Bundled Fragments of First-Order Modal Logic: (Un)Decidability

Anantha Padmanabha
Institute of Mathematical Sciences, HBNI, Chennai, India
ananthap@imsc.res.in
https://orcid.org/0000-0002-4265-5772

R Ramanujam
Institute of Mathematical Sciences, HBNI, Chennai, India
jam@imsc.res.in

Yanjing Wang
Department of Philosophy, Peking University, Beijing, China
y.wang@pku.edu.cn

Abstract

Quantified modal logic is notorious for being undecidable, with very few known decidable fragments such as the monodic ones. For instance, even the two-variable fragment over unary predicates is undecidable. In this paper, we study a particular fragment, namely the *bundled* fragment, where a first-order quantifier is always followed by a modality when occurring in the formula, inspired by the proposal of [15] in the context of non-standard epistemic logics of know-what, know-how, know-why, and so on.

As always with quantified modal logics, it makes a significant difference whether the domain stays the same across possible worlds. In particular, we show that the predicate logic with the bundle $\forall\Box$ alone is undecidable over constant domain interpretations, even with only monadic predicates, whereas having the $\exists\Box$ bundle instead gives us a decidable logic. On the other hand, over increasing domain interpretations, we get decidability with both $\forall\Box$ and $\exists\Box$ bundles with unrestricted predicates, where we obtain tableau based procedures that run in PSPACE. We further show that the $\exists\Box$ bundle cannot distinguish between constant domain and variable domain interpretations.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics

Keywords and phrases First-order modal logic, decidability, bundled fragments

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2018.43

Acknowledgements We thank the reviewers for the insightful and helpful comments.

1 Introduction

Propositional modal logics have been extensively used to reason about labelled transition systems in computer science. These have led to the advent of temporal logics which have been very successful in the formal specification and verification of a wide range of systems. While these have been best used in the context of finite state reactive systems (over infinite behaviours), in the last couple of decades, such logics have been developed for infinite state systems as well ([1]). Finding decidable logics with reasonable complexity over infinite state systems continues to be a challenge.

38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018).
Editors: Sumit Ganguly and Paritosh Pandya; Article No. 43; pp. 43:1–43:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 x, y refers to the two-variable fragment, P^1 refers to unary predicates. GF is the guarded fragment. $\Box_i$ is multi-modal logic. $\Box_i(x)$ refers to having only 1 free variable inside the scope of modalities (monodic fragment). p refers to propositions and / indicates having one of them.

Language	Decidability	Ref
P^1	undecidable	[10]
x, y, p, P^1	undecidable	[9, 6]
x, y, $\Box_i$, single P^1	undecidable	[11]
single x	decidable	[12, 4]
$x, y/P^1/\text{GF}$, $\Box_i(x)$	decidable	[17]

A natural candidate to describe systems with unbounded data is First Order Logic (FO), and it has been extensively used not only in reasoning about mathematical structures, but also about databases and knowledge representation systems. When we wish to describe data updates in such systems, we have labelled transition systems where each state carries information on data, thus making them infinite state systems. It is then easy to specify transitional properties of such systems in First Order Modal Logic (FOML).

However FOML is infamously hard to handle technically: usually you lose good properties of first-order logic and modal propositional logic when putting them together. (FOML is also the theatre in which numerous philosophical controversies have been played out.) On the one hand, the decidable fragments of first-order logic have been well mapped out during the last few decades ([3]). On the other hand, we have a thorough understanding of the robust decidability of propositional modal logics [13]. However, when it comes to finding decidable fragments of FOML, the situation seems quite hopeless: even the *two-variable fragment with one single monadic predicate* is (robustly) undecidable over almost all useful model classes [11].

On the positive side, certain guarded fragments of FOML that are decidable [17]. One promising approach has come from the study of the so-called *monodic fragment*, which requires that there be at most one free variable in the scope of any modal subformula. Combining the monodic restriction with a decidable fragment of FO we often obtain decidable fragments of FOML[17, 2], as Table 1 shows.

The reason behind this sad tale is not far to seek: the addition of $\Box$ gives implicitly an extra quantifier, over a fresh variable. Thus if we consider the two-variable fragment of FOML, with only unary predicates in the syntax, we can use $\Box$ to code up binary relations and we ride out of the two-variable fragment as well as the monadic fragment of FO. The monodic fragment restricts the use of free variables inside the scope of $\Box$ significantly to get decidability.

It is then natural to ask: apart from variable restrictions, is there some other way to obtain syntactic fragments of FOML that are yet decidable?

One answer came, perhaps surprisingly, from epistemic logic. In recent years, interest has grown in studying epistemic logics of knowing-how, knowing-why, knowing-what, and so on (see [16] for a survey). As observed there most of the new epistemic operators essentially share a unified *de re* semantic schema of $\exists x \Box$ where $\Box$ is an epistemic modality ($\mathsf{B}^{\exists\Box}$-FOML). For instance, $\exists x \Box \varphi$ may mean that there exists a mechanism which you know such that executing it will make sure that you end in a φ state [14]. Such reasoning leads to the proposal in [15] of a new fragment of FOML by packing $\exists$ and $\Box$ into a *bundle* modality, but without any restriction on predicates or the occurrences of variables. Formally, $\mathsf{B}^{\exists\Box}$-FOML fragment is given by the syntax:

$$\varphi ::= P\overline{x} \mid \neg\varphi \mid (\varphi \land \varphi) \mid \exists x \Box \varphi$$

Note that in this language, quantifiers have to always come with modalities. Such a language may seem weak but it already suffices to say many interesting things.

The following examples describe a database model that comes with a binary relation $R(x, y)$ to mean that x "is dominated by" y, and the states describe the possible updates of the database before/after updates.

- $\exists x\Box\ \neg\exists y\Box\ (R(x, y))$: There is a king element such that after any update, no element is sure to dominate it later.
- $\forall x\Diamond\ \exists y\Box\ (R(x, y))$: Every element can be updated in such a way that another can necessarily dominate it ($\forall x\Diamond$ is the dual of $\exists x\Box$).
- $\exists x\Box\ \big(\exists y\Box\ (R(x, y) \wedge \exists z\Box\ (R(x, z) \wedge R(y, z))\big)$: There is an element x that is dominated by some element y after any possible update and further, there exists z which will always dominate both x and y in any possible update from there on.

It may be noted that the domain does not need to be fixed uniformly at all states to interpret these formulas. For instance, when we consider the first formula above, we refer to some "active" element x present at the current state. It is necessary that x continues to be active at successor states where we compare it against other elements, but there could be new elements in the successor states. When we define the formal semantics in the next section, it will be clear that these (and other similar) formulas may be interpreted over constant domain or increasing domain models uniformly.

It is shown [15] that the $\mathsf{B}^{\exists\Box}$-FOML fragment with arbitrary predicates is in fact PSPACE-complete over FOML. Essentially, the idea is based on the 'secret of success' of modal logic: *guard the quantifiers*, now with a modality. On the other hand, the same fragment is undecidable over equivalence models, and this can be shown by coding first-order sentences in this language using the symmetry property of the accessibility relation.

There are curious features to observe in this tale of (partial) success. The fragment in [15] includes the $\exists\Box$ bundle but not its companion $\forall\Box$ bundle, and considers *only* increasing domain models. The latter observation is particularly interesting when we notice that equivalence models, where the fragment becomes undecidable, force constant domain semantics.

The last distinction is familiar to first-order modal logicians, but might come across as a big fuss to others. Since FOML extends FO, the models contain a first order structure at each state. Then it makes a significant difference whether we work with a single data domain fixed for the entire model, or whether this can vary across transitions (updates). In the latter case, each possible world has its own domain, and quantification extends only over objects that exist in the current world.

Given such subtlety, it is instructive to consider more general bundled fragments of FOML, including both $\exists\Box$ and $\forall\Box$ as the natural first step, and study them over constant domain as well as varying domain models. This is precisely the project undertaken in this paper, and the results are summarized in Table 2. In this paper, the only varying domains we consider are *increasing* ones, whereby the data domain may change across a transition but can only increase monotonically.

As we can see, the $\exists\Box$ bundle behaves better computationally than the $\forall\Box$ bundle. For $\forall\Box$, even the monadic fragment is undecidable over constant domain models: we can encode in this language, qua satisfiability, any first-order logic sentence with binary predicates by exploiting the power of $\forall\Box$.

On the other hand, we can actually give a tableau method for the $\exists\Box$ and $\forall\Box$ fragment together, similar to the tableau in [15], for increasing domain models. The crucial observation

Table 2 Satisfiability problem classification for Bundled FOML fragment, P refers to predicates of arbitrary arity and P^1 refers to monadic predicates. Models are either constant domain or increasing domain.

Language	Model	Decidability	Remark
$\forall\Box$, P^1	Constant	undecidable	
$\exists\Box$, P	Constant	decidable	PSPACE-complete
$\exists\Box, \forall\Box$, P	Increasing	decidable	PSPACE-complete

is that such models allow us to manufacture new witnesses for $\exists x\Box$ and $\exists x\Diamond$ formulas on the fly, giving considerable freedom in model construction, which is not available in constant domain models.

Indeed, the well-behavedness of the $\exists\Box$ bundle is further attested to by the fact that it is decidable over constant domain models as well. So constant domain is not the culprit for undecidability of this fragment over equivalence models. In fact, we show that the $\exists\Box$ bundle cannot distinguish increasing domain models and constant domain models.

The paper is structured as follows. After formal definitions of bundled fragments, we present undecidability results for unary predicate $\forall\Box$ fragment over FOML with constant domain semantics and then move on to tableaux procedures for the decidable fragments. We then show that the validities of $\exists\Box$ over increasing domain are exactly the same as its validities over constant domain models, and end the paper with another look at mapping the terrain of these fragments.

2 Bundled fragment of First order modal logic

Let Var be a countable set of variables, and $\mathbf{P}$ be a countable set of predicate symbols, with $\mathbf{P}_n \subseteq \mathbf{P}$ denoting the set of all predicate symbols of arity n. We use $\overline{x}$ to denote a finite sequence of variables in Var. We only consider the "pure" first order unimodal logic: that is, the vocabulary is restricted to Var (no equality and no constants and no function symbols).

▶ **Definition 1.** Given Var and $\mathbf{P}$, the bundled fragment of FOML denoted by B-FOML is defined as follows:

$$\varphi ::= P\overline{x} \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \exists x\Box\varphi \mid \forall x\Box\varphi$$

where $x \in$ Var, $P \in \mathbf{P}$. We denote the fragment B$^{\exists\Box}$-FOML to be the formulas which contains only $\exists\Box$(and its dual $\forall\Diamond$) formulas and B$^{\forall\Box}$-FOML which contains only $\forall\Box$ (and its dual $\exists\Diamond$) formulas.

