

# The Complexity of Symbolic Model Checking for Topological Evidence Models

**MSc Thesis** (*Afstudeerscriptie*)

written by

**Paul Fink**

(born December 26th, 1998)

under the supervision of **Dr. Malvin Gattinger** and **Dr. Ronald de Haan**, and submitted to the  
Examinations Board in partial fulfillment of the requirements for the degree of

**MSc in Logic**

at the *Universiteit van Amsterdam*.

**Date of the public defense:** **Members of the Thesis Committee:**

*March 19, 2026*

Dr. Aybüke Özgün (chair)

Dr. Gregor Behnke

Dr. Balder ten Cate



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

## Abstract

We analyze the computational complexity of symbolic model checking for Topological Evidence Models, where the state space is represented succinctly by BDDs. For the full language and already the single-agent fragment without restrictions on the models we establish PSPACE-completeness. We then identify fragments for which the complexity drops to  $\Theta_2^P$ : in the single-agent setting, both restricting the language to the  $\mathbb{M}$ -modality and bounding the evidence size to a constant lead to  $\Theta_2^P$ -complete symbolic model checking. Moreover, we obtain  $\Theta_2^P$ -membership for the belief-only fragment by reusing the proof scheme developed for the bounded-evidence case, indicating a reusable method for proving  $\Theta_2^P$  upper bounds in related symbolic model-checking problems. Finally, we establish PSPACE-completeness for symbolic model checking the knowledge fragment with unbounded evidence.

# Contents

|          |                                                                                           |           |
|----------|-------------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                                       | <b>2</b>  |
| <b>2</b> | <b>Background</b>                                                                         | <b>4</b>  |
| 2.1      | Symbolic model checking and BDDs . . . . .                                                | 4         |
| 2.1.1    | BDDs and BDD-operations . . . . .                                                         | 5         |
| 2.2      | Topo-E-Models . . . . .                                                                   | 6         |
| 2.3      | Symbolic Topo-E-Models . . . . .                                                          | 8         |
| 2.4      | Symbolic Model checking for Topo-E-Models . . . . .                                       | 9         |
| 2.4.1    | BDD-translation . . . . .                                                                 | 10        |
| <b>3</b> | <b>Complexity-results</b>                                                                 | <b>12</b> |
| 3.1      | Overview of complexity landscape . . . . .                                                | 12        |
| 3.2      | Complexity of model checking the evidence languages . . . . .                             | 13        |
| 3.2.1    | Notation and Conventions . . . . .                                                        | 13        |
| 3.2.2    | PSPACE-completeness for the full multi-agent language . . . . .                           | 14        |
| 3.2.3    | PSPACE-completeness for the one-agent fragment . . . . .                                  | 16        |
| 3.2.4    | $\Theta_2^P$ -completeness for single-agent $\mathcal{L}_{\mathbb{M}}$ fragment . . . . . | 22        |
| 3.2.5    | $\Theta_2^P$ -completeness for single-agent bounded evidence . . . . .                    | 25        |
| 3.3      | Knowledge and Belief as primitive . . . . .                                               | 36        |
| 3.3.1    | $\Theta_2^P$ -completeness for one-agent $\mathcal{L}_B$ . . . . .                        | 36        |
| 3.3.2    | PSPACE-completeness for one-agent $\mathcal{L}_K$ . . . . .                               | 40        |
| <b>4</b> | <b>Conclusion</b>                                                                         | <b>42</b> |
| 4.1      | Conclusion . . . . .                                                                      | 42        |
| 4.1.1    | Future work . . . . .                                                                     | 42        |
|          | <b>Bibliography</b>                                                                       | <b>44</b> |

# Chapter 1

## Introduction

Topological evidence models provide a mathematical framework for analysing evidence-based notions of knowledge and belief. The aim of this thesis is to study the computational complexity of deciding, for a given model  $M$ , a designated world  $w$ , and a formula  $\varphi$ , whether  $M, w \models \varphi$ . This decision problem is known as *model checking* and forms a bridge from philosophically motivated models of evidence-based knowledge and belief to applications in AI, where agents must reason about what they know or believe based on partial information.

A major obstacle for applying topological evidence models is that their explicit state spaces grow exponentially in the number of atomic propositions. Even models with few propositions can therefore become too large to represent by explicitly listing all worlds and the associated evidence/observation structure. Instead, we can represent the set of worlds and the structure on top *symbolically* by Boolean functions which can describe exponentially many worlds using only polynomial space.

This succinctness makes it feasible to represent interesting models, but it also changes the computational landscape: since complexity is measured in the size of the input representation, symbolic encodings can increase the worst-case complexity of model checking. In particular, while model checking on explicit finite models for standard epistemic languages is tractable, the corresponding symbolic model checking problem for the full language studied in this thesis is PSPACE-complete.

This thesis asks where the high complexity of symbolic model checking comes from, and how it depends on restrictions of the models and fragments of the language. Two complexity classes will repeatedly appear:  $\Theta_2^P$  and PSPACE. Our results draw a sharp distinction between two sources of hardness:  $\Theta_2^P$ -hardness arises from a succinct representation of the *set of worlds* (i.e. the Boolean constraint describing which valuations are admissible), whereas PSPACE-hardness arises from a succinct representation of the *epistemic structure* on those worlds (the evidence/observation components that determine the modal transitions). The rise from  $\Theta_2^P$  to PSPACE and its relation to the size of the epistemic structure will become clear in Sec. 3.2.5. While this at first might seem like some edge case it sheds light on the sources of complexity of symbolic model checking on topological evidence models.

Topological evidence models originate in epistemic logic. Classical epistemic logic is usually given Kripke semantics in the style of Hintikka, where knowledge and belief are interpreted via accessibility relations. In contrast, topological McKinsey-Tarski semantics interpret modalities via topological operations. In the evidence-based reading adopted here, open sets represent pieces of evidence: a set of worlds consistent with what an agent the piece of evidence. Topological closure conditions then naturally capture how evidence can be refined or combined.

The epistemic interpretation of topological semantics has started with Moss and Parikh [MP92] and was further developed by van Benthem and Pacuit [BP11]. The specific class of topological evidence models studied in this thesis was introduced by Özgün [Özg17] and recently extended by Gomes [San25] with a notion of shared (group) knowledge. A key step towards this thesis is the move from explicit to symbolic representations which was introduced to the topological setting by [San25]. Symbolic model checking was developed in the temporal logic community and represents state spaces and transition structure by Boolean formulas rather than explicit graphs. In epistemic logic, symbolic techniques were brought to dynamic epistemic logic by Gattinger et al. via SMCDEL [Gat18]. Gomes follows a very similar approach. She extends the work by Gattinger by adding a symbolic interpretation for the evidence modality.

Complexity-theoretic aspects of symbolic model checking in dynamic epistemic logic have been studied by Su [Su+09] and more recently by Behnke et al. [Beh+25]. We build on this line of work to analyse symbolic model checking for topological evidence models. We consider two families of languages: (i) an evidence-based modal language from which knowledge and belief can be defined, and (ii) languages that take knowledge and belief as primitive operators. For the full languages, symbolic model checking is PSPACE-complete: membership is shown via a polynomial-time alternating TM, and PSPACE-hardness follows directly from a reduction by Su [Su+09, Prop 30].

We then refine the picture by analysing fragments of the full language and restrictions of the models in order to isolate the origin of hardness. First, we show that merely restricting the number of agents does not suffice to reduce complexity. Next, we study two settings in which the epistemic *structure* is limited: (a) fragments that only use the observation modality, and (b) models in which the size of the evidence vocabulary is bounded by a constant. In both cases, model checking becomes  $\Theta_2^P$ -complete. A similar  $\Theta_2^P$ -completeness phenomenon appears for the belief-only fragment, whereas adding knowledge operators brings us back to PSPACE-completeness.

The outline of this thesis is as follows: we introduce the models, the language and the model checking problem in Chapter 2. Chapter 3 first proves the complexity results for the evidence language and then moves on to the knowledge and belief fragment. Table 3.1 summarizes the results established in this thesis.

# Chapter 2

## Background

This thesis is largely self-contained and centers on complexity results, but it relies on a small number of technical notions and definitions. In this chapter we introduce (symbolic) model checking, the symbolic representation of Topo-E-models, and the basic BDD machinery used both in the implementation and in several complexity arguments. We first explain the symbolic model checking perspective and why symbolic representations can speed up evaluation in practice. We then define Topo-E-models and their symbolic counterparts (STEMs), introduce the model checking problems studied in the thesis, and provide small examples that will be useful as intuition for the proofs in Chapter 3.

### 2.1 Symbolic model checking and BDDs

Model checking on a possible-worlds model asks whether a formula  $\varphi$  is true at a designated world  $w$  of a given model  $\mathfrak{M}$ , i.e. whether  $\mathfrak{M}, w \models \varphi$ . This differs from satisfiability (SAT), which asks whether there *exists* a model of a given class that satisfies  $\varphi$ . SAT has received more attention than model checking in part because, on *explicitly represented* finite models  $M = (W, R, V)$ , model checking for basic modal and epistemic languages is in P: one evaluates  $\varphi$  bottom-up by structural recursion, and modal clauses can be checked by iterating over the explicitly listed successors or observation cells. A standard dynamic programming algorithm computes, for each subformula  $\psi$  of  $\varphi$  and each world  $w$ , whether  $M, w \models \psi$  in polynomial time: we calculate the truth-sets for each non-modal sub-formulas and each world and save the results in a table. Then we calculate truth-sets for the modal formulas via set operations on the truth-sets. As the worlds and the relation are listed explicitly we can produce and fill these tables in polynomial-time. The problem becomes more interesting when models are represented *symbolically*, because the symbolic input can describe exponentially many worlds and structure over them succinctly.

BDD-based symbolic model checking was introduced in [Bur+92] and mainly applied in verification and temporal logic. In epistemic logic, symbolic techniques were introduced for dynamic epistemic logic by Gattinger et al. via SMCDEL [Gat18], and were adapted to Topo-E-models by Gomes [San25]. In a symbolic model one does not enumerate worlds but fixes a set of propositional variables and a *state law*  $\theta$  describing which assignments count as worlds (are legal). Also the modal structure is represented implicitly – in our case via equality on observables and inclusion on evidence variables. The central challenge is therefore to evaluate modal formulas without explicitly generating the exponentially large underlying state space, and BDD-based symbolic operations provide one practical way to achieve this.

### 2.1.1 BDDs and BDD-operations

We have already noted that we represent the state law as a BDD. Throughout, by BDD we mean *reduced, ordered* binary decision diagrams. For a fixed variable ordering, reduced ordered BDDs give a canonical representation of Boolean functions and allow efficient symbolic operations. All the results and definitions we need for this study can be found in Bryant's survey [Bry92]. After defining BDDs, we introduce complexity results for the operations we need in this thesis (conjunction, disjunction, and restriction/cofactoring, satisfiability).

**Definition 1** (Ordered and reduced BDDs [Bry92, Sec. 2]). *Let  $X = \{x_1, \dots, x_m\}$  be Boolean variables and let  $\prec$  be a total order on  $X$ . An ordered binary decision diagram (OBDD) is a rooted directed acyclic graph with:*

1. *exactly two terminal nodes labeled  $\mathbf{0}$  and  $\mathbf{1}$ ;*
2. *each non-terminal node  $v$  labeled by a variable  $\text{var}(v) \in X$  and having exactly two outgoing edges, a low-edge to  $\text{low}(v)$  (taken when  $\text{var}(v) = 0$ ) and a high-edge to  $\text{high}(v)$  (taken when  $\text{var}(v) = 1$ );*
3. *along every path from the root to a terminal, variables appear in strictly increasing order w.r.t.  $\prec$ .*

*An OBDD is reduced if it satisfies the two standard reduction rules: (i) no node  $v$  has  $\text{low}(v) = \text{high}(v)$  (eliminate redundant tests), and (ii) no two distinct nodes have the same variable label and the same low- and high-successors (merge isomorphic subgraphs).*

We can use the BDD to evaluate an assignment  $\alpha$  by traversing the diagram from the root: at a node  $v$  follow  $\text{high}(v)$  if  $\alpha(\text{var}(v)) = 1$  and otherwise follow  $\text{low}(v)$ , until reaching  $\mathbf{0}$  or  $\mathbf{1}$ . This is a polynomial time operation as it would be for a standard propositional formula. We now present some results that are essential to prove complexity bounds when building BDDs.

**Fact 2** (BDD operations and complexity bounds [Bry92, Sec. 3]). *Let  $F$  and  $G$  be ROBDDs over the same variable order  $\prec$ , representing Boolean functions  $f$  and  $g$ .*

1. **Conjunction / disjunction (Apply).** *Bryant's Apply algorithm constructs an ROBDD for  $f \text{ op } g$  for any binary Boolean operator  $\text{op}$ ; conjunction and disjunction are instances. With memoization, the number of distinct recursive calls is bounded by the number of argument-node pairs, giving a worst-case time bound  $O(|F| \cdot |G|)$ .*
2. **Cofactoring / restriction (Restrict).** *Bryant's Restrict operation computes cofactors  $f[x := b]$  (and more generally restriction by a partial assignment) by simplifying the diagram and then reducing it. Using memoization, restriction runs in time  $O(|F|)$  in the size of the input BDD.*

While conjoining two BDDs is polynomial in the sizes of the input BDDs (Fact 2), compiling an arbitrary Boolean formula into an ROBDD can cause exponential blow-up, and the outcome depends on the chosen variable order.

For our reductions we use well-structured formulas and are able to freely choose a variable order. To prove that a CNF can be constructed in polynomial-time in the number of variables, one can prove that the Gaifman graph has bounded tree-width. Tree decomposition will also become important in the  $\Theta_2^P$ -membership proof model checking given bounded evidence.

**Definition 3** (Tree decomposition and treewidth [Bod+22]). Let  $G = (V, E)$  be a graph, let  $T = (I, F)$  be a tree, and let  $X_i \subseteq V$  be a set of vertices (called a bag) associated to  $i$  for every node  $i \in I$ .

The pair  $(T, (X_i)_{i \in I})$  is a tree decomposition of  $G$  if it satisfies the following conditions:

1.  $\bigcup_{i \in I} X_i = V$ ;
2. for all  $e = (v, w) \in E$ , there is an  $i \in I$  with  $v, w \in X_i$ ;
3. for all  $v \in V$ , the set of nodes

$$I_v = \{i \in I \mid v \in X_i\}$$

is connected (that is, it induces a subtree of  $T$ ).

The width of the tree decomposition is

$$\max_{i \in I} |X_i| - 1.$$

The treewidth of  $G$  is the minimum width of a tree decomposition of  $G$ .

**Definition 4** (Gaifman graph of a CNF [FPV05, Def. 2.2]). Let  $C$  be a CNF formula over variables  $\text{Var}(C)$ . The Gaifman graph (or primal graph) of  $C$  is the graph  $G_C = (\text{Var}(C), E_C)$  where  $\{x, y\} \in E_C$  iff  $x$  and  $y$  occur together in some clause of  $C$ . We define the treewidth of  $C$  as  $\text{tw}(C) := \text{tw}(G_C)$ .

**Theorem 5** (OBDD size bound via treewidth [FPV05, Cor. 2.1]). Let  $C$  be a CNF formula with  $n := |\text{Var}(C)|$  variables and treewidth  $\text{tw}(C) = q$ . Then  $C$  has an OBDD of size polynomial in  $n$  and exponential in  $q$ .

Although the theorem is stated existentially, the proof provides a construction. It gives an algorithm that, given a width- $q$  tree decomposition, builds an OBDD in polynomial time. In the reduction to prove PSPACE-completeness for the one-agent fragment without restrictions, we will provide a tree decomposition and then apply theorem.

Last but not least, an important property of BDDs is that once we have built them, we can check unsatisfiability and satisfiability in polynomial time:

**Fact 6** (BDD satisfiability / emptiness test [Bry92]). An ROBDD is satisfiable iff there exists a path from the root to the **1** terminal; it is unsatisfiable iff it represents the constant **0** function. Thus satisfiability/emptiness can be checked in time polynomial (indeed linear) in the BDD size.

## 2.2 Topo-E-Models

In the following, we give a short overview of Topo-E-Models as defined by Özgün [Özg17] and further developed by Gomes [San25]. For a complete treatment including proof systems and completeness proofs see [BGG25]. We will only introduce the notions needed for the complexity analysis of symbolic model-checking.

Topo-E-Models provide a way to model knowledge and belief based on soft and hard evidence. The models are given as a state space, a valuation, a partition for the hard evidence and a topology representing the soft evidence. Hard evidence (also observables) can be seen as the observations an

agent has made and is certain about. This kind of evidence is considered infallible. Soft evidence on the other hand is fallible.

