Modal Definability of First-Order Formulas with Free Variables and Query Answering

Stanislav Kikot[a], Evgeny Zolin[b,*]

[a] *Department of Computer Science and Information Systems, Birkbeck College University of London, London, WC1E 7HX, United Kingdom*
[b] *Department of Mathematical Logic and Theory of Algorithms, Faculty of Mechanics and Mathematics, Moscow State University, Moscow, 119 991, Russia*

Abstract

We present an algorithmically efficient criterion of modal definability for first-order existential conjunctive formulas with several free variables. Then we apply it to establish modal definability of some family of first-order ∀∃-formulas. Finally, we use our definability results to show that, in any expressive description logic, the problem of answering modally definable conjunctive queries is polynomially reducible to the problem of knowledge base consistency.

Keywords: Modal logic, Modal definability, Correspondence theory, Description logic, Knowledge base, Conjunctive query

1. Introduction

The correspondence between modal and first-order (FO) formulas on Kripke frames is the heart of modal logic. Developed in the 1960s, it is still a common tool for establishing completeness of many modal calculi. A typical modern example of its application is given by various logics of multi-agent systems for reasoning about agents' knowledge, belief, intentions, and cooperative actions [12].

Traditionally, two kinds of correspondence are studied: the *global* one between modal formulas and closed FO formulas, and the *local* one between modal formulas and FO formulas with one free variable. It was Kracht who first introduced in [25] the notion of correspondence between *n-tuples* of modal formulas and FO formulas with n free variables, for arbitrary $n \geq 1$. He established basic properties of this notion and devised a special calculus (called "the calculus of internal descriptions") for deriving instances of such a correspondence. In [25] he used this notion of correspondence for proving the claim, known now as Kracht's theorem [5, 24], which describes a large class of FO formulas that are modally definable.

Typically, this notion of correspondence is used only as a technical tool for proving similar theorems (see, e.g., [20]). However, recently a query answering algorithm based on the local correspondence emerged [36, 37]. Its key idea is to replace a query (which is a FO formula) with a corresponding modal formula. Since this algorithm is based on the *local* correspondence, the range of its applications is limited to unary queries, i.e., to FO formulas with one free variable. Now, this limitation can be overcome by considering a more general kind of correspondence, i.e., modal definability of FO formulas with several free variables. This is the departing point for our research.

In this paper, we mainly focus on modal definability of FO formulas of a special kind, called *existential conjunctive formulas*, or ∃&-formulas, for short. An ∃&-formula is an existentially quantified conjunction of atomic formulas of the form $x\mathsf{R}y$; for instance, $\exists y\,(x\mathsf{R}y \wedge y\mathsf{R}y)$. The motivation for considering these formulas is not only that they form a natural fragment of FO logic, but also that they are closely related to so-called *conjunctive queries*, which play an important rôle in knowledge representation and reasoning.

The main result of our paper is the algorithmically efficient criterion of modal definability for ∃&-formulas with several free variables. Moreover, given a modally definable ∃&-formula, our algorithm produces, in polynomial time, the corresponding tuple of modal formulas. This contrasts to the general case, for it is undecidable whether an arbitrary FO formula (even with one free variable) is modally definable, due to Chagrova's result [9].

The paper is organized as follows. Sect. 2 recalls the notion of modal definability (and introduces its generalization, modal expressibility) for FO formulas with several free variables. Sect. 3 introduces the family of ∃&-formulas, together with their graph representation. Sect. 3.1 presents our main result — the criterion of modal definability of ∃&-formulas, formulated in graph-theoretic terms. The proofs of definability and undefinability results are given in Sect. 4 and 5, respectively. In Sect. 6 we use our results to prove modal definability of a large family of ∀∃-formulas, which arise in many-dimensional modal logics.

The last part of our paper, Sect. 7, gives an application of our definability results to the problem of answering conjunctive queries in description logic knowledge bases. There we recall all necessary definitions and cornerstone results in that field. Then, the Reduction Theorem (Theorem 7.2) establishes the relationship between modal definability and query answering in FO theories. Its consequence (Theorem 7.8) says that for modally definable conjunctive queries, the problem of query answering can be solved easier than that for arbitrary conjunctive queries, by a polynomial reduction to the problem of knowledge base consistency. The concluding Sect. 8 points out directions for further research.

*Corresponding author
Email addresses: staskikotx@gmail.com (Stanislav Kikot), ezolin@gmail.com (Evgeny Zolin)

2. Modal definability

Modal formulas are built up from propositional variables $\mathsf{PV} = \{p_0, p_1, \ldots\}$ and modal operators $\{\Box_\ell \mid \ell \in L\}$ according to the following syntax:

$$\varphi, \psi \ ::= \quad p_i \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \psi \ \mid \ \Box_\ell \varphi.$$

Other connectives are taken as standard abbreviations, e.g. $\Diamond_\ell \varphi = \neg\Box_\ell\neg\varphi$.

Let us recall the Kripke semantics. A *frame* $F = (W, (R_\ell)_{\ell\in L})$ consists of a nonempty set W of *points* or *worlds* and binary relations $R_\ell \subseteq W \times W$. A *model* is a pair $M = (F, \theta)$, where F is a frame and θ a *valuation* on F, i.e., a function that assigns to each variable p a set $\theta(p) \subseteq W$. The *truth* of a formula φ at a point $w \in W$ in a model M (denoted by $M, w \models \varphi$ or $F, \theta, w \models \varphi$) is defined in a standard way. In particular, $M, w \models \Box_\ell\varphi$ iff $M, u \models \varphi$ for all points $u \in W$ with $wR_\ell u$. A formula φ is called *valid* at a point w of a frame F (notation: $F, w \models \varphi$) if $F, \theta, w \models \varphi$ for all valuations θ.

Next fix a countable set Var of individual variables and consider the first-order (FO) language with equality in the signature of binary relation symbols[1] R_ℓ, for each $\ell \in L$. Observe that a frame F can serve as an interpretation for this language. Hence, given a FO formula $A(x_1, \ldots, x_n)$ with n free variables and n elements $e_1, \ldots, e_n \in W$, the relation $F \models A(e_1, \ldots, e_n)$ is well-defined.

Now we come to the central definition of our paper, first proposed by Kracht in [25]. Intuitively, it formalizes the notion of a first-order formula $A(x_1, \ldots, x_n)$ and a tuple of modal formulas $\langle\varphi_1, \ldots, \varphi_n\rangle$ being equivalent in some sense. Unless otherwise stated, below we assume that $n \geq 1$, so that we do not consider closed FO formulas.

Definition 2.1. A FO formula $A(x_1, \ldots, x_n)$ *corresponds* to a tuple of modal formulas $\langle\varphi_1, \ldots, \varphi_n\rangle$ if, for any frame F and any points $e_1, \ldots, e_n$ in F, the equivalence holds:

$$F \models A(e_1, \ldots, e_n) \iff \text{for every valuation } \theta \text{ there is } i \leq n \text{ with } F, \theta, e_i \models \varphi_i.$$

In this case we write $A(\vec{x}) \leftrightsquigarrow \langle\varphi_1, \ldots, \varphi_n\rangle$. A formula $A(\vec{x})$ is *modally definable* if it corresponds to some tuple of modal formulas. For $n = 1$ this yields the classical definition of local correspondence between a FO formula $A(x)$ with one free variable and a modal formula φ.

Let us write $F, \theta \models e{:}\,\varphi$ as a shortcut for $F, \theta, e \models \varphi$ and allow for disjunctions of expressions $e{:}\,\varphi$. Then we can rewrite the above equivalence as follows:

$$F \models A(e_1, \ldots, e_n) \iff \text{for every valuation } \theta \text{ we have } F, \theta \models e_1{:}\,\varphi_1 \vee \ldots \vee e_n{:}\,\varphi_n.$$

Or even shorter, using the notion of validity (see also Definition 2.4 below):

$$F \models A(e_1, \ldots, e_n) \quad \iff \quad F \models e_1{:}\,\varphi_1 \vee \ldots \vee e_n{:}\,\varphi_n.$$

Here are some examples (proofs are left to the reader):

- $A(x, y) = x\mathsf{R}y$ corresponds to the pair of modal formulas $\langle\Diamond p, \neg p\rangle$,
- $A(x, y) = \exists z\,(x\mathsf{R}z \wedge y\mathsf{R}z)$ corresponds to the pair $\langle\Diamond p, \Diamond\neg p\rangle$,
- $A(x, y) = \exists z\,(x\mathsf{R}z \wedge z\mathsf{R}x \wedge y\mathsf{R}z \wedge z\mathsf{R}y)$ corresponds to $\langle\neg p \vee \Diamond\neg q, \neg r \vee \Diamond(q \wedge \Diamond p \wedge \Diamond r)\rangle$.

The set of modally definable FO formulas is closed under disjunction. Indeed, if $A(x_1, \ldots, x_n)$ corresponds to $\langle\varphi_1, \ldots, \varphi_n\rangle$ and $B(x_1, \ldots, x_n)$ corresponds to $\langle\psi_1, \ldots, \psi_n\rangle$, then $A \vee B$ correspond to $\langle\varphi_1 \vee \psi_1, \ldots, \varphi_n \vee \psi_n\rangle$, provided that the tuples $\langle\varphi_1, \ldots, \varphi_n\rangle$ and $\langle\psi_1, \ldots, \psi_n\rangle$ have no common propositional variables (which can be assumed w.l.o.g.). On the contrary, conjunction of modally definable FO formulas is not always definable, as the following example shows.

Example 2.2. Reflexivity is known to be modally definable: the formula $A(x, y) = x\mathsf{R}x$ corresponds to the pair $\langle p \to \Diamond p, \bot\rangle$, and the formula $B(x, y) = y\mathsf{R}y$ corresponds to $\langle\bot, p \to \Diamond p\rangle$. Let us show that their conjunction $C(x, y) = x\mathsf{R}x \wedge y\mathsf{R}y$ is *not* modally definable. Assume the contrary, i.e., that C corresponds to a pair of modal formulas $\langle\varphi, \psi\rangle$.

Let $W = \{a, b\}$ and consider two frames $F_1 = (W, \{\langle a, a\rangle\})$ and $F_2 = (W, \{\langle b, b\rangle\})$. Obviously, $C(a, b)$ is false in both F_1 and F_2. Hence there exist valuations[2] $\theta_1, \theta_2\colon W \to 2^{\mathsf{PV}}$ such that, denoting $M_1 = (F_1, \theta_1)$ and $M_2 = (F_2, \theta_2)$, we have $M_1, a \not\models \varphi$ and $M_2, b \not\models \psi$.

Now consider a frame $F = (W, \{\langle a, a\rangle, \langle b, b\rangle\})$. Clearly, $C(a, b)$ is true in F. In order to obtain a contradiction with modal definability of C, let us show that $F \not\models a{:}\,\varphi \vee b{:}\,\psi$. Consider a model $M = (F, \theta)$, where we put $\theta(a) := \theta_1(a)$ and $\theta(b) := \theta_2(b)$. Then the following bisimulations ([5, p. 64]) hold: $M, a \sim M_1, a$ and $M, b \sim M_2, b$. Hence $M, a \not\models \varphi$ and $M, b \not\models \psi$. Thus, $F, \theta \not\models a{:}\,\varphi \vee b{:}\,\psi$.

[1] We use R_ℓ for a relation in a frame and R_ℓ for the corresponding predicate symbol.

[2] By definition, a valuation is a function $\theta\colon \mathsf{PV} \to 2^W$, but it can be also represented as a function $\theta'\colon W \to 2^{\mathsf{PV}}$, by putting $\theta'(w) = \{p \in \mathsf{PV} \mid w \in \theta(p)\}$.

2.1. Modal expressibility

Here we generalize the notion of modal definability so that the resulting notion is closed under both conjunction and disjunction. Recall that Var stands for the set of individual variables that are used in first-order formulas.

Definition 2.3 (Syntax)**.** *Modal expressions* have the following syntax:

$$\Phi, \Psi \ ::= \quad x\colon \varphi \ \mid \ \Phi \wedge \Psi \ \mid \ \Phi \vee \Psi,$$

where $x \in \mathsf{Var}$ and φ is an arbitrary modal formula. We can assume that negation (and hence implication) of modal expressions is available as well: $\neg$ can be pushed down through $\wedge$ and $\vee$, and $\neg(x\colon \varphi)$ can be taken as a shortcut for $x\colon \neg\varphi$. If a modal expression Φ contains individual variables $x_1, \ldots, x_n$, we indicate this as $\Phi(x_1, \ldots, x_n)$. Modal expressions can be regarded as formulas of the *hybrid logic* $\mathcal{H}(@)$, once we rewrite $x\colon \varphi$ as $@_x\varphi$ (cf. [4], p. 49, see also Chapter 14).