$\top, \bot, \vee, \rightarrow$ are defined in the standard way. $\forall x\Diamond\varphi = \neg\exists x\Box\neg\varphi$ is the dual of $\exists x\Box\varphi$, and $\exists x\Diamond\varphi = \neg\forall x\Box\neg\varphi$ is the dual of $\forall x\Box\varphi$. With both bundles we can say, that every element is guaranteed an update such that some element is updatable to dominate it: $\forall x\Box\, \exists y\Box\, (R(x,y))$.

The *free* and *bound* occurrences of variables are defined as in first-order logic, by viewing $\exists x\Box$ and $\exists x\Diamond$ as quantifiers. We denote Fv(φ) as the set of free variables of φ. We write $\varphi(\overline{x})$ if all the free variables in φ are included in $\overline{x}$. Given a B-FOML formula φ and $x, y \in$ Var, we write $\varphi[y/x]$ for the formula obtained by replacing every free occurrence of x by y. A formula is said to be a *sentence* if it contains no free variables. As we will see later, $\exists x\Box\varphi$ is equivalent to $\Box\varphi$ if x is not free in φ. Therefore B-FOML is indeed an extension of modal logic.

The semantics presented below is the standard *increasing domain* semantics of FOML.

▶ **Definition 2.** An (increasing domain) model $\mathcal{M}$ for B-FOML is a tuple (W, D, δ, R, ρ) where, W is a non-empty set of worlds, D is a non-empty domain, $R \subseteq (W \times W)$, $\delta : W \to 2^D$ assigns to each $w \in W$ a *non-empty* local domain s.t. wRv implies $\delta(w) \subseteq \delta(v)$ for any $w, v \in W$, and [1] $\rho : (W \times \mathbf{P}) \to \bigcup_{n \in \omega} 2^{D^n}$ such that ρ assigns to each n-ary predicate on each world an n-ary relation on D.

Given a model $\mathcal{M}$, we use $W^{\mathcal{M}}, D^{\mathcal{M}}, \delta^{\mathcal{M}}, \rho^{\mathcal{M}}$ to denote its corresponding components. We often write D_w for $\delta^{\mathcal{M}}(w)$. A *constant domain* model is one where $D_w = D^{\mathcal{M}}$ for any $w \in W^{\mathcal{M}}$. Note that constant domain models are special cases of increasing domain models. A *finite model* is one with both $W^{\mathcal{M}}$ finite and $D^{\mathcal{M}}$ finite.

▶ **Definition 3.** Consider a model $\mathcal{M} = (W, D, \delta, R, \rho)$, $w \in W$. To interpret free variables, we also need a variable assignment $\sigma : \mathsf{Var} \to D$. Given $\mathcal{M} = (W, D, \delta, R, \rho)$, $w \in W$, and an assignment σ, define $\mathcal{M}, w, \sigma \vDash \varphi$ inductively as follows:

$$\boxed{\begin{array}{lcl}
\mathcal{M}, w, \sigma \vDash P(x_1 \cdots x_n) & \Leftrightarrow & (\sigma(x_1), \cdots, \sigma(x_n)) \in \rho(P, w) \\
\mathcal{M}, w, \sigma \vDash \neg\varphi & \Leftrightarrow & \mathcal{M}, w, \sigma \nvDash \varphi \\
\mathcal{M}, w, \sigma \vDash (\varphi \wedge \psi) & \Leftrightarrow & \mathcal{M}, w, \sigma \vDash \varphi \text{ and } \mathcal{M}, w, \sigma \vDash \psi \\
\mathcal{M}, w, \sigma \vDash \exists x \Box \varphi & \Leftrightarrow & \text{there is some } d \in \delta(w) \text{ such that} \\
 & & \mathcal{M}, v, \sigma[x \mapsto d] \vDash \varphi \text{ for all } v \text{ s.t. } wRv \\
\mathcal{M}, w, \sigma \vDash \exists x \Diamond \varphi & \Leftrightarrow & \text{there is some } d \in \delta(w) \text{ and some } v \in W \\
 & & \text{ such that } wRv \text{ and } \mathcal{M}, v, \sigma[x \mapsto d] \vDash \varphi
\end{array}}$$

where $\sigma[x \mapsto d]$ denotes an assignment that is the same as σ except for mapping x to d.

Note that the standard $\Box\alpha(\Diamond\alpha)$ of FOML can be expressed in this logic as $\exists x \Box \alpha(\exists x \Diamond \alpha)$ where x does not occur in α.

In general, when considering the truth of φ in a model, it suffices to consider $\sigma : \mathsf{Fv}(\varphi) \to D$, assignment restricted to the free variables occurring free in φ. When $\mathsf{Fv}(\varphi) = \{x_1, \ldots, x_n\}$ and $\{d_1, \ldots, d_n\} \subseteq D$, We write $\mathcal{M}, w \vDash \varphi[\overline{d}]$ to denote $\mathcal{M}, w, \sigma \vDash \varphi(\overline{x})$ for any σ such that for all $i \leq n$ we have $\sigma(x_i) = d_i$. Finally, when φ is a sentence, we can simply write $\mathcal{M}, w \models \varphi$.

Call σ *relevant* at $w \in W$ if $\sigma(x) \in \delta^{\mathcal{M}}(w)$ for all $x \in \mathsf{Var}$. The increasing domain condition ensures that whenever σ is relevant at w and we have wRv, then σ is relevant at v as well. (In a constant domain model, every assignment σ is relevant at all the worlds.) We say φ is *valid*, if φ is true on any $\mathcal{M}, w$ w.r.t. any σ relevant at w. φ is *satisfiable* if $\neg\varphi$ is not valid. [2]

3 Undecidability results

In this section we prove that the satisfiability problem for the $\mathsf{B}^{\forall\Box}$-FOML fragment over the class of constant domain models is undecidable even when the atomic predicates are restricted to be unary.

Kripke[10] showed that full FOML with constant domain semantics is undecidable even when the atomic predicates are only unary. Gabbay and Shehtman [6] showed that 2-variable Monadic FOML with propositions is undecidable. Kontchakov et al [9] showed that

[1] Note that we do not impose the restriction $\rho(w, P) \subseteq [\delta(w)]^n$ where arity of P is n, since it is not needed for our technical development. For more details about this relaxation, refer Hughes and Creswell [8].

[2] Note that the classical first-order principle *dictum de omne*: $\forall x \psi \to \psi[y/x]$ is not expressible in our language, but validity over relevant assignments gives us classical expressible analogues.

propositions can be eliminated. We take another step in this journey. We show that over constant domain models, the satisfiability problem for $\mathsf{B}^{\forall\Box}$-FOML fragment is undecidable over unary predicates.

Consider $\mathsf{FO}(R)$, the first order logic with only variables as terms and no equality, and the single binary predicate R. We know that $\mathsf{FO}(R)$ satisfiability problem is undecidable [7]. To translate $\mathsf{FO}(R)$ sentences to $\mathsf{B}^{\forall\Box}$-FOML formulas, we use two unary predicate symbols P, Q in the latter. The main idea is that the atomic formula $R(x, y)$ is coded up as the $\mathsf{B}^{\forall\Box}$-FOML formula $\exists z \Diamond\big(P(x) \wedge Q(y)\big)$, where z is a new variable, distinct from x and y.[3]

For any *quantifier-free* $\mathsf{FO}(R)$ formula β, we define the translation of β to $\mathsf{B}^{\forall\Box}$-FOML formula φ_β inductively as follows.

- $Tr(R(x, y)) := \exists z \Diamond\big(P(x) \wedge Q(y)\big)$, where z is distinct from x and y.
- $Tr(\neg\beta) := \neg Tr(\beta)$ and $Tr(\beta_1 \wedge \beta_2) := Tr(\beta_1) \wedge Tr(\beta_2)$.

Note that a quantifier-free FO formula is translated to a $\mathsf{B}^{\forall\Box}$-FOMLformula with modal (quantifier) depth 1. Now consider an $\mathsf{FO}(R)$ sentence α (having no free variables) presented in prenex form: $Q_1x_1\ Q_2x_2 \cdots Q_nx_n(\beta)$ where β is quantifier-free. We define

$$\psi_\alpha := Q_1x_1\Delta_1\ Q_2x_2\Delta_2\ \cdots Q_nx_n\Delta_n\ (Tr(\beta))$$

where $Q_ix_i\Delta_i := \exists x_i\Diamond$ if $Q_i = \exists$ and $Q_ix_i\Delta_i := \forall x_i\Box$ if $Q_i = \forall$.

We claim that satisfiability is preserved over this translation with a few additional formulas. Ideally, we want α to be satisfiable iff ψ_α is satisfiable. However, the translated formula might be satisfiable simply because some $Q_i := \forall$ and there are no successors for the worlds at depth i and thus the corresponding subformula translation $\forall x_i\Box\psi'$ trivially holds. To avoid this, we use formula λ_n which ensures that for all $i \leq n$ and every world at depth i, there is at least one successor: $\lambda_n := \bigwedge\limits_{j=0}^{n} (\forall z\Box)^j(\exists z\Diamond\top)$

Finally to ensure that $\exists z\Diamond(P(x) \wedge Q(y))$ is evaluated uniformly at the "tail" worlds, we have: $\gamma_n := \forall z_1\Box\ \forall z_2\Box\ \big((\exists z\Diamond)^n\ (\exists z\Diamond\ (P(z_1) \wedge Q(z_2)) \rightarrow (\forall z\Box)^n(\exists z\Diamond(P(z_1) \wedge Q(z_2))\big)$ where z_1, z_2 and z do not appear in α.

Suppose $\mathcal{M}, u \models \gamma_n$ then notice that for any world w at a path length 2 from u, if there is one world at distance n starting from w where $\Diamond(P(z_1) \wedge Q(z_2))$ holds, then $\Diamond(P(z_1) \wedge Q(z_2))$ holds at all worlds at a distance n starting from w. Notice that we use two dummy variables z_1 and z_2 in γ_n. Hence to match the modal depths of the translated formulas, we need to append two $\Box's$ to ψ_α and we need to use λ_{n+2} instead of λ_n. Thus, the complete translation is given by:

▶ **Definition 4.** Given a $\mathsf{FO}(R)$ sentence $\alpha := Q_1x_1Q_2x_2 \cdots Q_nx_n\beta$ in prenex normal form, the translated $\mathsf{B}^{\forall\Box}$-FOML formula φ_α is given by: $\varphi_\alpha := (\forall z\Box)^2(\psi_\alpha) \wedge \lambda_{n+2} \wedge \gamma_n$ where z does not occur in α.

