}$ and $(\varphi)_\Delta = P_{i_{\kappa+1}} \cdots P_{i_\mu} \neg x$ with $1 \leq i_{\kappa+1} < \cdots < i_\mu \leq n$ and $1 \leq j_{\kappa+1} < \cdots < j_\nu < m$. It now holds that φ is satisfiable if and only if $(\varphi')_\Delta$ and $(\varphi)_\Delta$ meet one of the following conditions:

- (i) $\mu \neq \nu$,
- (ii) $\mu = \nu$ and there exists a ξ , $\kappa < \xi \leq \mu$, such that $P_{i_\xi} = P_{j_\xi} = \diamond$,
- (iii) $\mu = \nu$ and there exists a ξ , $\kappa < \xi \leq \mu$, such that $P_{i_\xi} = \diamond$, $P_{j_\xi} = \square$, $j_\xi < i_\xi$ (i.e., the position of P_{i_ξ} in φ is to the right of P_{j_ξ}),
- (iv) $\mu = \nu$ and there exists a ξ , $\kappa < \xi \leq \mu$, such that $P_{i_\xi} = \square$, $P_{j_\xi} = \diamond$, $i_\xi < j_\xi$ (i.e., the position of P_{j_ξ} in φ is the right of P_{i_ξ}).

For necessity of (i)–(iv), let φ be an $\mathcal{HL}(\{\diamond, \square, \downarrow, @\}, \mathbf{N})$ -formula without nominals and unbound state variables which is satisfiable in a total frame but does not satisfy any of (i)–(iv). From the converses of conditions (i) and (ii), we derive that $\mu = \nu$ and that for all $\kappa < \xi \leq \mu$, $P_{i_\xi} = \square$ or $P_{j_\xi} = \square$. Now, observe that in total structures if $P_{i_\xi} = P_{j_\xi} = \square$, then φ and φ with P_{i_ξ} and P_{j_ξ} removed are equisatisfiable.

Hence we may assume that for all $\kappa < \xi \leq \mu$, either $P_{i_\xi} = \square$ or $P_{j_\xi} = \square$, but not both. Moreover, the converses of (iii) and (iv) imply $i_\xi > j_\xi$ for the former and $j_\xi > i_\xi$ for the latter case. But if we evaluate φ on some total Kripke structure K , then every $P_{i_\xi} = \diamond$ (resp. $P_{j_\xi} = \diamond$) is processed prior to the corresponding modality $P_{j_\xi} = \square$ (resp. $P_{i_\xi} = \square$). This is a contradiction to the satisfiability of φ : suppose $P_{i_\xi} = \diamond$, $P_{j_\xi} = \square$, and $i_\xi < j_\xi$. Then $P_{i_\xi} = \diamond$ enforces a successor v behind which x is bound, but $P_{j_\xi} = \square$ causes the subformula starting at P_{j_ξ} to be satisfied in v . In the evaluation of that subformula, $\neg x$ will then be required to be satisfied in the state where x is bound.

For sufficiency of (i)–(iv), assume that φ meets one of the above conditions. It is not hard to see that the satisfaction of condition (i) implies that φ is satisfied in a Kripke structure consisting of a single chain of states. Otherwise, if one of conditions (ii)–(iv) is satisfied, then φ is satisfied in a Kripke structure consisting of two chains of states that share a common prefix of length $i_\xi - 1$.

For the case that φ contains unbound state variables or nominals, observe that φ can be transformed into an equisatisfiable formula ψ in polynomial time such that ψ is free of nominals and unbound state variables:

$$\psi := \downarrow r. \diamond \downarrow x_1. @_r \cdots \diamond \downarrow x_k. @_r \diamond \downarrow y_1. @_r \cdots \diamond \downarrow y_\ell. @_r \diamond \varphi[i_1/y_1, \dots, i_\ell/y_\ell],$$

where r, y_1, \dots, y_ℓ are fresh state variables, x_1, \dots, x_k enumerate the unbound state variables occurring in φ , and i_1, \dots, i_ℓ enumerate the nominals occurring in φ . If ψ is satisfiable, then so is φ . On the other hand, if φ is satisfiable in a total structure, then ψ is satisfied in the structure obtained by adding a spypoint [7], i.e., a state w that is a successor of every state of the original structure.

So far, an algorithm deciding satisfiability of φ needs to check which of the three cases for λ holds. In the third case, after freeing φ of nominals, unbound

state variables, and multiple binding of the same state variable, $(\varphi)_\Delta$ and $(\varphi')_\Delta$ need to be computed, and conditions (i)–(iv) need to be checked. In order to see that these tasks can be performed in logarithmic space, it needs to keep on mind that $(\varphi)_\Delta$ and $(\varphi')_\Delta$ as well as the initial transformations need not be computed explicitly; pointers to the current positions in φ , $(\varphi)_\Delta$ and $(\varphi')_\Delta$ suffice to determine the required information on-the-fly.