Definition 2.4 (Semantics)**.** The *truth* of a modal expression $\Phi(\vec{x})$ in a model M on an n-tuple of its elements $\vec{e}$ is denoted by $M \models \Phi(\vec{e})$ and defined by induction: $M \models e\colon \varphi$ iff $M, e \models \varphi$; the cases of $\wedge$ and $\vee$ are standard. *Validity* of a modal expression in a frame on a tuple of elements $F \models \Phi(\vec{e})$ is defined as usual.

Definition 2.5. A FO formula $A(\vec{x})$ *corresponds* to a modal expression $\Phi(\vec{x})$, written as $A(\vec{x}) \leftrightsquigarrow \Phi(\vec{x})$, if for every frame F and every n-tuple of its elements $\vec{e}$, the equivalence holds:

$$F \models A(\vec{e}) \quad \Longleftrightarrow \quad F \models \Phi(\vec{e}).$$

If such an expression $\Phi(\vec{x})$ exists, the FO formula $A(\vec{x})$ is called *modally expressible*.

For example, the formula $A(x, y) = x\mathsf{R}y$ corresponds to the modal expression $y\colon p \to x\colon \Diamond p$. Observe that the family of modally expressible FO formulas is closed under both conjunction and disjunction. Kracht's modal definability (Def. 2.1) is a special case of modal expressibility, with $\Phi(\vec{x}) = x_1\colon \varphi_1 \vee \ldots \vee x_n\colon \varphi_n$; modal expressions of this kind will be called *Kracht disjunctions*, since they essentially appeared, although implicitly, in [25].

A notion equivalent to our notion of a modal expression was considered even earlier by van Benthem in [3, Ch. 3] from a different perspective. Therein, he introduced a family of FO formulas with several free variables called *m-formulas* and proved that every m-formula is equivalent to a Boolean combination of (the standard translations of) expressions of the form $x_i\colon \varphi_i$, for some variables x_i and modal formulas φ_i (see our Sect. 7.1 below for the definition of the standard translation). Thus, van Benthem's m-formulas are exactly the FO translations of our modal expressions. In [3, Theorem 3.9] he proved the following result: *an arbitrary FO formula $A(\vec{x})$ with unary and binary predicate symbols is equivalent to some m-formula (i.e., $A(\vec{x})$ is equivalent on Kripke models, not frames, to some modal expression $\Phi(\vec{x})$) iff $A(\vec{x})$ is invariant for total bisimulations and generated submodels*.

3. Existential conjunctive first-order formulas

In the sequel, we investigate modal definability of existential conjunctive formulas (or $\exists\&$-formulas, for short), which are first-order formulas of the form $\exists\vec{y}\, B(\vec{x}, \vec{y})$, where B is a conjunction of atomic formulas. Whether such a formula is modally definable depends on its properties, which can be conveniently formulated in graph-theoretic terms. So, first let us introduce a graph representation of such formulas.

Suppose that we are given an $\exists\&$-formula $A(\vec{x}) = \exists\vec{y}\, B(\vec{x}, \vec{y})$, where B is a conjunction of formulas of the form $z\mathsf{R}z'$ with $\mathsf{R} \in \{\mathsf{R}_\ell \mid \ell \in L\}$, $z, z' \in (\vec{x}, \vec{y})$, $\vec{x} = (x_1, \ldots, x_n)$, $\vec{y} = (y_1, \ldots, y_m)$. To this formula we associate a two-color graph called the *diagram* (or $\exists$-diagram) of A, which is a tuple

$$D = (V, V_\bullet, V_\circ, (\Pi_\ell)_{\ell \in L}),$$

where $V = V_\bullet \cup V_\circ$, $V_\bullet = \{x_1, \ldots, x_n\}$ (*black* nodes), $V_\circ = \{y_1, \ldots, y_m\}$ (*white* nodes), $V_\bullet \cap V_\circ = \varnothing$, and binary relations $\Pi_\ell \subseteq V \times V$ are defined as follows:

$$\langle z, z' \rangle \in \Pi_\ell \iff \text{the formula } B(\vec{x}, \vec{y}) \text{ contains the conjunct } z\mathsf{R}_\ell z'.$$

So, in the diagram D, black and white nodes are the free and bound variables of the formula A, respectively, and edges correspond to conjuncts in the formula B. We will denote diagrams by $D = (V, V_\bullet, V_\circ, \vec{\Pi})$, where $\vec{\Pi} = (\Pi_\ell)_{\ell \in L}$.

Conversely, any diagram D (i.e., a graph of the above type) gives rise to an $\exists\&$-formula $A_D(\vec{x}) = \exists\vec{y}\, B_D(\vec{x}, \vec{y})$, where

$$B_D(\vec{x}, \vec{y}) = \bigwedge \Big\{ z\mathsf{R}_\ell z' \ \mid \ \langle z, z' \rangle \in \Pi_\ell,\ \ell \in L \Big\}.$$

For instance, the diagram in Figure 1 gives rise to the following $\exists\&$-formula:

$$A(x_1, x_2) = \exists y_1 \exists y_2 \exists y_3 (x_1\mathsf{R}y_1 \wedge y_1\mathsf{R}y_2 \wedge y_2\mathsf{R}y_1 \wedge y_2\mathsf{R}x_2 \wedge x_2\mathsf{R}y_3).$$

Now we can apply all graph-theoretic notions to $\exists\&$-formulas, meaning that we are talking about the associated diagrams. We assume the reader to be familiar with standard notions of graph theory, such as a (*directed* or *undirected*) *path* and *cycle*, etc.. A graph is called *connected* if any two distinct nodes are connected by a (possibly undirected) path, *acyclic* if it contains no cycles (even undirected). A node b is called *reachable* from a node a if there is a directed path from a to b.

Given a diagram D and a subset $Z \subseteq V$ of its nodes, by $D\restriction_Z$ we denote its subgraph (diagram) spanned by the set Z. A *white edge* (path, cycle) in a diagram is an edge (path, cycle) that contains only white nodes. A *white subgraph* of D is the graph (diagram) $D\restriction_{V_\circ}$. Similarly for the black color. Two diagrams D and D' with the same set of black nodes $\vec{x}$ are called *equivalent* if their formulas $A_D(\vec{x})$ and $A_{D'}(\vec{x})$ are equivalent. To any diagram $D = (V, V_\bullet, V_\circ, \vec{\Pi})$ we associate a frame $F_D = (V, \vec{\Pi})$ obtained from D by forgetting the colors of nodes.

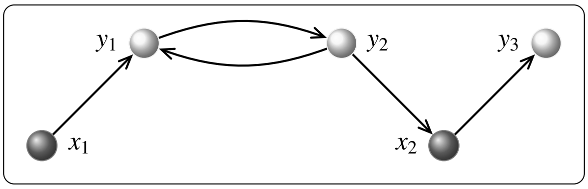


Figure 1: A diagram (minimal, accessible, and containing a white cycle).

<table>
<tr><td colspan="4">minimal ∃&-formula $A(\vec{x}) = \exists\vec{y}\, B(\vec{x},\vec{y})$</td></tr>
<tr><td>inaccessible</td><td colspan="3">accessible</td></tr>
<tr><td rowspan="3">inexpressible</td><td>white-cyclic</td><td colspan="2">white-acyclic</td></tr>
<tr><td rowspan="2">inexpressible</td><td>disconnected</td><td>connected</td></tr>
<tr><td>expressible, not definable</td><td>definable</td></tr>
</table>

Table 1: Criterion of modal definability for ∃&-formulas.

3.1. Criterion of modal definability

The main result of our paper is the criterion of modal definability (and expressibility) for ∃&-formulas. It is efficient, in the sense that there is an algorithm that, given an ∃&-formula, decides whether it is modally definable or expressible (and if so, produces the corresponding modal formulas or a modal expression). Note that in general the problem of determining whether a first-order formula (even with only one free variable) is modally definable is undecidable, due to Chagrova's result [9].

Definition 3.1. A diagram D (and its ∃&-formula) is called

- *minimal* if removing any of its edges yields a diagram not equivalent to D;
- *accessible* if every white node is reachable from some black node;
- *white-acyclic* if its white subgraph is acyclic.

The criterion is summarized in Table 1. Below we give explicit formulations of the results and provide some comments.

Criterion. Let $A(\vec{x}) = \exists\vec{y}\, B(\vec{x},\vec{y})$ be an ∃&-formula. Without loss of generality, we can assume that $A(\vec{x})$ is minimal.[3]

Indeed, given an ∃&-formula, one can efficiently build an equivalent minimal one. Simply, try to remove edges one by one and check whether the resulting formula $A'(\vec{x})$ implies $A(\vec{x})$. If it does, then the removed edge was redundant and we can repeat the process for the new formula. Note that checking whether one ∃&-formula implies another one amounts to checking the existence of a graph homomorphism (which is an NP-complete problem).

- *If $A(\vec{x})$ is minimal, but inaccessible, then it is not modally expressible, and hence not modally definable* (Lemma 5.7). This is intuitively clear: no modal formula can say anything about unreachable worlds in a Kripke frame.

- *If $A(\vec{x})$ is minimal,[4] accessible, but contains a white cycle, then it is not modally expressible, and hence not modally definable* (Theorem 5.15).

  This is one of the hardest (negative) results of our paper. Its proof employs the notion of the ultrafilter extension of a Kripke frame.

- *If $A(\vec{x})$ is accessible, white-acyclic, and connected, then it is modally definable* (Theorem 4.1). This is the most important positive result of our paper.

- *If $A(\vec{x})$ is accessible and white-acyclic, then it is modally expressible* (Theorem 4.2). *If, additionally, it is disconnected and minimal, then it is not modally definable* (Lemma 5.8). This is the only case where modal expressibility gives us more than modal definability.

4. Definable ∃&-formulas

This section is devoted to the proof of the positive results stated in Sect. 3.1.

Theorem 4.1. *If an ∃&-formula $A(\vec{x})$ is accessible, white-acyclic, and connected, then it is modally definable.*

[3] Minimality is only needed in negative (i.e., undefinability) results. For positive results it is redundant, as the algorithm that produces modal formulas or expressions does not require the minimality of an ∃&-formula.

[4] Minimality is essential here, as the formula $A(x) = \exists y\,(x\mathsf{R}x \wedge x\mathsf{R}y \wedge y\mathsf{R}y)$ illustrates: despite the presence of a white cycle $y\mathsf{R}y$, it is equivalent to $x\mathsf{R}x$ and so is modally definable. On the other hand, the formula $A'(x) = \exists y\,(x\mathsf{R}y \wedge y\mathsf{R}y)$ is minimal and known to be undefinable [19].

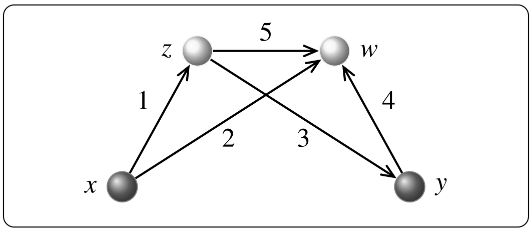


Figure 2: A modally definable diagram.

Theorem 4.2. *Every accessible and white-acyclic* $\exists\&$*-formula* $A(\vec{x})$ *is modally expressible.*

The latter theorem is a simple corollary of the former one. Indeed, the diagram of $A(\vec{x})$ is a disjoint union of connected components. Hence $A(\vec{x})$ is a conjunction of modally definable (and hence modally expressible) formulas, and so is modally expressible itself.

Before we proceed to the proof of Theorem 4.1, let us formulate this theorem for the case $n = 1$ more specifically, using the notion of a modal *generalized Sahlqvist formula* introduced by Goranko and Vakarelov [18] (we will not reproduce its rather complicated definition here).

Theorem 4.3. *If an* $\exists\&$*-formula* $A(x)$ *with one free variable is accessible and white-acyclic, then it corresponds to some modal generalized Sahlqvist formula.*

The proof is given at the end of Sect. 4.2, as it uses some notions introduced below. The remainder of Sect. 4 constitutes the proof of Theorem 4.1. Let us first illustrate the underlying idea with examples.

4.1. Examples

Here we demonstrate how to turn an $\exists\&$-formula that satisfies the conditions of Theorem 4.1 into a modal expression. Schematically, the process consists of the following stages:

$$\exists\&\text{-formula} \mapsto \text{diagram} \mapsto \text{term} \mapsto \text{system} \mapsto \text{solution} \mapsto \text{modal expression}.$$

For the time being, please do not look for the rigorous meaning of each step presented below, rather consider them as a “rule of thumb”. Everything will be explained in the subsequent subsections.

