Validity and Entailment in Modal and Propositional Dependence Logics

Miika Hannula

Department of Computer Science, University of Auckland, Auckland, New Zealand
m.hannula@auckland.ac.nz

Abstract

The computational properties of modal and propositional dependence logics have been extensively studied over the past few years, starting from a result by Sevenster showing $\text{NEXPTIME}$-completeness of the satisfiability problem for modal dependence logic. Thus far, however, the validity and entailment properties of these logics have remained uncharacterised to a great extent. This paper establishes a complete classification of the complexity of validity and entailment in modal and propositional dependence logics. In particular, we address the question of the complexity of validity in modal dependence logic. By showing that it is $\text{NEXPTIME}$-complete we refute an earlier conjecture proposing a higher complexity for the problem.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases modal logic, propositional logic, dependence logic, entailment, validity, complexity

Digital Object Identifier 10.4230/LIPIcs.CSL.2017.28

1 Introduction

The notions of dependence and independence are pervasive in various fields of science. Usually these concepts manifest themselves in the presence of multitudes (e.g., events or experiments). Dependence logic is a recent logical formalism which, in contrast to others, has exactly these multitudes as its underlying concept [26]. In this article we study dependence logic in the propositional and modal logic context and present a complete classification of the computational complexity of their associated entailment and validity problems.

In first-order logic, the standard formal language behind all mathematics and computer science, dependencies between variables arise strictly from the order of their quantification. Consequently, more subtle forms of dependencies cannot be captured, a phenomenon exemplified by the fact that first-order logic lacks expressions for statements of the form

"for all $x$ there is $y$, and for all $u$ there is $v$, such that $R(x, y, u, v)$"

where $y$ and $v$ are to be chosen independently from one another. To overcome this barrier, branching quantifiers of Henkin and independence-friendly logic of Hintikka and Sandu suggested the use of quantifier manipulation [13, 14]. Dependence logic instead extends first-order logic at the atomic level with the introduction of new dependence atoms

$$\text{dep}(x_1, \ldots, x_n)$$

which indicate that the value of $x_n$ depends only on the values of $x_1, \ldots, x_{n-1}$. Dependence atoms are evaluated over teams, i.e., sets of assignments which form the basis of team
semantics. The concept of team semantics was originally proposed by Hodges in refutation of the view of Hintikka that the logics of imperfect information, such as his independence-friendly logic, escape natural compositional semantics [15]. By the development of dependence logic it soon became evident that team semantics serves also as a connecting link between the aforementioned logics and the relational database theory. In particular, team semantics enables the extensions of even weaker logics, such as modal and propositional logics, with various sophisticated dependency notions known from the database literature [5, 7, 16, 17].

In this article we consider modal and propositional dependence logics that extend modal and propositional logics with dependence atoms similar to (1), the only exception being that dependence atoms here declare dependencies between propositions [27]. We establish a complete classification of the computational complexity of the associated entailment and validity problems, including a solution to an open problem regarding the complexity of validity in modal dependence logic.

Modal dependence logic was introduced by Väänänen in 2008, and soon after it was shown to enjoy a \textsc{Nexptime}-complete satisfiability problem [25]. Since then the expressivity, complexity, and axiomatizability properties of modal dependence logic and its variants have been exhaustively studied. Especially the complexity of satisfiability and model checking for modal dependence logic and its variants has been already comprehensively classified [3, 4, 9, 10, 12, 16, 17, 20, 22]. Furthermore, entailment and validity of modal and propositional dependence logics have been axiomatically characterized by Yang and Väänänen in [30, 31, 32] and also by Sano and Virtema in [24]. Against this background it is rather surprising that the related complexity issues have remained almost totally unaddressed. The aim of this article is to address this shortage in research by presenting a complete classification with regards to these questions. A starting point for this endeavour is a recent result by Virtema which showed that the validity problem for propositional dependence logic is \textsc{Nexptime}-complete [28, 29]. In that paper the complexity of validity for modal dependence logic remained unsettled, although it was conjectured to be harder than that for propositional dependence logic. This conjecture is refuted in this paper as the same exact \textsc{Nexptime} bound is shown to apply to modal dependence logic as well. Furthermore, we show that this result applies to the extension of propositional dependence logic with quantifiers as well as to the so-called extended modal logic which can express dependencies between arbitrary modal formulae (instead of simple propositions). We also extend our investigations to the entailment problem and show that for both (quantified) propositional and (extended) modal dependence logics this problem is \textsc{co-Nexptime-Np}-complete. Moreover, our investigations show that for modal logic extended with so-called intuitionistic disjunction the associated entailment, validity, and satisfiability problems are all \textsc{Pspace}-complete, which is, in all the three categories the complexity of the standard modal logic. The outcome of these investigations is a complete picture of the validity and entailment properties of modal and propositional dependence logics, summarized in Table 1 in the conclusion section.

The obtained results have interesting consequences. First, combining results from this paper and [25, 29] we observe that similarly to the standard modal logic case the complexity of validity and satisfiability coincide for (extended) modal dependence logic. It is worth pointing out here that satisfiability and validity cannot be seen as each other’s duals in the dependence logic context. Dependence logic cannot express negation nor logical implication which renders its associated validity, satisfiability, and entailment problems genuinely different. Secondly, it was previously known that propositional and modal dependence logics deviate on the complexity of their satisfiability problem (\textsc{NP}-complete vs. \textsc{Nexptime}-complete [20, 25], resp.) and that the standard propositional and modal logics differ from one another
on both satisfiability and validity (\(\text{NP}\)-complete/\(\text{co-NP}\)-complete vs. \(\text{PSPACE}\)-complete, resp. \([2, 18, 19]\)). Based on this it is somewhat surprising to find out that modal and propositional dependence logics correspond to one another in terms of the complexity of both their validity and entailment problems.

**Organization.** This article is organized as follows. In Section 2 we present some notation and background assumptions. In Section 3 we give a short introduction to modal dependence logics, followed by Section 4 which proves \(\text{co-NEXPTIME}^{\text{NP}}\)-membership for modal dependence logic entailment. In Section 5 we define (quantified) propositional dependence logics, and in the subsequent section we show \(\text{co-NEXPTIME}^{\text{NP}}\)-hardness for entailment in this logic. In Section 7 we draw our findings together, followed by Section 8 that is reserved for conclusions. For some of the proofs we refer the reader to Appendix.

## 2 Preliminaries

We assume that the reader is familiar with the basic concepts of propositional and modal logic, as well as those of computational complexity. Let us note at this point that all the hardness results in the paper are stated under polynomial-time reductions.