The most intuitive reading for belief then is truth in all maximally consistent pieces of evidence. That is, an agent rationally believes a proposition in a world if all maximally refined pieces of evidence that are inside the current observation cell are consistent with the proposition. An agent knows a proposition if there also is a piece of evidence that supports the proposition. We can formally characterize these definitions with topological notions. Formally, the models are defined as

**Definition 7** (Topo-E-Model). *A (multi-agent) Topo-E-Model for a finite set of agents  $I$  is a tuple*

$$\mathfrak{M} = (X, \{\Pi_i\}_{i \in I}, \{\tau_i\}_{i \in I}, \nu), \quad \text{where}$$

|                                                |                                                                                                         |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| $X$                                            | <i>is a non-empty set of worlds (states)</i>                                                            |
| $\Pi_i \subseteq \mathcal{P}(X)$               | <i>is a partition of <math>X</math> (hard evidence / observation cells for agent <math>i</math>)</i>    |
| $\tau_i \subseteq \mathcal{P}(X)$              | <i>is a topology on <math>X</math> (soft evidence for agent <math>i</math>)</i>                         |
|                                                | <i>with the constraint <math>\Pi_i \subseteq \tau_i</math> (every hard cell is <math>i</math>-open)</i> |
| $\nu : \text{Prop} \rightarrow \mathcal{P}(X)$ | <i>is a valuation for propositional atoms.</i>                                                          |

For  $w \in X$ , let  $\Pi_i(w)$  denote the unique cell of  $\Pi_i$  containing  $w$ . We write  $\text{Int}_i(\cdot)$  and  $\text{Cl}_i(\cdot)$  for interior and closure w.r.t.  $\tau_i$ .

Further, we define a modal language  $\mathcal{L}_{\mathbb{M}\Box}$  as

**Definition 8** (Full language  $\mathcal{L}_{\mathbb{M}\Box}$ ). *The multi-agent full language for Topo-E-Models is defined by*

$$\varphi ::= \top \mid \perp \mid p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid \Box_i \varphi \mid [\forall]_i \varphi$$

we use  $\langle \exists \rangle \equiv \neg \mathbb{M} \neg$  and  $\Diamond \equiv \neg \Box \neg$  as abbreviations.

The original language also contains a share operator. We omit it in this thesis. Gomes [San25] gives reduction axioms that eliminate the share operator in polynomial time, so it does not affect the complexity bounds considered here.

**Definition 9** (Topo-E-semantics). *For a Topo-E-Model  $\mathfrak{M}$  and an  $\mathcal{L}_{\mathbb{M}\Box}$ -formula  $\varphi$  we define the forcing relation  $\models$  inductively by*

|                                                 |                                                                                                                                                                                            |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mathfrak{M}, w \models \top$                  | $\Leftrightarrow$ always                                                                                                                                                                   |
| $\mathfrak{M}, w \models \perp$                 | $\Leftrightarrow$ never                                                                                                                                                                    |
| $\mathfrak{M}, w \models p$                     | $\Leftrightarrow w \in \nu(p) \quad (p \in \text{Prop})$                                                                                                                                   |
| $\mathfrak{M}, w \models \neg\varphi$           | $\Leftrightarrow \mathfrak{M}, w \not\models \varphi$                                                                                                                                      |
| $\mathfrak{M}, w \models (\varphi \wedge \psi)$ | $\Leftrightarrow \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi$                                                                                                |
| $\mathfrak{M}, w \models [\forall]_i \varphi$   | $\Leftrightarrow \forall t \in \Pi_i(w) \ \mathfrak{M}, t \models \varphi$                                                                                                                 |
| $\mathfrak{M}, w \models \Box_i \varphi$        | $\Leftrightarrow w \in \text{Int}_i(\llbracket \varphi \rrbracket^{\mathfrak{M}}) \Leftrightarrow \exists U \in \tau_i \ (w \in U \subseteq \llbracket \varphi \rrbracket^{\mathfrak{M}})$ |

Here  $\llbracket \varphi \rrbracket^{\mathfrak{M}} := \{x \in X \mid \mathfrak{M}, x \models \varphi\}$  is the truth set of  $\varphi$  in  $\mathfrak{M}$ .

We now define belief and knowledge modalities as abbreviations:

- **Belief** ( $B_i\varphi := [\forall]_i \Diamond_i \Box_i \varphi$ ): there is soft evidence for  $\varphi$  that is consistent with all other soft evidence in the current observation cell. Formally,  $\text{Int}_i(\llbracket \varphi \rrbracket)$  is locally dense at  $w$ :

$$\Pi_i(w) \subseteq \text{Cl}_i(\text{Int}_i(\llbracket \varphi \rrbracket)).$$

- **Knowledge** ( $K_i\varphi := B_i\varphi \wedge \Box_i\varphi$ ):  $\varphi$  is believed and also has factive soft evidence at the actual world, that is  $w \in \text{Int}_i(\llbracket \varphi \rrbracket)$ .

We further define the language  $\mathcal{L}_{KB}$  where knowledge and belief are primitives but are still interpreted with the same semantics as their current abbreviations above.

## 2.3 Symbolic Topo-E-Models

With symbolic Topo-E-Models (STEM) we avoid explicit listing of worlds and relations or topologies. Here, the model is given as a set of propositional variables together with a state law that restricts which worlds exist and propositional variables for observables and evidence. Formally,

**Definition 10** (Symbolic Topo-E-Model). *A symbolic Topo-E-Model for a finite set of agents  $I$  is a tuple*

$$\mathfrak{M} = (\mathcal{V}, \theta, \{O_i\}_{i \in I}, \{E_i\}_{i \in I}), \quad \text{where}$$

$$\begin{array}{ll} V = \{x_1, \dots, x_n\} & \text{is a set of propositional variables} \\ E_i \subseteq V & \text{is the evidence set for agent } i \\ O_i \subseteq V & \text{is the observable set for agent } i \\ \theta & \text{is the state-law given as BDD.} \end{array}$$

A world  $w \subseteq V$  is a set of propositional variables consistent with  $\theta$ , that is  $\theta(w) = 1$ . We are still working with the language  $\mathcal{L}_{\Box}$ .

**Definition 11** (Symbolic semantics). *For a symbolic Topo-E-Model  $\mathfrak{M}$  and a  $\mathcal{L}_{\Box}$ -formula  $\varphi$  we define the forcing relation  $\models$  inductively by*

$$\begin{array}{ll} \mathfrak{M}, w \models p & \Leftrightarrow p \in w \quad (p \in V) \\ \mathfrak{M}, w \models \neg\varphi & \Leftrightarrow \mathfrak{M}, w \not\models \varphi \\ \mathfrak{M}, w \models (\varphi \wedge \psi) & \Leftrightarrow \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi \\ \mathfrak{M}, w \models [\forall]_i \varphi & \Leftrightarrow \forall t \text{ s.t. } O_i \cap w = O_i \cap t \text{ and } t \models \theta, \text{ holds } \mathfrak{M}, t \models \varphi \\ \mathfrak{M}, w \models \Box_i \varphi & \Leftrightarrow \forall t \text{ s.t. } w \cap E_i \subseteq t \cap E_i \wedge O_i \cap w = O_i \cap t \text{ and } t \models \theta, \text{ holds } \mathfrak{M}, t \models \varphi \end{array}$$

$\Diamond_i$  and  $\langle \exists \rangle_i$  are defined by duality.

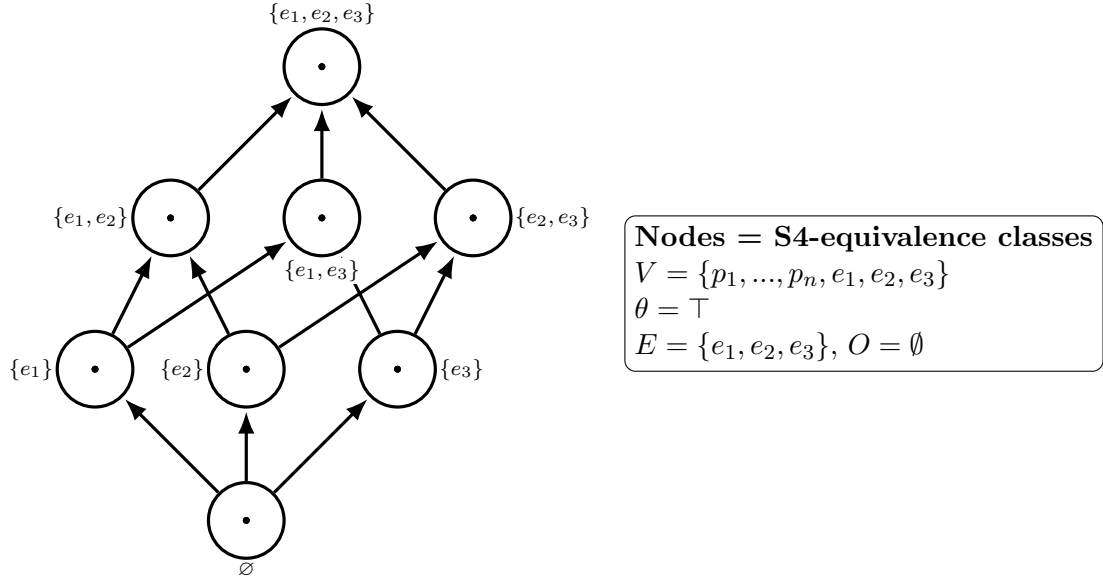


Figure 2.1: Evidence lattice with three evidence atoms and  $\top$  as state law. The nodes represent all assignments that are consistent with the state law and force the evidence atoms in its tag.

Gomes [San25, Thm. 6.11] proves that there is a truth-preserving translation between explicit and symbolic Topo-E-Models. The translation from symbolic to explicit models is inherently exponential as it enumerates all worlds consistent with the state law. Hence, it does not help for our complexity investigations.

To get a more intuitive picture note that the semantics for  $[\forall]_i$  corresponds to quantifying over an equivalence relation:

$$w \approx_i t \text{ iff } w \cap O_i = t \cap O_i$$

Now the semantics for  $\Box_i$  correspond to quantifying over a preorder inside the partition cells defined by the  $\approx_i$  equivalence relation.

$$w \preceq_i t \text{ iff } w \cap E_i \subseteq t \cap E_i \text{ and } w \approx_i t.$$

These relations are implicitly defined but never spelled out as explicitly listing them would amount to spelling out the worlds which in turn leads to explicit models. Still, it is an important tool for visualization/intuition as well as for proofs as many things are easier to prove when arguing over the underlying relations.

It is important to note that  $\approx_i$  partitions the state space into observation cells, while  $\preceq_i$  defines a preorder on the worlds within a cell by evidence extension. Two worlds are  $\preceq_i$ -equivalent exactly when they agree on the evidence variables  $E_i$ . For the trivial state law  $\top$  this gives us the lattice  $(2^{E_i}, \subseteq)$  inside each observation cell, as depicted in Figure 2.1. By restrictions through the state law, we can either restrict which equivalence clusters exists and/or which worlds are in each equivalence cluster. The preorder structure defined through  $\subseteq$  persists.

## 2.4 Symbolic Model checking for Topo-E-Models

We now give the formal definition of the model checking problems we are going to consider in this work.

**Definition 12** (Model checking on STEMs). *We define the model checking problems on symbolic Topo-E-Models as*

$$STEMC_{\mathcal{L}}^C(\mathfrak{M}, w, \varphi) = 1 \quad \Longleftrightarrow \quad \mathfrak{M}, w \models \varphi$$

where  $C$  is a tuple of constraints on the model and  $\mathcal{L}$  is a language.  $C$  contains the constraints on the number of agents, the maximal size of the evidence sets, and the maximal size of the observable sets. The language is indexed by the modalities that appear in the modal language.

As an example,  $STEMC_{\mathcal{L}_{\mathbb{M}\square}}^{n=1, |E| \leq k}$  is the problem of deciding, given a one-agent STEM  $\mathfrak{M} = (V, \theta, O, E)$  with  $|E| \leq k$ , a world  $w \subseteq V$  with  $w \models \theta$ , and a formula  $\varphi \in \mathcal{L}_{\mathbb{M}\square}$ , whether  $\mathfrak{M}, w \models \varphi$ .

Note, whenever we restrict the class of models the complexity stays equal or goes down. The same holds for restricting the language to a fragment.

### 2.4.1 BDD-translation

Gomes [San25, Def. 6.6] defines translations of the symbolic semantics into Boolean formulas necessary for the BDD implementation. Note that these translations are potentially exponential.

**Definition 13** (Local Boolean Translation). *Given a symbolic Topo-E-Model  $\mathfrak{M}$ , we define the local boolean translation*

$$\|\cdot\|_{\mathfrak{M}} : \mathcal{L}_{\mathbb{M}\square} \rightarrow \mathcal{L}_{Bool}(V)$$

on  $\mathfrak{M}$  as follows:

$$\begin{aligned} \|\top\|_{\mathfrak{M}} &:= \top \\ \|p\|_{\mathfrak{M}} &:= p \\ \|\neg\varphi\|_{\mathfrak{M}} &:= \neg \|\varphi\|_{\mathfrak{M}} \\ \|\varphi \wedge \psi\|_{\mathfrak{M}} &:= \|\varphi\|_{\mathfrak{M}} \wedge \|\psi\|_{\mathfrak{M}} \\ \|\Box_I \varphi\|_{\mathfrak{M}} &:= \forall E'_I \left( \left( \bigwedge_{e_i \in E_I} (e'_i \leftrightarrow e_i) \right) \rightarrow \forall (V \setminus O_I) \left( \left( \bigwedge_{e_i \in E_I} (e'_i \rightarrow e_i) \right) \wedge \theta \rightarrow \|\varphi\|_{\mathfrak{M}} \right) \right) \\ \|[\forall]_I \varphi\|_{\mathfrak{M}} &:= \forall (V \setminus O_I) (\theta \rightarrow \|\varphi\|_{\mathfrak{M}}), \end{aligned}$$

where  $p \in V$  is any propositional variable.

**Theorem 14** (Correctness of the Boolean translation [San25, Thm. 6.7]). *For every  $\varphi \in \mathcal{L}_{\mathbb{M}\square}$  and every pointed model  $(\mathfrak{M}, s)$ ,*

$$(\mathfrak{M}, s) \models \varphi \quad \text{iff} \quad s \models \|\varphi\|_{\mathfrak{M}}.$$

All cases but the one for  $\Box_I$  correspond to the work by Gattinger in [Gat18] for symbolic model checking on S5-models. These results justify the BDD-algorithm used in the implementation.

There are two things to note about this translation. Firstly, if we were to expand the quantifiers, the translation would be exponential and would correspond to explicitly enumerating all worlds. For example, the quantifier  $\forall E_i$  together with the conjunction over  $e_i \in E_i$  yields a conjunction over  $2^{|E_i|}$  assignments.

Secondly, this expansion is not necessary for producing the BDD, since we apply universal abstraction rather than explicit enumeration. Still, the BDD can blow up, even in the fragment without  $\Box_I$ . There

are several possible sources for this kind of exponential blow-up, and in some cases it is unavoidable. A simple example of a Boolean function that necessarily leads to exponential blow-up is the hidden weighted bit function [Bol+99]. Accordingly, the symbolic model checking approach has worst-case exponential time and space behaviour, since in the worst case the BDD construction may produce an exponential-size BDD, and hence the algorithm is not restricted to polynomial space.

This does not contradict our PSPACE-completeness result: PSPACE-completeness guarantees the existence of a polynomial-space decision procedure, but it does not imply that every practically useful symbolic procedure must stay within polynomial space on all instances. However, in many cases the translation is tame and can lead to much faster evaluation in practice.

## Chapter 3

# Complexity-results

### 3.1 Overview of complexity landscape

In this chapter we map out the complexity landscape for symbolic model checking on topological evidence models. We will look at the languages  $\mathcal{L}_{\mathbb{M}\square}$ ,  $\mathcal{L}_{K,B}$  and fragments of both. The results we are going to prove are summarized in table 3.1.

Table 3.1: Complexity of symbolic model checking for Topo-E-Models

| Problem                                                     | Setting     | Comments                                                      | Complexity             |                  |
|-------------------------------------------------------------|-------------|---------------------------------------------------------------|------------------------|------------------|
| $STEMC_{\mathbb{M}}^{n \geq 2}$                             | Multi-agent | $\mathcal{L}_{\mathbb{M}}$ and all extensions                 | PSPACE-complete        | [Su+09, Prop 30] |
| $STEMC_{\mathcal{L}_{\square}}^{n=1}$                       | One-agent   | $\mathcal{L}_{\square}$ and $\mathcal{L}_{\mathbb{M}\square}$ | PSPACE-complete        | Sec. 3.2.3       |
| $STEMC_{\mathcal{L}_{\mathbb{M}}}^{n=1}$                    | One-agent   | $\mathcal{L}_{\mathbb{M}}$ (no box)                           | $\Theta_2^P$ -complete | Sec. 3.2.4       |
| $STEMC_{\mathcal{L}_{\mathbb{M}\square}}^{n=1,  E  \leq k}$ | One-agent   | Bounded evidence                                              | $\Theta_2^P$ -complete | Sec. 3.2.5       |
| $STEMC_{\mathcal{L}_B}^{n \geq 2}$                          | Multi-agent | B-only to full $\mathcal{L}_{KB}^{n \geq 2}$                  | PSPACE-complete        | Sec. 3.3         |
| $STEMC_{\mathcal{L}_K}^{n=1}$                               | One-agent   | $K$ -modality                                                 | PSPACE-complete        | Sec. 3.3.2       |
| $STEMC_{\mathcal{L}_B}^{n=1}$                               | One-agent   | Belief only                                                   | $\Theta_2^P$ -complete | Sec. 3.3.1       |

The first important result presented here is that symbolic model-checking the full language is PSPACE-complete. As the S5-models investigated by Su et al. [Su+09] are a fragment ( $\mathbb{M}$  only, unbounded observables, multi-agent), we can directly conclude PSPACE-hardness. We get membership by providing an alternating Turing machine that runs in polynomial time. Already the 2-agent case with unrestricted observables,  $STEMC_{\mathcal{L}_{\mathbb{M}}}^{n=2}$  is PSPACE-complete as [Su+09, Prop. 30] shows. Thus, we will be mainly concerned with the one-agent case.