Note that for any $\mathsf{FO}(R)$ sentence α of quantifier depth n, we get a translated formula φ_α of modal (quantifier) depth $n + 3$.

Before stating the theorem, we define some useful notation.

▶ **Definition 5.** For any $\mathsf{FO}(R)$ sentence $\alpha := Q_1x_1Q_2x_2 \cdots Q_nx_n\beta$ in the prenex normal form with β being quantifier-free, we define the following:

[3] This is similar to the approach used by Kripke [10], specialized to the $\mathsf{B}^{\forall\Box}$-FOML fragment.

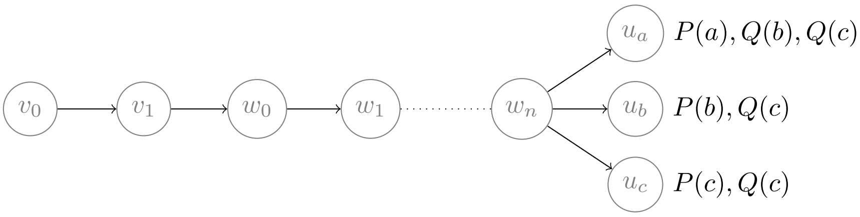

Figure 1 The translated model for (D, I) where $D = \{a, b, c\}$ and $I = \{(a, b),\ (a, c),\ (b, c),\ (c, c)\}$. For any sentence $\alpha \in FO(R)$ of quantifier depth n, $(D, I) \models \alpha$ iff $M, v_0 \models \varphi_\alpha$.

- For all $1 \le i \le n$ let $x_1 \cdots x_i$ be denoted by $\overline{x^i}$ and the vector $[d_1, d_2 \cdots d_i]$ be denoted by $\overline{d^i}$ where every $d_j \in D$.
- Let $[\overline{x^i} \mapsto \overline{d^i}]$ denote the interpretation where $\sigma(x_j) = d_j$.
- For $0 \le i < n$, let $\alpha[i] = Q_{i+1}x_{i+1} \cdots Q_n x_n \beta$ and $\psi_\alpha[i] = Q_{i+1}x_{i+1}\Delta_{i+1} \cdots Q_n x_n \Delta_n(\varphi_\beta)$ be the corresponding translated formula. Also, let $\alpha[n] = \beta$ and $\psi_\alpha[n] = Tr(\beta)$.

▶ **Theorem 6.** *For any* $\mathsf{FO}(R)$ *sentence* $\alpha := Q_1x_1Q_2x_2 \cdots Q_nx_n\beta$ *in prenex normal form,* α *is satisfiable iff* φ_α *is constant domain satisfiable.*

Proof. Let $\alpha := Q_1x_1 \cdots Q_nx_n\beta$, where β is quantifier-free. To prove $(\Rightarrow)$, assume that α is satisfiable. Let D be some domain such that $(D, I) \models \alpha$ where $I \subseteq (D \times D)$ is the interpretation for R. We use the same D as the domain and construct a FOML model. Define $\mathcal{M} = (W, R, D, \delta, \rho)$ where:

$W = \{v_0, v_1\} \cup \{w_i \mid 0 \le i \le n\} \cup \{u_d \mid d \in D\}$.
$R = \{(v_0, v_1), (v_1, w_0)\} \cup \{(w_i, w_{i+1}) \mid 0 \le i < n\} \cup \{(w_n, u_d) \mid u_d \in W\}$.
$\delta(u) = D$ for all $u \in W$.
For all $i \in \{0, 1\}$ and $0 \le j \le n$ and $v_i, w_j \in W$ define $\rho(v_i, P) = \rho(v_i, Q) = \rho(w_j, P) = \rho(w_j, Q) = \emptyset$ and for all $u_d \in W$, $\rho(u_d, P) = \{d\}$ and $\rho(u_d, Q) = \{c \mid (d, c) \in I\}$.

Note that $\mathcal{M}$ is a constant domain model. $\mathcal{M}$ is illustrated in Figure 1 for one such translation. Note that $\mathcal{M}$ has exactly one path of length $n + 2$ starting from v_0 which ends at w_n. Hence, $\mathcal{M}, v_1 \models \lambda_{n+2} \wedge \gamma_n$.

Finally, we claim that $\mathcal{M}, v_0 \models (\forall z\Box)^2\psi_\alpha$ which completes the proof of the forward direction. Again, since $v_0 \to v_1 \to w_0$ is the only path of length 2 starting from v_0, it is enough to verify that $\mathcal{M}, w_0 \models \psi_\alpha$. We set up an induction to prove this.

Claim. For all $0 \le i \le n$, $w_i \in W$, for all vectors $\overline{d^i} \in D^i$ of length i, we have $D, I, [\overline{x^i} \mapsto \overline{d^i}] \models \alpha[i]$ iff $\mathcal{M}, w_i, [\overline{x^i} \mapsto \overline{d^i}] \models \psi_\alpha[i]$.

The proof is by reverse induction on i. The base case, when $i = n$, we have $\alpha[n] = \beta$. Now we induct on the structure of β, to prove the claim. In the base case we have $R(x_i, x_j)$. By definition of ρ, if $(a, b) \in I$ then $\mathcal{M}, u_a, [x_i \to a, x_j \to b] \models (P(x) \wedge Q(y))$ and hence $\mathcal{M}, w_n, [x_i \to a, x_j \to b] \models \exists z\Diamond(P(x_1) \wedge Q(x_2))$. On the other hand if $\mathcal{M}, w_n, [x_i \to a, x_j \to b] \models \exists z\Diamond(P(x_1) \wedge Q(x_2))$ then since $\mathcal{M}, u_a \not\models P(b)$ for all $b \neq a$, it has to be the case that $\mathcal{M}, u_a, [x_i \to a, x_j \to b] \models (P(x) \wedge Q(y))$ and thus $(a, b) \in I$. The $\neg$ and $\wedge$ cases are standard.

For the induction step, we need to consider formulas $\alpha[i-1]$ and $\psi_\alpha[i-1]$ and the world w_{i-1}. Now $\alpha[i-1]$ is either $\exists x_i\alpha[i]$ or $\forall x_i\alpha[i]$.

For the case when $\alpha[i-1]$ is $\exists x_i\alpha[i]$ the corresponding $\psi_\alpha[i-1]$ is $\exists x_i\Diamond(\psi_\alpha[i])$. We have $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i\alpha[i]$ iff there is some $c \in D$ such that

$D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \alpha[i]$ iff (by induction hypothesis)
$\mathcal{M}, w_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \psi_\alpha[i]$ iff
$\mathcal{M}, w_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i \Diamond \psi_\alpha[i]$, as required.

For the case when $\alpha[i-1]$ is $\forall x_i \alpha[i]$, we have $\psi_\alpha[i-1] = \forall x_i \Box \psi_\alpha[i]$. Now,
$D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \alpha[i]$ iff
for all $c \in D$ we have $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \alpha[i]$
iff (by induction hypothesis) for all $c \in D$ we have
$\mathcal{M}, w_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \psi_\alpha[i]$ iff
$\mathcal{M}, w_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \Box \psi_\alpha[i]$ (since w_i is the only successor of w_{i-1}).
This completes ($\Rightarrow$) since $(D, I) \models \alpha[0]$ and we have $\alpha[0] = \alpha$. Thus $\mathcal{M}, w_0 \models \psi_\alpha$.

To prove ($\Leftarrow$), suppose that φ_α is satisfiable, and let $\mathcal{M} = (W, D, R, \gamma, V)$ be a constant domain model such that $\mathcal{M}, r \models \psi_\alpha$. Note that since $\mathcal{M}, r \models \lambda_{n+2}$, every path starting from r has length at least $n+2$ and there is at least one such path.

Let w be any world at height 2. Since $\mathcal{M}, r \models \lambda_{n+2} \wedge (\forall z \Box)^2 \psi_\alpha$, there is at least one path of length n starting from w and also we have $\mathcal{M}, w \models \psi_\alpha$. Further since, $\mathcal{M}, r \models \gamma_n$, for any $c, d \in D$ we have $\mathcal{M}, w, [z_1 \to c, z_2 \to d] \models (\exists z \Diamond)^n (\exists z \Diamond (P(z_1) \wedge Q(z_2)) \to (\forall z \Box)^n (\exists z \Diamond (P(z_1) \wedge Q(z_2))$.

Define $I = \{(c, d) \mid c, d \in D$ and $\mathcal{M}, w, [x \to c, y \to d] \models (\forall z \Box)^n \exists z \Diamond (P(x) \wedge Q(y))\}$.

For $0 \leq i \leq n$ let W_i denote the set of all worlds at distance i from w with $W_0 = \{w\}$. The $\mathsf{FO}(R)$ model for α is given by $\mathcal{M}' = (D, I)$. We now claim that the formula α is satisfied in this model, which is proved by induction on $n - i$. Again, the relevant claim is as follows:

Claim. For all $0 \leq i \leq n$ and for all $d_1 \cdots d_i \in D$, we have:

(a) if there is some $v_i \in W_i$ such that $\mathcal{M}, v_i, [\overline{x^i} \mapsto \overline{d^i}] \models \psi_\alpha[i]$ then for all $u_i \in W_i$ we have $\mathcal{M}, u_i, [\overline{x^i} \mapsto \overline{d^i}] \models \psi_\alpha[i]$

(b) $D, I, [\overline{x^i} \mapsto \overline{d^i}] \models \alpha[i]$ iff for all $v_i \in W_i$, $\mathcal{M}, v_i, [\overline{x^i} \mapsto \overline{d^i}] \models \psi_\alpha[i]$.

The proof is by induction on $n - i$. In the base case, $i = n$. Now we induct on the structure of β (assume that β is in negation normal form).

In the base case we have $R(x_i, x_j)$. To prove (a), if for some $v_n \in W_n$ suppose $\mathcal{M}, v_n, [x_i \to c, x_j \to d] \models (\exists z \Diamond)(P(x_i) \wedge Q(x_j))$. Recall that $\mathcal{M}, w, [z_1 \to c, z_2 \to d] \models (\exists z \Diamond)^n (\exists z \Diamond (P(z_1) \wedge Q(z_2)) \to (\forall z \Box)^n (\exists z \Diamond (P(z_1) \wedge Q(z_2))$. Hence we have $\mathcal{M}, w, [z_1 \to c, z_2 \to d] \models (\forall z \Box)^n (\exists z \Diamond (P(z_1) \wedge Q(z_2))$. Thus, for all $u_n \in W_n$, we have $\mathcal{M}, u_n, [x_i \to c, x_j \to d] \models (\exists z \Diamond)(P(x_i) \wedge Q(x_j))$.