Over ER frames, the above algorithm can be easily adapted. In case $\lambda \in \{0, 1, p, \neg p, x, i\}$ we proceed as above. Hence assume $\lambda \in \{\neg x, \neg i\}$.

First observe that the operator $(\cdot)_\Delta$ has been defined for formulae without nominals and unbound state variables only. We extend its definition to the general case by letting $(\psi)_\Delta$ be undefined whenever for some ψ_j in the family of formulae needed to define $(\psi)_\Delta$ there is no $\downarrow t$, $t \in \text{NOM} \cup \text{SVAR}$, such that the suffix starting at $@_t$ is free of $@$ -operators.

Now, if $(\varphi)_\Delta$ and $(\varphi')_\Delta$ are both defined, then all states that φ speaks about are situated in the same strongly connected component. It thus suffices to replace the conditions (i)–(iv) imposed on $(\varphi)_\Delta$ and $(\varphi')_\Delta$ with the following three. φ is satisfiable over ER frames if and only if

- (i') $P_{i_\mu} = P_{j_\nu} = \diamond$, or
- (ii') $P_{i_\mu} = \diamond$, $P_{j_\nu} = \square$ and $j_\nu < i_\mu$.
- (iii') $P_{i_\nu} = \diamond$, $P_{j_\mu} = \square$ and $i_\nu < j_\mu$.

If otherwise $(\varphi)_\Delta$ or $(\varphi')_\Delta$ is undefined, we may assume that the state labeled by the nominal (resp. by the state variable) corresponding to λ is bound in an equivalence class different from the equivalence class of the state that has to satisfy λ : assume, e. g., that $(\varphi')_\Delta$ is undefined due to $@_t$, $t \in \text{NOM}$. Then $K, g, w \models \varphi$ for the ER structure $K = (\{w, w'\}, \{(w, w), (w', w')\}, \eta)$ with $[\eta, g](t) = w'$. The case that $(\varphi)_\Delta$ is undefined is analogous. Hence φ is satisfiable.

Over arbitrary frames, it remains to extend the procedure for total-SAT to also recognize formulae φ being unsatisfiable over total but satisfiable over arbitrary or transitive frames (e. g., $\square 0$). This is the case if and only if φ contains a subformula $\square\psi$ such that none of the states in which φ is to be evaluated needs to have a successor (which might be enforced by preceding operators, consider, e. g., $@_i \diamond @_i \square 0$). More formally, for $1 < \ell \leq n$, let

$$(P_1 \cdots P_\ell 1)_\Delta = P_{i_1} \cdots P_{i_\ell} 1.$$

Then φ is satisfiable over arbitrary frames if and only if one of the conditions (i)–(iv) is satisfied or for some $1 \leq \ell \leq n$, $P_{i_\mu} = \square$ and for all $1 \leq k < \ell$, $k \in \{i_1, \dots, i_\mu\}$ implies that $(P_1 \cdots P_k 1)_\Delta = P_{j_1} \cdots P_{j_\nu}$ contains less than μ modalities (i. e. $\mu > \nu$) or there exists a position ξ such that $1 \leq \xi \leq \nu$ such that $P_{i_\xi} = P_{j_\xi} = \diamond$.

Over transitive frames, we can combine the ideas above to an algorithm deciding the satisfiability for B -formulae with $[B] \subseteq \mathbb{N}$. Hence the theorem applies. \square

3.3.2 All Boolean functions Finally, let us consider the clones between D , S_1 and BF , the clone of all Boolean functions. For the classes of all frames, transitive frames, and equivalence relations, we can transfer results obtained for the set

$\{\wedge, \vee, \neg\}$ of Boolean functions to these clones using the following technical Lemma. Additionally, we show that we can reduce the satisfiability problem over the class of all frames to the one over the class of total frames, establishing undecidability for all hybrid languages in this case.

Lemma 3.18 $\mathfrak{F}\text{-SAT}(O, \{\wedge, \vee, \neg\}) \leq_{\text{cd}} \mathfrak{F}\text{-SAT}(O, B)$ for all considered frame classes $\mathfrak{F} \in \{\text{all}, \text{trans}, \text{total}, \text{ER}\}$, if $[B \cup \{1\}] = \text{BF}$ and $O \cap \{\downarrow, @\} \neq \emptyset$.