**Notation.** Following the common convention we assume that all our formulae in the team semantics context appear in negation normal form (NNF). We use \(p, q, r, \ldots\) to denote propositional variables. For two sequences \(\pi\) and \(\bar{b}\), we write \(\pi\bar{b}\) to denote their concatenation. For a function \(f\) and a sequence \((a_1, \ldots, a_n)\) of elements from the domain of \(f\), we denote by \(f(a_1, \ldots, a_n)\) the sequence \((f(a_1), \ldots, f(a_n))\). Let \(\phi\) be any formula. Then \(\text{Var}(\phi)\) refers to the set of variables appearing in \(\phi\), and \(\text{Fr}(\phi)\) to the set of free variables appearing in \(\phi\), both defined in the standard way. We sometimes write \(\phi(p_1, \ldots, p_n)\) instead of \(\phi\) to emphasize that \(\text{Fr}(\phi) = \{p_1, \ldots, p_n\}\). For a subformula \(\phi_0\) of \(\phi\) and a formula \(\theta\), we write \(\phi(\theta/\phi_0)\) to denote the formula obtained from \(\phi\) by substituting \(\theta\) for \(\phi_0\). We use \(\phi^{\top}\) to denote the NNF formula obtained from \(\neg\phi\) by pushing the negation to the atomic level, and sometimes \(\phi^{\top}\) to denote \(\phi\).

## 3 Modal Dependence Logics

The syntax of modal logic (ML) is generated by the following grammar:

\[
φ ::= p \mid \neg p \mid (φ \land φ) \mid (φ \lor φ) \mid \Box φ \mid \Diamond φ.
\]  

(2)

Extensions of modal logic with different dependency notions are made possible via a generalization of the standard Kripke semantics by teams, here defined as sets of worlds. A Kripke model over a set of variables \(V\) is a tuple \(M = (W, R, π)\) where \(W\) is a non-empty set of worlds, \(R\) is a binary relation over \(W\), and \(π : V \rightarrow P(W)\) is a function that associates each variable with a set of worlds. A team \(T\) of a Kripke model \(M = (W, R, π)\) is a subset of \(W\). For the team semantics of modal operators we define the set of successors of a team \(T\) as \(R[T] := \{w \in W \mid \exists w' \in T : (w', w) \in R\}\) and the set of legal successor teams of a team \(T\) as \(R(T) := \{T' \subseteq R[T] \mid \forall w \in T \exists w' \in T' : (w, w') \in R\}\). The team semantics of modal logic is now defined as follows.

**Definition 1 (Team Semantics of ML).** Let \(φ\) be an ML formula, let \(M = (W, R, π)\) be a Kripke model over \(V \supseteq \text{Var}(φ)\), and let \(T \subseteq W\). The satisfaction relation \(M, T \models φ\) is defined as follows:
We write $\phi \equiv \psi$ to denote that $\phi$ and $\psi$ are equivalent, i.e., for all Kripke models $M$ and teams $T$, $M, T \models \phi$ iff $M, T \models \psi$. Let $\Sigma \cup \{ \phi \}$ be a set of formulae. We write $M, T \models \Sigma$ iff $M, T \models \phi$ for all $\phi \in \Sigma$, and say that $\Sigma$ entails $\phi$ if for all $M$ and $T$, $M, T \models \Sigma$ implies $M, T \models \phi$. Let $L$ be a logic in the team semantics setting. The entailment problem for $L$ is to decide whether $\Sigma$ entails $\phi$ (written $\Sigma \models \phi$) for a given finite set of formulae $\Sigma \cup \{ \phi \}$ from $L$. The validity problem for $L$ is to decide whether a given formula $\phi \in L$ is satisfied by all Kripke models and teams. The satisfiability problem for $L$ is to decide whether a given formula $\phi \in L$ is satisfied by some Kripke model and a non-empty team.$^1$

The following flatness property holds for all modal logic formulae. Notice that by $\models_{ML}$ we refer to the usual satisfaction relation of modal logic.

**Proposition 2 (Flatness [25]).** Let $\phi$ be a formula in $ML$, let $M = (W, R, \pi)$ be a Kripke model over $V \supseteq \text{Var}(\phi)$, and let $T \subseteq W$ be a team. Then:

$$M, T \models \phi \iff \forall w \in T : M, w \models_{ML} \phi.$$ 

Team semantics gives rise to different extensions of modal logic capable of expressing various dependency notions. In this article we consider dependence atoms that express functional dependence between propositions. To facilitate their associated semantic definitions, we first define for each world $w$ of a Kripke model $M$ a truth function $w_M$ from $ML$ formulae into $\{0, 1\}$ as follows:

$$w_M(\phi) = \begin{cases} 
1 & \text{if } M, \{w\} \models \phi, \\
0 & \text{otherwise}.
\end{cases}$$

**Dependence atom.** Modal dependence logic (MDL) is obtained by extending $ML$ with dependence atoms

$$\text{dep}(\overline{p}, q)$$ (3)

where $\overline{p}$ is a sequence of propositional atoms and $q$ is a single propositional atom. Furthermore, we consider extended dependence atoms of the form

$$\text{dep}(\overline{\phi}, \psi)$$ (4)

where $\overline{\phi}$ is a sequence of $ML$ formulae and $\psi$ is a single $ML$ formula. The extension of $ML$ with atoms of the form (4) is called extended modal dependence logic (EMDL). Atoms of

---

$^1$ The empty team satisfies all formulae trivially.
Then:

\[ \mathcal{M}, T \models \text{dep}(\phi, \psi) :\iff \forall w, w' \in T : w_{\mathcal{M}}(\phi) = w'_{\mathcal{M}}(\phi) \text{ implies } w_{\mathcal{M}}(\psi) = w'_{\mathcal{M}}(\psi). \]

For the sake of our proof arguments, we also extend modal logic with predicates. The syntax of relativational modal logic (RML) is given by the grammar:

\[ \phi ::= p | \sim \phi | (\phi \land \phi) | \Box \phi | S(\phi_1, \ldots, \phi_n). \]  \hfill (5)

The formulae of RML are evaluated over relational Kripke models \( \mathcal{M} = (W, R, \pi, S_1^{\mathcal{M}}, \ldots, S_n^{\mathcal{M}}) \) where each \( S_i^{\mathcal{M}} \) is a set of binary sequences of length \( \#S_i \), that is, the arity of the relation symbol \( S_i \). We denote by \( \mathcal{M}, w \models_{\text{RML}} \phi \) the satisfaction relation obtained by extending the standard Kripke semantics of modal logic as follows:

\[ \mathcal{M}, w \models_{\text{RML}} S(\phi_1, \ldots, \phi_n) :\iff (w_{\mathcal{M}}(\phi_1), \ldots, w_{\mathcal{M}}(\phi_n)) \in S^\mathcal{M}. \]

Notice also that by \( \sim \) we refer to the contradictory negation, e.g., here \( \mathcal{M}, w \models_{\text{RML}} \sim \phi :\iff \mathcal{M}, w \not\models_{\text{RML}} \phi \). The team semantics for \( \sim \) is also defined analogously. Recall that the negation symbol \( \sim \) is used only in front of propositions, and \( \neg p \) is satisfied by a Kripke model \( \mathcal{M} \) and a team \( T \) iff in the standard Kripke semantics it is satisfied by all pointed models \( \mathcal{M}, w \) where \( w \in T \).