For (b), $(D, I) \models R(c, d)$ iff $\mathcal{M}, w, [x_i \to c, x_j \to d] \models (\exists z \Box)^n (\exists z \Diamond z (P(x_i) \wedge Q(x_j))$ iff (by definition of R) for all $v_n \in W_n$ we have $\mathcal{M}, v_n, [x_i \to c, x_j \to d] \models (\exists z \Diamond)(P(x_i) \wedge Q(x_j))$.

For the case $\neg R(x, y)$ let $M, v_n, [x_i \to c, x_j \to d] \models \neg(\exists z \Diamond (P(x_i) \wedge Q(x_j))$ this implies $M, w, [x_i \to c, x_j \to d] \not\models (\exists z \Box)^n [\exists z \Box (P(x_i) \wedge Q(x_j))$. Now suppose (a) does not hold, then there is some v'_n such that $M, v'_n, [x_i \to c, x_j \to d] \models (\exists z \Diamond (P(x) \wedge Q(y))$ but this implies $M, w, [x_i \to c, x_j \to d] \models (\exists z \Diamond)^n (\exists z \Diamond (P(x) \wedge Q(y))$ and hence $M, w, [x_i \to c, x_j \to d] \models (\exists z \Box)^n (\exists z \Box (P(x) \wedge Q(y))$ which contradicts the assumption. Further (b) follows but routine induction.

The cases of $\vee$ and $\wedge$ are standard.

For the induction step, consider the case when $\alpha[i-1]$ is of the form $\exists x_i \alpha[i]$; the corresponding translated formula is $\exists x_i \Diamond \psi_\alpha[i]$.

To prove (a), suppose for some $v_{i-1} \in W_{i-1}$ we have $\mathcal{M}, v_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x \Diamond \psi_\alpha[i]$ then there is some $c \in D$ and some successor of v_{i-1}, $v'_i \in W_i$ such that $\mathcal{M}, v'_i, [\overline{x^{i-1}} \mapsto$

$\overline{d^{i-1}}, x^i \to c] \models \psi_\alpha[i]$. Now by induction, for all $u_i \in W_i$, we have $\mathcal{M}, u_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x^i \to c] \models \psi_\alpha[i]$. Further since $\mathcal{M}, r \models \lambda_{n+2}$, every $u_{i-1} \in W_{i-1}$ has at least one successor $u'_i \in W_i$. Thus, $\mathcal{M}, u_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i \Diamond \psi_\alpha[i]$.

For (b) suppose, $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i \alpha[i]$ then there is some $c \in D$ such that $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \alpha[i]$ iff (by induction hypothesis) $\mathcal{M}, v_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \psi_\alpha[i]$ for every $v_i \in W_i$ at height i. Now any $w_{i-1} \in W_{i-1}$ is at height $< n$ and since $\mathcal{M}, r \models \lambda_{n+2}$, there is some $v'_i \in W_i$ which is a successor of w_{i-1}. Hence, for all $w_{i-1} \in W_{i-1}$ we have $\mathcal{M}, w_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i \Diamond \psi_\alpha[i]$.

On the other hand, suppose for all $v_{i-1} \in W_{i-1}$ we have $\mathcal{M}, v_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i \Diamond \psi_\alpha[i]$. Choose arbitrary $w_{i-1} \in W_{i-1}$. By semantics, there is some $c \in D$ and $u_i \in W_i$ which is a successor of w_{i-1} such that $\mathcal{M}, u_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \psi_\alpha[i]$. Now by induction (a) at step i, for all $u' \in W_i$ we have $\mathcal{M}, u', [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \psi_\alpha[i]$ and hence $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \alpha[i]$. Hence $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \exists x_i \alpha[i]$.

For the case when $\alpha[i-1]$ is of the form $\forall x_i \alpha[i]$, to prove (a), suppose for some $v_{i-1} \in W_{i-1}$ we have $\mathcal{M}, v_{i-1}, [\overline{x^i} \mapsto \overline{d^i}] \models \forall x_i \Box \psi_\alpha[i]$. Choose arbitrary $c \in D$. Then for all $v'_i \in W_i$ which are successors of v_{i-1}, we have $\mathcal{M}, v'_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x^i \to c] \models \psi_\alpha[i]$. Since there is at least once such successor of v_{i-1}, by induction (a) for all $u_i \in W_i$, we have $\mathcal{M}, u_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x^i \to c] \models \psi_\alpha[i]$. Now, note that for all $w_{i-1} \in W_i$ we have successors of $w_{i-1} \subseteq W_i$ and c was chosen arbitrarily. Hence for all $w_{i-1} \in W_{i-1}$ we have $\mathcal{M}, w_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \Box \alpha[i]$.

To prove (b), suppose $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \alpha[i]$. Choose arbitrary $c \in D$. Then $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x^i \to c] \models \alpha[i]$ and by induction hypothesis, for all $v_i \in W_i$ we have $\mathcal{M}, v_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x^i \to c] \models \psi_\alpha[i]$. Again for any $w_{i-1} \in W_{i-1}$, since successors of w_i are in W_{i-1} and c was chosen arbitrarily we have $\mathcal{M}, w_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \Box \psi_\alpha[i]$.

Finally, suppose for all $w_{i-1} \in W_{i-1}$ we have $\mathcal{M}, w_{i-1}, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \Box \psi_\alpha[i]$. Choose arbitrary $c \in D$. Since every $u_i \in W_i$ is a successor of some $w_{i-1} \in W_{i-1}$, for all $u_i \in W_i$ we have $\mathcal{M}, u_i, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \psi_\alpha[i]$. Now by induction hypothesis, $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}, x_i \to c] \models \alpha[i]$. Since c was chosen arbitrarily, $D, I, [\overline{x^{i-1}} \mapsto \overline{d^{i-1}}] \models \forall x_i \alpha[i]$. ◀

4 Decidability results

Having seen that the $\mathsf{B}^{\forall\Box}$-FOML (and hence full B-FOML) fragment is undecidable over constant domain models, and noted that the $\exists\Box$ bundle is decidable over increasing domain models ([15]), it is natural to wonder whether the problem is undecidable because of $\exists\Diamond(\forall\Box)$ bundle or constant domain semantics, or both. In this section, we show that it is indeed the combination that is the culprit, by proving that relaxing either of the conditions leads to decidability. First, we show that the full B-FOML fragment is decidable over increasing domain models, and then show that the $\exists\Box$ bundle is decidable over constant domain models. For technical reasons, we consider formulas given in negation normal form (NNF):

$$\varphi ::= P\overline{x} \mid \neg P\overline{x} \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \exists x \Box \varphi \mid \exists x \Diamond \varphi \mid \forall x \Box \varphi \mid \forall x \Diamond \varphi$$

Formulas of the form $P\overline{x}$ and $\neg P\overline{x}$ are *literals*. Clearly, every B-FOML-formula φ can be rewritten into an equivalent formula in NNF. We call a formula *clean* if no variable occurs both bound and free in it and every use of a quantifier quantifies a distinct variable. A finite set of formulas is *clean* if their conjunction is clean. Note that every B-FOML-formula can be rewritten into an equivalent clean formula. (For instance, $\exists x\Box P(x) \vee \exists x\Box Q(x)$ and $P(x) \wedge \exists x\Box Q(x)$ are unclean formulas, whereas $\exists x\Box P(x) \vee \exists y\Box Q(y)$ and $P(x) \wedge \exists y\Box Q(y)$ are their clean equivalents.)

A tableau is a tree structure $T = (W, V, E, \lambda)$ where W is a finite set, (V, E) is a rooted tree and $\lambda : V \to L$ is a labelling map. Each element in L is of the form (w, Γ, F), where $w \in W$, Γ is a finite set of formulas and $F \subseteq \mathsf{Var}$ is a finite set. The intended meaning of the label is that the node constitutes a world w that satisfies the formulas in Γ with the "assignment" F, with each variable in F denoting one that occurs free in Γ and as we will see, the assignment will be the identity.

Tableau procedures offer an intuitive way of constructing a canonical model for the given formula. See Fitting and Mendelson [5] for details on tableau procedures for first order modal logics.

4.1 Increasing domain models

Tableau procedures for first order logics typically add witnesses for existential quantifiers using "new" elements (either variables or constants) while simultaneously instantiate universally quantified formulas by the newly added ones. Tableau procedures for modal logics add successor worlds for $\Diamond$ modalities that inherit formulas α when $\Box\alpha$ is in the parent node. Clearly, we need a combination of both. Increasing domain semantics enables us to easily add new witnesses "as we need", so we consider this first.

One complication with bundled quantifiers and modalities is that we need to ass witnesses for existential quantifiers and successor worlds "simultaneously", in the sense that any decision for one affects the choice of the other. To be specific, suppose that we are in an intermediate step of tableau construction when we have formulas $\{\exists x\Diamond\alpha, \exists y\Box\beta, \forall z\Diamond\varphi, \forall z'\Box\psi\}$ at a node w. We need new witnesses for x and y. Further, we need to add a new successor node wv_x; this new node inherits not only α but also β and ψ. But there is plenty more to consider. We already have "active" variables F, which has been updated to F' now. For each $y' \in F'$ we need a φ-successor (which inherits β and ψ as well).

The (BR) rule in the tableau formalizes this intuition when there are multiple occurrences of the bundled formulas. In general if we have formulas $\{\exists x_1\Diamond\alpha_1..\exists x_{n_1}\Diamond\alpha_{n_1}\} \cup \{\exists y_1\Box\beta_1..\exists y_{n_2}\Box\beta_{n_2}\} \cup \{\forall z_1\Diamond\varphi_1..\forall z_{m_1}\varphi_{m_1}\} \cup \{\forall z'_1\Box\psi_1..\forall z'_{m_2}\Box\psi'_{m_2}\}$ at a world w, we need two kinds of successors. The first kind is where a new successor wv_{x_i} is created for every α_i (where x_i is the witness). These successors should satisfy all $\Box$ formulas and hence we add β_j, ψ_l appropriately. The second kind are the ones that take care of $\forall\Diamond$ formulas and hence we have one successor $wv_{z_k}^{y'}$ for every φ_k and every $y' \in F'$. Again β_j, ψ_l are added appropriately to handle $\Box$ constraints.

The $(\vee)$ and $(\wedge)$ rules are standard and The rule (END) says that in the absence of any $Qx\Diamond$ formulas, with $Q \in \{\exists, \forall\}$, the branch does not need to be explored further, as only the literals remain.