The first technically interesting result is PSPACE-completeness for the one-agent case  $STEMC_{\mathcal{L}_{\mathbb{M}\square}}^{n=1}$ . We will present a reduction from TQBF that established PSPACE-hardness. The reduction works by encoding the strategy tree from the game view of TQBF in our symbolic models by means of evidence atoms, that is, we construct a symbolic binary tree in our model. PSPACE-membership simply follows by a restriction of the algorithm for the multi-agent language.

Then, we show that the symbolic model checking problem  $STEMC_{\mathcal{L}_{\mathbb{M}}}^{n=1}$  is  $\Theta_2^P$ -complete. This result is established by a reduction from the canonical  $\Theta_2^P$ -complete problem MAX-MODEL-PARITY. We establish  $\Theta_2^P$ -membership by an algorithm that first reduces modal-depth and evaluates all modal

formulas with a polynomial number of *non-adaptive* NP-oracle calls.

For symbolic model checking with restricted evidence  $STEMC_{\mathcal{L}_{\mathbb{M}\Box}}^{n=1, |E| \leq k}$ , we get  $\Theta_2^P$ -hardness from the hardness of  $STEMC_{\mathcal{L}_{\mathbb{M}}}^{n=1}$ . We then prove  $\Theta_2^P$ -membership by providing an algorithm that runs in polynomial-time with a tree-like NP-oracle query DAG. We bound the treewidth by an explicit tree decomposition and invoke [GR22, Thm 2] from Gharibian et al. to conclude  $\Theta_2^P$  membership.

Finally, we study variants where knowledge  $K$  and/or belief  $B$  are taken as primitives. We still keep the same semantics. All upper bounds transfer by reusing the model checking procedures for the evidence language. For knowledge, the one-agent PSPACE-hardness proof can be adapted from the unbounded-evidence TQBF reduction by adding maximal worlds so that the belief component of  $K$  becomes trivial and the reduction directly transfers. For belief, the picture changes: the special case  $B\varphi$  with propositional  $\varphi$  falls to CONP (under the assumption that the state law is given as a BDD). This enables a  $\Theta_2^P$  model-checking procedure for the belief-only language by an algorithm similar to the one for bounded evidence. Beyond complexity, this suggests that the model checker implemented by Gomes could be optimized by switching to the equivalent truth in all maximal worlds definition.

## 3.2 Complexity of model checking the evidence languages

### 3.2.1 Notation and Conventions

Throughout this chapter we work with symbolic Topo-E-models

$$\mathfrak{M} = (V, \theta, \{E_i\}_{i \in I}, \{O_i\}_{i \in I}),$$

where  $V$  is a finite set of propositional atoms,  $\theta$  is the *state law* (given as a BDD), and  $E_i, O_i \subseteq V$  are, respectively, the evidence and observable atoms of agent  $i$ . In the one-agent cases, we drop the indices. A *world* is a set  $w \subseteq V$ . We call  $w$  *legal* if  $\theta(w) = 1$ . Formulas are evaluated at pointed models  $(\mathfrak{M}, w, \varphi)$  with  $w$  legal.

**Projection formulas.** For any  $X \subseteq V$  and any  $S \subseteq X$ , we write

$$\pi_X(S) := \left( \bigwedge_{x \in S} x \right) \wedge \left( \bigwedge_{x \in X \setminus S} \neg x \right)$$

for the propositional formula that fixes exactly the valuation  $S$  on  $X$ . For a world  $w$  we abbreviate  $\pi_X(w) := \pi_X(w \cap X)$ . In particular, we use  $\pi_O(w)$  for observation-cell constraints and  $\pi_E(w)$  for evidence-block constraints.

**Fixed observation cell (one-agent).** In the one-agent case, the observation cell never changes along modal steps. Thus, for model checking at  $w$  we fix the observation cell by strengthening the state law to

$$\theta^w := \theta \wedge \pi_O(w),$$

and then work with  $\theta^w$  in all satisfiability tests below. We write  $\theta$  after this replacement to keep the notation simple.

**Size measure.** We measure the complexity in terms of the input variables  $|V|$ , the size of the BDD for  $\theta$ , and the formula size  $|\varphi|$ .

### 3.2.2 PSPACE-completeness for the full multi-agent language

We prove the upper bound for symbolic model checking in the full multi-agent language. Together with the hardness results proved in [Su+09], this yields PSPACE-completeness.

We prove membership via an alternating Turing machine. To avoid negative and positive occurrences of modalities (and hence a polarity parameter), we first put the input formula into negation normal form (NNF): negation occurs only in front of atoms. This can be done in polynomial time using the standard dualities

$$\neg\Box_i\psi \equiv \Diamond_i\neg\psi, \quad \neg\Diamond_i\psi \equiv \Box_i\neg\psi, \quad \neg[\forall]_i\psi \equiv \langle\exists\rangle_i\neg\psi, \quad \neg\langle\exists\rangle_i\psi \equiv [\forall]_i\neg\psi.$$

Thus, in what follows we assume  $\varphi$  is in NNF.

Our algorithm works with configurations  $(s, \psi)$  where  $s \subseteq V$  and  $\psi \in \mathcal{L}_{\forall\Box}$ . It then recurses on the formula. Boolean cases are handled in a standard way. The only interesting cases are how the ATM handles the modalities. The idea can be summarized as follows:

- For  $(s, \Diamond_i\psi)$ ,  $(s, \langle\exists\rangle_i\psi)$ : existentially guess  $t \subseteq V$ , check  $\theta(t)$  and the corresponding accessibility condition ( $s \preceq_i t$  or  $s \approx_i t$ ); if it holds, recurse on  $(t, \psi)$ , otherwise reject.
- For  $(s, \Box_i\psi)$ ,  $(s, [\forall]_i\psi)$ : universally branch over all  $t \subseteq V$ ; if  $\theta(t)$  and the accessibility condition hold, recurse on  $(t, \psi)$ , otherwise accept immediately.

We now give a fully spelled-out version of the algorithm and prove correctness:

---

**Algorithm 1:** ALTMC (full language, NNF)

---

**Input:**  $\mathfrak{M} = (V, \theta, \{E_i, O_i\}_{i \in I})$  with  $\theta$  as BDD, world  $w \subseteq V$  with  $w \models \theta$ , formula  $\varphi$

**Output:** *true* iff  $\mathcal{M}, w \models \varphi$

```
1  $\varphi \leftarrow \text{NNF}(\varphi)$ .
2 return MC( $w, \varphi$ ).

3 Function MC( $s, \psi$ ):
4   if  $\psi = \top$  then
5     return true
6   else if  $\psi = p$  then
7     return ( $p \in s$ )
8   else if  $\psi = \neg p$  then
9     return ( $p \notin s$ )
10  else if  $\psi = \psi_1 \wedge \psi_2$  then
11    universally branch to MC( $s, \psi_1$ ) and MC( $s, \psi_2$ );
12  else if  $\psi = \psi_1 \vee \psi_2$  then
13    existentially choose  $j \in \{1, 2\}$ ; return MC( $s, \psi_j$ )
14  else if  $\psi = \Diamond_i \chi$  then
15    existentially guess  $t \subseteq V$ ;
16    if BDD-EVAL( $\theta, t$ ) = 1  $\wedge (s \cap O_i = t \cap O_i) \wedge (s \cap E_i \subseteq t \cap E_i)$  then
17      return MC( $t, \chi$ )
18    else
19      return false
20  else if  $\psi = \langle \exists \rangle_i \chi$  then
21    existentially guess  $t \subseteq V$ ;
22    if BDD-EVAL( $\theta, t$ ) = 1  $\wedge (s \cap O_i = t \cap O_i)$  then
23      return MC( $t, \chi$ )
24    else
25      return false
26  else if  $\psi = \Box_i \chi$  then
27    universally branch over all  $t \subseteq V$ ;
28    if BDD-EVAL( $\theta, t$ ) = 1  $\wedge (s \cap O_i = t \cap O_i) \wedge (s \cap E_i \subseteq t \cap E_i)$  then
29      return MC( $t, \chi$ )
30    else
31      return true
32  else if  $\psi = [\forall]_i \chi$  then
33    universally branch over all  $t \subseteq V$ ;
34    if BDD-EVAL( $\theta, t$ ) = 1  $\wedge (s \cap O_i = t \cap O_i)$  then
35      return MC( $t, \chi$ )
36    else
37      return true
```

---

**Theorem 15** (PSPACE-membership). *Model checking the full language  $\mathcal{L}_{\Box, [\forall]}$  on symbolic Topo-E-Models is in PSPACE.*

*Proof.* We first show correctness of MC, then the PSPACE bound.

We prove by structural induction on  $\psi$  that for every state  $s$  and every NNF subformula  $\psi$  of  $\varphi$ ,

$$\text{MC}(\mathfrak{M}, s, \psi) \text{ accepts} \quad \text{iff} \quad \mathfrak{M}, s \models \psi.$$

*Atoms:* For  $\psi = p$  and  $\psi = \neg p$  this holds as the semantic and the acceptance conditions coincide.

*Boolean cases:* For  $\psi = \chi_1 \wedge \chi_2$ , MC universally branches to both children and accepts iff both children accept. By the induction hypothesis, this is equivalent to  $\mathfrak{M}, s \models \chi_1$  and  $\mathfrak{M}, s \models \chi_2$ , i.e. to  $\mathfrak{M}, s \models \chi_1 \wedge \chi_2$ . For  $\psi = \chi_1 \vee \chi_2$  the argument is analogous with existential branching.

*Modalities:* We treat  $[\forall]_i$ ; the cases  $\Box_i$ ,  $\Diamond_i$  and  $\langle \exists \rangle_i$  are analogous. By semantics,

$$\mathfrak{M}, s \models [\forall]_i \chi \iff \forall t \subseteq V \left[ (\theta(t) = 1 \wedge s \approx_i t) \rightarrow \mathfrak{M}, t \models \chi \right].$$

The ATM branches universally over all  $t \subseteq V$ . On each branch we have that if  $\theta(t) \neq 1$  or  $s \not\approx_i t$ , the branch accepts immediately; otherwise it continues with  $\text{MC}(\mathfrak{M}, t, \chi)$ . By the induction hypothesis, this accepts iff  $\mathfrak{M}, t \models \chi$ . Thus the universal acceptance condition of the TM coincides with the Topo-E-semantics.

For  $\langle \exists \rangle_i \chi$ , the same reasoning applies with existential guessing and the semantic clause  $\exists t (\theta(t) = 1 \wedge s \approx_i t \wedge \mathfrak{M}, t \models \chi)$ . For  $\Box_i$  and  $\Diamond_i$  replace  $s \approx_i t$  by  $s \preceq_i t$ .

A configuration of MC stores a state  $s \subseteq V$ , which needs  $\mathcal{O}(|V|)$  bits, and a pointer to a subformula of  $\varphi$ , which needs  $\mathcal{O}(|\varphi|)$  bits. As we recurse on the formulas, the recursion depth is  $\mathcal{O}(|\varphi|)$ . The checks  $\theta(t) = 1$ ,  $s \cap O_i = t \cap O_i$  and  $s \cap E_i \subseteq t \cap E_i$  can be performed in polynomial time in  $|V|$  and the size of the BDD for  $\theta$ . Thus the ATM runs in polynomial time. Therefore, the problem is in *APTIME*. Given the standard theorem *APTIME* = PSPACE, model checking is in PSPACE. [AB09, Thm 5.10]  $\square$

### 3.2.3 PSPACE-completeness for the one-agent fragment

In this section, we establish PSPACE-completeness for the one-agent fragment. PSPACE-membership we get by simply restricting the algorithm 1. Hence, it is only left to prove hardness. We do this by reducing from TQBF. The reduction works by encoding the strategy tree from the game semantics view on TQBF in symbolic Topo-E-Models. This is inspired by [Lad77] where PSPACE-completeness is proved for S4 satisfiability by doing a similar encoding of a tree in a Kripke model. In our reduction, we will use the state law, and the evidence atoms to symbolically represent a binary tree. We then construct a modal-formula that is true at the root of the tree if and only if the original QBF is true. This is done by matching the existential and universal quantifiers by  $\Diamond$ -modalities, respectively  $\Box$ -modalities, guarded to the next level. Each step encoded in the modalities then corresponds to picking a value for a variable.

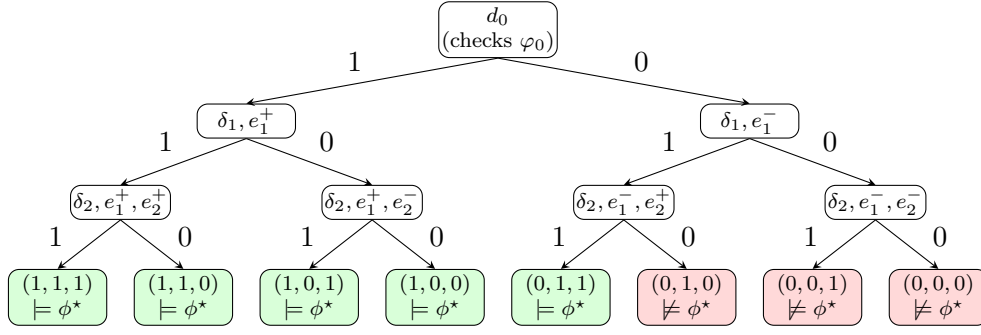


Figure 3.1: Kripke model for the TQBF instance  $\forall x_1 \forall x_2 \exists x_3 (x_1 \vee (x_2 \wedge x_3))$ . The prefix is encoded by the modal formula  $\varphi_2 = \Diamond(\delta_3 \wedge \phi^*)$ ,  $\varphi_1 = \Box(\delta_2 \rightarrow \varphi_2)$ ,  $\varphi_0 = \Box(\delta_1 \rightarrow \varphi_1)$ . Each world is labeled with the corresponding level formula  $\delta_i$  and the evidence literals  $e_j^\pm$  true there, encoding a partial assignment to  $x_1, x_2, x_3$ . Leaves are annotated with the corresponding total assignment  $(x_1, x_2, x_3)$  we get after substituting  $e^\pm$  for  $x_i, \neg x_i$ . The formulas  $\varphi_2, \varphi_1, \varphi_0$  mirror the quantifier prefix.

**Theorem 16** (PSPACE-hardness for one-agent fragment). *For every fully quantified Boolean formula*

$$\Phi = Q_1 x_1 \dots Q_n x_n \phi(x_1, \dots, x_n),$$

*we can construct in polynomial time a one-agent symbolic Topo-E-model  $\mathfrak{M}_\Phi$  and an  $\mathcal{L}_{\mathbb{M}\Box}$ -formula  $\varphi_\Phi$  such that*

$$\Phi \text{ is true} \quad \text{iff} \quad \mathfrak{M}_\Phi, s_0 \models \varphi_\Phi,$$

*where  $s_0$  is the root of the binary tree represented by  $\mathfrak{M}_\Phi$ . Thus, symbolic model checking in the one-agent case is PSPACE-hard.*

*Proof.* We reduce from TQBF. Let

$$\Phi = Q_1 x_1 \dots Q_n x_n \phi(x_1, \dots, x_n)$$

be a fully quantified Boolean formula. According to the game view on TQBF, the quantifiers induce a two-player game (Existential vs. Universal) in which the players choose values for the variables in order, and  $\Phi$  is true if and only if the existential player has a winning strategy in this game.

We work with a single agent and define:

$$V = \{d_0, d_1, \dots, d_n\} \cup \{e_i^+, e_i^- \mid 1 \leq i \leq n\}.$$

Intuitively,  $e_i^+$  encodes  $x_i = 1$ , and  $e_i^-$  encodes  $x_i = 0$ . The atoms  $d_i$  indicate that  $x_1, \dots, x_i$  have been decided.

We take  $E = V$  and  $O = \emptyset$  and define the state law  $\theta$  as:

$$\theta = d_0 \wedge \bigwedge_{i=0}^{n-1} (d_{i+1} \rightarrow d_i) \wedge \bigwedge_{i=1}^n (d_i \leftrightarrow (e_i^+ \vee e_i^-)) \wedge \bigwedge_{i=1}^n (\neg e_i^+ \vee \neg e_i^-).$$

A world is any subset  $w \subseteq V$  with  $\theta(w) = 1$ . For any such world  $w$  we have:

- $d_i$  holds iff exactly one of  $e_i^+, e_i^-$  holds,
- from  $d_{i+1} \rightarrow d_i$  we obtain the chain  $d_n \rightarrow d_{n-1} \rightarrow \dots \rightarrow d_0$ .

Thus,  $w$  “decides”  $x_i$  exactly when  $d_i$  holds. For an example of how the reduction works see Figure 3.1.