In addition to the aforementioned dependency notions we examine so-called intuitionistic disjunction \( \boxdot \) defined as follows:

\[ \mathcal{M}, T \models \phi_1 \boxdot \phi_2 :\iff \mathcal{M}, T \models \phi_1 \text{ or } \mathcal{M}, T \models \phi_2. \]  \hfill (6)

We denote the extension of ML with intuitionistic disjunction \( \boxdot \) by \( \text{ML}(\boxdot) \). Notice that the logics MDL and EMDL are expressively equivalent to \( \text{ML}(\boxdot) \) but exponentially more succinct as the translation of (3) to \( \text{ML}(\boxdot) \) involves a necessary exponential blow-up [11]. All these logics satisfy the following downward closure property which will be used in the upper bound result.

\begin{proposition}[Downward Closure [3, 27, 31]]
Let \( \phi \) be a formula in MDL, EMDL, or \( \text{ML}(\boxdot) \), let \( \mathcal{M} = (W, R, \pi) \) be a Kripke model over \( V \supseteq \text{Var}(\phi) \), and let \( T \subseteq W \) be a team. Then:

\[ T' \subseteq T \text{ and } \mathcal{M}, T \models \phi \implies \mathcal{M}, T' \models \phi. \]
\end{proposition}

We can now proceed to the upper bound result which states that the entailment problem for EMDL is decidable in \text{co-NEXPTIME}^\text{NP}.

## 4 Upper Bound for EMDL Entailment

In this section we show that EMDL entailment is in \text{co-NEXPTIME}^\text{NP}. The idea is to represent dependence atoms using witnessing functions guessed universally on the left-hand side and existentially on the right-hand side of an entailment problem \( \{\phi_1, \ldots, \phi_{n-1}\} \models \phi_n \).

This reduces the problem to validity of an RML formula of the form \( \phi_1^* \land \ldots \land \phi_{n-1}^* \to \phi_n^* \) where \( \phi_i^* \) is obtained by replacing in \( \phi_i \) all dependence atoms with relational atoms whose interpretations are bound by the guess. We then extend an Algorithm by Ladner that shows a \text{PSPACE} upper bound for the validity problem of modal logic [18]. As a novel...
algorithmic feature we introduce recursive steps for relational atoms that query to the guessed functions. The \textbf{co-NEXPTIME} upper bound then follows by a straightforward running time analysis.

We start by showing how to represent dependence atoms using intuitionistic disjunctions defined over witnessing functions. Let \(\pi = (\alpha_1, \ldots, \alpha_n)\) be a sequence of ML formulae and let \(\beta\) be a single ML formula. Then we say that a function \(f : \{\top, \bot\}^n \rightarrow \{\top, \bot\}\) is a witness of \(d := \text{dep}(\pi, \beta)\), giving rise to a witnessing ML formula

\[
D(f, d) := \bigvee_{a_1, \ldots, a_n \in \{\top, \bot\}} \alpha_1^{a_1} \land \cdots \land \alpha_n^{a_n} \land \beta^{f[a_1, \ldots, a_n]}.
\]  

(7)

The equivalence

\[
d \equiv \bigvee_{f : \{\top, \bot\}^n \rightarrow \{\top, \bot\}} D(f, d)
\]

has been noticed in the contexts of MDL and EMDL respectively in [27, 3]. Note that a representation of the sort (8) necessitates that the represented formula, in this case the dependence atom \(d\), has the downward closure property.

To avoid the exponential blow-up involved in both (7) and (8), we instead relate to RML by utilizing the following equivalence:

\[
(W, R, \pi) \vdash_{\text{RML}} D(f, d) \iff (W, R, \pi, S^M) \vdash_{\text{RML}} S(\pi\beta),
\]

(9)

where \(S^M := \{(a_1, \ldots, a_n, b) \in \{0, 1\}^{n+1} \mid f(a_1, \ldots, a_n) = b\}\). Before proceeding to the proof, we need the following simple proposition, based on [29, 31] where the statement has been proven for empty \(\Sigma\).

\textbf{Proposition 4.} Let \(\Sigma\) be a set of ML formulae, and let \(\phi_0, \phi_1 \in \text{ML}(\emptyset)\). Then \(\Sigma \models \phi_0 \otimes \phi_1\) iff \(\Sigma \models \phi_0\) or \(\Sigma \models \phi_1\).

\textbf{Proof.} It suffices to show the only-if direction. Assume first that \(\mathcal{L} = \text{ML}\), and let \(\mathcal{M}_0, T_0\) and \(\mathcal{M}_1, T_1\) be counterexamples to \(\Sigma \models \phi_0\) and \(\Sigma \models \phi_1\), respectively. W.l.o.g. we may assume that \(\mathcal{M}_0\) and \(\mathcal{M}_1\) are disjoint. Since the truth value of a \(\text{ML}(\emptyset)\) formula is preserved under taking disjoint unions of Kripke models (see Theorem 6.1.9. in [31], also Proposition 2.13. in [29]) we obtain that \(\mathcal{M}, T_0 \models \Sigma \cup \{\lnot \phi_0\}\) and \(\mathcal{M}, T_1 \models \Sigma \cup \{\lnot \phi_1\}\) where \(\mathcal{M} = \mathcal{M}_0 \cup \mathcal{M}_1\).

By the downward closure property of \(\text{ML}(\emptyset)\) (Proposition 3), and by the flatness property of \(\text{ML}\) (Proposition 2), we then obtain that \(\mathcal{M}, T \models \Sigma \cup \{\lnot \phi_0, \lnot \phi_1\}\) where \(T = T_0 \cup T_1\). \(\blacktriangleleft\)

The proof now proceeds via Lemmata 5 and 6 of which the former constitutes the basis for our alternating exponential-time algorithm. Note that if \(\phi\) is an EMDL formula with \(k\) dependence atom subformulæ, listed (possibly with repetitions) in \(d_1, \ldots, d_k\), then we call \(\mathcal{J} = (f_1, \ldots, f_k)\) a \textit{witness sequence} of \(\phi\) if each \(f_i\) is a witness of \(d_i\). Furthermore, we denote by \(\phi(\mathcal{J}/\mathcal{D})\) the ML formula obtained from \(\phi\) by replacing each \(d_i\) with \(D(f_i, d_i)\).