The corresponding tableau rules are given as follows:

▶ **Definition 7.** Tableau rules for increasing domain models for the B-FOML fragment are given by:

$$\frac{w : \varphi_1 \vee \varphi_2, \Gamma, F}{w : \varphi_1, \Gamma, F \mid w : \varphi_2, \Gamma, F}\ (\vee) \qquad \frac{w : \varphi_1 \wedge \varphi_2, \Gamma, F}{w : \varphi_1, \varphi_2, \Gamma, F}\ (\wedge)$$

Given $n_1 \geq 1$ or $m_1 \geq 1$; $n_2, m_2, s \geq 0$:

$$\frac{\begin{matrix} w : \exists x_1 \Diamond \alpha_1 \cdots, \exists x_{n_1} \Diamond \alpha_{n_1}, \exists y_1 \Box \beta_1, \cdots, \exists y_{n_2} \Box \beta_{n_2}, \\ \forall z_1 \Diamond \varphi_1, \cdots, \forall z_{m_1} \Diamond \varphi_{m_1}, \forall z_1' \Box \psi_1, \cdots, \forall z_{m_2}' \Box \psi_{m_2}, \end{matrix} r_1 \ldots r_s, F}{\begin{matrix} \langle wv_{x_i} : \alpha_i, \{\beta_j \mid 1 \leq j \leq n_2\}, \{\psi_l[z/z_l'] \mid z \in F', l \in [1, m_2]\}, F' \rangle \textit{ where } i \in [1, n_1] \\ \cup \langle wv_{z_k}^{y'} : \varphi_k[y'/z_k], \{\beta_j \mid 1 \leq j \leq n_2\}, \{\psi_l[z/z_l'] \mid z \in F', l \in [1, m_2]\}, F' \rangle \\ \textit{where } k \in [1, m_1], y' \in F' \end{matrix}}\ (\texttt{BR})$$

Given $n_2 \geq 1$ or $m_2 \geq 1$; $s \geq 0$:

$$\frac{w : \exists y_1 \Box \beta_1, \cdots, \exists y_{n_2} \Box \beta_{n_2}, \forall z_1' \Box \psi_1, \cdots, \forall z_{m_2}' \Box \psi_{m_2}, r_1 \ldots r_s, F}{w : r_1 \ldots r_s, F}\ (\texttt{END})$$

where $F' = F \cup \{x_i \mid i \in [1, n_1]\} \ \cup \{y_j \mid j \in [1, n_2]\}$ and $r_1 \cdots r_s \in \mathit{lit}$ (the literals).

Note that we use variables themselves as witnesses and F' extends F with one witness for each α_i (x_i) and one for each β_j (y_j). Further, there is an implicit ordering on how rules are applied: (BR) insists on the label containing no top level conjuncts or disjuncts, and hence may be applied only after the $\wedge$ and $\vee$ rules have been applied as many times as necessary.

For a given formula φ, we start building the tableau with the root node $(\{w\}, \{r\}, \emptyset, L)$ where $L(r) = (w, \{\varphi\}, \mathsf{Fv}(\varphi))$. A rule specifies that if a node labelled by the premise of the rule exists at a node, it can cause one or more new nodes to be created as children with the labels as given by the completion of the rule. A tableau is saturated when no more rules can be applied. For any formula φ, we refer to the saturated tableau of φ simply as tableau of φ.[4]

The rule (BR) looks complicated but actually asserts standard modal validities with multiplicity. To see how it works, consider a model $\mathcal{M}$, a world u and assignment σ such that $(\mathcal{M}, u, \sigma) \models \exists x \Diamond \alpha \wedge \exists y \Box \beta \wedge \forall z \Box \psi$. Then there are some domain elements $c, d \in \delta(u)$, and a successor world v_c such that $(\mathcal{M}, v_c, \sigma') \models \alpha \wedge \beta \wedge \psi$, where $\sigma'(x) = \sigma'(z) = c$ and $\sigma'(y) = d$. Further if $(\mathcal{M}, u, \sigma) \models \forall z \Diamond \varphi \wedge \forall z' \Box \psi$ then for all $d \in \delta(u)$, we have a successor world v^d such that for all $c \in \delta(u)$, $(\mathcal{M}, v^d, \sigma') \models \varphi \wedge \psi$, where $\sigma'(z) = d$ and $\sigma'(z') = c$. When the domain elements is a finite set (C) which are themselves variables, then we could as well write $(\mathcal{M}, v^d, \sigma') \models \varphi[z] \wedge \bigwedge_{z' \in C} \psi[z']$. The rule (BR) achieves just this, but has to do all this simultaneously for all the quantified formulas at the node "in one shot", and has to keep the formulas clean too.

We need to check that the rule (BR) is well-defined. Specifically, if the label in the premise contains only clean formulas, we need to check that the label in the conclusion does the same. To see this, observe the following, with Γ being the set of clean formulas in the premise. Let Δ, Δ' stand for any modality.

- Note that if $\exists x \Delta \varphi$ and $Q y \Delta' \psi$ are both in Γ, with Q any quantifier, then $x \neq y$ and neither x occurs free in ψ nor y occurs free in φ; also φ or ψ do not contain any subformula that quantifies over x or y.

[4] Refer Wang[15] for an illustration of a similar tableau construction.

- Hence, in the conclusion of (BR), every substitution of the form $\varphi[z/y]$ results in a clean formula, since z occurs free y does not occur at all. Similar argument holds for ψ. Hence the resulting set of formulas in the successors are always clean.

Thus, maintaining "cleanliness" allows us to treat existential quantifiers as giving their own witnesses. The "increase" in the domain is given by the added elements in F' in the conclusion. Note that with each node creation either the number of boolean connectives or the maximum quantifier rank of formulas in the label goes down, and hence repeated applications of the tableau rules must terminate, thus guaranteeing that the tableau generated is always finite.

A tableau is said to be *open* if it does not contain any node u such that its label contains a literal r as well as its negation. Given a tableau T, we say a node $(w : \Gamma, F)$ is a *branching node* if it is branching due to the application of BR. We call $(w : \Gamma, F)$ the *last node of* w, if it is a leaf node or a branching node. Clearly, given any label w appearing in any node of a tableau T, the last node of w uniquely exists. If it is a non-leaf node, every child of w is labelled wu for some u.

Let t_w denote the last node of w in tableau T and let $\lambda(t_w) = (w : \Gamma, F)$. If it is a non-leaf node, then it is a branching node with rule (BR) applying to it with F' as its conclusion. We let $Dom(t_w)$ denote the set F' in this case and $Dom(t_w) = F$ otherwise.

▶ **Theorem 8.** *For any clean* B-FOML*-formula θ in NNF let $F_r = \{x \mid x \text{ is free in } \theta\} \cup \{z\}$, where $z \in$ Var, z does not appear in θ. Then:*
There is an open tableau from $(r : \{\theta\}, F_r)$ iff θ is satisfiable in an increasing domain model.

Proof. Note that we include a new variable $z \in F_r$ to ensure that the domain is always non-empty.

Let T be any (saturated) tableau T starting from $(r : \{\theta\}, F_r)$ where θ is clean. We observe that for any node t with label $(w : \Gamma, F)$ in T, we have the following. If $x \in F$ and occurs in a formula in Γ then every occurrence of x is free. Further, every variable x occurring free in a formula in Γ is in F. These are proved by induction on the height of t using the fact that the rule (BR), when applied to clean formulas, results in clean formulas.

To prove the theorem, given an open tableau T starting from $(r : \{\theta\}, F_r)$, we define $\mathcal{M} = (W, D, \delta, R, \rho)$ where: $W = \{w \mid (w : \Gamma, F)$ occurs in some label of T for some $\Gamma, F\}$; $D = \mathsf{Var}$; wRv iff $v = wv'$ for some v'; $\delta(w) = Dom(t_w)$; $\overline{x} \in \rho(w, P)$ iff $P\overline{x} \in \Gamma$, where $\lambda(t_w) = (w, \Gamma, F)$. Clearly, if wRv then $Dom(t_w) \subseteq Dom(t_v)$, and hence $\mathcal{M}$ is indeed an increasing domain model.

Moreover ρ is well-defined due to openness of T. We now show that $\mathcal{M}, r$ is indeed a model of θ, and this is proved by the following claim.

Claim. For any $w \in W$ if $\lambda(t_w) = (w : \Gamma, F)$ and if $\alpha \in \Gamma$ then $(\mathcal{M}, w, id_F) \models \alpha$. (Below, we abuse notation and write $(\mathcal{M}, w, F) \models \alpha$ for $(\mathcal{M}, w, id_F) \models \alpha$ where $id_F = \{(x, x) \mid x \in F\}$.)

The proof proceeds by reverse induction on the height of the node at which w occurs as label. The base case is when the node considered is a leaf node and hence it is also the last node with that label. The definition of ρ ensures that the literals are evaluated correctly in the model and hence the base case follows.

For the induction step, the conjunction and disjunction cases, the current node is not the last node. Thus the induction applies to its successor which will also have the same label w and the claim follows.

Now consider the application of rule (BR) at a branching node t_w with label $(w : \Gamma, F)$. Let [5]

$$\begin{aligned}\Gamma = \quad & \{\exists x_i \Diamond \alpha_i \mid i \in [1, n_1]\} \cup \{\exists y_j \Box \beta_j \mid j \in [1, n_2]\} \cup \{\forall z_k \Diamond \varphi_k \mid k \in [1, m_1]\} \\ & \cup \{\forall z'_l \Box \psi_l \mid l \in [1, m_2]\} \cup \{r_1 \dots r_s\}.\end{aligned}$$

By induction hypothesis, we have that for every $i \leq n_1$, $\mathcal{M}, wv_{x_i}, F' \vDash \alpha_i \wedge \bigwedge_{j \leq n_2} \beta_j \wedge \psi'$ and for every $y \in F'$ and $k \in [1, m_1]$, $\mathcal{M}, wv^y_{z_k}, F' \vDash \varphi_k[y/z_k] \wedge \psi'$, where $\psi' = \bigwedge^{z \in F'}_{l \leq m_2} \psi_l[z/z'_l]$.

Note that $D_w = Dom(t_w) = F'$. We need to show that $\mathcal{M}, w, F \vDash \alpha$ for each $\alpha \in \Gamma$. Every such α is either a literal or a bundle formula. The assertion for literals follows from the definition of ρ. For $\exists x_i \Diamond \alpha_i \in \Gamma$ we have the successor wv_{x_i} where $\mathcal{M}, wv_{x_i}, F' \models \alpha_i$ and (by observation at the beginning of the proof) $\mathsf{Fv}(\alpha_i) \subseteq F$ and hence we have $\mathcal{M}, w, F \models \exists x_i \Diamond \alpha_i$. Similarly for every $\forall z_k \Diamond \varphi_k \in \Gamma$ and $y \in D_w$ we have the successor $wv^y_{z_k}$ where $\mathcal{M}, wv^y_{z_k}, F \models \varphi_k[y/z_k]$ and thus $\mathcal{M}, w, F \models \forall z_k \Diamond \varphi_k$.