Example 4.4. Consider an $\exists\&$-formula whose diagram D is depicted in Fig. 2:

$$A(x, y) \;=\; \exists z \exists w\, (x\mathsf{R}_1 z \wedge x\mathsf{R}_2 w \wedge z\mathsf{R}_3 y \wedge y\mathsf{R}_4 w \wedge z\mathsf{R}_5 w).$$

Observe that it is accessible, white-acyclic, and connected, thus satisfies the preconditions of Theorem 4.1. First, we build a term that “describes” the diagram D “from the viewpoint” of the node x:

$$t \;=\; \Diamond_1\big(\Diamond_3 y \wedge \Diamond_5(\ominus_2 x \wedge \ominus_4 y)\big).$$

Intuitively, it says that the node x can 1-see a node (labeled by z) that 3-sees y and 5-sees a node (labeled by w) that is 2-seen from x and 4-seen from y. In order to obtain modal formulas, we need to eliminate $\ominus$’s and the nominals x and y. To this end, we introduce propositional variables p, q, r that will “stand for” the corresponding subterms of t:

$$t \;=\; \Diamond_1\big(\Diamond_3 \underbrace{\;y\;}_{p} \wedge \Diamond_5(\underbrace{\ominus_2 x}_{q} \wedge \underbrace{\ominus_4 y}_{r})\big).$$

Thus we obtain the following “system” and “solve” it as described in Sect. 4.4.1:

$$\begin{cases} \;\;y \;\Rightarrow\; p \\ \ominus_2 x \;\Rightarrow\; q \\ \ominus_4 y \;\Rightarrow\; r \end{cases} \quad\rightsquigarrow\quad \begin{cases} y \;\Rightarrow\; p \\ x \;\Rightarrow\; \Box_2 q \\ y \;\Rightarrow\; \Box_4 r \end{cases}$$

Now we use this to build a modal expression: its antecedent is the conjunction of the lines of the above solution, whereas its consequent is $x{:}\, t'$, where t' is a modal formula obtained by substituting the variables p, q, r for the appropriate subterms into t:

$$\big(y{:}\, p \;\wedge\; x{:}\, \Box_2 q \;\wedge\; y{:}\, \Box_4 r\big) \;\rightarrow\; x{:}\, \Diamond_1(\Diamond_3 p \wedge \Diamond_5(q \wedge r)).$$

Clearly, it can be equivalently rewritten into a modal expression of the form $x{:}\,\varphi \vee y{:}\,\psi$, where φ and ψ are ordinary modal formulas. So, the FO formula $A(x, y)$ is modally definable indeed.

Example 4.5. For the same formula $A(x, y)$, we could proceed alternatively. Let us start our "traversal" of the diagram D with the edge xR_2w. Then we obtain a different term t_1 that again "describes" the diagram D "from the viewpoint" of the node x:

$$t_1 \;=\; \Diamond_2\Big(\underbrace{\diamondminus_4 y}_{q} \;\wedge\; \underbrace{\diamondminus_5(\Diamond_3 \overbrace{\;\;y\;\;}^{p} \;\wedge \diamondminus_1 x)}_{r}\Big).$$

This time we obtain (and solve) the following "system":

$$\begin{cases} \quad\; y \;\Rightarrow\; p \\ \quad \diamondminus_4 y \;\Rightarrow\; q \\ \diamondminus_5(\Diamond_3 p \wedge \diamondminus_1 x) \;\Rightarrow\; r \end{cases} \quad\rightsquigarrow\quad \begin{cases} y \;\Rightarrow\; p \\ y \;\Rightarrow\; \Box_4 q \\ x \;\Rightarrow\; \Box_1(\Diamond_3 p \to \Box_5 r) \end{cases}$$

Finally, we obtain a modal expression similarly to the above:

$$\Big(y\colon p \;\wedge\; y\colon \Box_4 q \;\wedge\; x\colon \Box_1(\Diamond_3 p \to \Box_5 r)\Big) \;\to\; x\colon \Diamond_2(q \wedge r).$$

Although the resulting modal expression differs from that in the previous example, the reader is encouraged to verify that both expressions correspond to our $\exists$&-formula $A(x, y)$.

4.2. Safe terms

Here we describe a wide family of first-order formulas with several free variables that are modally definable (see Theorem 4.13 below). They are obtained from the so called safe terms[5] introduced (in different notation) in [20].

Definition 4.6 (Syntax)**.** *Terms* are built up according to the following syntax, where x ranges over the set Var of *nominals* and $\ell \in L$:

$$t, s \;::=\; \top \mid \bot \mid x \mid t \wedge s \mid t \vee s \mid \Diamond_\ell t \mid \Box_\ell t \mid \diamondminus_\ell t.$$

One can see that terms are what is known as *hybrid formulas* [4, p. 49] extended with converse diamonds $\diamondminus$, but containing no propositional variables, negations, or converse boxes $\boxminus$. Terms *with* propositional variables will appear later in Sect. 4.4.

Definition 4.7 (Semantics)**.** Given a term $t(\vec{x})$ that contains nominals $\vec{x} = (x_1, \ldots, x_n)$, a frame $F = (W, (R_\ell)_{\ell \in L})$, and its elements b and $\vec{a} = (a_1, \ldots, a_n)$, we define the *truth relation* $F, b \models t(\vec{a})$ inductively as follows (the cases for the Boolean connectives $\top, \bot, \wedge, \vee$ are standard):

$$\begin{array}{lcl} F, b \models a & \rightleftharpoons & b = a \\ F, b \models \Diamond_\ell t & \rightleftharpoons & \exists c \in W\,(\,bR_\ell c \;\;\&\;\; F, c \models t\,) \\ F, b \models \Box_\ell t & \rightleftharpoons & \forall c \in W\,(\,bR_\ell c \;\Rightarrow\; F, c \models t\,) \\ F, b \models \diamondminus_\ell t & \rightleftharpoons & \exists c \in W\,(\,cR_\ell b \;\;\&\;\; F, c \models t\,) \end{array}$$

Definition 4.8 (Standard translation)**.** To every term $t(\vec{x})$ we associate a FO formula $t^*(\vec{x}; y)$ (or $t^*(y)$ for short, when $\vec{x}$ is clear from context); its free variables are y and those nominals x_i that occur in t. The translation t^* is defined by induction (here z is a fresh variable; the cases for the Boolean connectives $\top, \bot, \wedge, \vee$ are standard):

$$\begin{array}{rcl} x_i^*(y) & := & (y = x_i) \\ (\Diamond_\ell t)^*(y) & := & \exists z\,(\,y\mathsf{R}_\ell z \;\wedge\; t^*(z)\,) \\ (\Box_\ell t)^*(y) & := & \forall z\,(\,y\mathsf{R}_\ell z \to t^*(z)\,) \\ (\diamondminus_\ell t)^*(y) & := & \exists z\,(\,z\mathsf{R}_\ell y \;\wedge\; t^*(z)\,) \end{array}$$

Since the standard translation "mimics" the semantics of terms, it is clear that $F, b \models t(\vec{a})$ iff $F \models t^*(\vec{a}; b)$. Below, we omit the subscript ℓ in operators $\Diamond, \Box, \diamondminus$ whenever possible, assuming that it ranges over L.

Definition 4.9. *Simple*[6] terms are defined by induction (notice 'or' in the third item):

- each nominal x_i is a simple term;
- if t is simple, then so is $\diamondminus t$;
- if t ***or*** s is simple, then so is $t \wedge s$.

A term is called *safe* if all its subterms of the form $\diamondminus t$ are simple.

Examples of simple terms are x, $\diamondminus x$, $\diamondminus(\diamondminus x \wedge \Diamond\diamondminus y)$; they are also safe. The term $\Diamond x$ is safe but not simple; the term $\diamondminus x \wedge \diamondminus\Diamond x$ is simple, but not safe; the term $\diamondminus\Diamond x$ is neither simple, nor safe. Simple terms can be characterized in terms of their syntactic trees as follows. A $\diamondminus\wedge$*-path* in the syntactic tree of a term is a path from its root to a leaf in which all nodes (except for its end) are labeled by either $\wedge$ or $\diamondminus$.

[5] Originally, terms represent (in some sense not discussed here) minimal valuations in van Benthem's substitution algorithm for computing first-order equivalents of generalized Sahlqvist formulas; the latter were introduced in [18].

[6] Our terminology differs from that in [20]. What we call here: (a) a term, (b) a simple term, (c) a safe term, corresponds in [20] to: (a) an L-expression, (b) an L-expression ϕ safe for ϕ, (c) a positive combination of L-expressions or a quasi-safe L-expression.

Lemma 4.10. *A term is simple iff its syntactic tree contains a* $\diamondminus\wedge$*-path from the root to some nominal.*

Let us write $F \models b\colon t(\vec{a})$ as an alternative notation for $F, b \models t(\vec{a})$. This gives us the semantics for "expressions" of the form $y\colon t(\vec{x})$. Note that $y\colon t(\vec{x})$ can be regarded as a hybrid formula $@_y t(\vec{x})$.

Definition 4.11. We say that a FO formula $C(\vec{x}, y)$ is *equivalent* to $y\colon t(\vec{x})$, written as $C(\vec{x}, y) \leftrightarrow y\colon t(\vec{x})$, if for every frame F and all its points $\vec{a}, b$, we have $F \models C(\vec{a}, b)$ iff $F \models b\colon t(\vec{a})$.

We say that $y\colon t(\vec{x})$ *corresponds* to a modal expression $\Phi(\vec{x}, y)$ and write $y\colon t(\vec{x}) \leftrightsquigarrow \Phi(\vec{x}, y)$ if, for every frame F and all its elements $\vec{a}, b$, we have $F \models b\colon t(\vec{a})$ iff $F \models \Phi(\vec{a}, b)$. A term $t(\vec{x})$ is called *modally expressible* if $y\colon t(\vec{x})$ corresponds to some modal expression $\Phi(\vec{x}, y)$; if $\Phi(\vec{x}, y)$ here is a Kracht disjunction $x_1\colon \varphi_1 \vee \ldots \vee x_n\colon \varphi_n \vee y\colon \varphi$, the term t is called *modally definable*.

Now we are ready to outline our plan of the proof of Theorem 4.1. For a given $\exists\&$-formula $A(\vec{x})$ that satisfies the conditions of the theorem, we build a safe term $t(\vec{x})$ such that $A(\vec{x})$ is equivalent to $x_i\colon t(\vec{x})$, for some i. Then for this safe term t we build a corresponding modal expression $\Phi(\vec{x}, y)$. As a consequence, $A(\vec{x})$ corresponds to $\Phi(\vec{x}, x_i)$, so we are done. Formally, we establish the following two theorems.

Theorem 4.12. *Given an accessible, white-acyclic, and connected* $\exists\&$*-formula* $A(\vec{x})$ *and any* $i \le n$*, one can build in polynomial time a safe term* $t(\vec{x})$ *such that* $A(\vec{x})$ *is equivalent to* $x_i\colon t(\vec{x})$*.*

For the proof, see Sect. 4.3. Note that the resulting safe term t will contain only nominals and operators $\top, \wedge, \Diamond, \diamondminus$.

Theorem 4.13. *Every safe term is modally definable. Moreover, given a safe term* $t(\vec{x})$*, one can build in polynomial time a modal expression (even a Kracht disjunction)* $\Phi(\vec{x}, y)$ *that corresponds to* $y\colon t(\vec{x})$*.*

The proof is given in Sect. 4.4. We prove this theorem for safe terms t that may additionally involve the operators $\bot, \vee, \Box$ (as it costs us almost nothing). Note that Theorem 4.13 has already been proved in Lemma 35 of [20]. Here we re-establish this result and present an explicit polynomial algorithm that produces a modal expression Φ for a given safe term t.

Thus, FO formulas of the form $t^*(\vec{x}; y)$, for safe terms t, yield a family of modally definable FO formulas with several free variables.

We are ready to prove Theorem 4.3, which says: *If an* $\exists\&$*-formula* $A(x)$ *with one free variable is accessible and white-acyclic, then it corresponds to some generalized Sahlqvist modal formula.*

Proof. Since $A(x)$ has a single free variable, accessibility of its diagram implies connectivity. Then, by Theorem 4.12, $A(x)$ is equivalent to $x\colon t(x)$, for some safe term t. Now, the (standard translation of the) expression $x\colon t(x)$ is a special case of a first-order *generalized Kracht formula* introduced in [20, Def. 29]. Therefore, by Theorem 30 from [20], $x\colon t(x)$, and hence $A(x)$, corresponds to some generalized Sahlqvist modal formula. $\Box$

4.3. From diagrams to safe terms

Here we prove Theorem 4.12. Let D be a diagram, $A_D(\vec{x})$ its $\exists\&$-formula, x_i its black node. A pair (D, x_i) will be called a $\bullet$*-diagram*; it will be called acyclic, accessible, etc., if D is so.

Definition 4.14. We say that a term $t(\vec{x})$ *represents* a $\bullet$-diagram (D, x_i) if the formula $A_D(\vec{x})$ is equivalent to $x_i\colon t(\vec{x})$.

Our aim is to show that any accessible, *white-acyclic*, connected $\bullet$-diagram is representable by a safe term. To this end, we first show that any such diagram can be obtained from an *acyclic* one by merging some black nodes (Lemma 4.17). Secondly, for an acyclic $\bullet$-diagram, its representing term can be built easily (Lemma 4.18).