\textbf{Lemma 5.} Let \(\phi_1, \ldots, \phi_n\) be formulae in EMDL. Then \(\{\phi_1, \ldots, \phi_{n-1}\} \models \phi_n\) iff for all witness sequences \(\mathcal{J}_1, \ldots, \mathcal{J}_{n-1}\) of \(\phi_1, \ldots, \phi_{n-1}\) there is a witness sequence \(\mathcal{J}\) of \(\phi_n\) such that

\[
\{\phi_1(\mathcal{J}_1/\mathcal{D}_1), \ldots, \phi_{n-1}(\mathcal{J}_{n-1}/\mathcal{D}_{n-1})\} \models \phi_n(\mathcal{J}_n/\mathcal{D}_n).
\]

\textbf{Proof.} Assume first that \(\phi\) is an arbitrary formula in EMDL, and let \(d := \text{dep}(\pi, \beta)\) be a subformula of \(\phi\). It is straightforward to show that \(\phi\) is equivalent to

\[
\bigvee_{f : \{\top, \bot\}^n \rightarrow \{\top, \bot\}} \phi(D(f, d)/d).
\]
Iterating these substitutions we obtain that \( \{ \phi_1, \ldots, \phi_{n-1} \} \models \phi_n \) iff
\[
\bigvee_{\mathcal{F}_1} \phi_1(\mathcal{F}_1/\bar{\alpha}_1), \ldots, \bigvee_{\mathcal{F}_{n-1}} \phi_1(\mathcal{F}_{n-1}/\bar{\alpha}_{n-1}) \models \bigvee_{\mathcal{F}_n} \phi_1(\mathcal{F}_n/\bar{\alpha}_n),
\]
(10)
where \( \mathcal{F}_i \) ranges over the witness sequences of \( \phi_i \). Then (10) holds iff for all \( \mathcal{F}_1, \ldots, \mathcal{F}_{n-1} \),
\[
\{ \phi_1(\mathcal{F}_1/\bar{\alpha}_1), \ldots, \phi_{n-1}(\mathcal{F}_{n-1}/\bar{\alpha}_{n-1}) \} \models \bigvee_{\mathcal{F}_n} \phi_1(\mathcal{F}_n/\bar{\alpha}_n).
\]
(11)
Notice that each formula \( \phi_1(\mathcal{F}_i/\bar{\alpha}_i) \) belongs to ML. Hence, by Proposition 4 we conclude that (11) holds iff for all \( \mathcal{F}_1, \ldots, \mathcal{F}_{n-1} \) there is \( \mathcal{F}_n \) such that
\[
\{ \phi_1(\mathcal{F}_1/\bar{\alpha}_1), \ldots, \phi_{n-1}(\mathcal{F}_{n-1}/\bar{\alpha}_{n-1}) \} \models \phi_1(\mathcal{F}_n/\bar{\alpha}_n).
\]
(12)

The next proof step is to reduce an entailment problem of the form (12) to a validity problem of an RML formula over relational Kripke models whose interpretations agree with the guessed functions. For the latter problem we then apply Algorithm 1 whose lines 1-14 and 19-26 constitute an algorithm of Ladner that shows the \( \text{PSPACE} \) upper bound for modal logic satisfiability [18]. Lines 15-18 consider those cases where the subformula is relational. Lemma 6 now shows that, given an oracle \( A \), this extended algorithm yields a \( \text{PSPACE}^A \) decision procedure for satisfiability of RML formulae over relational Kripke models whose predicates agree with \( A \). For an oracle set \( A \) of words from \( \{0,1,\#\}^* \) and \( k \)-ary relation symbol \( R_i \), we define \( R_i^A := \{(a_1, \ldots, a_k) \in \{0,1\}^k \mid \text{bin}(i) \# a_1 \ldots a_k \in A\} \). Note that by \( a \vdash b \) we denote the concatenation of two strings \( a \) and \( b \).

▶ Lemma 6. Given an RML-formula \( \phi \) over a vocabulary \( \{S_1, \ldots, S_n\} \) and an oracle set of words \( A \) from \( \{0,1,\#\}^* \), Algorithm 1 decides in \( \text{PSPACE}^A \) whether there is a relational Kripke model \( \mathcal{M} = (W,R,\pi,S_1^A,\ldots,S_n^A) \) and a world \( w \in W \) such that \( \mathcal{M},w \models_{\text{RML}} \phi \).

Proof. We leave it to the reader to show (by a straightforward structural induction) that, given an input \( (A, B, C, D) \) where \( A, B, C, D \subseteq \text{RML} \), Algorithm 1 returns \( \text{Sat}(A,B,C,D) \) true iff there is a relational Kripke model \( \mathcal{M} = (W,R,\pi,S_1^A,\ldots,S_n^A) \) such that
\[
\mathcal{M},w \models_{\text{RML}} \phi \text{ if } \phi \in A; \\
\mathcal{M},w \not\models_{\text{RML}} \phi \text{ if } \phi \in B; \\
\mathcal{M},w \not\models_{\text{RML}} \Box \phi \text{ if } \phi \in C; \text{ and} \\
\mathcal{M},w \not\models_{\text{RML}} \Diamond \phi \text{ if } \phi \in D.
\]

Hence, \( \text{Sat}((\psi),\emptyset,\emptyset,\emptyset) \) returns true iff \( \psi \) is satisfiable by \( \mathcal{M},w \) with relations \( S_1^M \) obtained from the oracle. We note that the selection of subformulae \( \phi \) from \( A \cup B \) can be made deterministically by defining an ordering for the subformulae. Furthermore, we note that this algorithm algorithm runs in \( \text{PSPACE}^A \) as it employs \( O(n) \) recursive steps that each take space \( O(n) \). A detailed space analysis (following that in [18]) can be found in Appendix. ◀

Using Lemmata 5 and 6 we can now show the co-NEXPTIME\(^{\text{NP}} \) upper bound. In the proof we utilize the following connection between alternating Turing machines and the exponential time hierarchy at the level co-NEXPTIME\(^{\text{NP}} \) = \( \Pi_2^{\text{EXP}} \).

▶ Theorem 7 ([1, 21]). \( \Sigma_1^{\text{EXP}} \) (or \( \Pi_2^{\text{EXP}} \)) is the class of problems recognizable in exponential time by an alternating Turing machine which starts in an existential (universal) state and alternates at most \( k \) many times.
Validity and Entailment in Modal and Propositional Dependence Logics

Algorithm 1: PSPACE\(^A\) algorithm for deciding validity in RML. Notice that queries to \(S_I^A\) range over \((b_1,\ldots,b_k) \in \{0,1\}^k\).

<table>
<thead>
<tr>
<th>Input</th>
<th>((A,B,C,D)) where (A,B,C,D \subseteq \text{RML} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>(\text{Sat}(A,B,C,D))</td>
</tr>
</tbody>
</table>