Now for the case $\exists y_j \Box \beta_j$: by induction hypothesis, for all successors $wv^{\#}_z$ of w where $\#$ is either empty or $\# \in F'$ we have $\mathcal{M}, wv^{\#}_z, F' \vDash \beta_j$. Since $\mathsf{Fv}(\beta_j) \subseteq F \cup \{y_j\}$, we have $\mathcal{M}, wv^{\#}_z, id_F[y_j \mapsto y_j] \vDash \beta_j$ for each $wv^{\#}_z$. Finally note that $y_j \in F' = D_w$ and hence we have $\mathcal{M}, w, id_F \vDash \exists y_j \Box \beta_j$.

The case $\forall z'_l \Box \psi_l$ is similar. By induction hypothesis, we have $\mathcal{M}, wv^{\#}_z, F' \vDash \psi_l[z/z'_l]$ for every $z \in F'$ and again by cleanliness preservation, $\mathcal{M}, wv^{\#}_z, F'[z_l \mapsto z] \vDash \psi_l$ for all $z \in F' = D_w$.

Hence $\mathcal{M}, w, id_F \vDash \forall z'_l \Box \psi_l$.

Finally note that for the root r, if $t_r = (r : \Gamma, F)$ then $F = F_r$ since domain changes only for a (BR) rule which will not be the t_r. Hence it follows that $\mathcal{M}, r, F_r \vDash \theta$.

Completeness of tableau construction. For the other direction, we show that all rule applications preserve the satisfiability of the formula sets in the labels. This would ensure that there is an open tableau since satisfiability of formula sets ensures lack of contradiction among literals. It is easy to see that the rules ($\wedge$) preserves satisfiability and so does the (END) rule, since F is non-empty at every step. If one of the conclusions of the ($\vee$) rule is satisfiable then so is the premise. It remains only to show that (BR) preserves satisfiability. Consider a label set Γ of clean formulas at a branching node. Let

$$\begin{aligned}\Gamma = \quad & \{\exists x_i \Diamond \alpha_i \mid i \in [1, n_1]\} \cup \{\exists y_j \Box \beta_j \mid j \in [1, n_2]\} \cup \{\forall z_k \Diamond \varphi_k \mid k \in [1, m_1]\} \\ & \cup \{\forall z'_l \Box \psi_l \mid l \in [1, m_2]\} \cup \{r_1 \dots r_s\}.\end{aligned}$$

be satisfiable at a model $\mathcal{M} = \{W, D, \delta, R, \rho\}$, $w \in W$ and a relevant assignment η such that $\eta(x) \in D_w$ for all $x \in \mathsf{Fv}(\Gamma)$ and $\mathcal{M}, w, \eta \vDash \bigwedge_{\chi \in \Gamma} \chi$.

By the semantics, we have the following:

(A) There exist $a_1, \dots, a_{n_1} \in D_w$ and $v_1 \dots v_{n_1} \in W$ where wRv_i such that $\mathcal{M}, v_i, \eta[x_i \mapsto a_i] \vDash \alpha_i$ for all $i \leq n_1$.

(B) For all $c \in D_w$ there exist $v^c_1 \dots v^c_{m_1} \in W$, where $wRv^c_{m_i}$ such that $\mathcal{M}, v^c_k, \eta[z_k \mapsto c] \models \varphi_k$ for all for all $k \leq m_1$.

(C) There exist $b_1, \dots b_{n_2} \in D_w$ such that for all $v \in W$ if wRv then $\mathcal{M}, v, \eta[y_j \mapsto b_j] \models \beta_j$ for all $j \leq n_2$.

(D) For all $d \in D_w$ and for all $v \in W$ if wRv then $\mathcal{M}, v, \eta[z'_l \mapsto d] \models \psi_l$ for all $l \leq m_2$.

Moreover, due to the fact that Γ is clean, we observe that:

(O) $\overline{x}, \overline{y}, \overline{z}$ and $\overline{z'}$ only occur in $\alpha_i, \beta_j, \varphi_k$ and ψ_l respectively.

[5] Note that the argument holds even if either of n_1 or m_1 is 0.

We now need to show:

(1) $\{\alpha_i\} \cup \{\beta_j \mid 1 \leq j \leq n_2\} \cup \{\psi_l[z/z'_l] \mid z \in F', 1 \leq l \leq m_2\}$ is satisfiable for all $i \leq n_1$.

(2) $\{\varphi_k[y'/z_k]\} \cup \{\beta_j \mid 1 \leq j \leq n_2\}, \{\psi_l[z/z'_l] \mid z \in F', 1 \leq l \leq m_2\}$ is satisfiable for all $k \leq m_1$, $y' \in F'$.

For (1): given $i \leq n_1$, due to (A), (C) and (O), we can pick an $a_i \in D_w$ and a successor v_i of w, and some $\overline{b} \in D_w$ such that

$$\mathcal{M}, v_i, \eta[x_i \mapsto a_i;\ \overline{y} \mapsto \overline{b}] \vDash \alpha_i \wedge \bigwedge_j \beta_j$$

By (D), (O) and the fact that η only assigns variables the elements in D_w, we can also show that

$$\mathcal{M}, v_i, \eta[x_i \mapsto a_i;\ \overline{y} \mapsto \overline{b}] \vDash \bigwedge\{\psi_l[z/z'_l] \mid z \in F', 1 \leq l \leq m_2\}.$$

Note that $\eta[x_i \mapsto a_i;\ \overline{y} \mapsto \overline{b}]$ is relevant for v_i since $\mathcal{M}$ is an increasing domain model and wRv_i. This completes the proof for (1).

For (2): Given $k \leq m_1$ and $y' \in F'$. Suppose $\eta(y') = c \in D_w$, then due to (B) we have a successor v^c_k of w such that $\mathcal{M}, v^c_k, \eta \models \varphi_k[y'/z_k]$. Now again, due to (C), (D), (O) and the fact that η is a relevant assignment for w, we have:

$$\mathcal{M}, v^c_k, \eta[\overline{y} \mapsto \overline{b}] \models \varphi_k[y'/z_k] \wedge \bigwedge_j \beta_j \wedge \bigwedge\{\psi_l[z/z'_l] \mid z \in F', 1 \leq l \leq m_2\}.$$

Again, $\eta[\overline{y} \mapsto \overline{b}]$ is also a relevant assignment for v^c_k, and this completes the proof for (2). ◀

The theorem offers us a decision procedure for checking satisfiability. Note that not only is the depth of the tableau linear in the size of the formula, but also that labels are never repeated across siblings. Hence an algorithm can explore the tableau depth wise and reuse the same space when exploring other branches. The techniques are standard as in tableau procedures for modal logics. The extra space overhead for keeping track of domain elements is again only linear in the size of the formula. Further, observe that every B-FOML formula has an equivalent formula in negation normal form with linear blow-up. This way, we can get a PSPACE-algorithm for checking satisfiability. The PSPACE lower bound follows from propositional modal logic, of which our language is an extension.

▶ **Corollary 9.** *Satisfiability of* B-FOML *fragment over increasing domain models is* PSPACE*-complete.*

4.2 Constant domain models

We now take up the second task, to show that over constant domain models, the culprit is the $\forall\Box$ bundle, by proving that the satisfiability problem for the $\mathsf{B}^{\exists\Box}$-FOML is decidable over constant domain models.

In these models, we need to fix the domain right at the start of the tableau construction and use only these elements as witnesses. We do this by calculating a precise bound on how many new elements need to be added for each subformula of the form $\exists x\Box\varphi$ and include as many as needed at the beginning of the tableau construction.

Let $\mathsf{Sub}(\theta)$ stand for the finite set of subformulas of θ. Given a clean formula $\theta \in \mathsf{B}^{\exists\Box}$-FOML in NNF, for every $\exists x_j\Box\varphi \in \mathsf{Sub}(\theta)$ let $\mathsf{Var}^{\exists}(\theta) = \{x \mid \exists x\Box\varphi \in \mathsf{Sub}(\theta)\}$. Now, cleanliness has its advantages: every subformula of a clean formula is clean as well. Hence, when θ_1 and θ_2 are both in $\mathsf{Sub}(\theta)$, $\mathsf{Var}^{\exists}(\theta_1) \cap \mathsf{Var}^{\exists}(\theta_2) = \emptyset$. Similarly, when $\theta_1 \in \mathsf{Sub}(\theta)$ and $\theta_2 \in \mathsf{Sub}(\theta_1)$, again $\mathsf{Var}^{\exists}(\theta_1) \cap \mathsf{Var}^{\exists}(\theta_2) = \emptyset$.

Fix a clean formula θ in NNF with modal depth h. For every $x \in \mathsf{Var}^{\exists}(\theta)$ define Var_x to be the set of h *fresh* variables $\{x^k \mid 1 \leq k \leq h\}$, and let $\mathsf{Var}^{+}(\theta) = \bigcup\{\mathsf{Var}_x \mid x \in \mathsf{Var}^{\exists}(\theta)\}$ be the set of new variables to be added. Note that $\mathsf{Var}_x \cap \mathsf{Var}_y = \emptyset$ when $x \neq y$. Fix a variable z not occurring in $\mathsf{Var}^{+}(\theta)$. Define $D_\theta = \mathsf{Fv}(\theta) \cup \mathsf{Var}^{+}(\theta) \cup \{z\}$.

The tableau rules for constant domain models for $\mathsf{B}^{\exists\Box}$-FOML fragment are given by:

$$\frac{w : \varphi_1 \vee \varphi_2, \Gamma, C}{w : \varphi_1, \Gamma, C \mid w : \varphi_2, \Gamma, C}\ (\vee) \qquad \frac{w : \varphi_1 \wedge \varphi_2, \Gamma, C}{w : \varphi_1, \varphi_2, \Gamma, C}(\wedge)$$

Given $n, s \geq 0;\ m \geq 1$:

$$\frac{w : \exists x_1 \Box \varphi_1, \ldots, \exists x_n \Box \varphi_n, \forall y_1 \Diamond \psi_1, \ldots, \forall y_m \Diamond \psi_m, r_1 \ldots r_s, C}{\langle (wv^{y}_{y_i} : \{\varphi_j[x_j^{k_j}/x_j] \mid 1 \leq j \leq n\}, \psi_i[y/y_i], C') \rangle \textit{where } y \in D_\theta, i \in [1, m]}\ (\mathtt{BR_C})$$

Given $n \geq 1, s \geq 0$:

$$\frac{w : \exists x_1 \Box \varphi_1, \cdots, \exists x_n \Box \varphi_n, r_1, \cdots r_s, C}{w : r_1 \cdots r_s, C}\ (\mathtt{END_C})$$

where $C \subseteq D_\theta$ and $C' = C \cup \{x_j^{k_j} \mid 1 \leq j \leq n\}$ where k_j is the smallest number such that $x_j^{k_j} \in \mathsf{Var}_{x_j} \setminus C$ and $r_1 \ldots r_s \in lit$.