Let $D = (V, V_\bullet, V_\circ, \vec{\Pi})$ and $D' = (V', V_\bullet', V_\circ, \vec{\Pi}')$ be diagrams with the same white subgraph: $\Pi_\ell\restriction_{V_\circ} = \Pi'_\ell\restriction_{V_\circ}$, for each $\ell \in L$. Here $V_\bullet = \{x_1, \ldots, x_n\}$ and $V_\bullet' = \{x'_1, \ldots, x'_\kappa\}$.

Definition 4.15. We say that D' is obtained by *merging black nodes* in D if there is a function $f\colon V \to V'$ (we call it a $\bullet$*-merging function*) that is identical on white nodes: $f(y) = y$ for all $y \in V_\circ$, surjectively maps $V_\bullet$ onto $V_\bullet'$, and the induced function on pairs of nodes defined by $\langle u, v\rangle \mapsto \langle f(u), f(v)\rangle$ surjectively maps Π_ℓ onto Π'_ℓ, for each $\ell \in L$.

Observe that merging black nodes in a diagram D corresponds to substituting in its $\exists\&$-formula $A_D(\vec{x})$ some free variables for some other free variables, and then removing duplicate conjuncts. To be more precise, if $f(x_i) = z_i$ for all $1 \le i \le n$, where $z_i \in \vec{x}' = (x'_1, \ldots, x'_\kappa)$, then the formula $A_{D'}(\vec{x}')$ is equivalent to $A_D(\vec{z})$.

Lemma 4.16. *Any* $\bullet$*-merging function preserves representability of* $\bullet$*-diagrams by safe terms.*

Proof. Assume that a safe term $t(\vec{x})$ represents a $\bullet$-diagram (D, x_i), so, $A_D(\vec{x}) \leftrightarrow x_i\colon t(\vec{x})$. Substituting $\vec{z}$ for $\vec{x}$ yields $A_D(\vec{z}) \leftrightarrow z_i\colon t(\vec{z})$. By the above remark, $A_{D'}(\vec{x}') \leftrightarrow A_D(\vec{z})$. Therefore, $A_{D'}(\vec{x}') \leftrightarrow z_i\colon t(\vec{z})$, hence the term $t(\vec{z})$ represents the $\bullet$-diagram (D', z_i). It remains to note that $t(\vec{z})$ is a safe term, for renaming nominals preserves safety. $\Box$

Lemma 4.17. *Any* ***white-acyclic*** *diagram* D *can be obtained by merging black nodes in some* ***acyclic*** *diagram* D'*. If additionally* D *is accessible and connected, then* D' *can be chosen so too. The diagram* D' *and the* $\bullet$*-merging function from* D' *onto* D *can be built efficiently (in polynomial time).*

Proof. By induction on the number of (undirected) simple[7] cycles in D. Let us show how to reduce this number by one. Assume that D has an simple cycle. Since D is white-acyclic, this cycle contains a black node, say $x \in V_\bullet$, and so has a form:

$$x = v_0 \xrightarrow{\ell_1} v_1 \dots \xrightarrow{\ell_\kappa} v_\kappa = x,$$

where all nodes v_i are distinct except for that $v_0 = v_\kappa$, and $\kappa \geq 1$.

Now we add to D a fresh black node x' and replace the edge $x \xrightarrow{\ell_1} v_1$ with the edge $x' \xrightarrow{\ell_1} v_1$ of the same orientation.[8] The resulting diagram D' has at least one simple cycle less than D has. Indeed, every simple cycle in D' belongs to D (since the node x' has degree 1 and so cannot occur in simple cycles), and the above cycle does not belong to D'. The projection from D' onto D is built in the obvious way: let $f(x') = x$ and let f be identical on all other nodes. Note that the diagram D' is accessible and connected if D was so. □

Lemma 4.18. *Every accessible, acyclic, connected* $\bullet$*-diagram is representable by a safe term, which can be built in polynomial (linear) time.*

Proof. Given such a $\bullet$-diagram (D, r), we introduce a non-transitive relation $\prec$ on the set V of its nodes: $x \prec y$ iff x belongs to the (unique undirected) path from r to y, and the nodes x and y are adjacent (i.e., linked by an edge in any direction). Clearly, $(V, \prec)$ is a directed tree with the root r.

Now, to each node z of D, we associate a term t_z by induction from leaves to the root of the tree $(V, \prec)$:

- For z a leaf of the tree, we put $t_z = \begin{cases} z, & \text{if } z \text{ is a black node,} \\ \top, & \text{if } z \text{ is a white node.} \end{cases}$
- For z not a leaf of the tree, we put $t_z = \begin{cases} z \wedge s_z, & \text{if } z \text{ is a black node,} \\ s_z, & \text{if } z \text{ is a white node.} \end{cases}$

  Here
$$s_z \quad = \quad \bigwedge \{ \Diamond_\ell t_v \mid z \prec v,\ \langle z, v\rangle \in \Pi_\ell,\ \ell \in L \} \wedge \bigwedge \{ \diamondminus_\ell t_v \mid z \prec v,\ \langle v, z\rangle \in \Pi_\ell,\ \ell \in L \}.$$

It remains to prove that the term t_r is safe and represents the $\bullet$-diagram (D, r).

Claim 1. *The term t_r is safe.*

Indeed, take any subterm of t_r of the form $\diamondminus_\ell t$. By the above construction, $t = t_v$ for some node v. Moreover, $\diamondminus_\ell t_v$ is a conjunct in s_z for some node z with $z \prec v$ and $\langle v, z\rangle \in \Pi_\ell$ (see the second line of the definition of s_z). We need to show that $\diamondminus_\ell t$ (or, equivalently, t itself) is simple.

If v is a black node, then $t_v = v \wedge s_v$ is simple, since v is a nominal.

If v is a white node, then since D is accessible, there is a directed path p from some black node x to v. Note that $x \neq r$; indeed, the only (undirected) path from r to v goes through z (since $z \prec v$), and the last edge in that path is directed from v to z, not vice versa (since $\langle v, z\rangle \in \Pi_\ell$). Then, by induction on the length of the path p, we can show that, for every node y in this path (including $y = v$), the term t_y is simple. Induction base ($v = x$) is trivial. As for induction step, t_y is a conjunction with at least one conjunct of the form $\diamondminus_\kappa t_u$, where the node u is closer to x and, by induction hypothesis, t_u is simple, hence so is $\diamondminus_\kappa t_u$ and the whole conjunction t_y.

Claim 2. *The term t_r represents the* $\bullet$*-diagram (D, r).*

For every node z in D, denote by D_z the diagram obtained from D by taking the z-rooted subtree of $(V, \prec)$ and making the node z black (if it was white in D); the colors of other nodes of D_z and the edges between nodes of D_z are the same as in D. It suffices to prove, by induction from leaves to the root of $(V, \prec)$, the following statement: *for every node z in D, the term t_z represents the* $\bullet$*-diagram (D_z, z)*. In symbols, we need to prove the equivalence: $z\colon t_z(\vec{x}) \leftrightarrow A_{D_z}(\vec{x})$. Below, we omit $\vec{x}$. Since an "expression" of the form $v\colon t(\vec{x})$ is equivalent to the FO formula $t^*(\vec{x}; v)$, below we freely use $v\colon t$ in FO formulas.

Induction base. If z is a leaf of the tree $(V, \prec)$, then t_z is either z or $\top$, hence $z\colon t_z$ is equivalent to $\top$. At the same time, the formula A_{D_z} is an empty conjunction and hence equivalent to $\top$, too.

Induction step. Assume that z is not a leaf. For simplicity, let z have only two children in $(V, \prec)$, a black one $x \in V_\bullet$ and a white one $y \in V_\circ$, linked to z by the edges $\langle z, x\rangle \in \Pi_\ell$ and $\langle y, z\rangle \in \Pi_\kappa$, for some $\ell, \kappa \in L$. (If z has more children or edges are oriented differently, the argument is the same, but notation becomes cumbersome.) To avoid trivial cases, let us also assume that neither x nor y is a leaf. So, we have $s_z = \Diamond_\ell t_x \wedge \diamondminus_\kappa t_y$, and the formula A_{D_z} is related to the formulas A_{D_x} and A_{D_y} as follows:

$$A_{D_z} \quad \leftrightarrow \quad (\, z\mathsf{R}_\ell x \wedge A_{D_x}\,) \wedge \exists y\, (\, y\mathsf{R}_\kappa z \wedge A_{D_y}\,). \tag{1}$$

In order to prove that $z\colon t_z \leftrightarrow A_{D_z}$, observe that the following chain of equivalences holds:

$$z\colon t_z \overset{(a)}{\longleftrightarrow} z\colon s_z \overset{(b)}{\longleftrightarrow} z\colon (\Diamond_\ell t_x \wedge \diamondminus_\kappa t_y) \overset{(c)}{\longleftrightarrow} \exists u\, (\, z\mathsf{R}_\ell u \wedge u\colon t_x\,) \wedge \exists v\, (\, v\mathsf{R}_\kappa z \wedge v\colon t_y\,)$$
$$\overset{(d)}{\longleftrightarrow} (\, z\mathsf{R}_\ell x \wedge x\colon t_x\,) \wedge \exists y\, (\, y\mathsf{R}_\kappa z \wedge y\colon t_y\,). \tag{2}$$

[7] A cycle is called *simple* if all its nodes are distinct. Of course, we have to count only simple cycles, since a cyclic graph always has infinitely many non-simple cycles.

[8] In particular, if we had a loop in D, i.e., a cycle of the length $\kappa = 1$, then the edge $x\Pi_\ell x$ is replaced with $x'\Pi_\ell x$.

Here (a) holds since t_z is either s_z or $z \wedge s_z$; (b) holds due to $s_z = \Diamond_\ell t_x \wedge \Diamond\!\!\!\!-_\kappa t_y$; (c) uses the equivalence $z\colon \Diamond t \leftrightarrow \exists w\,(z\mathsf{R}w \wedge w\colon t)$ and a similar one for $\Diamond\!\!\!\!-$; (d) uses two facts: first, x is black and so $t_x = (x \wedge s_x)$, hence $u\colon t_x \leftrightarrow (u = x) \wedge u\colon t_x$, thus we can drop $\exists u$ and replace u with x; secondly, y is white and hence does not occur in t_y, so we can rename v into y.

By induction hypothesis, $x\colon t_x \leftrightarrow A_{D_x}$ and $y\colon t_y \leftrightarrow A_{D_y}$. Therefore, the last formulas in (2) and (1) are equivalent, so we are done. □

4.4. From safe terms to modal formulas

Here we prove Theorem 4.13. Let $t(\vec{x})$ be a safe term. We show how to transform $y\colon t(\vec{x})$ into a corresponding modal expression of the form (which is obviously equivalent to a Kracht disjunction defined in Sect. 2.1):

$$(x_1\colon \varphi_1 \wedge \ldots \wedge x_n\colon \varphi_n) \;\to\; y\colon \varphi \tag{3}$$

Idea. We shall eliminate nominals and $\Diamond\!\!\!\!-$'s from t one by one and substitute propositional variables p_i for some of its subterms, until t contains no nominals or $\Diamond\!\!\!\!-$'s and hence is an ordinary modal formula. At intermediate steps of the transformation of a term into a modal formula we will obtain "terms with variables" (let us call them *terms* too) whose syntax is:

$$\top \mid \bot \mid x \mid p \mid t \wedge s \mid t \vee s \mid \Diamond t \mid \Box t \mid \Diamond\!\!\!\!-\, t$$

Simple and *safe* terms are defined as in Def. 4.9. In particular, p is a safe but not simple term.

Furthermore, we will have to consider "mixed" expressions of the form $\Psi(\vec{x}) \to y\colon t(\vec{x})$, where Ψ is an ordinary modal expression (see Def. 2.3) and t is a term (possibly with variables). In particular, $y\colon t(\vec{x})$ can be regarded as a "mixed" expression. Semantics for them (the notions of *truth* in a model and *validity* in a frame, on a given tuple of worlds) can be given just by combining Definitions 2.4 and 4.7. For these expressions, we introduce the following notion.

Definition 4.19. Two expressions $\mathcal{E}(\vec{x})$ and $\mathcal{E}'(\vec{x})$ are called *equi-valid* if, for every frame F and all its points $\vec{a}$, we have: $F \models \mathcal{E}(\vec{a}) \iff F \models \mathcal{E}'(\vec{a})$.

Now we are ready to prove our theorem. By induction on the number of *occurrences* of nominals in t, we prove a slightly more general statement: a "mixed" expression $\Phi(\vec{x}, y)$ of the form

$$(x_1\colon \varphi_1 \wedge \ldots \wedge x_n\colon \varphi_n) \to y\colon t(\vec{x}) \tag{4}$$

where $t(\vec{x})$ is a safe term, can be transformed into an *equi-valid* modal expression of the form (3).

Induction base. If the term t in (4) contains no nominals, then (since t is safe) it contains no $\Diamond\!\!\!\!-$'s either and thus is an ordinary modal formula. Therefore, Φ is already of the form (3).

Induction step. It suffices to show, given an expression $\Phi(\vec{x}, y)$ of the form (4), how to reduce the number of occurrences of nominals in t by 1. We need an auxiliary notion.