1. if \(A \cup B \not\subseteq \text{Prop} \) then
   2. choose \(\phi \in (A \cup B) \setminus \text{Prop} \);
   3. if \(\phi = \sim \psi \) and \(\phi \in A \) then
      4. return \(\text{Sat}(A \setminus \{\phi\}, B \cup \{\psi\}, C, D)\);
   5. else if \(\phi = \sim \psi \) and \(\phi \in B \) then
      6. return \(\text{Sat}(A \cup \{\psi\}, B \setminus \{\phi\}, C, D)\);
   7. else if \(\phi = \psi \wedge \theta \) and \(\phi \in A \) then
      8. return \(\text{Sat}((A \cup \{\psi, \theta\}) \setminus \{\phi\}, B, C, D)\);
   9. else if \(\phi = \psi \wedge \theta \) and \(\phi \in B \) then
      10. return \(\text{Sat}(A, (B \cup \{\psi\}) \setminus \{\phi\}, C, D) \lor \text{Sat}(A, (B \cup \{\theta\}) \setminus \{\phi\}, C, D)\);
   11. else if \(\phi = \Box \psi \) and \(\phi \in A \) then
      12. return \(\text{Sat}(A \setminus \{\phi\}, B \cup \{\psi\}, C, D)\);
   13. else if \(\phi = \Box \psi \) and \(\phi \in B \) then
      14. return \(\text{Sat}(A, B \setminus \{\phi\}, C \cup \{\psi\})\);
   15. else if \(\phi = S_i(\psi_1,\ldots,\psi_k) \) and \(\phi \in A \) then
      16. return \(\bigvee_{(b_1,\ldots,b_k) \in S_i} \text{Sat}((A \cup \{\psi_j : b_j = 1\}) \setminus \{\phi\}, B \cup \{\psi_j : b_j = 0\}, C, D)\);
   17. else if \(\phi = S_i(\psi_1,\ldots,\psi_k) \) and \(\phi \in B \) then
      18. return \(\bigvee_{(b_1,\ldots,b_k) \not\in S_i} \text{Sat}(A \cup \{\psi_j : b_j = 1\}, (B \cup \{\psi_j : b_j = 0\}) \setminus \{\phi\}, C, D)\);

19. end
20. else if \((A \cup B) \subseteq \text{Prop} \) then
   21. if \(A \cap B \neq \emptyset \) then
      22. return false;
   23. else if \(A \cap B = \emptyset \) and \(C \cap D \neq \emptyset \) then
      24. return \(\bigwedge_{D \in D} \text{Sat}(C, \{D\}), \emptyset, \emptyset\);
   25. else if \(A \cap B = \emptyset \) and \(C \cap D = \emptyset \) then
      26. return true;
   27. end
28. end

Theorem 8. The entailment problem for EMDL is in co-NEXPTIME\(^{NP}\).

Proof. Assuming an input \(\phi_1,\ldots,\phi_n\) of EMDL-formulae, we show how to decide in \(\Pi^P_2\) whether \(\{\phi_1,\ldots,\phi_{n-1}\} \models \phi_n\). By Theorem 7 it suffices to construct an alternating exponential-time algorithm that switches once from an universal to an existential state. By Lemma 5, \(\{\phi_1,\ldots,\phi_{n-1}\} \models \phi_n\) iff for all \(\overline{f}_1,\ldots,\overline{f}_{n-1}\) there is \(\overline{f}_n\) such that

\[
\{\phi_1(\overline{f}_1/\overline{a}_1),\ldots,\phi_{n-1}(\overline{f}_{n-1}/\overline{a}_{n-1})\} \models \phi(\overline{f}_n/\overline{a}_n).
\] (13)

Recall from the proof of Lemma 5 that all the formulae in (13) belong to ML. Hence by the flatness property (Proposition 2) \(\models\) is interchangeable with \(\models_{\text{ML}}\) in (13). It follows that (13) holds iff

\[
\phi := \phi_1(\overline{f}_1/\overline{a}_1) \land \ldots \land \phi_{n-1}(\overline{f}_{n-1}/\overline{a}_{n-1}) \land \sim \phi(\overline{f}_n/\overline{a}_n)
\] (14)
We will show that Algorithm 1 on with the grammar rule

\[ M \]

where \( M \) where \( M \) where \( M \) where \( M \)

The syntax of propositional dependence logic (PDL) is now defined as follows:

\[ \phi ::= p \mid \neg p \mid (\phi \land \phi) \mid (\phi \lor \phi) \]  

The syntax of propositional dependence logic (PDL) is obtained by extending the syntax of PL with dependence atoms of the form \( (\cdot) \). Furthermore, the syntax of PL(\( \Diamond \)) extends (15) with the grammar rule \( \phi ::= \phi \Diamond \phi \).

The formulas of these logics are evaluated against propositional teams. Let \( V \) be a set of variables. We say that a function \( s: V \to \{0,1\} \) is a (propositional) assignment over \( V \), and a (propositional) team \( X \) over \( V \) is a set of propositional assignments over \( V \). A team \( X \) over \( V \) induces a Kripke model \( \mathcal{M}_X = (T_X, \emptyset, \pi) \) where \( T_X = \{w_s \mid s \in X\} \) and \( w_s \in \pi(p) \iff s(p) = 1 \) for \( s \in X \) and \( p \in V \). The team semantics for propositional formulae is now defined as follows:

\[ X \models \phi \iff X :\!\!: \mathcal{M}_X, T_X \models \phi, \]

where \( \mathcal{M}_X, T_X \models \phi \) refers to the team semantics of modal formulae (see Sect. 3). If \( \phi^* \) is a formula obtained from \( \phi \) by replacing all propositional atoms \( p \) (except those inside a dependence atom) with predicates \( A(p) \), then we can alternatively describe that \( X \models \phi \iff \mathcal{M} = \{[0,1], A := \{1\}\} \) and \( X \) satisfy \( \phi^* \) under the lax team semantics of first-order dependence logics [5].

We will also examine validity and entailment in quantified propositional dependence logic which is a team semantics adaptation and generalization of the dependency quantified
Boolean formula problem [8]. This problem, shown to be NEXPTIME-complete in [23], extends the quantified Boolean formula problem, the standard PSPACE-complete problem, with the introduction of additional quantification constraints. To this end, we start with the introduction of quantified propositional logic. The syntax quantified propositional logic (QPL) is obtained by extending that of PL with universal and existential quantification over propositional variables. Their semantics is given in terms of so-called duplication and supplementation teams. Let \( p \) be a propositional variable and \( s \) an assignment over \( V \). We denote by \( s(a/p) \) the assignment over \( V \cup \{p\} \) that agrees with \( s \) everywhere, except that it maps \( p \) to \( a \). Universal quantification of a propositional variable \( p \) is defined in terms of duplication teams \( X[\{0,1\}/p] := \{s(a/p) \mid s \in X, a \in \{0,1\}\} \) that extend teams \( X \) with all possible valuations for \( p \). Existential quantification is defined in terms of supplementation teams \( X[F/p] := \{s(a/p) \mid s \in X, a \in F(s)\} \) where \( F \) is a mapping from \( X \) into \( \{\{0\}, \{1\}, \{0,1\}\} \). The supplementation team \( X[F/p] \) extends each assignment of \( X \) with a non-empty set of values for \( p \). The satisfaction relations \( X = \exists p \phi \) and \( X = \forall p \phi \) are now given as follows:

\[
X \models \exists p \phi \iff \exists F \in X \{\{0\}, \{1\}, \{0,1\}\} : X[F/p] \models \phi,
X \models \forall p \phi \iff X[\{0,1\}/p] \models \phi.
\]

We denote by QPDL the extension of PDL with quantifiers. Observe that the flatness and downward closure properties of modal formulae (Propositions 2 and 3, resp.) apply now analogously to propositional formulae. Note that by \( \models_{PL} \) we refer to the standard semantics of propositional logic.