Proof. Take $\varphi \in \mathcal{H}\mathcal{L}(O, \{\wedge, \vee, \neg\})$. Since $[B \cup \{1\}] = [\{\wedge, \vee, \neg\}] = \text{BF}$, we can rewrite φ as an $\mathcal{H}\mathcal{L}(O, B \cup \{1\})$ -formula φ' , leaving modal and hybrid operators untouched, ue to [16,26]. It can easily be seen, that this can be done in constant depth. Now we can easily transform φ' into an $\mathcal{H}\mathcal{L}(O, B)$ -formula φ'' , replacing all occurrences of 1 with $\downarrow x.x$ or $@_x x$. Clearly, φ and φ'' are equisatisfiable over \mathfrak{F} . \square

Lemma 3.19 $\text{SAT}(O, \{\wedge, \vee, \neg\}) \leq_{\text{cd}} \text{total-SAT}(O, \{\wedge, \vee, \neg\})$, for every set of operators $O \subseteq \{\diamond, \square, \downarrow, @\}$.

Proof. Let B denote the set $\{\wedge, \vee, \neg\}$. We recursively define the reduction function $(\cdot)^r : \mathcal{H}\mathcal{L}(O, B) \rightarrow \mathcal{H}\mathcal{L}(O, B)$ as follows,

$$\begin{aligned} a^r &= a, & a \in \text{ATOM} & & (@_t \varphi)^r &= @_t(p \wedge \varphi^r) \\ (c(\varphi_1, \dots, \varphi_n))^r &= c(\varphi_1^r, \dots, \varphi_n^r) & & & (\downarrow x.\varphi)^r &= \downarrow x.\varphi^r \\ (\diamond \varphi)^r &= \diamond(p \wedge \varphi^r) & & & & \end{aligned}$$

where p is a fresh atomic proposition and c refers to an arbitrary n -ary Boolean operator. We show that, for any $\varphi \in \mathcal{H}\mathcal{L}(O, B)$, $\varphi \in \text{SAT}(O, B)$ if and only if $\varphi^r \in \text{total-SAT}(O, B)$.

For the “ \Rightarrow ” direction, assume $K, g, w \models \varphi$ for $K = (W, R, \eta)$. From K , we construct a total Kripke structure $K' = (W', R', \eta')$ with $W' = W \cup \{\tilde{w} \mid w \in W\}$, $R' = R \cup \{(w, \tilde{w}), (\tilde{w}, \tilde{w}) \mid w \in W\}$, $\eta'(p) = W$, and $\eta'(x) = \eta(x)$ for all other atomic propositions and nominals x . It is straightforward to show inductively that $K', g, w \models \varphi^r$.

For the “ \Leftarrow ” direction, assume that $K, g, w \models \varphi^r$ for a total Kripke structure $K = (W, R, \eta)$. From K , we construct a Kripke structure $K' = (W', R', \eta')$ with $W' = \{w\} \cup \eta(p) \cup \bigcup_{i \in \text{NOM}} \eta(i) \cup \bigcup_{v \in \text{SVAR}} \{g(v)\}$, $R' = R \cap W'$, $\eta'(x) = \eta(x) \cap W'$ for all other atomic propositions and nominals x . It is straightforward to show inductively that $K', g, w \models \varphi$. \square

Lemmata 3.18 and 3.19, together with Theorem 2.1, yield the following theorem.

Theorem 3.20 Let $[B \cup \{1\}] = \text{BF}$. Then:

- (1) $\text{SAT}(O, B)$ and $\text{total-SAT}(O, B)$ are coRE -complete, for any $O \supseteq \{\diamond, \downarrow\}$.
- (2) $\text{trans-SAT}(\{\diamond, \downarrow, @\}, B)$ is coRE -complete.
- (3) $\text{trans-SAT}(\{\diamond, \downarrow\}, B)$ is NEXPTIME -complete.
- (4) $\text{ER-SAT}(\{\diamond, \downarrow\}, B)$ and $\text{ER-SAT}(\{\diamond, \downarrow, @\}, B)$ are NEXPTIME -complete.

4 Conclusions

We have almost completely classified the complexity of hybrid binder logics over four frame classes with respect to all possible combinations of Boolean operators, see Figure 1.

The main open question is the one for tight upper bounds for the monotone fragments including the \Box -operator over the classes of all and of transitive frames. A classification for the monotone fragments restricting the frames to transitive trees, total transitive trees, linear orders, and the natural numbers is provided by [17] which has been submitted for publication.

Another open questions concerns the hybrid languages including the \Box -operator but not the $@$ -operator over the class of transitive frames. The complexity for the respective satisfiability problems based on V , E , and M is open; in the case of V even for the class of all frames. For I , containment in AC^0 follows from an analysis of the proof of Theorem 3.15.

Finally, we could not obtain any bounds on the complexity for problems based on L , besides the LOGSPACE lower bound inherited from Theorem 3.3.