As we have set  $O = \emptyset$ , the induced preorder reduces to

$$w \preceq t \quad \text{iff} \quad w \cap E \subseteq t \cap E.$$

As we have set  $E = V$  this further reduces to

$$w \preceq t \quad \text{iff} \quad w \subseteq t.$$

Thus, each world (corresponding to an assignment) is related to all its extensions.

To guard modal operators to minimal extensions (our model only provides the transitive closure of the strategy tree) we define

$$\delta_i := \begin{cases} d_i \wedge \neg d_{i+1}, & \text{if } 0 \leq i < n, \\ d_n, & \text{if } i = n, \end{cases}$$

so that  $w \models \delta_i$  expresses that  $w$  decides *exactly* the variables  $x_1, \dots, x_i$ . More precisely, for legal world  $w$  we have:

- if  $w \models \delta_i$ , then for every  $k \leq i$  exactly one of  $e_k^+, e_k^-$  is in  $w$   
(as  $\bigwedge_{i=1}^n (d_i \leftrightarrow (e_i^+ \vee e_i^-)) \wedge \bigwedge_{i=1}^n (\neg e_i^+ \vee \neg e_i^-)$ )
- if  $w \models \delta_i$ , then for every  $k > i$  neither  $e_k^+$  nor  $e_k^-$  is in  $w$  ( $w \models \neg d_k$ ).

Note further, there is a root world  $\{d_0\}$  and if  $w \models \delta_{i-1}$  for some  $1 \leq i \leq n$ , then there are exactly two  $\preceq$ -minimal extensions  $t \supseteq w$  such that  $t \models \delta_i$ , namely:

- $t_w^{(0)}$  with  $e_i^- \in t_w^{(0)}$  (encoding  $x_i = 0$ ),
- $t_w^{(1)}$  with  $e_i^+ \in t_w^{(1)}$  (encoding  $x_i = 1$ ).

We will use these minimal extensions to mirror a player choosing the value for the next variable in the game view on TQBF.

Now, for each world  $w$ , we define a partial assignment  $\alpha_w$  on  $x_1, \dots, x_n$  as

$$\alpha_w(x_i) := \begin{cases} 1 & \text{if } e_i^+ \in w, \\ 0 & \text{if } e_i^- \in w, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

This is well-defined as we have  $\neg(e_i^+ \wedge e_i^-)$  in  $\theta$ . The chain of  $d_i$ 's ensures that if  $x_{i+1}$  is decided, then so are  $x_1, \dots, x_i$ . We will need these partial assignments to prove by induction that each modal step in our formula corresponds to a quantifier step in the QBF. Next we will define a formula  $\varphi_\Phi$  such that

$$\mathfrak{M}, \{d_0\} \models \varphi_\Phi \iff \Phi \text{ is true.}$$

Let  $\phi^*$  be the propositional formula obtained from  $\phi$  by replacing each  $x_i$  by  $e_i^+$  and each  $\neg x_i$  by  $e_i^-$ . We now define formulas  $\varphi_i$  that mirror the TQBF game choices by backward induction on  $i$ :

$$\varphi_n := \phi^*$$

and for  $i = n, \dots, 1$ ,

$$\varphi_{i-1} := \begin{cases} \Diamond(\delta_i \wedge \varphi_i), & \text{if } Q_i = \exists, \\ \Box(\delta_i \rightarrow \varphi_i), & \text{if } Q_i = \forall. \end{cases}$$

Now, define  $\varphi_\Phi := \varphi_0$ .

For the downward induction we need one further definition. For  $0 \leq i \leq n$  and any partial assignment  $\alpha$  to  $x_1, \dots, x_i$ , let

$$\Phi_i(\alpha) := \begin{cases} Q_{i+1}x_{i+1} \dots Q_n x_n \phi(\alpha, x_{i+1}, \dots, x_n), & \text{if } i < n, \\ \phi(\alpha), & \text{if } i = n, \end{cases}$$

be the TQBF suffix starting at position  $i + 1$ . Note that  $\Phi_0(\emptyset) = \Phi$  and  $\Phi_n(\alpha)$  is  $\phi$  evaluated under the total assignment  $\alpha$ .

**Invariant.** We prove by downward induction on  $i$  an invariant that says that truth is preserved under eliminating quantifier  $Q_i$  by substituting in the values from the partial assignment  $\alpha_w$  for the variable  $x_i$  where  $w \models \delta_i$ :

*Invariant  $J(i)$ .* For every state  $w$  with  $w \models \delta_i$  (i.e.  $w$  decides exactly  $x_1, \dots, x_i$ ),

$$\mathfrak{M}_\Phi, w \models \varphi_i \quad \text{iff} \quad \Phi_i(\alpha_w) \text{ is true.}$$

**Base case  $i = n$ .** Let  $w$  be such that  $w \models \delta_n$ , so  $\alpha_w$  decides all variables. Then

$$\varphi_n = \phi^*$$

and, by construction of  $\phi^*$ ,

$$\mathfrak{M}_\Phi, w \models \varphi_n \iff \phi^* \text{ is true in } w \iff \phi(\alpha_w) \text{ is true} \iff \Phi_n(\alpha_w) \text{ is true.}$$

Thus  $J(n)$  holds.

**Inductive step.** Assume  $J(i)$  for some  $1 \leq i \leq n$  and consider  $J(i - 1)$ . Let  $w$  be such that  $w \models \delta_{i-1}$ .

*Case  $Q_i = \exists$ :* Then

$$\varphi_{i-1} = \Diamond(\delta_i \wedge \varphi_i).$$

Intuitively, at a state  $w \models \delta_{i-1}$  the next variable to be chosen is  $x_i$ . The  $\Diamond$ -step represents the existential player selecting one of the two possible values for  $x_i$ , i.e. moving to a  $\preceq$ -successor that decides  $x_i$  and from there satisfies the continuation  $\varphi_i$ .

Formally, by the semantics of  $\Diamond$  and the fact that  $w \preceq t$  iff  $w \subseteq t$ ,

$$\mathfrak{M}_\Phi, w \models \varphi_{i-1} \iff \exists t \supseteq w (t \models \delta_i \wedge \mathfrak{M}_\Phi, t \models \varphi_i).$$

Now fix such a state  $w$  with  $w \models \delta_{i-1}$ . By the definition of  $\delta_i$  (and the state law  $\theta$ ), the condition  $t \models \delta_i$  means that  $t$  extends  $w$  by deciding *exactly* the next variable  $x_i$  (and no later one). Hence there are precisely two  $\preceq$ -minimal successors  $t \supseteq w$  with  $t \models \delta_i$ :

$$t_w^{(0)} := w \cup \{e_i^-\} \cup \{d_i\} \quad \text{and} \quad t_w^{(1)} := w \cup \{e_i^+\} \cup \{d_i\},$$

encoding the choices  $x_i = 0$  and  $x_i = 1$ , respectively. Any further extension of  $t_w^{(a)}$  decides variables  $x_{i+1}, x_{i+2}, \dots$  and is therefore not  $\preceq$ -minimal among the  $\delta_i$ -states above  $w$ .

Therefore the existential quantification over  $t$  can be restricted to these two states:

$$\mathfrak{M}_\Phi, w \models \varphi_{i-1} \iff (\mathfrak{M}_\Phi, t_w^{(0)} \models \varphi_i) \text{ or } (\mathfrak{M}_\Phi, t_w^{(1)} \models \varphi_i).$$

For  $a \in \{0, 1\}$ , the partial assignment induced by  $t_w^{(a)}$  is exactly the extension of  $\alpha_w$  obtained by setting  $x_i$  to  $a$ :

$$\alpha_{t_w^{(a)}} = \alpha_w[x_i \mapsto a].$$

By the induction hypothesis  $J(i)$  (which applies since  $t_w^{(a)} \models \delta_i$ ),

$$\mathfrak{M}_\Phi, t_w^{(a)} \models \varphi_i \iff \Phi_i(\alpha_{t_w^{(a)}}) \text{ is true} \iff \Phi_i(\alpha_w[x_i \mapsto a]) \text{ is true}.$$

Combining these gives us

$$\mathfrak{M}_\Phi, w \models \varphi_{i-1} \iff \exists a \in \{0, 1\} \Phi_i(\alpha_w[x_i \mapsto a]) \text{ is true}.$$

This is exactly the truth condition for the QBF suffix

$$\Phi_{i-1}(\alpha_w) = \exists x_i \Phi_i(\alpha_w, x_i),$$

and hence  $J(i-1)$  holds when  $Q_i = \exists$ .

Case  $Q_i = \forall$ : Now

$$\varphi_{i-1} = \Box(\delta_i \rightarrow \varphi_i).$$

Intuitively, at a state  $w \models \delta_{i-1}$  the next variable to be chosen is  $x_i$ . The  $\Box$ -step represents the universal player choosing the value of  $x_i$ ; hence  $\varphi_i$  must hold after *both* possible choices. Again we guard the universal step by  $\delta_i$  so that we only quantify over the two immediate  $x_i$ -extensions.

Formally, by the semantics of  $\Box$  and the fact that  $w \preceq t$  iff  $w \subseteq t$ ,

$$\mathfrak{M}_\Phi, w \models \varphi_{i-1} \iff \forall t \supseteq w (t \models \delta_i \Rightarrow \mathfrak{M}_\Phi, t \models \varphi_i).$$

Now fix such a state  $w$  with  $w \models \delta_{i-1}$ . As in the existential case, the condition  $t \models \delta_i$  means that  $t$  extends  $w$  by deciding *exactly* the next variable  $x_i$ . Hence the only  $\preceq$ -minimal successors  $t \supseteq w$  with  $t \models \delta_i$  are the two states

$$t_w^{(0)} := w \cup \{e_i^-\} \cup \{d_i\} \quad \text{and} \quad t_w^{(1)} := w \cup \{e_i^+\} \cup \{d_i\},$$

encoding the choices  $x_i = 0$  and  $x_i = 1$ , respectively. The universal condition reduces to checking  $\varphi_i$  at both guarded successors:

$$\mathfrak{M}_\Phi, w \models \varphi_{i-1} \iff (\mathfrak{M}_\Phi, t_w^{(0)} \models \varphi_i) \text{ and } (\mathfrak{M}_\Phi, t_w^{(1)} \models \varphi_i).$$

For  $a \in \{0, 1\}$ , we again have

$$\alpha_{t_w^{(a)}} = \alpha_w[x_i \mapsto a].$$

By the induction hypothesis  $J(i)$  (which applies since  $t_w^{(a)} \models \delta_i$ ),

$$\mathfrak{M}_\Phi, t_w^{(a)} \models \varphi_i \iff \Phi_i(\alpha_{t_w^{(a)}}) \text{ is true} \iff \Phi_i(\alpha_w[x_i \mapsto a]) \text{ is true.}$$

Combining these gives

$$\mathfrak{M}_\Phi, w \models \varphi_{i-1} \iff \forall a \in \{0, 1\} \Phi_i(\alpha_w[x_i \mapsto a]) \text{ is true.}$$

This is exactly the truth condition for the QBF suffix

$$\Phi_{i-1}(\alpha_w) = \forall x_i \Phi_i(\alpha_w, x_i),$$

and hence  $J(i-1)$  holds when  $Q_i = \forall$ .

Now it is only left to show that the reduction runs in polynomial time:

- *Model size and BDD for  $\theta$ .* We have that the number of variables  $n = |\text{Var}(\phi)| \leq |\phi|$ . We introduce  $3n + 1$  propositional variables,

$$V_\theta = \{d_0, \dots, d_n\} \cup \{e_i^+, e_i^- \mid 1 \leq i \leq n\},$$

so  $|V_\theta| = \mathcal{O}(n)$ . Further,  $E = V$ ,  $O = \emptyset$  for the unique agent. The state law  $\theta$  is the conjunction

$$\theta = d_0 \wedge \bigwedge_{i=0}^{n-1} (d_{i+1} \rightarrow d_i) \wedge \bigwedge_{i=1}^n \left( d_i \leftrightarrow (e_i^+ \vee e_i^-) \right) \wedge \bigwedge_{i=1}^n (\neg e_i^+ \vee \neg e_i^-).$$

Expanding this into CNF, we get clauses of the types:

1.  $(d_0)$ .
2.  $(\neg d_{i+1} \vee d_i)$
3.  $(\neg d_i \vee e_i^+ \vee e_i^-)$ ,  $(\neg e_i^+ \vee d_i)$  and  $(\neg e_i^- \vee d_i)$ .
4.  $(\neg e_i^+ \vee \neg e_i^-)$

These are  $\mathcal{O}(n)$  clauses with a total of  $\mathcal{O}(n)$  variables.

We now prove that this CNF admits a polynomial-size BDD. Under the ordering

$$d_0 \prec e_1^+ \prec e_1^- \prec d_1 \prec \dots \prec e_n^+ \prec e_n^- \prec d_n$$

the Gaifman graph (see Definition 4) consists of triangles  $\{d_i, e_i^+, e_i^-\}$  (induced by clause types (3) and (4)), which are linked by edges  $\{d_i, d_{i+1}\}$  (induced by clause type (2)). Hence, it has constant treewidth by the obvious tree decomposition with bags  $\{d_i, e_i^+, e_i^-\}$  and  $\{d_i, d_{i+1}\}$ . By [FPV05, Thm 2] (see our Theorem 5), we can construct the OBDD  $\theta$  in polynomial-time.

- *Translation of the formula.* The formula  $\phi^*$  is obtained from  $\phi$  by replacing each occurrence of  $x_i$  by  $e_i^+$  and each  $\neg x_i$  by  $e_i^-$ . This does not increase the size of the formula. Thus,  $|\phi^*| = \mathcal{O}(|\phi|)$ .

We set  $\varphi_n := \phi^*$  and define for  $i = n, \dots, 1$ :

$$\varphi_{i-1} := \begin{cases} \Diamond(\delta_i \wedge \varphi_i), & \text{if } Q_i = \exists, \\ \Box(\delta_i \rightarrow \varphi_i), & \text{if } Q_i = \forall, \end{cases}$$

where  $\delta_i$  is of constant size (either  $d_i \wedge \neg d_{i+1}$  or  $d_n$ ). Each step adds a constant amount of symbols around  $\varphi_i$ , so

$$|\varphi_0| \leq |\phi^*| + \mathcal{O}(n) = \mathcal{O}(|\phi| + n),$$

which is polynomial in the size of  $\Phi$ .

Thus the reduction is a polynomial-time many-one reduction from TQBF to symbolic model checking in the one-agent case.  $\square$

### 3.2.4 $\Theta_2^P$ -completeness for single-agent $\mathcal{L}_{\mathbb{M}}$ fragment

In this section, we prove membership of model-checking the 1-agent  $[\forall]$  only fragment  $\mathcal{L}_{[\forall]}$  of  $\mathcal{L}_{\mathbb{M}\Box}$  on symbolic Topo-E-Models. This result is also a consequence of  $\Theta_2^P$ -membership of the bounded evidence case (next section) but it can be proved easily and provides a good introduction to what follows. First we recall a standard result: S5-formulas can syntactically be reduced to equivalent formulas with modal depth at most 1 [CH96, Ch. 5]. Flattening a formula first allows us to model-check it with polynomially many parallel NP-oracle queries. This corresponds to the complexity class  $P_{\parallel}^{\text{NP}} = \Theta_2^P$ . To spell out the algorithm cleanly we use the following (polynomial-time) sub-routine:

- FLATTENS5( $\psi$ ) takes a NNF  $\mathcal{L}_{[\forall]}$ -formula and reduces it to one of modal depth at most 1.

For propositional  $\Phi$  we define the oracles:

$$\text{SAT}(o, \Phi) = 1 \iff (\theta \wedge \pi_O(o) \wedge \Phi) \text{ is satisfiable}, \quad \text{UNSAT}(o, \Phi) := \neg \text{SAT}(o, \Phi).$$

---

**Algorithm 2:** MC- $\mathcal{L}_{[\forall]}$  (1-agent  $[\forall]$ -only fragment,  $\Theta_2^P$  membership)

---

**Input:** STEM  $\mathcal{M} = (V, \theta, O)$  with  $\theta$  as a BDD, world  $w \subseteq V$  with  $\text{BDD-EVAL}(\theta, w) = 1$ ,  
formula  $\psi \in \mathcal{L}_{[\forall]}$

**Output:** *true* iff  $\mathcal{M}, w \models \psi$

```

1  $\psi \leftarrow \text{FLATTENS5}(\psi)$ 
2  $o \leftarrow w \cap O$ 
3  $\Phi \leftarrow \{ \varphi \mid \varphi \text{ is a modal subformula of } \psi \}$ 
4 foreach  $\varphi = [\forall]\alpha \in \Phi$  do
5    $b_\varphi \leftarrow \text{UNSAT}(o, \neg\alpha)$ 
6 foreach  $\varphi = \langle \exists \rangle \alpha \in \Phi$  do
7    $b_\varphi \leftarrow \text{SAT}(o, \alpha)$ 
8  $\psi^* \leftarrow \psi$ 
9 foreach  $\varphi \in \Phi$  do
10   $\psi^* \leftarrow \psi^*[\varphi \mapsto b_\varphi]$ 
11 return  $\text{EVALPROP}(w, \psi^*)$ 
```

---

**Theorem 17** ( $\Theta_2^P$ -membership). *Given a one-agent Topo-E-Model  $\mathfrak{M}$ , a  $\mathcal{L}_{[\forall]}$ -formula  $\psi$  and a world  $w \subseteq V$  checking whether*

$$\mathfrak{M}, w \models \psi$$

*is in  $\Theta_2^P$*

*Proof.* We prove the results by showing that Algorithm 2 is correct and in  $P_{\parallel}^{\text{NP}}$ .