**Proposition 10** (Flatness [26]). Let \( \phi \) be a formula in QPL, and let \( X \) be a team over \( V \supseteq Fr(\phi) \). Then:

\[
X \models \phi \iff \forall s \in X : s \models_{PL} \phi.
\]

**Proposition 11** (Downward Closure [26]). Let \( \phi \) be a formula in QPDL or QPL(\( \otimes \)), and let \( X \) be a team over a set \( V \supseteq Fr(\phi) \) of propositional variables. Then:

\[
Y \subseteq X \text{ and } X \models \phi \implies Y \models \phi.
\]

We denote the restriction of an assignment \( s \) to variables in \( V \) by \( s \upharpoonright V \), and define the restriction of a team \( X \) to \( V \), written \( X \upharpoonright V \), as \( \{s \upharpoonright V \mid s \in X\} \). We conclude this section by noting that, similarly to the first-order case, quantified propositional dependence logic satisfies the following locality property.

**Proposition 12** (Locality [26]). Let \( \phi \) be a formula in \( L \) where \( L \in \{QPL, \text{QPL(} \otimes \text{)}\} \), let \( X \) be a team over a set \( V \supseteq Fr(\phi) \), and let \( Fr(\phi) \subseteq V' \subseteq V \). Then:

\[
X \models \phi \iff X \upharpoonright V' \models \phi.
\]

## 6 Lower Bound for PDL Entailment

In this section we prove that the entailment problem for PDL is co-NEXPTIME\( ^{NP} \)-hard. This result is obtained by reducing from a variant of the quantified Boolean formula problem, that is, the standard complete problem for PSPACE.
Definition 13 ([8]). A $\Sigma_k$-alternating dependency quantified Boolean formula ($\Sigma_k$-ADQBF) is a pair $(\phi, \mathcal{C})$ where $\phi$ is an expression of the form

$$\phi := (\exists f^1 \ldots \exists f^s)(\forall f^1 \ldots \forall f^s)(\exists f^{s+1}_1 \ldots \exists f^{s+1}_{k}) \ldots (Q f^1 \ldots Q f^k) \forall p_1 \ldots \forall p_n \theta,$$

where $Q \in \{\exists, \forall\}, \mathcal{C} = (\tau^1, \ldots, \tau^s)$ lists sequences of variables from $\{p_1, \ldots, p_n\}$, and $\theta$ is a quantifier-free propositional formula in which only the quantified variables $p_i$ and function symbols $f_j$ with arguments $\tau^i$ may appear. Analogously, a $\Pi_k$-alternating dependency quantified Boolean formula ($\Pi_k$-ADQBF) is a pair $(\phi, \mathcal{C})$ where $\phi$ is an expression of the form

$$\phi := (\forall f^1 \ldots \forall f^s)(\exists f^1 \ldots \exists f^s)(\forall f^{s+1}_1 \ldots \forall f^{s+1}_{k}) \ldots (Q f^1 \ldots Q f^k) \forall p_1 \ldots \forall p_n \theta,$$

The sequence $\mathcal{C}$ is called the constraint of $\phi$.

The truth value of a $\Sigma_k$-ADQBF or a $\Pi_k$-ADQBF instance is determined by interpreting each $Q f_j$ where $Q \in \{\exists, \forall\}$ as existential/universal quantification over Skolem functions $f_j : \{0, 1\}^{\mid \mathcal{C} \mid} \rightarrow \{0, 1\}$. Let us now denote the associated decision problems by $\text{TRUE}(\Sigma_k$-ADQBF) and $\text{TRUE}(\Pi_k$-ADQBF). These problems characterize levels of the exponential hierarchy in the following way.

Theorem 14 ([8]). Let $k \geq 1$. For odd $k$ the problem $\text{TRUE}(\Sigma_k$-ADQBF) is $\Sigma_k^{\text{EXP}}$-complete. For even $k$ the problem $\text{TRUE}(\Pi_k$-ADQBF) is $\Pi_k^{\text{EXP}}$-complete.

Since $\text{TRUE}(\Pi_2$-ADQBF) is $\text{co-NEXPTIME}^{\text{NP}}$-complete, we can show the lower bound via an reduction from it. Notice that regarding the validity problem of PDL, we already have the following lower bound.

Theorem 15 ([29]). The validity problem for PDL is $\text{NEXPTIME}$-complete, and for MDL and EMDL it is $\text{NEXPTIME}$-hard.

This result was shown by a reduction from the dependence quantified Boolean formula problem (i.e. $\text{TRUE}(\Sigma_1$-ADQBF)) to the validity problem of PDL. We use essentially the same technique to reduce from $\text{TRUE}(\Pi_2$-ADQBF) to the entailment problem of PDL.

Theorem 16. The entailment problem for PDL is $\text{co-NEXPTIME}^{\text{NP}}$-hard.

Proof. By Theorem 14 it suffices to show a reduction from $\text{TRUE}(\Pi_2$-ADQBF). Let $(\phi, \mathcal{C})$ be an instance of $\Pi_2$-ADQBF in which case $\phi$ is of the form

$$\forall f_1 \ldots \forall f_m \exists f_{m+1} \ldots \exists f_{m+m'} \forall p_1 \ldots \forall p_n \theta$$

and $\mathcal{C}$ lists tuples $\tau_i \subseteq \{p_1, \ldots, p_n\}$, for $i = 1, \ldots, m + m'$. Let $q_i$ be a fresh propositional variable for each Skolem function $f_i$. We define $\Sigma := \{\text{dep}(\tau_i, q_i) \mid i = 1, \ldots, m\}$ and

$$\psi := \theta \lor \bigvee_{i = m+1}^{m+m'} \text{dep}(\tau_i, q_i).$$

Clearly, $\Sigma$ and $\psi$ can be constructed from $(\phi, \mathcal{C})$ in polynomial time. It suffices to show that $\Sigma \models \psi$ iff $\phi$ is true.

Assume first that $\Sigma \models \psi$ and let $f_i : \{0, 1\}^{\mid \tau_i \mid} \rightarrow \{0, 1\}$ be arbitrary for $i = 1, \ldots, m$. Construct a team $X$ that consists of all assignments $s$ that map $p_1, \ldots, p_n, q_{m+1}, \ldots, q_{m+m'}$ into $\{0, 1\}$ and $q_1, \ldots, q_m$ respectively to $f_1(s(\tau_1)), \ldots, f_m(s(\tau_m))$. Since $X \models \Sigma$ we find $Z, Y_1, \ldots, Y_{m'} \subseteq X$ such that $Z \cup Y_1 \cup \ldots \cup Y_{m'} = X$, $Z \models \theta$, and $Y_i \models \text{dep}(\tau_{m+i}, q_{m+i})$ for
i = 1, \ldots, m'$. We may assume that each $Y_i$ is a maximal subset satisfying $\text{dep}(\overline{t}_{m+i}, q_{m+i})$, i.e., for all $s \in X \setminus Y_i$, $Y_i \cup \{s\} \neq \text{dep}(\overline{t}_{m+i}, q_{m+i})$. By downward closure (Proposition 11) we may assume that $Z$ does not intersect any of the subsets $Y_1, \ldots, Y_{m'}$. It follows that there are functions $f_i : \{0, 1\}^{\|s\|} \to \{0, 1\}$, for $i = m + 1, \ldots, m + m'$, such that

\[ Z = \{ s(f_{m+1}(s(\overline{t}_{m+1}))/q_{m+1}, \ldots, f_{m+m'}(s(\overline{t}_{m+m'}))/q_{m+m'}) \mid s \in X \}. \]

Notice that $Z$ is maximal with respect to $p_1, \ldots, p_n$, i.e., $Z \upharpoonright \{p_1, \ldots, p_n\} = \{p_1, \ldots, p_n\} \{0, 1\}$. Hence, by the flatness property (Proposition 10), and since $Z \models \theta$, it follows that $\theta$ holds for all values of $p_1, \ldots, p_n$ and for the values of $q_1, \ldots, q_{m+m'}$ chosen respectively according to $f_1, \ldots, f_{m+m'}$. Therefore, $\phi$ is true which shows the direction from left to right.