We are currently investigating the same problems over frame classes important for representing modal properties, such as transitive trees, linear frames and the natural numbers. Here, satisfiability for \downarrow , $@$, and arbitrary Boolean operators is already decidable, but with a nonelementary lower bound; hence, a complexity analysis is worthwhile as well. Because each such frame is acyclic, the fact that certain formulae are always satisfied in the singleton reflexive frame is not helpful any longer. This makes obtaining upper bounds more difficult. On the other hand, we can also express the constant 0 by $\downarrow x.\Diamond x$, which reduces the sets of Boolean operators to consider. Results for some of these cases, namely for monotone fragments over acyclic frame classes, are submitted for publication [17].

References

- [1] Areces, C., P. Blackburn and M. Marx, *Hybrid logic is the bounded fragment of first-order logic*, in: *Proc. WOLLIC-99*, 1999, pp. 35–50.
- [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., E. Hemaspaandra, H. Schnoor and I. Schnoor, *Generalized modal satisfiability.*, in: *Proc. STACS*, 2006, pp. 500–511.
- [4] 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* (2007), pp. 125–140.
- [5] Bauland, M., T. Schneider, H. Schnoor, I. Schnoor and H. Vollmer, *The complexity of generalized satisfiability for linear temporal logic*, *Logical Methods in Computer Science* **18** (2009), pp. 493–514.
- [6] Blackburn, P., *Representation, reasoning, and relational structures: a hybrid logic manifesto*, *Logic Journal of the IGPL* **8** (2000).
- [7] Blackburn, P. and J. Seligman, *Hybrid languages*, *J. of Logic, Language and Inform.* **4** (1995), pp. 41–62.
- [8] Böhler, E., N. Creignou, S. Reith and H. Vollmer, *Playing with Boolean blocks, part I: Post’s lattice with applications to complexity theory*, *ACM-SIGACT Newsletter* **34** (2003), pp. 38–52.
- [9] 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.

- [10] Donini, F., B. Hollunder, M. Lenzerini, D. Nardi, W. Nutt and A. Spaccamela, *The complexity of existential quantification in concept languages*, Artificial Intelligence **53** (1992), pp. 309–327.
- [11] Etessami, K., *Counting quantifiers, successor relations, and logarithmic space*, Journal of Computer and System Sciences **54** (1997), pp. 400–411.
- [12] Franceschet, M., M. de Rijke and B. Schlingloff, *Hybrid logics on linear structures: Expressivity and complexity*, in: *Proc. 10th TIME* (2003), pp. 166–173.
- [13] Goranko, V., *Hierarchies of modal and temporal logics with reference pointers*, Journal of Logic, Language and Information **5** (1996), pp. 1–24.
- [14] Hemaspaandra, E., *The complexity of poor man’s logic*, Journal of Logic and Computation **11** (2001), pp. 609–622, corrected version available at arXiv (2005).
- [15] Ladner, R., *The computational complexity of provability in systems of modal propositional logic*, SIAM Journal on Computing **6** (1977), pp. 467–480.
- [16] Lewis, H., *Satisfiability problems for propositional calculi*, Math. Sys. Theory **13** (1979), pp. 45–53.
- [17] Meier, A., M. Mundhenk, T. Schneider, M. Thomas and F. Weiss, *The complexity of satisfiability for monotone fragments of hybrid logic* (2010), submitted for publication.
- [18] Mundhenk, M. and T. Schneider, *The complexity of hybrid logics over equivalence relations*, J. of Logic, Language and Information. (2009), pp. 493–514.
- [19] Mundhenk, M., T. Schneider, T. Schwentick and V. Weber, *Complexity of hybrid logics over transitive frames*, Logic J. of the IGPL, To appear.
- [20] Papadimitriou, C. H., “Computational Complexity,” Addison-Wesley, 1994.
- [21] Pippenger, N., “Theories of Computability,” Cambridge University Press, 1997.
- [22] Post, E., *The two-valued iterative systems of mathematical logic*, Annals of Mathematical Studies **5** (1941), pp. 1–122.
- [23] Regan, K. and H. Vollmer, *Gap-languages and log-time complexity classes*, Theoretical Computer Science **188** (1997), pp. 101–116.
- [24] Schaefer, T. J., *The complexity of satisfiability problems*, in: *Proc. STOC* (1978), pp. 216–226.
- [25] Schneider, T., “The Complexity of Hybrid Logics over Restricted Classes of Frames,” Ph.D. thesis, Univ. of Jena (2007).
- [26] Schnoor, H., “Algebraic Techniques for Satisfiability Problems,” Ph.D. thesis, Univ. of Hannover (2007).
- [27] Schwentick, T. and V. Weber, *Bounded-variable fragments of hybrid logics*, in: *Proc. 24th STACS*, LNCS **4393** (2007), pp. 561–572.
- [28] 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.
- [29] Vollmer, H., “Introduction to Circuit Complexity,” Springer, 1999.
- [30] Weber, V., *Branching-time logics repeatedly referring to states*, Journal of Logic, Language and Information **18** (2009), pp. 593–624.