Usually, the *depth* of an occurrence of a subterm in a term is defined as the number of operators in the scope of which this subterm lies. Here we need a similar measure, which however ignores the operators $\wedge$ and $\Diamond\!\!\!\!-$. Formally, the *$\Diamond\!\!\!\!-\wedge$-ignoring depth* of an occurrence[9] of a nominal x in a term t is denoted by $d(x, t)$ and defined by induction:

$$\begin{array}{rcrcll} & & d(x,x) & := & 0, & \\ d(x, t \wedge s) & = & d(x, s \wedge t) & := & d(x,t), & \text{where } x \text{ is in } t, \\ d(x, t \vee s) & = & d(x, s \vee t) & := & d(x,t)+1, & \text{where } x \text{ is in } t, \\ d(x, \Box t) & = & d(x, \Diamond t) & := & d(x,t)+1, & \\ & & d(x, \Diamond\!\!\!\!-\, t) & := & d(x,t). & \end{array}$$

For example, for a term $t = \Diamond\!\!\!\!-(\Diamond x \wedge \Diamond\!\!\!\!-\, y \wedge (\Diamond\!\!\!\!-\Diamond\!\!\!\!-\, z \vee \Box x))$ we have $d(x,t) = 1$ for the first occurrence of x, $d(y,t) = 0$, $d(z,t) = 1$ and $d(x,t) = 2$ for the second occurrence of x.

Now, given an expression $\Phi(\vec{x}, y)$ of the form (4), let us run the following procedure.

1. Find in t the deepest with respect to $d(\cdot, t)$ occurrence of a nominal; let it be $x \in \vec{x}$.
2. Find in t the maximal (with respect to the *subterm–term* relation) *simple* subterm $s = s(x)$ containing this occurrence of x and no other occurrences of any nominals (including x). Such a term s exists, since x itself is simple.
3. Replace in t this occurrence of s with a fresh variable p, thus obtaining a term $t' = t'(\vec{x}, p) = t[s \mapsto p]$.
4. Solve the "equation" $s(x) \Rightarrow p$, i.e., transform it into $x \Rightarrow \varphi(p)$, where φ is a modal formula, as described in Sect. 4.4.1 below.
5. Finally, transform $\Phi(\vec{x}, y)$ as follows: add $x\colon \varphi$ into its premise and replace t with t' in its conclusion. The resulting expression $\Phi'(\vec{x}, y)$ looks as follows:

$$(x_1\colon \varphi_1 \wedge \ldots \wedge x_n\colon \varphi_n) \wedge x\colon \varphi(p) \;\to\; y\colon t'(\vec{x}, p)$$

   It is of the form (4), but with a fewer number of occurrences of nominals in t' than in t.

[9]Below, we deal with *occurrences* of subterms, nominals, etc., although we do not introduce special notation for occurrences, and even omit the word 'occurrence' sometimes.

It remains to show that the term t' obtained at Step 3 is safe (see Lemma 4.20), explain how to "solve equations" at Step 4 (see Sect. 4.4.1), and finally prove that the expressions $\Phi(\vec{x}, y)$ and $\Phi'(\vec{x}, y)$ are equi-valid (see Lemma 4.23).

Lemma 4.20. *The term t' obtained at Step 3 is safe.*

Proof. Assume on the contrary that t' has a non-simple subterm of the form $\diamondminus r$. Clearly, $\diamondminus r$ contains p, since otherwise $\diamondminus r$ is inherited from t, where all subterms of this form are simple. So, the subterm $\diamondminus r(s)$ of t is simple, but the subterm $\diamondminus r(p)$ of t' is not. Let us consider the syntactic trees of t and t'.

Claim. *The path from $\diamondminus r(s)$ to the chosen occurrence of x is a $\diamondminus\wedge$-path.*

Assume the contrary. Since the term $\diamondminus r(s)$ is simple, by Lemma 4.10 there is a $\diamondminus\wedge$-path γ from it to some occurrence of nominal, say y (possibly, $y = x$). This occurrence of y is outside s, because s contains no occurrences of any nominals other than the chosen occurrence of x, and the path from $\diamondminus r(s)$ to x is not a $\diamondminus\wedge$-path, by our assumption. But then the substitution $[s \mapsto p]$ does not affect the path γ. Thus, γ is a $\diamondminus\wedge$-path in t' from $\diamondminus r(p)$ to y, and so $\diamondminus r(p)$ is simple, contrary to its choice.

Let us call a subterm of t *nice* if it is simple, contains the chosen occurrence of the nominal x and no other occurrences of any nominals (including x). So, by assumption, s is the maximal nice subterm of t.

Consider the immediate superterm σ of s in t. Since s is a proper subterm of $\diamondminus r(s)$, the term σ is a subterm of $\diamondminus r(s)$. By the Claim, the main connective of σ is either $\diamondminus$ or $\wedge$, because it is on the $\diamondminus\wedge$-path from $\diamondminus r(s)$ to x. But the case $\sigma = \diamondminus s$ is impossible, since otherwise σ is nice and bigger than s. Hence $\sigma = s \wedge s'$, for some term s'. Then s' contains a nominal, say y (possibly, $y = x$), for otherwise σ is nice and bigger than s. The path from $\diamondminus r(s)$ to y is not a $\diamondminus\wedge$-path, since otherwise we would have the same path in t' and hence $\diamondminus r(p)$ would be simple. At the same time, the path from $\diamondminus r(s)$ to x *is* a $\diamondminus\wedge$-path, by the Claim. Therefore, y is deeper than x with respect to the $\diamondminus\wedge$-ignoring depth measure $d(\cdot, t)$, which contradicts the choice of x. □

4.4.1. Solving "equations"

Suppose that $s(x)$ is a simple term with a single occurrence of a single nominal x, and p is a propositional variable. By *solving* the "equation" $s(x) \Rightarrow p$ we mean transforming it into an "equation" of the form $x \Rightarrow \varphi(p)$, where φ is a modal formula, using the following rules (we omit the other rule for conjunction with $\psi \wedge t$ in the premise):

$$\frac{(t \wedge \psi) \;\Rightarrow\; \varphi}{t \;\Rightarrow\; (\psi \to \varphi)} \qquad\qquad \frac{\diamondminus t \;\Rightarrow\; \varphi}{t \;\Rightarrow\; \Box\varphi}$$

Since the term s is simple, these rules are sufficient to transform $s(x) \Rightarrow p$ into $x \Rightarrow \varphi(p)$. The following lemma reveals the meaning of these rules.

Lemma 4.21. *For any model M, any term t, and any modal formulas φ, ψ, we have:*

$$\begin{array}{llcl} \textbf{(A)} & M \models (t \wedge \psi) \to \varphi & \Longleftrightarrow & M \models t \to (\psi \to \varphi) \\ \textbf{(B)} & M \models \diamondminus t \to \varphi & \Longleftrightarrow & M \models t \to \Box\varphi. \end{array}$$

Proof. (A) is trivial. Let us prove (B).
($\Rightarrow$) For any point $a \in M$ assume that $a \models t$. To prove that $a \models \Box\varphi$, take any point $b \in M$ with aRb. Then we have $b \models \diamondminus t$ and hence $b \models \varphi$.
($\Leftarrow$) For any point $b \in M$ assume that $b \models \diamondminus t$. This means that, for some $a \in M$ with aRb, we have $a \models t$. Then we have $a \models \Box\varphi$ and hence $b \models \varphi$. □

Corollary 4.22. *If the equation $s(x) \Rightarrow p$ is transformed into $x \Rightarrow \varphi(p)$ using the above rules, then for any model M and any its point a, we have*

$$M \models s(a) \to p \qquad \Longleftrightarrow \qquad M, a \models \varphi(p). \tag{5}$$

So, the term $s(x)$ always represents the minimal valuation of the variable p under which the formula $\varphi(p)$ is true at x.

By now, we have explained Step 4 of our algorithm that transforms safe terms into modal expressions, and thus the modal expression $\Phi'(\vec{x}, y)$ at Step 5 is well defined.

Lemma 4.23. *The expressions $\Phi(\vec{x}, y)$ and $\Phi'(\vec{x}, y)$ are equi-valid.*

Proof. Denote $\Psi(\vec{x}) := (x_1\colon \varphi_1 \wedge \ldots \wedge x_n\colon \varphi_n)$. Then Φ and Φ' look as follows:

$$\begin{array}{rrcl} \Phi(\vec{x}, y) = & \Psi(\vec{x}) & \to & y\colon t(\vec{x}) \\ \Phi'(\vec{x}, y) = & x\colon \varphi(p) \wedge \Psi(\vec{x}) & \to & y\colon t'(\vec{x}, p). \end{array}$$

Take any frame F and any its points $\vec{a}, b$. We need to prove the equivalence:

$$F \models \Phi(\vec{a}, b) \qquad \Longleftrightarrow \qquad F \models \Phi'(\vec{a}, b).$$

Below $a \in \vec{a}$ is the point that is assigned to the chosen nominal $x \in \vec{x}$.

($\Rightarrow$) Assume that $F \models \Phi(\vec{a}, b)$. Take any model $M = (F, \theta)$ based on F in which the premise of $\Phi'(\vec{a}, b)$ is true: $M \models a{:}\,\varphi(p) \wedge \Psi(\vec{a})$. Then $M \models s(a) \to p$ by (5), and $M \models b{:}\,t(\vec{a})$ due to $M \models \Phi(\vec{a}, b)$. Finally, since $\theta(s(a)) \subseteq \theta(p)$ by the above and $t' = t[s \mapsto p]$, we obtain $M \models b{:}\,t'(\vec{a}, p)$, by monotonicity of terms (as they are built using monotonic operators $\wedge$, $\vee$, $\Diamond$, $\Box$, $\diamondminus$ and hence replacing in t a smaller $s(a)$ with a larger p preserves the truth of the term).

($\Leftarrow$) Assume that $F \models \Phi'(\vec{a}, b)$. Take any model $M = (F, \theta)$ based on F in which the premise of $\Phi(\vec{a}, b)$ is true: $M \models \Psi(\vec{a})$. Since p does not occur in Ψ, we can freely change the valuation of p in M. Let us put $\theta(p) := \theta(s(a))$; this is well-defined, since p does not occur in $s(x)$ either. Now, $M \models s(a) \to p$ and hence $M \models a{:}\,\varphi(p)$, by (5). Thus, the premise of $\Phi'(\vec{a}, b)$ is true in M. Then so is its conclusion: $M \models b{:}\,t'(\vec{a}, p)$. Finally, we can substitute $[p \mapsto s]$ in t' and conclude that $M \models b{:}\,t(\vec{a})$, because the valuations of p and $s(a)$ are equal. □

5. Undefinable ∃&-formulas

This section is devoted to the proof of undefinability results stated in Sect. 3.1. In Sect. 5.1 we establish some properties of ∃&-formulas. Sect. 5.2 contains simple results on modal undefinability of ∃&-formulas. In Sect. 5.3, we recall the well-known notion of ultrafilter extension and the related anti-preservation result (and slightly generalize it). Finally, in Sect. 5.4 we use this technique in order to obtain our most difficult result on modal undefinability of ∃&-formulas.

In this section we assume that we are given a diagram $D = (V, V_\bullet, V_\circ, \vec{\Pi} = (\Pi_\ell)_{\ell\in L})$, where $V = V_\bullet \cup V_\circ$, $V_\bullet = \{x_1, \dots, x_n\}$, $V_\circ = \{y_1, \dots, y_m\}$, $V_\bullet \cap V_\circ = \varnothing$, and $\Pi_\ell \subseteq V{\times}V$ for each $\ell \in L$. Also recall that D gives rise to the FO formulas $B_D(\vec{x}, \vec{y})$ and $A_D(\vec{x})$ defined in Sect. 3 and the frame $F_D = (V, (\Pi_\ell)_{\ell\in L})$ defined in Sect. 3.1. Lemmas below must be understood in this context.

Also note that, in order to facilitate the notation, elements of V play a dual rôle here: on the one hand, they are variables in the formulas A_D and B_D, on the other hand, they are points of the frame F_D. The meaning is clear from context. In particular, when we are talking about the formulas A_D and B_D per se, $\vec{x}$ and $\vec{y}$ are just variables, whereas when we write $F_D \models B_D(\vec{x}, \vec{y})$, we mean that $\vec{x}$ and $\vec{y}$ are points of F_D substituted for the corresponding variables.

5.1. Properties of ∃&-formulas

Lemma 5.1. *$F_D \models B_D(\vec{x}, \vec{y})$ and hence $F_D \models A_D(\vec{x})$.*

Lemma 5.2. *Let $B(\vec{x}, \vec{y})$ be any conjunction of atoms of the form $z\mathsf{R}_\ell z'$ with $z, z' \in (\vec{x}, \vec{y})$.*
Then $F_D \models B(\vec{x}, \vec{y})$ iff each conjunct of $B(\vec{x}, \vec{y})$ occurs in $B_D(\vec{x}, \vec{y})$.