Note that the rule $\mathtt{BR_C}$ starts off one branch for each $y \in D_\theta$, since the $\forall\Diamond$ connective requires this over the fixed constant domain D_θ. C keeps track of the variables used already along the path from the root till the current node. These are now fixed, so the witness for $\exists x \Box \varphi$ is picked from the remaining variables in $\mathsf{Var}_x(\theta)$. The variables in Var_{x_j} are introduced only by applying $\mathtt{BR}$. Since $|\mathsf{Var}_{x_j}|$ is the modal depth, we always have a fresh x_j^k to choose.

▶ **Theorem 10.** *For any clean* $\mathsf{B}^{\exists\Box}$-FOML*-formula θ in NNF, there is an open constant tableau from* $(r, \{\theta\}, \mathsf{Fv}(\theta))$ *iff θ is satisfiable in a constant domain model.*

The structure of the proof is very similar to that of Theorem 8. But we need to be careful to check that sufficient witnesses exist as needed, since the domain is fixed at the beginning of tableau construction. The proof details are presented in the appendix.

▶ **Corollary 11.** *The satisfiability problem for* $\mathsf{B}^{\exists\Box}$-FOML*-formulas over constant domain models is* PSPACE*-complete.*

5 Between Constant Domain and Increasing Domain

We now show that the $\mathsf{B}^{\exists\Box}$-FOML fragment cannot distinguish increasing domain models and constant domain models. Note that in FOML this distinction is captured by the Barcan formula $\forall x \Box \varphi \to \Box \forall x \varphi$; but this is not expressible in $\mathsf{B}^{\exists\Box}$-FOML.[6]

[6] However, with equality added in the language, we can distinguish the two by:
$\exists x_1 \Box(\forall x_2 \Diamond(\forall z \Diamond(x_1 = x_2)) \wedge \forall y_1 \Diamond(\exists z \Box(\exists y_2 \Box(y_1 \neq y_2))$.
We can also accomplish this in the $\forall\Box$ fragment: $\forall x \Box \forall y \Box \neg P(x) \wedge \forall z \Box \exists x \Diamond \neg P(x)$.

The tableau construction of $\mathsf{B}^{\exists\Box}$-FOML fragment over increasing domain models is a restriction of the BR rule in the last section presented in [15].

$$\text{Given } n, s \geq 0;\ m \geq 1:$$

$$\frac{w : \exists x_1 \Box \varphi_1, \ldots, \exists x_n \Box \varphi_n, \forall y_1 \Diamond \psi_1, \ldots, \forall y_m \Diamond \psi_m, r_1 \ldots r_s, F}{\langle (wv^y_{y_i} : \{\varphi_j \mid 1 \leq j \leq n\}, \psi_i[y/y_i], F') \rangle where\ y \in F', i \in [1, m]} \ (\mathtt{BR_W})$$

$$\text{where } F' = F \cup \{x_j \mid j \in [1, n]\}.$$

Note that $\mathtt{BR_C}$ produces a constant domain tableau whereas $\mathtt{BR_W}$ produces an increasing domain tableau. Now, to prove that the $\mathsf{B}^{\exists\Box}$-FOML fragment cannot distinguish increasing domain models and constant domain models, it is sufficient to show that any formula $\varphi \in \mathsf{B}^{\exists\Box}$-$\mathsf{FOML}$ is satisfiable over increasing domain model is also satisfiable in a constant domain model. We prove this by showing that any $\varphi \in \mathsf{B}^{\exists\Box}$-$\mathsf{FOML}$ fragment that has an open tableau also has a constant domain tableau. From this tableau, we can extract the constant domain model where φ is satisfiable.

▶ **Theorem 12.** *For any* $\mathsf{B}^{\exists\Box}$-FOML *formula* φ *satisfiable on some increasing domain model, the constant domain tableau of* φ *is open.*

The proof is sketched in the appendix.

6 Discussion

We have considered a decidable fragment of FOML by bundling quantifiers together with modalities, retaining the same complexity as propositional modal logic, while yet admitting arbitrary k-ary predicates.

We note that choice of how this bundling is done is crucial. The $\exists\Box$ bundle is shown to be robustly decidable, for both constant domain and increasing domain semantics, whereas the $\forall\Box$ bundle is undecidable over constant domain models. Indeed, other ways of "bundling" quantifiers and modalities is possible. For instance, the $\Box\forall$ bundle seems to be similar to the $\forall\Box$ that we have considered (over constant domain models) but $\Box\exists$ seems to be interestingly different. Indeed, we could proceed further and consider bundles determined by a shape of quantifier prefix: $\exists x_1 \ldots \exists x_n \Box$ or $\exists x_1 \ldots \exists x_n \forall z_1 \ldots \forall z_n \Box$ might be worthy of study as a bundle as well. In this sense, this paper is envisaged as a study of "bundling" quantifiers and modalities and its impact on decidability rather than proposing the definitive bundled fragment.

An obvious extension is to consider the language with constants, function symbols and equality. This would be of importance in the study of systems with unbounded data. A crucial direction for further development is to consider the *transitive closure* modality so that reachability properties are specified. The tableau procedures presented already give us a basis for exploring model checking algorithms, but working with finite presentations of data domains needs some care.

References

1 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Logic in Computer Science, 1996. LICS'96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 313–321. IEEE, 1996.

2 Francesco Belardinelli and Alessio Lomuscio. Interactions between knowledge and time in a first-order logic for multi-agent systems: completeness results. *Journal of Artificial Intelligence Research*, 45:1–45, 2012.

3 Egon Börger, Erich Grädel, and Yuri Gurevich. *The classical decision problem.* Springer Science & Business Media, 2001.

4 Gisèle Fischer-Servi et al. The finite model property for MIPQ and some consequences. *Notre Dame Journal of Formal Logic*, 19(4):687–692, 1978.

5 Melvin. Fitting and Richard L Mendelsohn. *First-order modal logic.* Synthese Library. Springer, 1998.

6 D. M. Gabbay and V. B. Shehtman. Undecidability of Modal and Intermediate First-Order Logics with Two Individual Variables. *J. Symbolic Logic*, 58(3):800–823, September 1993. URL: `https://projecteuclid.org:443/euclid.jsl/1183744299`.

7 Kurt Gödel. Zum entscheidungsproblem des logischen funktionenkalküls. *Monatshefte für Mathematik und Physik*, 40(1):433–443, 1933.

8 G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

9 Roman Kontchakov, Agi Kurucz, and Michael Zakharyaschev. Undecidability of first-order intuitionistic and modal logics with two variables. *Bull. Symbolic Logic*, 11(3):428–438, September 2005. `doi:10.2178/bsl/1122038996`.

10 Saul A. Kripke. The Undecidability of Monadic Modal Quantification Theory. *Mathematical Logic Quarterly*, 8(2):113–116, 1962. `doi:10.1002/malq.19620080204`.

11 Mikhail N. Rybakov and Dmitry Shkatov. Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter. *CoRR*, abs/1706.05060, 2017. `arXiv:1706.05060`.

12 Krister Segerberg. Two-Dimensional Modal Logic. *Journal of Philosophical Logic*, 2(1):77–96, 1973. URL: `http://www.jstor.org/stable/30226970`.

13 Moshe Y Vardi. Why is modal logic so robustly decidable? Technical report, Rice University, 1997.

14 Yanjing Wang. A logic of goal-directed knowing how. *Synthese*, 2017. forthcoming.

15 Yanjing Wang. A New Modal Framework for Epistemic Logic. In *Proceedings Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2017, Liverpool, UK, 24-26 July 2017.*, pages 515–534, 2017. `doi:10.4204/EPTCS.251.38`.

16 Yanjing Wang. Beyond knowing that: a new generation of epistemic logics. In *Jaakko Hintikka on knowledge and game theoretical semantics*, pages 499–533. Springer, 2018.

17 Frank Wolter and Michael Zakharyaschev. Decidable Fragments of First-Order Modal Logics. *The Journal of Symbolic Logic*, 66(3):1415–1438, 2001. URL: `http://www.jstor.org/stable/2695115`.

Appendix

Proof of Theorem 10

▶ **Theorem 10.** *For any clean* $\mathsf{B}^{\exists\Box}$-FOML*-formula* θ *in NNF, there is an open constant tableau from* $(r, \{\theta\}, \mathsf{Fv}(\theta))$ *iff* θ *is satisfiable in a constant domain model.*

We show that existence of a constant open tableau is equivalent to satisfiability over constant domain models. First, the following observation on the rule $(\mathtt{BR})$ is useful.

▶ **Proposition 13.** *The rule* $(\mathtt{BR_C})$ *preserves cleanliness of formulas: if a tableau node is labelled by* $(w : \Gamma, C)$*,* Γ *is clean, and a child node labelled* $(wv : \Gamma', C')$ *is created by* $(\mathtt{BR})$ *then* Γ' *is clean as well.*

An important corollary of this proposition is that for all $x \in D_\theta$, at any tableau node all occurrences of x in Γ are free. Therefore, for any formula of the form $\psi_i[z/y_i]$ in the conclusion of the rule, z is free and y_i does not occur at all.

The following fact, familiar from first order logic, will be handy in the proof.

▶ **Proposition 14.** *For any* FOML *formula φ and any model $\mathcal{M}, w$ and any variable z:*

$$\mathcal{M}, w, \sigma \vDash \varphi[z/x] \iff \mathcal{M}, w, \sigma'[x \mapsto \sigma(z)] \vDash \varphi$$

if $\sigma(y) = \sigma'(y)$ for all $y \neq x$ with y occurring free in φ.

Now we shall prove Theorem 10.

Proof. To prove the soundness of tableau construction, given an open tableau T from the root node labelled $(r : \{\theta\}, \mathsf{Fv}(\theta))$, we define $\mathcal{M} = \{W, D_\theta, R, \rho\}$ where $W = \{w \mid (w : \Gamma, C)$ is a label at some node in $T\}$. and wRv iff $v = wv'$ for some v'. For the valuation, we have $\overline{x} \in \rho(w, P)$ iff $P\overline{x} \in \Gamma$, where $\lambda(t_w) = (w, \Gamma)$.