Assume then that $\phi$ is true, and let $X$ be a team satisfying $\Sigma$. Then there are functions $f_i : \{0, 1\}^{\|s\|} \to \{0, 1\}$ such that $f(s(\overline{t}_i)) = s(q_i)$ for $s \in X$ and $i = 1, \ldots, m$. Since $\phi$ is true we find functions $f_i : \{0, 1\}^{\|s\|} \to \{0, 1\}$, for $i = m + 1, \ldots, m + m'$, such that for all $s \in X$:

\[ s(f_{m+1}(s(\overline{t}_{m+1}))/q_{m+1}, \ldots, f_{m+m'}(s(\overline{t}_{m+m'}))/q_{m+m'}) \models \theta. \]  

Clearly, $Y_i : = \{ s \in X \mid s(q_i) \neq f(s(\overline{t}_i)) \}$ satisfies $\text{dep}(\overline{t}_i, q_i)$ for $i = m + 1, \ldots, m + m'$. Then it follows by (16) and flatness (Proposition 10) that $X \setminus (Y_{m+1} \cup \ldots \cup Y_{m+m'})$ satisfies $\theta$. Therefore, $\Sigma \models \psi$ which concludes the direction from right to left.

7 Entailment in Modal and Propositional Dependence Logics

We may now draw together the main results of Sections 4 and 6. There it was shown that in terms of the entailment problem $\text{co-NEXPTIME}^{\text{NP}}$ is both an upper bound for $\text{EMDL}$ and a lower bound for $\text{PDL}$. Therefore, we obtain in Theorem 17 that for all the logics inbetween it is also the exact complexity bound. Theorem 17 indicates that we can count $\text{QPDL}$ in this set of logics. The proof of this uses standard reduction methods and is located in Appendix.

\begin{theorem}
The satisfiability, validity, and entailment problems for $\text{QPDL}$ are polynomial-time reducible to the satisfiability, validity, and entailment problems for $\text{MDL}$, respectively.
\end{theorem}

\begin{proof}

The upper bound for $\text{EMDL}$ and $\text{MDL}$ was shown in Theorem 8, and by Theorem 17 the same upper bound applies to $\text{QPDL}$ and $\text{PDL}$. The lower bound for all of the logics comes from Theorem 16.

We also obtain that all the logics inbetween $\text{PDL}$ and $\text{EMDL}$ are $\text{NEXPTIME}$-complete in terms of their validity problem. The proof arises analogously from Corollary 9 and Theorem 15.

\begin{theorem}
The validity problem for $\text{EMDL}$, $\text{MDL}$, $\text{QPDL}$, and $\text{PDL}$ is $\text{NEXPTIME}^{\text{NP}}$-complete.
\end{theorem}

\begin{proof}

Recall that this close correspondence between propositional and modal dependence logics only holds with respect to their entailment and validity problems. Satisfiability of propositional dependence logic is only $\text{NP}$-complete whereas it is $\text{NEXPTIME}$-complete for its modal variant. It is also worth noting that the proof of Theorem 8 gives rise to an alternative proof for the $\text{NEXPTIME}$ upper bound for $\text{MDL}$ (and $\text{EMDL}$) satisfiability, originally proved in [25]. Moreover, the technique can be successfully applied to $\text{ML}(\emptyset)$. The following theorem entails that $\text{ML}(\emptyset)$ is no more complex than the ordinary modal logic.

\end{proof}
Table 1 Summary of results. The stated complexity classes refer to completeness results.

<table>
<thead>
<tr>
<th></th>
<th>satisfiability</th>
<th>validity</th>
<th>entailment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>PSPACE [18]</td>
<td>PSPACE [18]</td>
<td>PSPACE [18]</td>
</tr>
<tr>
<td>QPDL, MDL, EMDL</td>
<td>NEXPTIME [23, 25], [Thm. 17]</td>
<td>NEXPTIME [Thm. 19]</td>
<td>co-NEXPTIMENP [Thm. 18]</td>
</tr>
</tbody>
</table>

Theorem 20. The satisfiability, validity, and entailment problems for ML(⊔) are PSPACE-complete.

Proof. The lower bound follows from the Flatness property of ML (Proposition 2) and the PSPACE-hardness of satisfiability and validity problems for ML [18]. For the upper bound, it suffices to consider the entailment problem. The other cases are analogous. As in the proof of Lemma 5 (see also Theorem 5.2 in [29]) we reduce ML(⊔)-formulae to large disjunctions with the help of appropriate witness functions. For an ML(⊔)-formula ϑ, denote by Fϑ the set of all functions that map subformulae α ⊔ β of ϑ to either α or β. For each f ∈ Fϑ, we then denote by ϑf the formula obtained from ϑ by replacing each subformula of the form α ⊔ β with f(α ⊔ β). It is straightforward to show that ϑ is equivalent to ϑf ∈ Fϑ. Let now ϕ1, . . . , ϕn be a sequence of ML(⊔) formulae. Analogously to the proof of Lemma 5 we can show using Proposition 4 that {ϕ1, . . . , ϕn−1} |= ϕn iff for all f1 ∈ Fϕ1, . . . , fn−1 ∈ Fϕn−1 there is fn ∈ Fϕn such that {ϕ1f1, . . . , ϕn−1fn−1} |= ϕnf. Notice that the number of intuitionistic disjunctions appearing in ϕ1, . . . , ϕn is polynomial, and hence any single sequence of functions f1 ∈ Fϕ1, . . . , fn ∈ Fϕn can be stored using only a polynomial amount of space. It follows that the decision procedure presented in the proof of Theorem 8 can be now implemented in polynomial space. We immediately obtain the PSPACE upper bound for validity. For satisfiability, notice that ⋃ f∈Fϑ f f is satisfiable iff f f is satisfiable for some f ∈ Fϑ. Checking the right-hand side can be done as described above. This concludes the proof.

Combining the proofs of Theorem 8 and Theorem 20 we also notice that satisfiability, validity, and entailment can be decided in PSPACE for EMDL-formulae whose dependence atoms are of logarithmic length.