Proof. By definition of F_D and B_D, we have: $F_D \models x\mathsf{R}_\ell y$ iff $\langle x, y\rangle \in \Pi_\ell$, iff the formula $B_D(\vec{x}, \vec{y})$ contains the conjunct $x\mathsf{R}_\ell y$. □

Lemma 5.3 says that $A_D(\vec{x})$ is the strongest ∃&-formula true in F_D.

Lemma 5.3. *Let $A(\vec{x})$ be any ∃&-formula with free variables among $\vec{x}$.*
If $F_D \models A(\vec{x})$, then $A_D(\vec{x})$ implies $A(\vec{x})$.

Proof. Let $A_D(\vec{x}) = \exists\vec{y}\, B_D(\vec{x}, \vec{y})$ and $A(\vec{x}) = \exists\vec{z}\, B(\vec{x}, \vec{z})$. Suppose that $F_D \models A(\vec{x})$. Then $F_D \models B(\vec{x}, \vec{v})$ for some elements $\vec{v}$ of F_D. To prove that $A_D(\vec{x})$ implies $A(\vec{x})$, assume that $F \models A_D(\vec{a})$ for some frame F and its points $\vec{a}$. Then $F \models B_D(\vec{a}, \vec{b})$ for some points $\vec{b}$ of F. We need to show that $F \models A(\vec{a})$; to this end, we will find points $\vec{c}$ of F such that $F \models B(\vec{a}, \vec{c})$.

For convenience, let us denote $x_i^F := a_i$ and $y_j^F := b_j$. Now we put $c_k := v_k^F$ (recall that each v_k is either x_i or y_j, so that v_k^F is well-defined). We claim that $F \models B(\vec{a}, \vec{c})$. Indeed, since $F_D \models B(\vec{x}, \vec{v})$, Lemma 5.2 implies that each conjunct of $B(\vec{x}, \vec{v})$ occurs in $B_D(\vec{x}, \vec{y})$. Therefore, since $F \models B_D(\vec{a}, \vec{b})$, we conclude that $B(\vec{a}, \vec{c})$. □

Combining the above results, we obtain the following equivalences (cf. [11, Lemma 13]):

$$\begin{aligned} F_D \models B(\vec{x}, \vec{y}) \quad &\Longleftrightarrow \quad B_D(\vec{x}, \vec{y}) \text{ implies } B(\vec{x}, \vec{y}),\\ F_D \models A(\vec{x}) \quad &\Longleftrightarrow \quad A_D(\vec{x}) \text{ implies } A(\vec{x}). \end{aligned}$$

Lemma 5.4. *Let D be a minimal accessible diagram with $V_\bullet = \{x_1, \dots, x_n\}$, $V_\circ = \{y_1, \dots, y_m\}$. Assume $F_D \models B_D(\vec{x}, \vec{z})$, for some variables $\vec{z} = (z_1, \dots, z_m)$ from $V = V_\bullet \cup V_\circ$. Then*
(a) $\{z_1, \dots, z_m\} = \{y_1, \dots, y_m\}$ (hence $\vec{z}$ is a permutation of $\vec{y}$);
(b) $\langle z_i, z_j\rangle \in \Pi_\ell$ iff $\langle y_i, y_j\rangle \in \Pi_\ell$, for each $i, j \in \{1, \dots, m\}$ and $\ell \in L$.

Proof. (*a*) Denote $Z := \{z_1, \dots, z_m\}$. We need to prove that $Z = V_\circ$. Assume the contrary: $Z \neq V_\circ$. Since $|Z| = |V_\circ|$, this implies $Z \cap V_\circ \subsetneq V_\circ$. Without loss of generality, $Z \cap V_\circ = \{y_1, \dots, y_k\}$ for some $k < m$.

So, the free variables of the formula $B_D(\vec{x}, \vec{z})$ are $\vec{x}$ and $\{y_1, \dots, y_k\}$. Denote this formula by $B'(\vec{x}, y_1, \dots, y_k)$, let $A'(\vec{x}) = \exists y_1 \dots \exists y_k\, B'$ be the corresponding ∃&-formula, and let D' be the diagram associated to $A'(\vec{x})$. Since $F_D \models B'(\vec{x}, y_1, \dots, y_k)$, by Lemma 5.2, each conjunct of B' occurs in B_D; in other words, D' is a subgraph of D. Moreover, since D is accessible, it has no isolated white nodes; but D' lacks y_m, therefore, D' is obtained from D by removing *at least one* edge.

Now let us prove that $A'(\vec{x})$ implies $A_D(\vec{x})$, which will contradict the minimality of D. By Lemma 5.3, it suffices to show that $F_{D'} \models A_D(\vec{x})$. But this is easy: by Lemma 5.1, $F_{D'} \models B'(\vec{x}, y_1, \dots, y_k)$, in other words, $F_{D'} \models B_D(\vec{x}, \vec{z})$, which implies $F_{D'} \models A_D(\vec{x})$.

(b) By (a), the function $f\colon V_\circ \to V_\circ$ defined by $f(y_i) = z_i$ is a bijection. It induces a bijection $f'\colon V_\circ{\times}V_\circ \to V_\circ{\times}V_\circ$ defined by $f'(\langle y_i, y_j\rangle) = \langle z_i, z_j\rangle$.

As shown in (a), each conjunct of $B_D(\vec{x},\vec{z})$ occurs in $B_D(\vec{x},\vec{y})$. In particular, if $y_i\mathsf{R}_\ell y_j$ is in $B_D(\vec{x},\vec{y})$, then, by substitution, $z_i\mathsf{R}_\ell z_j$ is in $B_D(\vec{x},\vec{z})$ and hence in $B_D(\vec{x},\vec{y})$ by the above. Denoting $S_\ell := \Pi_\ell \cap (V_\circ{\times}V_\circ)$, we have that if $\langle y_i, y_j\rangle \in S_\ell$ then $\langle z_i, z_j\rangle \in S_\ell$.

Thus, f' maps S_ℓ into S_ℓ. But f' is injective and S_ℓ is finite. Hence f' is a bijection from S_ℓ to S_ℓ. Therefore, $\langle y_i, y_j\rangle \in S_\ell$ iff $\langle z_i, z_j\rangle \in S_\ell$, as required. □

Given a diagram $D = (V, V_\bullet, V_\circ, \vec{\Pi})$, denote $\Pi = \bigcup_{\ell\in L} \Pi_\ell$. A *covering relation* for D is a binary relation $S \subseteq V{\times}V$ that satisfies the following three conditions (where S^* stands for the reflexive-transitive closure of S):

(1) $S \subseteq \Pi$;
(2) S is acyclic (i.e., the graph (V, S) has no cycles, even undirected);
(3) $\forall y \in V_\circ\ \exists x \in V_\bullet\colon\ xS^*y$ (i.e., all white nodes are S-reachable from black nodes).

By definition, in an accessible diagram, white nodes are Π-reachable from black nodes. Lemma 5.5 says that the same can be achieved by an acyclic relation $S \subseteq \Pi$.

Lemma 5.5. *Every accessible diagram has a covering relation.*

Proof. We build a sequence of binary relations $S_0 \subset S_1 \subset \ldots \subset S_r$, where each S_i satisfies (1) and (2) and the last one, S_r, also satisfies (3) and hence is a covering relation for D. **Induction base:** $S_0 := \varnothing$ trivially satisfies (1) and (2).

Induction step. Assume that S_i satisfies (1) and (2), but not (3). This means that some white node y is not S_i-reachable from black nodes: $y \notin S_i^*(V_\bullet)$, where $S_i^*(V_\bullet) = \{z \in V \mid \exists x \in V_\bullet\colon\ xS_i^*z\}$.

Let Γ be the set of all directed paths in D from $S_i^*(V_\bullet)$ to y. Since D is accessible, there is a path from $V_\bullet$ to y; but $V_\bullet \subseteq S_i^*(V_\bullet)$, thus $\Gamma \neq \varnothing$. Then pick a path $\gamma \in \Gamma$ of a minimal length, and let $\langle z, z'\rangle \in \Pi_\ell$ be its first edge.

Now put $S_{i+1} = S_i \cup \{\langle z, z'\rangle\}$. Since $S_i \subseteq \Pi$ and $\langle z, z'\rangle \in \Pi_\ell$, we have $S_{i+1} \subseteq \Pi$. Moreover, $z' \notin S_i^*(V_\bullet)$ due to the minimality of γ, hence $S_i \subsetneq S_{i+1}$ and S_{i+1} has no cycles. Thus, S_{i+1} satisfies (1) and (2). Since W is finite and S_i are all distinct, the above process will eventually terminate. The resulting relation S_r will satisfy (1), (2), and (3). □

Lemma 5.6. *For every accessible diagram D with an undirected white cycle C, there is an edge* e *in C such that, after removing* e*, the diagram remains to be accessible.*

Proof. By Lemma 5.5, D has a covering relation S. Then let e be any edge that belongs to the cycle C but not to S (it exists, since S is acyclic). Now, if we remove e from D, the remaining diagram will be accessible, since all white nodes are still S-reachable from black nodes (as removing the edge e does not affect the relation S). □

5.2. Simple cases of undefinability

Lemma 5.7. *Every minimal inaccessible ∃&-formula is not modally expressible (and hence not modally definable).*

Proof. Let $A_D(\vec{x}) = \exists\vec{y}\, B_D(\vec{x},\vec{y})$ be a minimal inaccessible ∃&-formula, and $D = (V, V_\bullet, V_\circ, \vec{\Pi})$ be its diagram. Then there is a variable $y_{i_0} \in V_\circ$ unreachable from black nodes $V_\bullet$ in D. In order to prove that $A_D(\vec{x})$ is not modally expressible, assume the contrary, i.e., that it corresponds to a modal expression $\Phi(\vec{x})$. This means that, for any frame F and any tuple of its points $\vec{e}$,

$$F \models A_D(\vec{e}) \quad\Longleftrightarrow\quad F \models \Phi(\vec{e}). \tag{6}$$

Consider the frame $F_D = (V, \vec{\Pi})$ and the tuple of its points $\vec{x}$. By Lemma 5.1, $F_D \models A_D(\vec{x})$. Hence, by (6), we have $F_D \models \Phi(\vec{x})$.

Recall that the validity of modal formulas (and hence expressions) is preserved under taking generated subframes ([5, Theorem 3.14]). Let F' be the subframe of F_D generated by the set of black nodes $V_\bullet$. Denote the set of its worlds by V'. Note that $y_{i_0} \notin V'$, since the variable y_{i_0} is unreachable from $V_\bullet$, by assumption.

So, $F' \models \Phi(\vec{x})$. Then $F' \models A_D(\vec{x})$, by (6). Hence $F' \models B_D(\vec{x},\vec{z})$, for some $z_1,\ldots,z_m \in V'$. The formula B_D is a conjunction of atomic formulas, and F' is a FO submodel of F_D, so we also have $F_D \models B_D(\vec{x},\vec{z})$. Since the diagram D is minimal, we can apply Lemma 5.4 and conclude that $\{z_1,\ldots,z_m\} = V_\circ$. This contradicts the fact that $y_{i_0} \in (V_\circ \setminus V') \subseteq V_\circ \setminus \{z_1,\ldots,z_m\}$. □

Lemma 5.8. *The ∃&-formula associated to the disjoint union of two (or more) diagrams is not modally definable.*

Proof. The proof is an easy generalization of the argument given in Example 2.2. Let $D = D_1 \cup D_2$, where the diagrams D_1 and D_2 are disjoint (and both have edges), and let $\vec{x}$ and $\vec{x}'$ be the lists of black nodes in D_1 and D_2 respectively. Clearly, $A_D(\vec{x},\vec{x}')$ is equivalent to $A_{D_1}(\vec{x}) \wedge A_{D_2}(\vec{x}')$. We need to prove that A_D is not modally definable. Assume the contrary: there are tuples of modal formulas $\vec{\varphi}$ and $\vec{\varphi}'$ such that, for any frame F and its worlds $\vec{e}$ and $\vec{e}'$,

$$F \models A_D(\vec{e},\vec{e}') \quad\Longleftrightarrow\quad F \models \textstyle\bigvee_i e_i\colon \varphi_i \ \vee\ \bigvee_j e'_j\colon \varphi'_j. \tag{7}$$

Consider a frame[10] $F_1 = F_{D_1} \cup (\vec{x}', \varnothing)$. Clearly, $F_1 \not\models A_D(\vec{x},\vec{x}')$, since D_2 has edges, whereas F_1 does not have any edges related to $\vec{x}'$. Then by (7), there is a valuation $\theta_1\colon \{\vec{x},\vec{x}'\} \to \mathsf{PV}$ such that, denoting $M_1 = (F_1, \theta_1)$, we have $M_1, x_i \not\models \varphi_i$ for all i.

[10] Here $(\vec{x}', \varnothing)$ is a frame whose worlds are all the variables x'_j and all relations are empty.