Truth-preservation under  $\text{FLATTENS5}(\psi)$  and polynomial-time of it are standard results [CH96, Ch. 5]. Thus, after the flattening step we obtain an equivalent formula of modal depth at most 1, and hence every modal subformula is of the form  $[\forall]\alpha$  or  $\langle\exists\rangle\alpha$  with propositional  $\alpha$ . Let  $o := w \cap O$ . Then the oracle  $\text{SAT}(o, \cdot)$  mirrors the semantics of the depth-1 modalities:

$$\begin{aligned} \mathfrak{M}, w \models \langle\exists\rangle\alpha &\iff \exists s \subseteq V : (s \models \theta \wedge s \cap O = o \wedge s \models \alpha) && \iff \text{SAT}(o, \alpha). \\ \mathfrak{M}, w \models [\forall]\alpha &\iff \neg \exists s \subseteq V : (s \models \theta \wedge s \cap O = o \wedge s \models \neg\alpha) && \iff \text{UNSAT}(o, \neg\alpha). \end{aligned}$$

The number of modal subformulas is linear in  $|\psi|$ , hence we only need polynomially many oracle calls. Substituting the resulting truth-values gives us a propositional formula, which can be evaluated by  $\text{EVALPROP}(w, \cdot)$  in polynomial time. Therefore we have a polynomial-time algorithm with one batch of polynomially many non-adaptive NP-oracle calls, i.e. in  $P_{\parallel}^{\text{NP}}$ . This shows membership in  $\Theta_2^P$ .  $\square$

We prove hardness by a reduction from the canonical  $\Theta_2^P$ -complete problem MAX-MODEL-PARITY:

**Definition 18** (MAX-MODEL-PARITY). *Given a satisfiable propositional formula  $\varphi$  decide whether*

$$\max_{\alpha \models \varphi} \left( \sum_i \alpha_i \right) \quad \text{is odd.}$$

For our reduction, we will need counting circuits to express "among the variables  $x_1, \dots, x_n$  at most  $k$  are true". These have to be constructible in polynomial-time. There are many options for this. We will use the one defined in [Sin05].

**Definition 19** (Sinz sequential counter). *Fix  $n \geq 2$  and  $k \geq 1$ . Let*

$$X = \{x_1, \dots, x_n\}, \quad S = \{s_{i,j} \mid 1 \leq i \leq n-1, 1 \leq j \leq k\}.$$

*Define  $LT_{n,k}^{\text{SEQ}}$  as the following set (conjunction) of clauses:*

$$\begin{aligned} LT_{n,k}^{\text{SEQ}} := & \{(\neg x_1 \vee s_{1,1})\} \cup \{(\neg s_{1,j}) \mid 2 \leq j \leq k\} \\ & \cup \bigcup_{i=2}^{n-1} \left( \{(\neg x_i \vee s_{i,1}), (\neg s_{i-1,1} \vee s_{i,1}), (\neg x_i \vee \neg s_{i-1,k})\} \right. \\ & \quad \left. \cup \bigcup_{j=2}^k \{(\neg x_i \vee \neg s_{i-1,j-1} \vee s_{i,j}), (\neg s_{i-1,j} \vee s_{i,j})\} \right) \\ & \cup \{(\neg x_n \vee \neg s_{n-1,k})\}. \end{aligned}$$

Sinz proves in [Sin05, Thm. 1] that

1.  $LT_{n,k}^{\text{SEQ}}$  is a clausal encoding of  $\leq k(x_1, \dots, x_n)$ , i.e. among  $x_1, \dots, x_n$  at most  $k$  variables are set to true.

2. spelling this out requires  $\mathcal{O}(n * k)$  clauses and  $\mathcal{O}(n * k)$  auxiliary variables.

We do not want to express that at most but that at least  $j$  variables are true. We can do this by using the duality defined in Lemma 2 in [Sin05]:

$$\geq j(x_1, \dots, x_n) \text{ iff } \leq (n - j)(\neg x_1, \dots, \neg x_n) \text{ iff } LT_{n, n-j}^{\text{SEQ}}[x_i \mapsto \neg x_i \text{ for all } i].$$

**Theorem 20** ( $\Theta_2^P$ -hardness). *For every MAX-MODEL-PARITY instance  $\varphi$  over variables  $x_1, \dots, x_n$ , we can construct in polynomial time a one-agent symbolic Topo-E-Model  $\mathfrak{M}$  and an  $\mathcal{L}_{[\forall]}$ -formula  $\Psi$  such that for every world  $w \subseteq V$ ,*

$$\max_{\alpha \models \varphi} \left( \sum_i \alpha_i \right) \text{ is odd} \quad \text{iff} \quad \mathfrak{M}, w \models \Psi,$$

*Reduction.* Given  $\varphi$ , construct the symbolic Topo-E-Model  $\mathfrak{M} = (V, \theta, O, E)$  as follows. Let  $X := \{x_1, \dots, x_n\}$  and let  $S$  be a set of auxiliary variables used by a sequential-counter encoding (as defined above). Define

$$V := X \cup S, \quad O := \emptyset, \quad E := \emptyset, \quad \theta := \top.$$

Thus  $\theta$  is a constant BDD and since  $O = \emptyset$   $[\forall]$  and  $\langle \exists \rangle$  quantify over all worlds  $t \subseteq V$ .

Define

$$C_{\geq j} := LT_{n, n-j}^{\text{SEQ}}[x_i \mapsto \neg x_i \text{ for all } i].$$

By correctness of the Sinz encoding, for every assignment  $\beta$  to  $X$ ,

$$\beta \models C_{\geq j} \quad (\text{for some extension to } S) \quad \iff \quad \sum_i \beta_i \geq j.$$

In particular,  $C_{\geq j}$  has size polynomial in  $n$ .

Now, for each  $1 \leq j \leq n$  define the  $\mathcal{L}_{[\forall]}$ -formula

$$\psi_j := \langle \exists \rangle (\varphi \wedge C_{\geq j}) \wedge [\forall] (C_{\geq (j+1)} \rightarrow \neg \varphi).$$

Intuitively,  $\psi_j$  says: there exists a satisfying assignment with at least  $j$  ones, and there is no satisfying assignment with at least  $j+1$  ones; hence the maximum number of ones is exactly  $j$ .

Finally, let

$$\Psi := \bigvee_{\substack{1 \leq j \leq n \\ j \text{ odd}}} \psi_j.$$

**Correctness.** Let  $OPT := \max_{\alpha \models \varphi} \sum \alpha_i$ . Fix any world  $w \subseteq V$ . Since  $O = \emptyset$  and  $\theta = \top$ , the modal operators quantify over all worlds  $t \subseteq V$ . By Sinz results, we have

- $\mathfrak{M}, w \models \langle \exists \rangle (\varphi \wedge C_{\geq j})$  iff there exists a satisfying assignment of  $\varphi$  with at least  $j$  ones, i.e. iff  $OPT \geq j$ .
- $\mathfrak{M}, w \models [\forall] (C_{\geq (j+1)} \rightarrow \neg \varphi)$  iff there is no satisfying assignment of  $\varphi$  with at least  $j+1$  ones, i.e. iff  $OPT \leq j$ .

Hence  $\mathfrak{M}, w \models \psi_j$  iff  $OPT = j$ , and therefore  $\mathfrak{M}, w \models \Psi$  iff  $OPT$  is odd.

**Size bound.** The model has  $\theta = \top$  and  $O = \emptyset$ , and  $|V|$  is polynomial in  $n$ . The formula  $\Psi$  contains  $\mathcal{O}(n)$  many  $\psi_j$ , each using a polynomial-size Sinz encoding. Thus,  $|\Psi|$  is polynomial in  $n$  and therefore in  $|\varphi|$ .  $\square$

### 3.2.5 $\Theta_2^P$ -completeness for single-agent bounded evidence

Fix a one-agent symbolic Topo-E-model  $\mathfrak{M} = (V, \theta, O, E)$  and a legal world  $w$ . As in Sec. 3.2.1 we fix the observation cell once by replacing  $\theta$  with  $\theta \wedge \pi_O(w)$ , and write  $\theta$  thereafter. Assume  $|E| \leq k$  for a fixed constant  $k$ . First note that we get hardness by simply setting evidence and observables to empty. Then, the reduction for the  $\forall$ -only case applies verbatim. We show that symbolic model checking in this fragment is in  $\Theta_2^P$ . We design a poly-time algorithm that has an NP-DAG with bounded separator number as query structure. By theorem 2 from Gharibian et al. [GR22] this suffices to conclude  $\Theta_2^P$  membership. The theorem invoked is a strengthening of Gottlob’s classical result about  $\Theta_2^P$  completeness for NP-Trees [Got95]. We will further explain this result below and first provide an intuitive introduction to how our algorithm works.

Algorithm 3 evaluates  $\varphi$  at  $w$  by (i) enumerating the (constant-size) domain of consistent evidence blocks, (ii) decomposing  $\varphi$  into *modal gadgets* whose dependencies form a tree, and (iii) evaluating this dependency tree bottom-up using local SAT tests on blocks. Bounded evidence restricts the modalities to a constant-size domain  $L \subseteq 2^E$  of size at most  $2^k$  on which they are invariant, so every modal-rooted subformula is equivalent to a constant-size *circuit*  $X \subseteq L$ .

We first present the modal gadgets we will use in the algorithm.

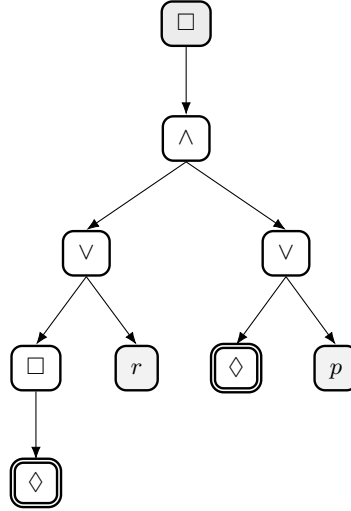


Figure 3.2: A modal gadget

**Definition 21** (modal gadgets (leaf-modal form)). *Let  $\varphi$  be in NNF and let  $T_\varphi$  be its syntax tree. A node is modal if its label is in  $\{\Diamond, \Box\}$ . A node  $u$  is propositional if the subtree rooted at  $u$  contains no modal nodes.*

*We define gadgets only for modal nodes  $m$  whose child is non-modal; let  $b(m)$  be that child.*

*Let  $\text{Bool}(b(m))$  be obtained by starting from  $b(m)$  and closing under  $\wedge, \vee$ -children as long as we do not reach a modal node and do not expand below propositional nodes (maximal propositional subformulas are treated as atoms).*

*Let  $\text{MC}(m)$  be the set of modal children of nodes in  $\text{Bool}(b(m))$ .*

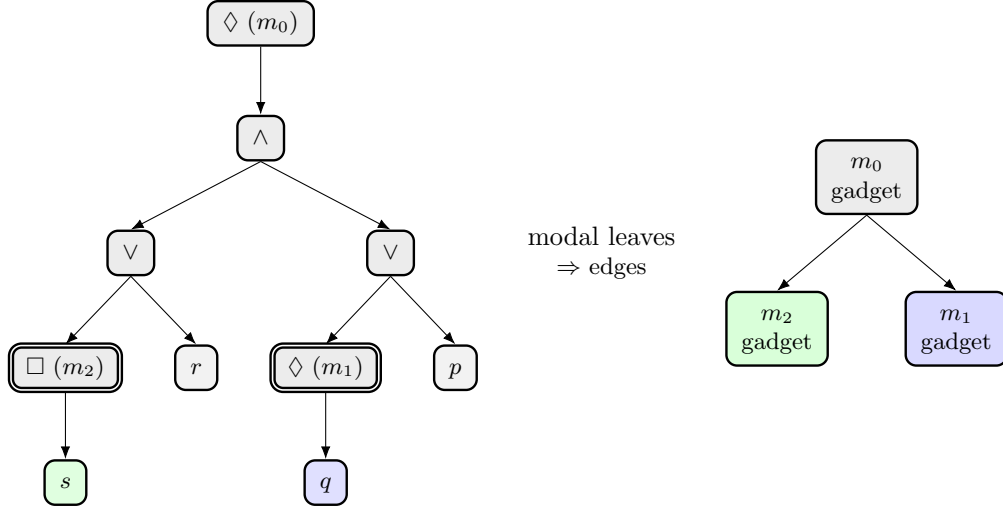


Figure 3.3: From syntax tree to dependency tree; constructing the 3 gadgets  $m_0, m_1, m_2$ .

For each  $m' \in MC(m)$ , let

$$m' = m_0 \rightarrow m_1 \rightarrow \dots \rightarrow m_r$$

be the maximal downward chain of modal nodes starting at  $m'$  (each  $m_i$  is modal and  $m_r$  has a non-modal child).

The gadget of  $m$  is

$$\text{Gad}(m) := \{m\} \cup \text{Bool}(b(m)) \cup \bigcup_{m' \in MC(m)} \{m_0, \dots, m_r\}.$$

Note that the leaves of these gadgets are either propositional formulas or modal gadgets (modal rooted formula). Propositional formulas can be evaluated by an NP-oracle. The modal leaves we will evaluate by substituting pre-computed circuits. This leads to the dependency tree shown in Figure 3.3 between the gadgets.

**Definition 22** (Dependency forest). For modal node  $m$  in  $T_\varphi$  where  $T_\varphi$  is the syntax tree of  $\varphi$  for which there exists  $\text{Gad}(m)$ , let  $\text{Leaves}(m)$  be the gadget leaves of  $\text{Gad}(m)$  and define the modal leaves

$$\text{ModLeaves}(m) := \{\ell \in \text{Leaves}(m) \mid \text{lab}(\ell) \in \{\diamond, \square\}\}.$$

We define a directed edge between modal nodes by

$$m \rightarrow m' \quad :\Longleftrightarrow \quad m' \in \text{ModLeaves}(m).$$

Hence, we get a tree for each outermost modal node  $r$ . The resulting rooted directed structure

$$\text{DepTree}(r) := (\text{Mod}(\varphi), \rightarrow, r)$$

is the dependency tree of  $r$ . It is a tree because edges always point to strict descendants in the syntax tree, and a modal node has at most one parent gadget in the dependency forest. Let  $F_\varphi$  be the forest we get by taking the union over the dependency trees for all maximal modal roots.

Let

$$L := \{\ell \subseteq E \mid \text{SAT}(\theta \wedge \pi_E(\ell)) = 1\}$$

be the set of consistent (that is non-empty) evidence blocks. Since  $|E| \leq k$  is fixed,  $|L| \leq 2^k$  is a constant.

We define for some  $X \subseteq L$

$$R^{-1}(X) := \{\ell \in L \mid \exists \ell' \in X : \ell \subseteq \ell'\}.$$

which amounts to the inverse on the subset induced relation on  $L$  corresponding to the relation induced by the  $w \cap E \subseteq v \cap E$  in the semantics clauses for  $\Box$ .

**Lemma 23** (Block invariance of modal roots). *Let  $\psi$  be modal-rooted and let  $u, v \models \theta$  with  $u \cap E = v \cap E$  (i.e.  $\ell_u = \ell_v$ ). Then*

$$\mathfrak{M}, u \models \psi \iff \mathfrak{M}, v \models \psi.$$

*Thus, for every modal-rooted  $\alpha$  in the syntax tree of  $\varphi$  there exists a circuit  $X_\alpha \subseteq L$  with size at most  $|L| \leq 2^k$  such that, for all  $\theta(u) = 1$ ,*

$$\mathfrak{M}, u \models \text{Form}(\alpha) \iff \ell_u \in X_\alpha.$$

*Proof.* If  $u \cap E = v \cap E$ , then  $u$  and  $v$  have exactly the same successor blocks, because successors are determined solely by evidence extension  $\subseteq$  at the block level. Since  $\psi$  is modal-rooted, its evaluation only quantifies over successors all of which depend only on the current evidence block. Hence the truth of  $\psi$  is identical at  $u$  and  $v$ . The circuit  $X_\alpha$  is obtained by collecting those blocks  $\ell \in L$  for which  $\text{Form}(\alpha)$  is true at some (equivalently, at every) world in block  $\ell$ .  $\square$

We will freely switch between the set view  $X \subseteq L$  and the propositional formula  $\xi_X$  over  $E$  representing it:

$$\xi_X := \bigvee_{\ell \in X} \pi_E(\ell), \quad \text{so that} \quad \text{Ext}(\xi_X) = \{\ell \in L \mid \text{SAT}(\ell, \xi_X) = 1\} = X.$$

This translation is constant-size under bounded evidence.

We now present the bounded evidence model checking algorithm (ALG 3) in pseudo code and prove correctness. Figure 3.4 shows the high level scheme of the algorithm. For our algorithm, we need two subroutines: MODALELIMINATION (ALG 4) and EVALGADGET (ALG 5). MODALELIMINATION is used to eliminate stacked modalities using set operations on circuits. We will need it for the other two subroutines.