8 Conclusion

We have examined the validity and entailment problem for modal and propositional dependence logics (see Table 1). We showed that the entailment problem for (extended) modal and (quantified) propositional dependence logic is co-NEXPTIMENP-complete, and that the corresponding validity problems are NEXPTIME-complete. We also showed that modal logic extended with intuitionistic disjunction is PSPACE-complete with respect to its satisfiability, validity, and entailment problems, therefore being not more complex than the standard modal logic.

References

28:14 Validity and Entailment in Modal and Propositional Dependence Logics


A.1 From Quantified Propositional to Modal Logics

In this section we show how to generate simple polynomial-time reductions from quantified propositional dependence logics to modal dependence logics with respect to their entailment and validity problem. First we present Lemma 21 which is a direct consequence of [6, Lemma 14] that presents prenex normal form translations in the first-order dependence logic setting over structures with universe size at least 2. The result follows by the obvious first-order interpretation of quantified propositional formulae: satisfaction of a quantified propositional formula \( \phi \) by a binary team \( X \) can be replaced with satisfaction of \( \phi^* \) by
\[ M := \langle \{0, 1\}, t^M := 1, f^M := 0 \rangle \text{ and } X, \text{ where } \phi^* \text{ is a formula obtained from } \phi \text{ by replacing atomic propositional formulae } p \text{ and } \neg p \text{ respectively with } p = t \text{ and } p = f. \]

**Lemma 21** ([6]). Any formula \( \phi \) in L, where \( L \in \{ \text{QPDL, QPLInc, QPLInd} \}, \) is logically equivalent to a polynomial size formula \( Q_1 \psi_1 \ldots Q_n \psi_n \in L \) where \( \psi \) is quantifier-free and \( Q_i \in \{ \exists, \forall \} \) for \( i = 1, \ldots, n. \)

Next we show how to describe in modal terms a quantifier block \( Q_1 \psi_1 \ldots Q_n \psi_n \). Using the standard method in modal logic we construct a formula \( \text{tree}(V, p, n) \) that enforces the complete binary assignment tree over \( p_1, \ldots, p_n \) for a team over \( V \) [18]. The formulation of \( \text{tree}(V, p, n) \) follows the presented in [8]. We define \( \text{store}_n(p) := (p \land \square^n p) \lor (\neg p \land \square^n \neg p), \) where \( \square^n \) is a shorthand for \( \square \ldots \square \), to impose the existing values for \( p \) to successors in the tree. We also define \( \text{branch}_n(p) := \Diamond p \land \Diamond \neg p \land \square \text{store}_n(p) \) to indicate that there are \( \geq 2 \) successor states which disagree on the variable \( p \) and that all successor states preserve their values up to branches of length \( n \). Then we let

\[
\text{tree}(V, p, n) := \bigwedge_{q \in V} n \text{ store}_n(q) \land \bigwedge_{i=0}^{n-1} \square^i \text{branch}_{n-(i+1)}(p_{i+1}).
\]

Notice that \( \text{tree}(V, p, n) \) is an ML-formula and hence has the flatness property by Proposition 2.

**Theorem 17.** The satisfiability, validity, and entailment problems for QPDL are polynomial-time reducible to the satisfiability, validity, and entailment problems for MDL, respectively.

**Proof.** Consider first the entailment problem, and assume that \( \Sigma \cup \{ \phi \} \) is a finite set of formulae in either QPDL, QPLInc, or QPLInd. By Lemma 21 each formula in \( \theta \in \Sigma \cup \{ \phi \} \) can be transformed in polynomial time to the form \( \theta_0 = Q_1 \psi_1 \ldots Q_n \psi_n \) where \( \psi \) is quantifier-free. Moreover, by locality principle (Proposition 12) we may assume that the variable sequences \( p_1, \ldots, p_n \) corresponding to these quantifier blocks are initial segments of a shared infinite list \( p_1, p_2, p_3, \ldots \) of variables. Assume \( m \) is the maximal length of the quantifier blocks that appear in any of the translations, and let \( V \) be the set of variables that appear free in some of them. W.l.o.g. we may assume that \( \{ p_1, \ldots, p_m \} \) and \( V \) are disjoint. We let \( \theta_1 \) be obtained from \( \theta_0 \) by replacing quantifiers \( \exists \) and \( \forall \) respectively with \( \Diamond \) and \( \square \). It follows that \( \Sigma \models \phi \iff \{ \theta_1 \mid \theta \in \Sigma \} \cup \{ \text{tree}(V, p, n) \} \models \phi_1. \)

For the validity problem, we observe that \( \models \phi \iff \models \text{tree}(V, p, n) \lor (\text{tree}(V, p, n) \land \phi_1) \). Furthermore, for the satisfiability problem we have that \( \phi \) is satisfiable iff \( \text{tree}(V, p, n) \land \phi_1 \) is. Since the reductions are clearly polynomial, this concludes the proof.

**B Space Analysis for Algorithm 1**

Following [18] we show that Algorithm 1 requires only \( O(n^2) \) space on an input of the form \( \text{Sat}(\{ \phi \}, 0, 0, 0) : \) it takes \( O(n) \) recursive steps, each taking space \( O(n) \).

**Size of each recursive step.** At each recursive step \( \text{Sat}(A, B, C, D) \) is stored onto the work tape by listing all subformulae in \( A \cup B \cup C \cup D \) in such a way that each subformula \( \psi \) has

---

2 Notice that the direction from left to right does not hold under the so-called strict team semantics where \( \exists \) and \( \Diamond \) range over individuals. These two logics are not downwards closed and the modal translation does not prevent the complete binary tree of having two distinct roots that agree on the variables in \( V \).
its major connective (or relation/proposition symbol for atomic formulae) replaced with a special marker which also points to the position of the subset where \( \psi \) is located. In addition we store at each disjunctive/conjunctive recursive step the subformula or binary number that points to the disjunct/conjunct under consideration. Each recursive step takes now space \( \mathcal{O}(n) \).

**Number of recursive steps.** Given a set of formulae \( \mathcal{A} \), we write \( |\mathcal{A}| \) for \( \Sigma_{\phi\in \mathcal{A}} |\phi| \) where \( |\phi| \) is the length of \( \phi \). We show by induction on \( n = |\mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \cup \mathcal{D}| \) that \( \text{Sat}(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}) \) has \( 2n + 1 \) levels of recursion. Assume that the claim holds for all natural numbers less than \( n \), and assume that \( \text{Sat}(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}) \) calls \( \text{Sat}(\mathcal{A}', \mathcal{B}', \mathcal{C}', \mathcal{D}') \). Then \( |\mathcal{A}' \cup \mathcal{B}' \cup \mathcal{C}' \cup \mathcal{D}'| < n \) except for the case where \( \mathcal{A} \cap \mathcal{B} \) is empty and \( \mathcal{C} \cap \mathcal{D} \) is not. In that case it takes at most one extra recursive step to reduce to a length \( < n \). Hence, by the induction assumption the claim follows. We conclude that the space requirement for Algorithm 1 on \( \text{Sat}(\{\phi\}, \emptyset, \emptyset, \emptyset) \) is \( \mathcal{O}(n^2) \).