Similarly, a frame $F_2 = (\vec{x}, \varnothing) \cup F_{D_2}$ does not satisfy $A_D(\vec{x}, \vec{x}')$, hence by (7) there is a valuation $\theta_2\colon \{\vec{x}, \vec{x}'\} \to \mathsf{PV}$ such that, denoting $M_2 = (F_2, \theta_2)$, we have $M_2, x'_j \not\models \varphi'_j$ for all j.

Now consider the frame $F_D = F_{D_1} \cup F_{D_2}$. Define a valuation θ by putting $\theta(x_i) := \theta_1(x_i)$ and $\theta(x'_j) := \theta_2(x'_j)$, for all i, j. Let $M := (F_D, \theta)$. Then observe that the following bisimulations hold: $M, x_i \sim M_1, x_i$ and $M, x'_j \sim M_2, x'_j$. Hence $M, x_i \not\models \varphi_i$ and $M, x'_j \not\models \varphi'_j$. Therefore, M (and hence F_D) does not satisfy $\bigvee_i x_i\colon \varphi_i \vee \bigvee_j x'_j\colon \varphi'_j$. However, F_D satisfies $A_D(\vec{x}, \vec{x}')$, by Lemma 5.1. This contradicts (7). □

5.3. *Ultrafilter extension*

It is known [5] that the validity of modal formulas (and hence the truth of modally definable FO formulas with one free variable) is *anti*-preserved under taking ultrafilter extensions of frames (i.e., whenever a modal formula is valid in F^{ue} defined below, it is valid in F). Here we generalize this result to FO formulas with several free variables (see Theorem 5.14 below). Let us recall necessary definitions.

Definition 5.9. A set $u \subseteq 2^W$ is an *ultrafilter* over a set W if, for all $X, Y \subseteq W$,

(1) if $X, Y \in u$, then $X \cap Y \in u$;
(2) if $X \in u$ and $X \subset Y$, then $Y \in u$;
(3) $X \notin u$ iff $\overline{X} \in u$, where $\overline{X} = W \setminus X$.

From the definition it follows that $\varnothing \notin u$ and $W \in u$, for any ultrafilter u over a set W.

Definition 5.10. Given a frame $F = (W, (R_\ell)_{\ell\in L})$, its *ultrafilter extension* is defined as the frame $F^{\mathrm{ue}} = (W^{\mathrm{ue}}, (R^{\mathrm{ue}}_\ell)_{\ell\in L})$, where W^{ue} is the set of all ultrafilters over W, and $uR^{\mathrm{ue}}_\ell u'$ holds for ultrafilters u and u' iff $R^{-1}_\ell(X) \in u$ for all $X \in u'$. Here $R^{-1}_\ell(X) = \{z \mid zR_\ell x \text{ for some } x \in X\}$.

Given a model $M = (F, \theta)$, its *ultrafilter extension* is the model $M^{\mathrm{ue}} = (F^{\mathrm{ue}}, \theta^{\mathrm{ue}})$, where $u \in \theta^{\mathrm{ue}}(p)$ iff $\theta(p) \in u$, for every variable p and every ultrafilter u over W.

Given a world $a \in W$, the set $\pi_a = \{X \subseteq W \mid a \in X\}$ is obviously an ultrafilter; it is called the *principal ultrafilter* generated by a. A frame F can be seen as a (not necessarily generated) subframe of F^{ue}, if we identify worlds a of F with their principal ultrafilters π_a, as the following lemma shows.

Lemma 5.11 ([5], p. 95)**.** *For all worlds a, b in every frame F, we have $aR_\ell b$ iff $\pi_a R^{\mathrm{ue}}_\ell \pi_b$.*

The next lemma says that (*i*) a point a in M and the corresponding point π_a in M^{ue} are indistinguishable by any modal formula, and hence (*ii*) validity of modal formulas is *anti*-preserved under taking ultrafilter extension of frames.

Lemma 5.12 ([5], p. 96, 142)**.** *For all frames F, models M, worlds a, formulas φ:*
(*i*) $M^{\mathrm{ue}}, \pi_a \models \varphi \iff M, a \models \varphi$;
(*ii*) $F^{\mathrm{ue}}, \pi_a \models \varphi \implies F, a \models \varphi$.

We generalize this result to modal expressions[11] (introduced in Def. 2.3).

Lemma 5.13. *For all frames F, models M, worlds $\vec{a}$, modal expressions $\Phi(\vec{x})$:*
(*i*) $M^{\mathrm{ue}} \models \Phi(\pi_{a_1}, \dots, \pi_{a_n}) \iff M \models \Phi(a_1, \dots, a_n)$;
(*ii*) $F^{\mathrm{ue}} \models \Phi(\pi_{a_1}, \dots, \pi_{a_n}) \implies F \models \Phi(a_1, \dots, a_n)$.

Proof. Item (*i*) follows immediately from Lemma 5.12(*i*). As for item (*ii*), denote $\vec{e} = (\pi_{a_1}, \dots, \pi_{a_n})$. To prove that $F \models \Phi(\vec{a})$, take any valuation θ on F and put $M = (F, \theta)$. By assumption, $F^{\mathrm{ue}} \models \Phi(\vec{e})$, so $M^{\mathrm{ue}} \models \Phi(\vec{e})$. Using (*i*), we conclude that $M \models \Phi(\vec{a})$ as desired. □

We are ready to prove that modally expressible first-order formulas are anti-preserved under ultrafilter extensions.

Theorem 5.14 (Anti-preservation)**.** *Let $A(\vec{x})$ be a modally expressible first-order formula. Then for any frame F and its worlds $\vec{a}$, the implication holds:*

$$F^{\mathrm{ue}} \models A(\pi_{a_1}, \dots, \pi_{a_n}) \quad\Longrightarrow\quad F \models A(a_1, \dots, a_n).$$

Proof. Denote $\vec{e} = (\pi_{a_1}, \dots, \pi_{a_n})$. By assumption, $A(\vec{x})$ corresponds to some modal expression $\Phi(\vec{x})$. Then the claim follows from the chain of implications:

$$F^{\mathrm{ue}} \models A(\vec{e}) \quad\overset{(*)}{\Longrightarrow}\quad F^{\mathrm{ue}} \models \Phi(\vec{e}) \quad\overset{(**)}{\Longrightarrow}\quad F \models \Phi(\vec{a}) \quad\overset{(*)}{\Longrightarrow}\quad F \models A(\vec{a}),$$

where implications $(*)$ hold since $A(\vec{x})$ corresponds to $\Phi(\vec{x})$, and implication $(**)$ is due to Lemma 5.13(*ii*). □

Thus, in order to show that a FO formula $A(x_1, \dots, x_n)$ is *not* modally expressible, it suffices to find a frame F and its points $a_1, \dots, a_n$ such that $F \not\models A(a_1, \dots, a_n)$, but $F^{\mathrm{ue}} \models A(\pi_{a_1}, \dots, \pi_{a_n})$.

[11] Since modal expressions are hybrid formulas in the language $\mathcal{H}(@)$, this follows from a similar result for this language [8, Prop. 4.2.6].

5.4. Proof of undefinability

This subsection is devoted to the proof of the following theorem.

Theorem 5.15. *If an $\exists\&$-formula $A(\vec{x})$ is minimal, accessible, and white-cyclic, then it is **not** modally expressible.*

Assume that we are given an $\exists\&$-formula $A(\vec{x}) = \exists\vec{y}\, B(\vec{x}, \vec{y})$ such that its diagram $D = (V, V_\bullet, V_\circ, (\Pi_\ell)_{\ell\in L})$ is minimal, accessible, and contains a white cycle[12] C. To prove that $A(\vec{x})$ is *not* modally expressible, taking into account Theorem 5.14, we will build a frame F and its worlds $a_1, \ldots, a_n$ such that

$$F \not\models A(a_1, \ldots, a_n) \quad \text{and} \quad F^{\mathrm{ue}} \models A(\pi_{a_1}, \ldots, \pi_{a_n}). \tag{8}$$

By Lemma 5.6, the cycle C contains a white edge[13] $\mathsf{e} = \langle z, z'\rangle_{\ell_0}$ such that the diagram D' obtained from D by removing this edge is accessible. Denote by V_z the *white connected component* of the node z in D, i.e., the set of white nodes that are reachable from z through (undirected) white paths. Obviously, $z' \in V_z$ and V_z contains the cycle C. Let us denote by $V_\circ' := V_\circ \setminus V_z$ the set of the remaining white nodes in D. We need a simple lemma.

Lemma 5.16. *In the diagram D, every two distinct nodes from V_z are connected by a path contained in V_z and not containing the edge e.*

Proof. Let ε be the path obtained by removing the edge e from the cycle C. Clearly, ε connects z and z' and is contained in V_z. Since the set V_z is connected, any two distinct nodes from V_z are connected by a path inside V_z. If that path contains the edge e, then replace e with ε in it, thus obtaining a path inside V_z that connects the same nodes and does not contain the edge e. □

Now we are ready to build a frame $F = (W, (R_\ell)_{\ell\in L})$. Put $W := (V_\bullet \cup V_\circ') \cup (V_z \times \mathbb{N})$, and for each $\ell \in L$, put $R_\ell := R_\ell^{(1)} \cup R_\ell^{(2)} \cup R_\ell^{(3)} \cup R_\ell^{(4)} \cup R_\ell^{(5)}$, where

$$\begin{aligned}
R_\ell^{(1)} &= \{\langle x, y\rangle \mid x\Pi_\ell y;\ x, y \in (V_\bullet \cup V_\circ')\} \ = \ \Pi_\ell{\restriction}_{(V_\bullet \cup V_\circ)} \\
R_\ell^{(2)} &= \{\langle x, (y, i)\rangle \mid x\Pi_\ell y;\ x \in V_\bullet;\ y \in V_z;\ i \in \mathbb{N}\} \\
R_\ell^{(3)} &= \{\langle (x, i), y\rangle \mid x\Pi_\ell y;\ x \in V_z;\ y \in V_\bullet;\ i \in \mathbb{N}\} \\
R_\ell^{(4)} &= \{\langle (x, i), (y, i)\rangle \mid x\Pi_\ell y;\ x, y \in V_z;\ i \in \mathbb{N};\ \langle x, y\rangle_\ell \neq \langle z, z'\rangle_{\ell_0}\} \\
R_\ell^{(5)} &= \{\langle (z, i), (z', j)\rangle \mid i < j;\ i, j \in \mathbb{N}\},\ \text{if } \ell = \ell_0;\ \text{otherwise } R_\ell^{(5)} = \varnothing.
\end{aligned}$$

Intuitively, the frame F is obtained from the diagram D as follows:

- edges within $V_\bullet \cup V_\circ'$ are inherited from D, see the definition of $R_\ell^{(1)}$;
- every edge $\langle x, y\rangle$ from $V_\bullet$ to V_z turns into a countable family of edges $\langle x, (y, i)\rangle$ from $V_\bullet$ to $V_z \times \mathbb{N}$, see the definition of $R_\ell^{(2)}$;
- similarly for edges from V_z to $V_\bullet$, see the definition of $R_\ell^{(3)}$;
- all edges within V_z, except for the edge $\langle z, z'\rangle_{\ell_0}$, are reproduced at every i-th level of the set $V_z \times \mathbb{N}$, see the definition of $R_\ell^{(4)}$;
- the edge $\langle z, z'\rangle_{\ell_0}$ turns into a countable family of the corresponding edges from the i-th to the j-th layer, for all $i < j$, see the definition of $R_\ell^{(5)}$.

Consider a natural projection $p\colon W \to V$ defined, for any $w \in W$, as follows:

$$p(w) = \begin{cases} w, & \text{if } w \in V_\bullet \cup V_\circ', \\ y, & \text{if } w = (y, i) \in V_z \times \mathbb{N}. \end{cases}$$

Observe that p is a monotone function from the frame F to the frame F_D, in the sense that if $\langle w, w'\rangle \in R_\ell$ in F, then $\langle p(w), p(w')\rangle \in \Pi_\ell$ in F_D. Indeed, for $\langle w, w'\rangle \in R_\ell^{(j)}$ with $j \leq 4$, this is due to the presence of $x\Pi_\ell y$ in their definition; for $\langle w, w'\rangle \in R_\ell^{(5)}$, the claim is trivial, since $\langle z, z'\rangle \in \Pi_{\ell_0}$.

Example 5.17. Let us consider again the diagram D shown in Figure 1. We apply Lemma 5.6 and find an edge $\langle z, z'\rangle$, see Figure 3 (left). Thus the set of white nodes $V_\circ$ is partitioned into V_z and $V_\circ'$. Then we build an infinite frame F shown in Figure 3 (right). The subgraphs on $V_\bullet$ and $V_\circ$ are the same as in the diagram D, whereas the subgraph on V_z is copied countably many times, except for its edge $\langle z, z'\rangle$. The edge $\langle z, z'\rangle$ turns into a countable set of edges that connect (z, i) to (z', j), for all $i < j$.

Finally, let us prove (8) for the formula $A = A_D(\vec{x})$, the frame F, and its worlds $\vec{x}$ (since each x_i is a black node of D, it belongs to the domain W of F).