EVALGADGET evaluates the gadgets. As input it takes a modal gadget and circuits for the modal leaves. The first step then is to eliminate stacked modalities at the leaves using MODALELIMINATION. It then proceeds by producing a propositional formula that is equivalent to the modal gadget. This allows us to eliminate the gadget by substituting an equivalent *propositional* formula for it which then allows us to solve for the blocks where it is true using  $2^{|L|}$  SAT oracle calls. We save the result in a table and delete the node.

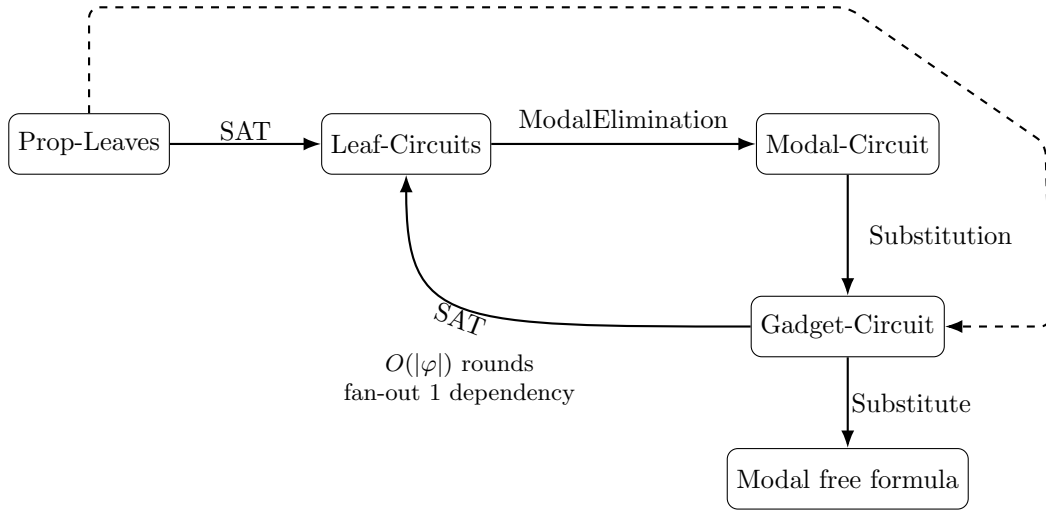


Figure 3.4: High-level scheme: SAT gives us circuits for leaves; modal elimination eliminates modalities; substitution gives us gadget circuits; repeating this process for  $O(|\varphi|)$  rounds gives equivalent circuits for all modal rooted formulas.

We start by proving correctness of  $\text{BOUNDEDEVIDENCEMC}^1$  given correctness of the sub-routines and afterwards prove the correctness lemmas for the sub-routines.

---

**Algorithm 3:**  $\text{BOUNDEDEVIDENCEMC}$

---

**Input:**  $\mathfrak{M} = (V, \theta, E, O)$ , legal  $w \models \theta$ , NNF  $\varphi$ ,  $|E| \leq k$

**Output:** *true* iff  $\mathfrak{M}, w \models \varphi$

```

1  $\theta \leftarrow \theta \wedge \pi_O(w)$ 
2  $\ell_w \leftarrow w \cap E$ 
3  $L \leftarrow \{\ell \subseteq E \mid \text{SAT}(\theta \wedge \pi_E(\ell)) = 1\}$ 
4 Build the gadget dependency forest  $F$  (nodes = gadget roots, edges as in Def. 22)
5  $\text{Val} \leftarrow$  empty map
6 while  $F$  nonempty do
7   foreach  $m \in \text{Leaves}(F)$  do
8      $\text{Val}[m] \leftarrow \text{EVALGADGET}(m, \text{Val}, L, \theta)$ 
9   delete all leaves from  $F$ 
10  $\varphi^* \leftarrow \varphi$ 
11 foreach maximal modal-root occurrence  $m$  in  $\varphi$  do
12    $\varphi^* \leftarrow \varphi^*[m \mapsto (\ell_w \in \text{Val}[m])]$ 
13 return  $\text{EVALPROP}(w, \varphi^*)$ 

```

---

**Theorem 24** (Correctness of  $\text{BOUNDEDEVIDENCEMC}$ ). *For every one-agent symbolic Topo-E-model  $\mathfrak{M} = (V, \theta, O, E)$ , every legal world  $w$ , and every NNF formula  $\varphi$  with  $|E| \leq k$ ,  $\text{BOUNDEDEVIDENCEMC}(\mathfrak{M}, w, \varphi)$  returns true iff*

$$\mathfrak{M}, w \models \varphi.$$

---

<sup>1</sup>It is easy to adjust the algorithm such that it only takes  $\log n$  rounds by using a tree contraction scheme. After Raking all leaves, we can compress all unary chains of modal gadgets (Caterpillar like syntax tree) by function composition: as each modal gadget only depends on one child gadget, we can produce tables  $\mathcal{F}_{m_i} : 2^L \rightarrow 2^L$  for each gadget  $m_i$  in the chain by evaluating them for all possible input circuits. Then, we can evaluate the chain by function composition and evaluate by taking the value for the lowest gadget as input. While this shows the spirit of parallelizing the calls, we avoid this here as it is not necessary for the proof we provide here.

Moreover, the algorithm terminates after  $\mathcal{O}(|\varphi|)$  rounds.

*Proof.* Let  $F$  be the gadget dependency forest built from the maximal modal roots of  $\varphi$ . For each gadget root  $m$ , let  $X_m \subseteq L$  denote the (unique) truth circuit of  $\varphi_m$ :

$$\ell_u \in X_m \iff \mathfrak{M}, u \models \varphi_m \quad \text{for all legal } u \models \theta.$$

(Existence and uniqueness follow from block invariance of modal roots.)

We prove the following invariant by induction over the rounds of the algorithm:

**Invariant.** Whenever a gadget root  $m$  is deleted from the current forest  $F$ , the table entry  $\text{Val}[m]$  is defined and equals  $X_m$ .

**Rake step:** Let  $m$  be a leaf of the current forest. Then all gadget dependencies of  $m$  have already been deleted in earlier rounds. Then, by the induction hypothesis, the entries  $\text{Val}[c]$  are correct for every  $c \in \text{ModalChildren}(m)$ . By Lemma 26, EVALGADGET computes  $\text{Val}[m] = X_m$  correctly given the correctness of all children. Hence every leaf deleted in the rake step satisfies the invariant.

By induction we conclude correctness of the algorithm. At termination, all gadget roots have been deleted and therefore all maximal modal-root occurrences  $m$  in  $\varphi$  satisfy

$$\text{Val}[m] = X_m.$$

The final substitution replaces each maximal modal-root  $m$  by the circuit  $(\ell_w \in \text{Val}[m])$ , which is equivalent to the propositional formula  $\xi_{X_m}$  and true when  $\mathfrak{M}, w \models \varphi_m$ . Hence the resulting formula  $\varphi^*$  is a propositional formula equivalent to  $\varphi$  at  $w$ :

$$w \models \varphi^* \iff \mathfrak{M}, w \models \varphi.$$

It remains to show that the algorithm terminates after at most  $|F|$  rounds as it deletes all leaves of the tree per round.  $\square$

---

**Algorithm 4:** MODALELIMINATION (extended)

---

**Input:** Block domain  $L$ , modal prefix  $P = M_1 \cdots M_n$  with  $M_i \in \{\Box, \Diamond, \Box, \langle \exists \rangle\}$ , base circuit

$$C \subseteq L$$

**Output:** Circuit  $C_{P\psi} \subseteq L$

```
1 for  $i \leftarrow n$  to 1 do
2    $M \leftarrow M_i$ 
3   if  $M = \Diamond$  then
4      $C \leftarrow R^{-1}(C)$ 
5   else if  $M = \Box$  then
6      $C \leftarrow L \setminus R^{-1}(L \setminus C)$ 
7   else if  $M = \Box$  then
8     if  $C = L$  then
9        $C \leftarrow L$ 
10    else
11       $C \leftarrow \emptyset$ 
12  else if  $M = \langle \exists \rangle$  then
13    if  $C \neq \emptyset$  then
14       $C \leftarrow L$ 
15    else
16       $C \leftarrow \emptyset$ 
17 return  $C$ 
```

---

**Lemma 25** (Correctness of MODALELIMINATION). *Let  $P = M_1 \cdots M_n$  with  $M_i \in \{\Diamond, \Box, \Box, \langle \exists \rangle\}$  be a modal prefix, and let  $\psi$  be a modal-rooted formula. Assume  $C_\psi \subseteq L$  is the truth circuit of  $\psi$ , i.e.*

$$\ell_u \in C_\psi \iff \mathfrak{M}, u \models \psi \quad \text{for all legal } u \models \theta.$$

Then MODALELIMINATION( $P, C_\psi$ ) returns a circuit  $C_{P\psi} \subseteq L$  such that

$$\ell_u \in C_{P\psi} \iff \mathfrak{M}, u \models P\psi \quad \text{for all legal } u \models \theta.$$

*Proof.* For  $M \in \{\Diamond, \Box, \Box, \langle \exists \rangle\}$  we define

$$T_M(X) := \begin{cases} R^{-1}(X), & M = \Diamond, \\ L \setminus R^{-1}(L \setminus X), & M = \Box, \\ L, & M = \Box \text{ and } X = L, \\ \emptyset, & M = \Box \text{ and } X \neq L, \\ L, & M = \langle \exists \rangle \text{ and } X \neq \emptyset, \\ \emptyset, & M = \langle \exists \rangle \text{ and } X = \emptyset. \end{cases}$$

Then MODALELIMINATION( $P, C_\psi$ ) computes

$$C_{P\psi} = T_{M_1}(T_{M_2}(\cdots T_{M_n}(C_\psi))).$$

Now, for every modal-rooted formula  $\chi$  with truth circuit  $C_\chi$  and every legal world  $u \models \theta$  we have

$$\ell_u \in T_M(C_\chi) \iff \mathfrak{M}, u \models M\chi. \quad (\dagger)$$

For  $M = \Diamond$ , we have

$$\begin{aligned} \ell_u \in T_\Diamond(C_\chi) &\iff \ell_u \in R^{-1}(C_\chi) \\ &\iff \exists v. u \cap E \subseteq v \cap E \text{ and } \ell_v \in C_\chi \\ &\iff \exists v. u \cap E \subseteq v \cap E \text{ and } \mathfrak{M}, v \models \chi \\ &\iff \mathfrak{M}, u \models \Diamond\chi. \end{aligned}$$

For  $M = \Box$ , we have

$$\begin{aligned} \ell_u \in T_\Box(C_\chi) &\iff \ell_u \notin R^{-1}(L \setminus C_\chi) \\ &\iff \nexists v. u \cap E \subseteq v \cap E \text{ and } \ell_v \notin C_\chi \\ &\iff \forall v. u \cap E \subseteq v \cap E \Rightarrow \mathfrak{M}, v \models \chi \\ &\iff \mathfrak{M}, u \models \Box\chi. \end{aligned}$$

For  $M = \boxtimes$ , we have

$$\begin{aligned} \ell_u \in T_{\boxtimes}(C_\chi) &\iff C_\chi = L \\ &\iff \forall \ell \in L, \forall v \models \theta \wedge \pi_E(\ell) : \mathfrak{M}, v \models \chi \\ &\iff \forall v \models \theta : \mathfrak{M}, v \models \chi \\ &\iff \mathfrak{M}, u \models \boxtimes\chi \end{aligned}$$

and for  $M = \langle \exists \rangle$ ,

$$\begin{aligned} \ell_u \in T_{\langle \exists \rangle}(C_\chi) &\iff C_\chi \neq \emptyset \\ &\iff \exists \ell \in L, \exists v \models \theta \wedge \pi_E(\ell) : \mathfrak{M}, v \models \chi \\ &\iff \exists v \models \theta : \mathfrak{M}, v \models \chi \\ &\iff \mathfrak{M}, u \models \langle \exists \rangle\chi. \end{aligned}$$

Thus  $(\dagger)$  holds for all  $M \in \{\Diamond, \Box, \boxtimes, \langle \exists \rangle\}$ , and repeated application gives correctness.  $\square$

---

**Algorithm 5:** EVALGADGET (extended)

---

**Input:** gadget root  $m$ , map  $\text{Val}$ , block domain  $L$ , constraint  $\theta$

**Output:** circuit  $\text{Val}[m] \subseteq L$

```
1  $W \leftarrow \emptyset$ 
2  $V \leftarrow \emptyset$ 
3 foreach  $c \in \text{ModalChildren}(m)$  do
4    $P_c \leftarrow \text{prefix of } c$ 
5    $C_c \leftarrow \text{MODALELIMINATION}(L, P_c, \text{Val}[c])$ 
6    $m \leftarrow m[c \mapsto C_c]$ 
7  $\beta \leftarrow \text{Form}(m)$ 
8 foreach  $\ell \in L$  do
9   if  $\text{SAT}(\theta \wedge \pi_E(\ell) \wedge \beta) = 1$  then
10     $W \leftarrow W \cup \{\ell\}$ 
11   if  $\text{SAT}(\theta \wedge \pi_E(\ell) \wedge \neg\beta) = 0$  then
12     $V \leftarrow V \cup \{\ell\}$ 
13  $r \leftarrow \text{root}(m)$ 
14 if  $r \in \{\Diamond, \langle \exists \rangle\}$  then
15    $\text{Val}[m] \leftarrow \text{MODALELIMINATION}(L, \text{root}(m), W)$ 
16 if  $r \in \{\Box, \Box\}$  then
17    $\text{Val}[m] \leftarrow \text{MODALELIMINATION}(L, \text{root}(m), V)$ 
18 return  $\text{Val}[m]$ 
```

---

**Lemma 26** (Correctness of EVALGADGET). *Let  $m$  be a gadget root. Assume that for every modal leaf  $c \in \text{ModLeaves}(m)$ , the table entry  $\text{Val}[c] \subseteq L$  is already correct:*

$$\ell_u \in \text{Val}[c] \iff \mathfrak{M}, u \models \text{Form}(c) \quad \text{for all legal } u \models \theta.$$

*Then  $\text{EVALGADGET}(m, \text{Val}, L, \theta)$  returns a circuit  $\text{Val}[m] \subseteq L$  such that*

$$\ell_u \in \text{Val}[m] \iff \mathfrak{M}, u \models \text{Form}(m) \quad \text{for all legal } u \models \theta.$$

*Proof.* Write  $M := \text{lab}(m) \in \{\Diamond, \Box, \langle \exists \rangle, \Box\}$ . By the definition of gadgets, the formula below the root of  $m$  has the form

$$\text{Form}(b(m)) = B_m[\phi_1, \dots, \phi_r, P_1\text{Form}(c_1), \dots, P_s\text{Form}(c_s)],$$

where:

- $B_m[\dots]$  is a Boolean combination (using only  $\wedge, \vee$ ),
- each  $\phi_j$  is propositional, and
- $c_1, \dots, c_s$  are exactly the modal leaves of the gadget of  $m$ , with  $P_i$  the modal prefix above  $c_i$  inside the gadget (possibly empty).

The evaluation starts with eliminating the prefix using MODALELIMINATION. For each  $i \in \{1, \dots, s\}$ ,

the value  $\text{Val}[c_i]$  is the exact truth circuit of  $\text{Form}(c_i)$  by assumption. Hence, by Lemma 25,

$$C_i := \text{MODALELIMINATION}(P_i, \text{Val}[c_i])$$

is the exact truth circuit of  $P_i\text{Form}(c_i)$ :

$$\ell_u \in C_i \iff \mathfrak{M}, u \models P_i\text{Form}(c_i) \quad \text{for all legal } u \models \theta.$$

In the next step, we substitute the modal parts by the computed circuits: let  $\beta_m$  be the formula obtained from  $\text{Form}(b(m))$  by replacing each subformula  $P_i\text{Form}(c_i)$  by the (equivalent) propositional representation  $\xi_{C_i}$ . Then  $\beta_m$  is modal-free, and each substitution preserves truth on legal worlds. Therefore, for every legal world  $u \models \theta$ ,

$$\mathfrak{M}, u \models \beta_m \iff \mathfrak{M}, u \models \text{Form}(b(m)).$$

The SAT loop in EVALGADGET computes

$$W = \{\ell \in L \mid \text{SAT}(\theta \wedge \pi_E(\ell) \wedge \beta_m) = 1\},$$

$$V = \{\ell \in L \mid \text{SAT}(\theta \wedge \pi_E(\ell) \wedge \neg\beta_m) = 0\}.$$

Hence,

$$\ell \in W \iff \exists v \models \theta \wedge \pi_E(\ell) : \mathfrak{M}, v \models \text{Form}(b(m)),$$

and

$$\ell \in V \iff \nexists v \models \theta \wedge \pi_E(\ell) : \mathfrak{M}, v \models \neg\text{Form}(b(m)).$$

That is,  $\ell \in W$ , resp.  $\ell \in V$ , if and only if  $\text{Form}(b(m))$  is satisfiable in  $\ell$ , resp. valid in  $\ell$ .

Finally, EVALGADGET applies the root modality  $M$  using MODALELIMINATION.