By definition, D_θ is not empty. Further, ρ is well-defined due to the openness of T. As before, we prove that $\mathcal{M}, r$ is indeed a model of θ, and this is proved by the following claim.

Claim. For any tree node w in T if $\lambda(t_w) = (w : \Gamma, C)$ and if $\alpha \in \Gamma$ then $(\mathcal{M}, w, id_C) \models \alpha$. (Again, we abuse notation and write $(\mathcal{M}, w, C) \models \alpha$ for $(\mathcal{M}, w, id_C) \models \alpha$ and denote $C(w)$ to be the C associated with the node labelled w.)

The proof proceeds exactly as before for all the rules except for a slight modification for the $(\mathtt{BR_C})$ rule. We shall consider only this rule in the proof here.

Suppose $(w : \Gamma, C)$ is a branching node where

$$\Gamma = \{\exists x_1 \Box \varphi_1 \ldots \exists x_n \Box \varphi_n, \forall y_1 \Diamond \psi_1 \ldots \forall y_m \Diamond \psi_m, r_1, \ldots r_s\}.$$

By induction hypothesis,

$$\mathcal{M}, wv^y_{y_i}, C'(wv^y_{y_i}) \vDash \psi_i[y/y_i] \wedge \bigwedge_1^n \varphi_j[x_j^{k_j}/x_j]$$

for every $y \in D_\theta$ and $i \in [1, m]$. We need to show that $\mathcal{M}, w, C \vDash \chi$ for each $\chi \in \Gamma$.

The assertion for literals in Γ follows from the definition of ρ. For each $\exists x_j \Box \varphi_j \in \Gamma$ and each $wv^y_{y_i}$, with $y \in D_\theta$, we have $\mathcal{M}, wv^y_{y_i}, C' \vDash \varphi_j[x_j^{k_j}/x_j]$ by induction hypothesis. It is clear that $\{x_j^{k_j} \mid 1 \leq j \leq n\}$ are not free in φ_j since they are chosen to be new. Further, since $x_j^{k_j}$ are not free in φ_j, by Proposition 14, $\mathcal{M}, wv^y_{y_i}, id_C[x_j \mapsto x_j^{k_j}] \vDash \varphi_j$ for all $wv^y_{y_i}$. Therefore $\mathcal{M}, w, C \vDash \exists x_j \Box \varphi_j$.

For $\forall y_i \Diamond \psi_i \in \Gamma$, and $y \in D_\theta$, by induction hypothesis, we have $\mathcal{M}, wv^y_{y_i}, C' \models \psi_i[y/y_i]$. By Proposition 13 and its corollary, y_i is not free in $\psi_i[y/y_i]$ and hence by Proposition 14, $\mathcal{M}, wv^y_{y_i}, id_C[y_i \mapsto y] \vDash \psi_i$. Since this holds for each $y \in D_\theta$, we get $\mathcal{M}, w, id_C \vDash \forall y_i \Diamond \psi_i$ for each i.

Thus, it follows that $\mathcal{M}, r, \sigma(r) \vDash \theta$.

To prove the completeness of the tableau construction, we show that rule applications preserve the satisfiability of the formula set. Again, we only discuss the $\mathtt{BR_C}$ case.

Consider a label set Γ of clean formulas at a branching node. Let

$$\Gamma = \{\exists x_j \Box \varphi_j \mid j \in [1, n]\} \cup \{\forall y_i \Diamond \psi_i \mid j \in [1, m]\} \cup \{r_1 \ldots r_s\}$$

be satisfiable in a model $\mathcal{M} = \{W, D, R, \rho\}$, $w \in W$ and an assignment η such that $\mathcal{M}, w, \eta \vDash \varphi$ for all $\varphi \in \Gamma$.

By the semantics:

(A) for all $c \in D^{\mathcal{M}}$ there exist $v_1^c \dots v_m^c \in W$, successors of w such that $\mathcal{M}, v_i^c, \eta[y_i \mapsto c] \models \psi_i$ for each $i \in [1, m]$.

(B) there exist $c_1, \dots c_n \in D$ such that for all $v \in W$, if wRv then $\mathcal{M}, v, \eta[x_j \mapsto c_j] \models \varphi_j$.

By cleanliness of formulas in Γ, each x_j is free only in φ_j, and each y_i is free only in ψ_i. Thus we can merge the assignments without changing the truth values of φ_j and ψ_i, and obtain:

(A') for all $c \in D$ there exist $v_1^c \dots v_m^c \in W$, successors of w, such that

$$\mathcal{M}, v_i^c, \eta[\overline{x_j} \mapsto \overline{c_j}, y_i \mapsto c] \models \varphi_1 \wedge \dots \wedge \varphi_n \wedge \psi_i$$

where $i \in [1, m]$.

Fixing a $y \in D_\theta$ and an $i \in [1, m]$, in the following we show that $\{\varphi_j[x_j^{k_j}/x_j] \mid 1 \leq j \leq n\} \cup \{\psi_i[y/y_i]\}$ is satisfiable. There are two cases to be considered:

1. y is not one of $x_j^{k_j}$. First since η is an assignment for all the variables in Var, we can suppose $\eta(y) = b \in D$. By (A') above, there exists a successor v_i^b of w such that $\mathcal{M}, v_i^b, \eta[\overline{x_j} \mapsto \overline{c_j}, y_i \mapsto b] \models \varphi_1 \wedge \dots \wedge \varphi_n \wedge \psi_i$.
Note that $\overline{x_j}$ and y_i are not in D_θ thus they are different from y. On the other hand, by cleanliness of Γ, y_i does not occur in φ_j and $\eta(y) = b$, hence $\mathcal{M}, v_i^b, \eta[\overline{x_j} \mapsto \overline{c_j}] \models \varphi_1 \wedge \dots \wedge \varphi_n \wedge \psi_i[y/y_i]$.
Finally, since each x_j only occurs in φ_j and each $x_j^{k_j}$ does not occur in $\varphi_1 \dots \varphi_j$ and $\psi_i[y/y_i]$, we have: $\mathcal{M}, v_i^b, \eta[\overline{x_j^{k_j}} \mapsto \overline{c_j}] \models \varphi_1[x_1^{k_1}/x_1] \wedge \dots \wedge \varphi_n[x_n^{k_n}/x_n] \wedge \psi_i[y/y_i]$.
2. y is $x_j^{k_j}$ for some j. Then we pick c_j, the witness for x_j, and by (A'), $\mathcal{M}, v_i^{c_j}, \eta[\overline{x_j} \mapsto \overline{c_j}, y_i \mapsto c_j] \models \varphi_1 \wedge \dots \wedge \varphi_n \wedge \psi_i$.
Since y is $x_j^{k_j}$, we have $\mathcal{M}, v_i^{c_j}, \eta[\overline{x_j} \mapsto \overline{c_j}, x_j^{k_j} \mapsto c_j] \models \varphi_1 \wedge \dots \wedge \varphi_n \wedge \psi_i[y/y_i]$.
Now proceeding similarly as in the case above we can show that: $\mathcal{M}, v_i^{c_j}, \eta[\overline{x_j^{k_j}} \mapsto \overline{c_j}] \models \varphi_1[x_1^{k_1}/x_1] \wedge \dots \wedge \varphi_n[x_n^{k_n}/x_j] \wedge \psi_i[y/y_i]$.

Finally note that all formulas resulting after applying $(\mathtt{BR_C})$ rule will be of the form A' and is satisfiable as argued above. This completes the proof of the theorem. ◀

Proof of Theorem 12

▶ **Theorem 12.** *For any* $\mathsf{B}^{\exists\Box}$-FOML *formula* φ *satisfiable on some increasing domain model, the constant domain tableau of* φ *is open.*

Proof. (Sketch) We give a proof sketch. Consider a clean $\mathsf{B}^{\exists\Box}$-FOML formula φ, and let $\varphi' = \varphi \wedge \bigwedge\{\exists x' \Box \top \mid x' \in \mathsf{Var}^+(\varphi)\}$ (recall that $\mathsf{Var}^+(\varphi) = \bigcup_{x \in \mathsf{Var}^\exists(\varphi)} \mathsf{Var}_x$). Clearly φ is satisfiable in an increasing domain model iff φ' is as well. Let T be an open tableau for φ'. We show that T can be transformed into a constant open tableau T' for φ.

Suppose T has no applications of $(\mathtt{BR})$, it is also a constant tableau and we are done, so suppose that T has at least one application of the rule $(\mathtt{BR})$. By construction, all the $x' \in \mathsf{Var}^+(\varphi)$ are added to the domain of the root, thus they are also at all the local domains in T. Note that we may have more elements in the local domains, such as x that get added when we apply $\mathtt{BR}$ to $\exists x \Box \varphi$, and therefore there are more branches than needed for a constant domain tableau of φ (such as those for x).

We can get rid of them by the following process:

- Fix $\psi = \exists x \Box \theta \in \mathsf{Sub}(\varphi)$:
    - Fix a node s where `BR` rule is applied and ψ is in s. Since φ is clean, there is no other node in any path of T from the root passing through s such that $\exists x \Box \theta' \in \mathsf{Sub}(\varphi)$ occurs for some θ'. Let m be the modal depth of φ. The path from the root to the predecessor of s can use at most $m-1$ different variables in $\mathsf{Var}_x(\varphi)$ when generating successors by applying the `BR` rule to some $\forall y \Diamond \theta$ formula. Pick the first $x^h \in \mathsf{Var}_x$ which is not used in the path up to this node.
    - Delete all the descendent nodes of s that are named using x^h when applying `BR` to some $\forall y \Diamond$ formula, i.e., the nodes named in the form of $stv^{x^h}_y$ where t can be empty. It is not hard to see that the resulting sub-tableau rooted at s has no occurrence of x^h at all since x^h could only be introduced among the children of s using `BR`.
    - Rename all the occurrences of x by x^h (in formulas and node names) in all the descendent nodes of s. Then the branching structure from the sub-tableau rooted at s will comply with the `BR` rule for constant-domain tableau.
    - Repeat the above for all the application nodes of the `BR` rule w.r.t. ψ
- Repeat the above procedure for all ψ of the form $\exists x \Box \theta \in \mathsf{Sub}(\varphi)$.

The core idea is to simply use the newly introduced variable x as if it were x^h in a constant-domain tableau. Note that each branch-cutting operation and renaming operation (by new variables) above will preserve openness, since openness is merely about contradictions among literals. We then obtain a constant domain tableau by setting the domain as D_φ. ◀

Note that the constant domain tableau T of φ constructed can be viewed as a subtree embedded inside the increasing domain tableau T' of φ'. However, showing that it is generated precisely by the tableau rules in Section 4.2 involves some tedious detail.