By the correctness of MODALELIMINATION we get

$$\mathfrak{M}, u \models M\text{Form}(b(m)) \iff \mathfrak{M}, u \models \text{Form}(m).$$

Thus,

$$\ell_u \in \text{Val}[m] \iff \mathfrak{M}, u \models \text{Form}(m).$$

This gives us correctness of EVALGADGET.  $\square$

Now that we have established correctness, we still have to show that our algorithm fulfills the requirements to prove  $\Theta_2^P$ -membership. All operations are polynomial, but the dependency structure of the oracles has to be clarified. Note that in general  $\Theta_2^P$  membership is proved by algorithms that either (1) do a polynomial number of non-adaptive NP-oracle calls or (2) do a logarithmic number of adaptive NP-oracle calls. Our algorithm issues a linear number of oracle rounds where each round contains a polynomial number of non-adaptive NP-oracle calls. Without the specific tree-like dependency structure, we would land in a higher complexity class than  $\Theta_2^P$ . But in our case, we have something close to an NP-TREE in the sense of Gottlob [Got95]. Gottlob shows that whenever we have a tree like query graph with NP-oracle nodes, we can simulate it by  $\log n$  oracle queries.

While the gadgets have tree-like dependency satisfying with fan-out 1, the results of the NP-rounds are used by each of the  $2^k$  NP-calls for the evaluation of the next gadget. Thus, each SAT-call has fan-out  $2^k$ . We make use of a refinement of Gottlob's theorem: Gharibian et al. [GR22] prove that given a NP-DAG with a constant (balanced) separator number, we can simulate the oracle queries with a logarithmic number of adaptive NP-oracle queries, i.e. land in  $\Theta_2^P$ . Instead of proving the constant size separator number directly, we prove that we have a constant treewidth for the underlying undirected graph which upper bounds the separator number. Gharibian et al. prove a more general result for a class of oracles. We will only refer to the NP-versions. They define by  $\text{NP-DAG}_s$  a NP-DAG of separator number  $s$  and write  $\text{NP-DAG}_1$  for  $s \in \mathcal{O}(1)$ .

**Theorem 27** (Theorem 2 in [GR22]).

$\text{NP-DAG}_1$  is  $\text{P}^{\text{NP}[\log]}$ -complete.

*In particular, any  $\text{P}^{\text{NP}}$  computation whose query graph has constant separator number is decidable in  $\text{P}^{\text{NP}[\log]}$ .*

We prove an upper (constant) bound on the separator number by showing that the underlying undirected graph has constant treewidth (see Lemma 19[GR22]).

**Lemma 28** (Constant treewidth of the query graph under bounded evidence). *Fix  $k$ . Let  $G_q$  be the NP-query DAG produced by  $\text{BOUNDEDEVIDENCEMC}$  on input  $(\mathfrak{M}, w, \varphi)$  with  $|E| \leq k$ . Then the underlying undirected graph  $U(G_q)$  has treewidth  $\mathcal{O}(1)$  (with the constant depending only on  $k$ ).*

*Proof.* The query graph produced by  $\text{BOUNDEDEVIDENCEMC}$  can be derived from the dependency forest  $F$ . Whenever we evaluate a gadget  $m$ , we do  $|L| \leq 2^k$  oracle calls. These oracle calls depend only on the calls for all children  $m'$  of  $m$  in the dependency tree. The query tree DAG  $G_q = (V_q, E_q)$  can thus be constructed by substituting each gadget  $m$  for  $1 \leq i \leq |L|$  nodes  $m_i$  and connecting  $m_i \rightarrow m'_i$  if and only if  $m'_i$  is a child of  $m_i$ . Intuitively,  $U(G_q)$  is the result of substituting each node in the dependency forest by a constant size cluster  $Q_m$ .

Let  $U(G_q)$  be the corresponding undirected graph. We prove an upper-bound on the treewidth by providing a tree decomposition.

Let  $Q_m$  be the set of query nodes of  $G_q$  generated for gadget  $m$ . The sets  $Q_m$  form a partition of the vertices of  $G_q$  and we have

$$|Q_m| \leq 2^k = b \quad \text{for every gadget root } m.$$

We now define a tree decomposition  $(T_{\text{dec}}, \beta)$  of  $U(G_q)$ .

**Decomposition tree.** Start with the forest  $U(F)$  and subdivide every edge  $\{m, m'\} \in E(F)$  once by a new node  $a_{m,m'}$ . Let  $a_m$  denote the node corresponding to gadget  $m$ . So for each edge  $\{m, m'\}$  in the original forest we get a path

$$a_m - a_{m,m'} - a_{m'}.$$

If  $F$  contains several trees, we add one extra dummy node  $\perp$  and connect it to each root. This turns the forest into a single tree  $T_{\text{dec}}$ .

**Bags.** Define the bag map  $\beta$  by

$$\beta(a_m) := Q_m, \quad \beta(a_{m,m'}) := Q_m \cup Q_{m'}, \quad \beta(\perp) := \emptyset.$$

Verifying the three tree decomposition axioms for  $U(G_q)$  is immediate as  $U(G_q)$  is already almost a tree: Every  $q \in V_q$  belongs to one batch  $Q_m$  thus (1) is fulfilled. Further, edges exist only between direct ancestors and children. Hence, for all connected  $q, q'$

$$\{q, q'\} \subseteq Q_m \cup Q_{m'} = \beta(a_{m,m'}).$$

For the third condition, fix a query vertex  $q \in V(U(G_q))$ , and let  $Q_m$  be its batch.

The bags containing  $q$  are exactly:

- the bag  $\beta(a_m) = Q_m$ , and
- the bags  $\beta(a_{m,m'})$  over neighbors  $m'$  of  $m$  for which  $q$  has a neighbor in  $Q_{m'}$ .

All these bags are incident to the node  $a_m$  in  $T_{\text{dec}}$ . Thus  $(T_{\text{dec}}, \beta)$  is a valid tree decomposition of  $U(G_q)$ .

Every batch has size at most  $b$ . Therefore

$$|\beta(a_m)| = |Q_m| \leq b, \quad |\beta(a_{m,m'})| = |Q_m \cup Q_{m'}| \leq |Q_m| + |Q_{m'}| \leq 2b,$$

and  $\beta(\perp) = \emptyset$ . Hence every bag has size at most  $2b$ , so the width of this decomposition is at most  $2b - 1$ . Thus

$$\text{tw}(U(G_q)) \leq 2b - 1 = \mathcal{O}(1).$$

□

Having established the fixed treewidth of the query DAG, we can easily prove the  $\Theta_2^P$ -membership:

**Theorem 29** ( $\Theta_2^P$ -membership for bounded evidence). *For every fixed  $k$ , symbolic model checking on one-agent symbolic Topo-E-models with  $|E| \leq k$  is in*

$$\Theta_2^P = P^{NP[\log]}.$$

*Proof.* Fix an input  $(\mathfrak{M}, w, \varphi)$ . The algorithm BOUNDEVIDENCEMC runs in deterministic polynomial time except for SAT-oracle calls. Correctness of the algorithm follows from Theorem 24.

One run induces an NP-query DAG  $G_q$ . By Lemma 28, the underlying undirected graph  $U(G_q)$  has constant treewidth (with constant depending only on  $k$ ). By [GR22, Lemma 10] (treewidth bounds separator number),  $U(G_q)$  has a constant separator number. Thus,  $G_q$  is an NP-query DAG of constant separator number (an NP-DAG<sub>1</sub> in [GR22] notation).

By [GR22, Thm. 2] (our Theorem 27), NP-DAG<sub>1</sub>  $\subseteq P^{NP[\log]}$ . Hence the oracle part of BOUNDEVIDENCEMC is computable in  $P^{NP[\log]}$ , and since the remaining work is deterministic polynomial time, the whole model-checking problem lies in

$$P^{NP[\log]} = \Theta_2^P.$$

□

### 3.3 Knowledge and Belief as primitive

The idea to define knowledge and belief based on evidence naturally leads to defining knowledge and belief based on the evidence modalities. This is the main approach by Baltag et al. [BGG25]. From a computational perspective it is also interesting to see what happens if we take knowledge and belief as primitives. We already have upper bounds from the definition through evidence modalities, but, potentially, we can get lower complexity classes for model checking with knowledge and belief as primitives (without evidence modalities).

In the multi-agent case the PSPACE bounds cleanly translates. This is because we can set the evidence to empty and then refer to [Su+09] which proves that the S5, two-agent case is in PSPACE. For the one-agent knowledge case, we can adjust the reduction we have used for unbounded evidence with one-agent to get a PSPACE bound.

In the belief case, the complexity drops notably. Instead of directly evaluating  $[\forall]\Diamond\Box$ , we can use an alternative maximality definition to achieve a  $\Theta_2^P$  lower bound without any restrictions on the models. Making use of BDD-properties, we show that model checking a single belief formula  $B\varphi$  with  $\varphi$  propositional is coNP-complete. Hence, inside an  $P$  algorithm with NP-oracle access, we can solve it by transforming the oracle answer. This allows us to apply a variation of the Algorithm 3 and conclude  $\Theta_2^P$  completeness. We start with the belief fragment as the proof directly builds on the last chapter.

#### 3.3.1 $\Theta_2^P$ -completeness for one-agent $\mathcal{L}_B$

We prove the  $\Theta_2^P$  completeness of the belief only fragment in two steps. First, we prove that model checking a single  $B\varphi$  with  $\varphi$  being propositional is coNP-complete. Having established coNP-completeness for symbolic model checking  $B\varphi$  with  $\varphi$  propositional, we have established the basis to prove  $\Theta_2^P$  completeness by reusing the algorithm we have designed for model checking the bounded evidence case. We have already noted that the algorithm does not use S4 specific properties at any point. The only conditions for it to be correct are that (1) the modal gadgets are equivalent to constant size circuits, (2) modalities with propositional scope can be evaluated by an NP-oracle and (3) that stacked modalities can be eliminated. All conditions are also fulfilled in the belief case *without* bounded evidence.

#### $B\phi$ is coNP-complete

We first show that model checking a formulas of the form  $B\varphi$  with  $\varphi$  being propositional is coNP-complete. The proof works by using the certificate definition of  $NP$  to show that the complement problem, "is there a maximal world that falsifies  $\varphi$ ", is an NP problem. Recall that we have

$$B\varphi := [\forall]\Diamond\Box\varphi.$$

In the topological evidence literature it standard that this corresponds to maximally refined open sets. We still give a short proof that this corresponds to truth in all  $\preceq$ -maximal worlds in the relevant partition cell:

**Lemma 30** (Maximality Belief). *Let  $o := v \cap O$ . Let  $\text{Max}_o(u)$  denote that  $u$  is  $\preceq$ -maximal among the*

legal worlds in the observation cell  $o$ . Then

$$\mathfrak{M}, v \models B\varphi \iff \forall u (\text{Max}_o(u) \Rightarrow \mathfrak{M}, u \models \varphi).$$

*Proof.* Let  $o := v \cap O$ . By definition,

$$B\varphi = [\forall]\Diamond\Box\varphi.$$

Thus  $\mathfrak{M}, v \models B\varphi$  iff for every legal world  $x$  with  $x \cap O = o$  there is a legal world  $y$  with  $x \preceq y$  and  $\mathfrak{M}, z \models \varphi$  for all legal  $z$  with  $y \preceq z$  and  $z \cap O = o$ .

Since  $E$  is finite, every legal world in the cell  $o$  has a maximal  $\preceq$ -extension in that cell. Thus the above is equivalent to  $\varphi$  being true in all  $\preceq$ -maximal legal worlds in the cell  $o$ , that is

$$\forall u (\text{Max}_o(u) \Rightarrow \mathfrak{M}, u \models \varphi).$$

□

Using the BDD representation of the state law  $\theta$ , we can test maximality efficiently. Fix a world  $u \subseteq V$ . Consider the propositional formula

$$M(u) := \theta \wedge \pi_O(u) \wedge \left( \bigwedge_{e \in u \cap E} e \right) \wedge \left( \bigvee_{e \in E \setminus u} e \right).$$

Here,

$$\bigwedge_{o \in u \cap O} o \wedge \bigwedge_{o \in O \setminus u} \neg o \wedge \bigwedge_{e \in u \cap E} e$$

fixes the partition and evidence atoms of  $u$ , while the final disjunction forces at least one additional evidence atom from  $E$  to be true. Thus  $M(u)$  is satisfiable iff  $u$  has a strict  $\preceq$ -extension within its partition cell, i.e., iff  $u$  is *not*  $\preceq$ -maximal in that cell.

We can represent this efficiently as a BDD in the following way:

Let  $\rho_u$  be the partial assignment that fixes the  $O$ -atoms to match the partition cell of  $u$  and fixes exactly the evidence atoms in  $u$  to true:

$$\rho_u(o) = \begin{cases} 1 & \text{if } o \in u \cap O, \\ 0 & \text{if } o \in O \setminus u, \end{cases} \quad \rho_u(e) = \begin{cases} 1 & \text{if } e \in u \cap E, \\ \text{unset} & \text{if } e \in E \setminus u. \end{cases}$$

Then the conjunction of literals that fixes the  $O$ -atoms and the evidence atoms of  $u$  is equivalently implemented by restricting (cofactoring) the BDD for  $\theta$  which is a polynomial time operation:

$$\theta \wedge \pi_O(u) \wedge \left( \bigwedge_{e \in u \cap E} e \right) \equiv \theta[\rho_u].$$

Hence we can define the maximality test BDD as

$$M(u) := \theta[\rho_u] \wedge \left( \bigvee_{e \in E \setminus u} e \right).$$

Since satisfiability/unsatisfiability of a BDD can be checked in time polynomial in the BDD size, this yields a polynomial-time maximality test. We will use this to define a certificate.

We fix the observation cell of the input world  $w$  by replacing  $\theta$  with  $\theta \wedge \pi_O(w)$ . We therefore write  $\text{Max}(u)$  as shorthand for maximality within this fixed cell.

Consider the complement of  $B\varphi$ :

$$\neg B\varphi \iff \exists u (\text{Max}(u) \wedge \mathfrak{M}, u \not\models \varphi).$$

This is an NP property via a standard certificate argument. As certificate, guess a world  $u \subseteq V$  and verify in polynomial time that:

1.  $u$  is a legal world:  $\theta(u) = 1$  (evaluate  $\theta$  on  $u$ ),
2.  $u$  is maximal:  $M(u)$  is unsatisfiable (BDD emptiness test),
3.  $u$  falsifies  $\varphi$ :  $u \not\models \varphi$  (propositional).

All three checks are polynomial in the input size (BDD sizes plus  $|\varphi|$ ).

It only remains to prove that the BDD-construction of  $M(u)$  works in polynomial time.

Computing  $\theta[\rho_u]$  from the BDD for  $\theta$  is a restriction (cofactoring) operation: for every variable assigned by  $\rho_u$  we follow the corresponding low/high edges and then reduce the resulting diagram. Bryant shows that restriction can be implemented in time linear in the size of the input BDD [Bry92, Sec. 3.2]. Hence  $\theta[\rho_u]$  is computable in time polynomial in  $|\theta|$  and  $|\rho_u| \leq |V|$ .

We now show that building the disjunction runs in polynomial time. Let  $k := |E \setminus w| \leq |E|$ . We construct a BDD  $D_w$  for the formula

$$D_w \equiv \bigvee_{e \in E \setminus w} e$$

under the same variable order  $\pi$ . This can be done by starting from the constant **0** and repeatedly applying disjunction with the corresponding variable-literal BDDs:

$$D_w := e_1 \vee e_2 \vee \dots \vee e_k.$$

Using Bryant's APPLY procedure for  $\vee$  [Bry92, Sec. 3.1], each step runs in time polynomial in the sizes of the operand BDDs. Moreover, the BDD for a disjunction of  $k$  variables has size  $\mathcal{O}(k)$  for any fixed ordering (it can be represented as a simple chain that tests the variables until one is true), so  $|D_w| = \mathcal{O}(k) = \mathcal{O}(|V|)$ .

Finally, we compute

$$\theta[\rho_u] \wedge D_u$$

using APPLY for  $\wedge$ . The running time is  $\mathcal{O}(|\theta[\rho_u]| \cdot |D_u|)$ , hence polynomial in  $|\theta|$  and  $|V|$  because  $|\theta[\rho_u]| \leq |\theta|$  and  $|D_u| = \mathcal{O}(|V|)$ .

Thus, the BDD for  $M(u)$  can be constructed in time polynomial in the input size. Combined with the polynomial-time satisfiability test for BDDs, this completes the certificate argument and yields model checking a single  $B\varphi$  with  $\varphi$  being propositional is in  $\text{coNP}$ .<sup>2</sup>

---

<sup>2</sup>It is important to note that not only the proof relies on the state law being a BDD, but that it is also a necessary condition for  $\text{coNP}$  membership. If the state law is not given as a BDD it is easy to show  $\Pi_2^P$ -hardness by reducing from a problem proved  $\Pi_2^P$ -complete in [EG93]. We translate the minimal model relation using evidence and then model check over all maximal models with the belief operator. This reduction relies on using an arbitrary propositional theory as state law. When the state law is required to be a BDD this does not work in polynomial time.

coNP-hardness for  $B\varphi$  is immediate: if we set the  $O = \emptyset$ ,  $E = \emptyset$  and  $\theta = \top$ , then

$$\mathfrak{M}, w \models B\varphi \iff \forall t (\mathfrak{M}, t \models \varphi),$$

and propositional validity/non-validity reduces to  $B\varphi$ .

### Adjusting the bounded-evidence algorithm to the belief fragment

In contrast to the bounded-evidence case, the modal-rooted formulas are not equivalent to block-circuits  $X \subseteq L$ , but collapse to Booleans  $\top, \perp$  because belief modalities are global over the maximality domain of the fixed observation cell. Note that we do not have bounded evidence anymore; our  $L \subseteq 2^{|E|}$  is *not* of constant size. Still the BDD representation of the state law allows us to solve the quantification over the maximal worlds with an coNP/NP-oracle. This collapses the belief modality to the  $\forall$ -case.

To prove  $\Theta_2^P$ -completeness, we can use a simplified version of Algorithm 3. The crucial observation here is that belief is invariant on the partition cell:

$$\mathfrak{M}, w \models B\varphi \iff \forall u (\text{Max}(u) \Rightarrow \mathfrak{M}, u \models \varphi).$$

Instead of saving circuits that are equivalent to a modal rooted formula in  $Val$ , we can now just save the partition global truth-values in the table (note again that in the model checking case we can just fix the partition cell for  $w$ ). To fill the table  $Val$ , we use the NP oracle

$$\text{MAXSAT}(\theta, \beta) \iff \exists u \subseteq V (u \models \theta \wedge \text{Max}(u) \wedge \mathfrak{M}, u \models \beta).$$

Belief elimination is much simpler than modal elimination in the bounded-evidence case: we only substitute the already computed Boolean values  $Val[c] \in \{\perp, \top\}$  for modal children  $c$ . This preserves truth as  $\top, \perp$  are invariant under the modalities. These observations justify correctness of EVALGADGET-BELIEF:

---

#### Algorithm 6: EVALGADGET-BELIEF

---

**Input:** gadget root  $m$ , map  $Val : \text{gadgets} \rightarrow \{\perp, \top\}$ , constraint  $\theta$

**Output:** truth value  $Val[m] \in \{\perp, \top\}$

```

1 foreach  $c \in \text{ModalChildren}(m)$  do
2    $b_c \leftarrow Val[c]$            // simplified modal elimination, we just cut the prefix.
3    $m \leftarrow m[c \mapsto b_c]$ 
4  $\beta \leftarrow \text{Form}(m)$ 
5  $r \leftarrow \text{root}(m)$ 
6 if  $r = \hat{B}$  then
7    $Val[m] \leftarrow \text{MAXSAT}(\theta, \beta)$ 
8 else if  $r = B$  then
9    $Val[m] \leftarrow \neg \text{MAXSAT}(\theta, \neg \beta)$ 
10 return  $Val[m]$ 

```

---

**Theorem 31** ( $\text{STEMC}_{\mathcal{L}_B}$  is  $\Theta_2^P$ -complete).

*Proof.* Now the only difference is that instead of  $2^k$  queries per gadget, we have exactly one oracle query per gadget. Hence the query graph has one node per gadget and edges induced by the dependency forest  $F$ . Its underlying undirected graph is therefore a forest, so it has treewidth at most 1. [GR22, Thm. 2] applies and gives us  $\Theta_2^P$ -membership.

For  $\Theta_2^P$ -hardness, we observe that the  $\mathcal{L}_M$ -case is a sub-case. If we set  $O = E = \emptyset$  the maximal worlds are the whole partition. Hence,  $\Theta_2^P$  hardness carries over.  $\square$

This treewidth=1 version of [GR22, Thm 2] is Gottlob's classical NP-Tree result [Got95].

### 3.3.2 PSPACE-completeness for one-agent $\mathcal{L}_K$

We close the complexity study by evaluating the complexity for the one-agent knowledge and belief case. We reuse the reduction of Theorem 16. The only difference is that universal steps will be written with  $K$  and existential steps with the dual of  $K$ . To justify this, we construct the model such that, for all  $w$  and all relevant  $i$ ,

$$\mathfrak{M}, w \models K(\delta_i \rightarrow \psi) \iff \mathfrak{M}, w \models \Box(\delta_i \rightarrow \psi), \quad \text{and} \quad \mathfrak{M}, w \models \langle K \rangle(\delta_i \wedge \psi) \iff \mathfrak{M}, w \models \Diamond(\delta_i \wedge \psi).$$

Then the proof from Theorem 16 applies.

Let  $\mathfrak{M}_\Phi$  be the one-agent model from Theorem 16. We extend it by a single fresh evidence atom  $d_{n+1}$ , set  $V' := V \cup \{d_{n+1}\}$ ,  $E' := V'$ ,  $O' := \emptyset$ , and strengthen the state law by the conjunct

$$d_{n+1} \rightarrow d_n.$$

Write  $\mathfrak{M}'_\Phi$  for the resulting model. Since still  $w \preceq t$  iff  $w \subseteq t$ , the  $\preceq$ -maximal (evidence-)worlds are exactly those containing  $d_{n+1}$ , and hence they satisfy  $d_{i+1}$  for every  $0 \leq i \leq n$ . Note that we defined the guards  $\delta_i$  such that we have for all  $0 \leq i \leq n$ ,

$$\delta_i := d_i \wedge \neg d_{i+1}.$$

Therefore every maximal world satisfies  $\neg \delta_i$ .

Note that we have the dualities

$$K\chi \equiv B\chi \wedge \Box\chi, \quad \langle K \rangle\chi := \neg K\neg\chi, \quad \text{so} \quad \langle K \rangle\chi \equiv \neg(B\neg\chi \wedge \Box\neg\chi) \equiv \neg B\neg\chi \vee \Diamond\chi.$$

By the alternative belief reading,  $B\eta$  holds iff  $\eta$  holds in all maximal evidence worlds. Since every maximal world satisfies  $\neg \delta_i$ , we have for all  $\psi$ :

$$(i) \quad B(\delta_i \rightarrow \psi) \text{ is true everywhere} \quad \text{and} \quad (ii) \quad B\neg(\delta_i \wedge \psi) \text{ is true everywhere.}$$

Hence, for all  $w$ ,

$$\mathfrak{M}'_\Phi, w \models K(\delta_i \rightarrow \psi) \iff \mathfrak{M}'_\Phi, w \models \Box(\delta_i \rightarrow \psi),$$

and using (ii),

$$\mathfrak{M}'_\Phi, w \models \langle K \rangle(\delta_i \wedge \psi) \iff \mathfrak{M}'_\Phi, w \models \Diamond(\delta_i \wedge \psi),$$

because  $\neg B \neg(\delta_i \wedge \psi)$  is everywhere false and the disjunction reduces to  $\Diamond$ .

Now, define  $\widehat{\varphi}_\Phi$  from  $\varphi_\Phi$  (as in Theorem 16) by replacing each universal clause  $\Box(\delta_i \rightarrow \cdot)$  by  $K(\delta_i \rightarrow \cdot)$  and each existential clause  $\Diamond(\delta_i \wedge \cdot)$  by  $\langle K \rangle(\delta_i \wedge \cdot)$ . By the equivalences above,  $\mathfrak{M}'_\Phi, \{d_0\} \models \widehat{\varphi}_\Phi$  iff  $\mathfrak{M}'_\Phi, \{d_0\} \models \varphi_\Phi$ , and the latter holds iff  $\Phi$  is true by Theorem 16. The extension adds one variable and a constant-size conjunct, so the reduction remains polynomial-time. Thus the one-agent  $K$ -language is PSPACE-hard. Membership follows directly as we can use the algorithm for the evidence language to solve the model checking problem. Hence, we can conclude PSPACE-completeness.

# Chapter 4

## Conclusion

### 4.1 Conclusion

In this work, we have mapped out most of the complexity landscape for symbolic model checking on topological evidence models. The (unrestricted) multi-agent cases have already been treated in the work on knowledge structures and DEL by Su [Su+09] and Behnke et al. [Beh+25]. It only remained to show that the whole language is indeed in PSPACE which required writing down a standard polynomial-time alternating Turing machine

The first novel result of this work is that the single-agent case is PSPACE-hard. We achieved this result by a reduction from TQBF in which we encoded the TQBF-tree in the symbolic model – a technique inspired by Ladner’s PSPACE-hardness proof for S4-SAT [Lad77]. We further adjusted this reduction to prove hardness for the languages that have no evidence modalities but contain the knowledge modality as primitive. Here, we showed that one can simulate the  $\Box$ -modality by trivializing the belief part in  $K\varphi = B\varphi \wedge \Box\varphi$ .

In the knowledge and belief setting, we further proved  $\Theta_2^P$ -completeness for the belief only fragment. This result is based on an efficient way to represent maximality in the BDD setting and Algorithm 3 we defined earlier for the bounded evidence case.

This algorithm allows us to prove  $\Theta_2^P$  model checking for a broad class of symbolic model checking problems. The key idea is to order the NP-oracle calls we need to evaluate the modal formula in a tree-like query structure. Then one can invoke [GR22, Thm. 2] to conclude that the original problem is in  $\Theta_2^P$ .

The algorithm applies in all symbolic model checking cases where (i) modalities over propositional formulas can be evaluated by an NP/CONP-oracle, (ii) modal-rooted formulas are equivalent to some finite circuit that allows for propositional representation, and (iii) modalities on these circuits can be efficiently eliminated. This already leads us to future work.

#### 4.1.1 Future work

First and foremost it would be interesting to properly define a class of symbolic model checking problems that can be proved  $\Theta_2^P$ -complete by this proof scheme. The central requirements are already listed above. Many adjacent cases like using  $\subsetneq$  instead of  $\subseteq$  to define relations on bounded evidence follow immediately, but as we have seen with belief the class should be quite a bit broader.

Further, it would be interesting to develop a parametrized version of the results achieved for

bounded evidence. We have seen that the separator number for the query graph is bounded by  $k = |E|$  and not the complete input size  $n$ . Gharibian et al. [GR22, Cor. 3] now seems to imply that there is an intermediate case between constant and non-constant evidence size: when we bound evidence as  $2^k \leq \log n$ , we land in quasi-polynomial-time with  $\log^2$  adaptive oracle calls. By standard complexity assumptions this is between  $\Theta_2^P$  and PSPACE and gives us an intermediate case worth investigating.

Regarding the Topo-E-Model language, we have left one interesting fragment open: the multi-agent fragment with bounded observable/evidence size, that is  $\sum_{i \in I} |O_i| \leq k$  and  $\sum_{i \in I} |E_i| \leq k$  for some constant. We conjecture  $\Theta_2^P$ -completeness for this case. One should be able to prove this using the proof scheme we developed for bounded evidence. In this case, one could define a finite accessibility relation based on the overlap between agents observable partitions. Given a bounded size of the observables for all agents, this could give a relation of bounded size that can be evaluated efficiently.

From a more practical perspective, it would be interesting to implement the alternative belief definition. Instead of calculating belief operators by evaluating the modality stack  $\forall \Diamond \Box$  as in the current implementation by Gomes [San25, Sec. 6.4.1], one could calculate a maximality BDD. This would have two advantages: firstly, one would solve a CONP problem instead of the stack of modalities. Secondly, it allows for easy precomputation of the maximality BDD. The heavy computation would be in computing the maximality BDD. Checking a belief formula afterwards would be cheap. This framework would allow for efficient caching and could improve the performance of the model checker implemented by Gomes.

# Bibliography

- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009 (cited on page 16).
- [Beh+25] Gregor Behnke, Malvin Gattinger, Avijeet Ghosh, and Haitian Wang. “Comparing State-Representations for DEL Model Checking”. In: *Proceedings of the 21st Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2025)*. EPTCS. To appear. 2025. <https://ccc.cs.uni-duesseldorf.de/tark-2025/accepted-papers/> (cited on pages 3, 42).
- [BGG25] Alexandru Baltag, Malvin Gattinger, and Djanira Gomes. *Virtual Group Knowledge and Group Belief in Topological Evidence Models (Extended Version)*. 2025. <https://arxiv.org/abs/2509.00184> (cited on pages 6, 36).
- [Bod+22] Hans L. Bodlaender, Josse van Dobben de Bruyn, Dion Gijswijt, and Harry Smit. “Constructing tree decompositions of graphs with bounded gonality”. In: *Journal of Combinatorial Optimization* 44 (2022), pages 2681–2699. <https://doi.org/10.1007/s10878-021-00762-w> (cited on page 6).
- [Bol+99] Beate Bollig, Martin Löbbling, Martin Sauerhoff, and Ingo Wegener. “On the Complexity of the Hidden Weighted Bit Function for Various BDD Models”. In: *Informatique théorique et applications / Theoretical Informatics and Applications (RAIRO-ITA)* 33.2 (1999), pages 103–115. [https://www.numdam.org/item/ITA\\_1999\\_\\_33\\_2\\_103\\_0/](https://www.numdam.org/item/ITA_1999__33_2_103_0/) (cited on page 11).
- [BP11] Johan van Benthem and Eric Pacuit. “Dynamic Logics of Evidence-Based Beliefs”. In: *Studia Logica* 99 (2011), pages 61–92. <https://doi.org/10.1007/s11225-011-9347-x> (cited on page 3).
- [Bry92] Randal E. Bryant. “Symbolic Boolean manipulation with ordered binary-decision diagrams”. In: *ACM Comput. Surv.* 24.3 (Sept. 1992), pages 293–318. ISSN: 0360-0300. <https://doi.org/10.1145/136035.136043> (cited on pages 5, 6, 38).
- [Bur+92] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. “Symbolic Model Checking:  $10^{20}$  States and Beyond”. In: *Information and Computation* 98.2 (1992), pages 142–170. [https://doi.org/10.1016/0890-5401\(92\)90017-A](https://doi.org/10.1016/0890-5401(92)90017-A) (cited on page 4).
- [CH96] M. J. Cresswell and G. E. Hughes. *A New Introduction to Modal Logic*. New York: Routledge, 1996 (cited on pages 22, 23).

- [EG93] Thomas Eiter and Georg Gottlob. “Propositional circumscription and extended closed-world reasoning are P2-complete”. In: *Theoretical Computer Science* 114.2 (1993), pages 231–245. ISSN: 0304-3975. <https://www.sciencedirect.com/science/article/pii/0304397593900733> (cited on page 38).
- [FPV05] Andrea Ferrara, Guoqiang Pan, and Moshe Y. Vardi. “Treewidth in Verification: Local vs. Global”. In: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2005)*. Volume 3835. Lecture Notes in Computer Science. Springer, 2005, pages 489–503. [https://doi.org/10.1007/11591191\\_34](https://doi.org/10.1007/11591191_34) (cited on pages 6, 21).
- [Gat18] Malvin Gattinger. “New Directions in Model Checking Dynamic Epistemic Logic”. PhD thesis. University of Amsterdam, 2018. <https://malv.in/phdthesis> (cited on pages 3, 4, 10).
- [Got95] Georg Gottlob. “NP trees and Carnap’s modal logic”. In: *J. ACM* 42.2 (Mar. 1995), pages 421–457. ISSN: 0004-5411. <https://doi.org/10.1145/201019.201031> (cited on pages 25, 33, 40).
- [GR22] Sevag Gharibian and Dorian Rudolph. “On Polynomially Many Queries to NP or QMA Oracles”. In: *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Edited by Mark Braverman. Volume 215. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 75:1–75:27. <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.75> (cited on pages 13, 25, 34, 35, 40, 42, 43).
- [Lad77] Richard E. Ladner. “The Computational Complexity of Provability in Systems of Modal Propositional Logic”. In: *SIAM Journal on Computing* 6.3 (Sept. 1977), pages 467–480. <https://doi.org/10.1137/0206033> (cited on pages 16, 42).
- [MP92] Lawrence S. Moss and Rohit Parikh. “Topological Reasoning and The Logic of Knowledge”. In: *Proceedings of the 4th Conference on Theoretical Aspects of Reasoning about Knowledge, Monterey, CA, USA, March 1992*. Edited by Yoram Moses. Morgan Kaufmann, 1992, pages 95–105 (cited on page 3).
- [Özg17] Aybüke Özgün. “Evidence in Epistemic Logic: A Topological Perspective”. PhD thesis. University of Amsterdam, 2017. <https://eprints.illc.uva.nl/id/eprint/2147/> (cited on pages 3, 6).
- [San25] Djanira dos Santos Gomes. “Virtual Group Knowledge on Topological Evidence Models”. Master’s thesis. Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 2025. <https://msclogic.illc.uva.nl/theses/archive/publication/5466/Virtual-Group-Knowledge-on-Topological-Evidence-Models> (cited on pages 3, 4, 6, 7, 9, 10, 43).
- [Sin05] Carsten Sinz. “Towards an Optimal CNF Encoding of Boolean Cardinality Constraints”. In: *Principles and Practice of Constraint Programming - CP 2005*. Edited by Peter van Beek. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pages 827–831 (cited on pages 23, 24).
- [Su+09] Kaile Su, Abdul Sattar, Guanfeng Lv, and Yan Zhang. “Variable Forgetting in Reasoning about Knowledge”. In: *Journal of Artificial Intelligence Research* 35 (Aug. 2009). <https://doi.org/10.1613/jair.2750> (cited on pages 3, 12, 14, 36, 42).