Lemma 5.18. $F \not\models A(\vec{x})$.

Proof. Assume the contrary. Then $F \models B_D(\vec{x}, \vec{a})$, for some elements $\vec{a} \in W$. Since the formula B_D is a conjunction of atomic formulas and the projection $p\colon F \to F_D$ is monotone, we obtain:

$$F_D \models B_D(p(x_1), \ldots, p(x_n), p(a_1), \ldots, p(a_m)).$$

[12] In this section, all cycles and paths are assumed to be undirected.

[13] It is convenient to denote by $\langle x, y\rangle_\ell$ the edge $\langle x, y\rangle$ in D that belongs to the relation Π_ℓ.

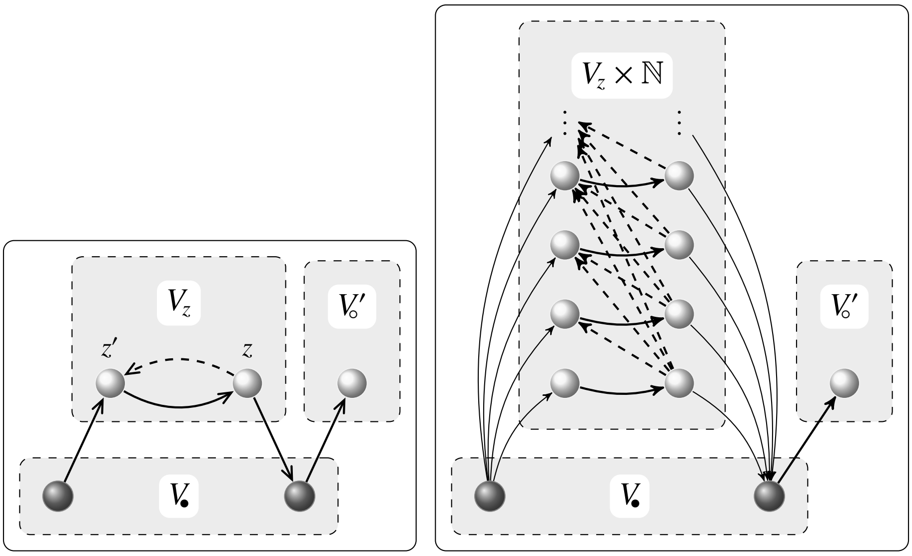


Figure 3: On the left: the diagram D with a white cycle (the edge $\mathsf{e} = \langle z, z' \rangle$ is dashed); its set of nodes is partitioned into $V_\bullet$, V_z, and $V_\circ'$. On the right: the Kripke frame F built from the diagram D.

Recall that $p(x_i) = x_i$. Denote $z_j := p(a_j)$, so that $z_j \in V$. So, $F_D \models B_D(\vec{x}, \vec{z})$. By Lemma 5.4(a), $\{z_1, \ldots, z_m\} = \{y_1, \ldots, y_m\}$. Take those a_j whose projection is in V_z:

$$H = \{ a_j \mid z_j \in V_z,\ 1 \le j \le m \}.$$

Clearly, $H \subseteq V_z \times \mathbb{N}$, since only elements from $V_z \times \mathbb{N}$ may have their projection in V_z. Hence, each $a_j \in H$ has the form $a_j = \langle z_j, n_j \rangle$, for some $n_j \in \mathbb{N}$.

Claim 1. *All points of H lie in the same layer of $V_z \times \mathbb{N}$, i.e., all the numbers n_j are equal.*

Indeed, take any distinct elements $a_i = \langle z_i, n_i \rangle$ and $a_j = \langle z_j, n_j \rangle$ from H and let us show that $n_i = n_j$. By Lemma 5.16, the nodes z_i and z_j are connected by a path γ_1 inside V_z not containing the edge e. By Lemma 5.4(b), for each edge $\langle z_s, z_t \rangle_\ell$ in γ_1, there is a corresponding edge $\langle y_s, y_t \rangle_\ell$ in the diagram D. These edges constitute a path γ_2 connecting y_i and y_j.

By the definition of B_D, the formula $B_D(\vec{x}, \vec{y})$ contains the conjuncts $y_s \mathsf{R}_\ell y_t$ that correspond to edges in γ_2. Since $F \models B_D(\vec{x}, \vec{a})$, the corresponding conjuncts $a_s \mathsf{R}_\ell a_t$ are true in F. Therefore, F contains a path γ_3 connecting our chosen elements a_i and a_j. Note that the projection of γ_3 is exactly the path γ_1, which does not contain the edge e.

By construction of $R_\ell^{(4)}$, if $a, b \in V_z \times \mathbb{N}$ and $\langle a, b \rangle_\ell$ is an edge in F such that its projection does not coincide with the edge e, then $\langle a, b \rangle \in R_\ell^{(4)}$, and hence a and b belong to the same layer. As shown above, all edges in γ_3 are of this kind. Therefore, all points in the path γ_3, including its ends a_i and a_j, belong to the same layer.

Claim 2. *The set H contains two points that lie in different layers of $V_z \times \mathbb{N}$.*

Indeed, the diagram D contains the edge $\mathsf{e} = \langle z, z' \rangle_{\ell_0}$. Recall that $z, z' \in V_z$. Pick those points $a_i, a_j \in H$ whose projections are $p(a_i) = z$ and $p(a_j) = z'$. This means that $a_i = \langle z_i, n_i \rangle$ and $a_j = \langle z_j, n_j \rangle$, where $z_i = z$, $z_j = z'$, for some numbers $n_i, n_j \in \mathbb{N}$. We claim that $n_i < n_j$.

Indeed, since $\langle z, z' \rangle \in \Pi_{\ell_0}$, or equivalently, $\langle z_i, z_j \rangle \in \Pi_{\ell_0}$, we have $\langle y_i, y_j \rangle \in \Pi_{\ell_0}$ by Lemma 5.4(b). Hence the formula $B_D(\vec{x}, \vec{y})$ contains the conjunct $y_i \mathsf{R}_{\ell_0} y_j$. Since $F \models B_D(\vec{x}, \vec{a})$, the conjunct $a_i \mathsf{R}_{\ell_0} a_j$ is true in F. Thus we have an edge $\langle a_i, a_j \rangle_{\ell_0}$ in F whose projection is the edge e. This implies that $\langle a_i, a_j \rangle \in R_{\ell_0}^{(5)}$, which means that $n_i < n_j$, so we are done.

Obviously, Claim 1 contradicts Claim 2, so this completes the proof of the lemma. □

Lemma 5.19. $F^{\mathrm{ue}} \models A(\pi_{x_1}, \ldots, \pi_{x_n})$.

Proof. It suffices to show that $F^{\mathrm{ue}} \models B_D(\pi_{x_1}, \ldots, \pi_{x_n}, u_1, \ldots, u_m)$, for some ultrafilters $u_1, \ldots, u_m$ over W. Fix any non-principal[14] ultrafilter u over $\mathbb{N}$.

If $y_i \in V_\circ'$, we simply put $u_i = \pi_{y_i}$.

Now take any $y_i \in V_z$. For any $X \subseteq W$, define the set $f_i(X) \subseteq \mathbb{N}$ as follows: $k \in f_i(X)$ iff $\langle y_i, k \rangle \in X$, for every $k \in \mathbb{N}$. Then f_i has the following properties, for all $X, Y \subseteq W$:

- $f_i(X \cap Y) = f_i(X) \cap f_i(Y)$;
- if $X \subseteq Y$ then $f_i(X) \subseteq f_i(Y)$;
- $f_i(\overline{X}) = \overline{f_i(X)}$.

Finally, we define u_i as follows: $X \in u_i$ iff $f_i(X) \in u$, for any $X \subseteq W$.

Claim 1. *Each u_i is an ultrafilter over W.*

[14] An ultrafilter is called *non-principal* if it is not principal, i.e., not of the form π_a. It exists over any infinite set. It is an easy exercise that $\varnothing$ does not belong to any ultrafilter, and no finite set is a member of any non-principal ultrafilter.

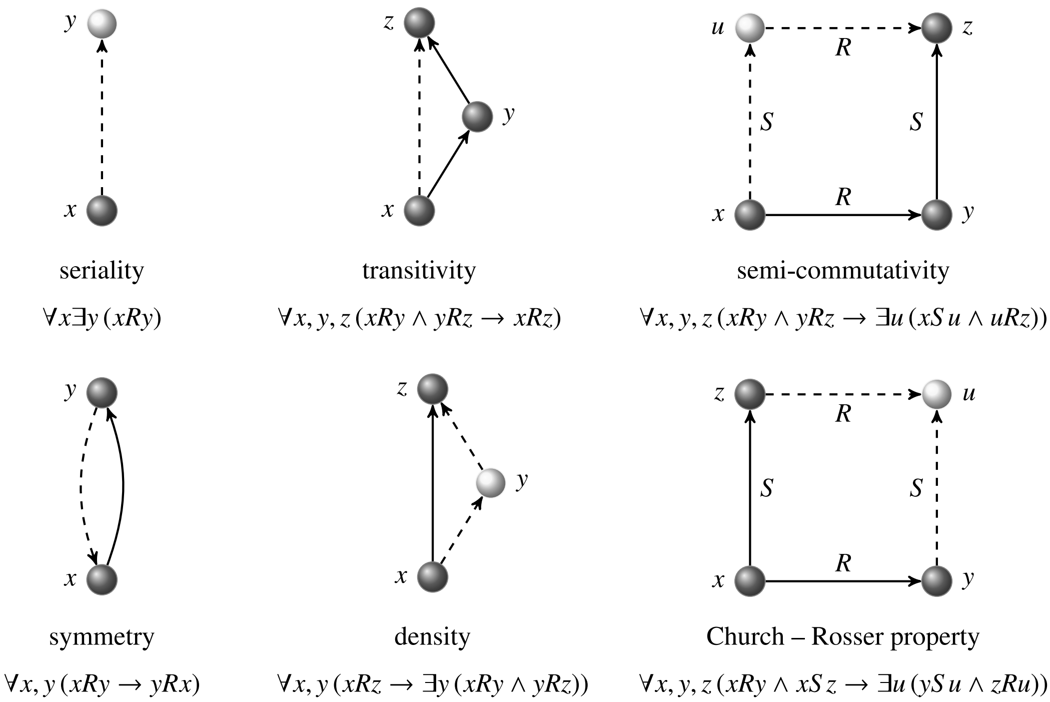


Figure 4: Examples of ∀∃-diagrams.

We need to check the three conditions from the definition of an ultrafilter.

(1) If $X, Y \in u_i$, then $f_i(X), f_i(Y) \in u$. Hence $f_i(X \cap Y) = f_i(X) \cap f_i(Y) \in u$ and so $(X \cap Y) \in u_i$.

(2) If $Y \supseteq X \in u_i$, then $f_i(X) \in u$, hence $f_i(X) \subseteq f_i(Y) \in u$, thus $Y \in u_i$.

(3) $X \in u_i$ iff $f_i(X) \in u$ iff $\overline{f_i(X)} \notin u$ iff $f_i(\overline{X}) \notin u$ iff $\overline{X} \notin u_i$.

Claim 2. $F^{\mathrm{ue}} \models B_D(\pi_{x_1}, \ldots, \pi_{x_n}, u_1, \ldots, u_m)$.

To show this, consider any conjunct $\alpha \mathsf{R}_\ell \beta$ from B_D, where $\alpha, \beta \in V$ are variables and $\alpha \Pi_\ell \beta$ is an edge in the diagram D. Then one of the following five cases takes place:

1) $\alpha, \beta \in (V_\bullet \cup V_\circ')$. Then $\langle \alpha, \beta \rangle \in R_\ell^{(1)} \subseteq R_\ell$, so $\langle \pi_\alpha, \pi_\beta \rangle \in R_\ell^{\mathrm{ue}}$, by Lemma 5.11.

2) $\alpha \in V_\bullet$, $\beta \in V_z$. Let $\alpha = x_i$, $\beta = y_j$. To prove that $\langle \pi_{x_i}, u_j \rangle \in R_\ell^{\mathrm{ue}}$, take any $X \in u_j$. Then $f_j(X) \in u$, hence $f_j(X) \neq \varnothing$. Fix any $k \in f_j(X)$, so that $(y_j, k) \in X$. Since $x_i \Pi_\ell y_j$, we have $\langle x_i, (y_j, k) \rangle \in R_\ell^{(2)} \subseteq R_\ell$. So $x_i \in R_\ell^{-1}(X)$, or equivalently, $R_\ell^{-1}(X) \in \pi_{x_i}$.

3) $\alpha \in V_z$, $\beta \in V_\bullet$. Let $\alpha = y_j$, $\beta = x_i$. To prove that $\langle u_j, \pi_{x_i} \rangle \in R_\ell^{\mathrm{ue}}$, take any $X \in \pi_{x_i}$. Then $x_i \in X$. Since $y_j \Pi_\ell x_i$, we have $\langle (y_j, k), x_i \rangle \in R_\ell^{(3)} \subseteq R_\ell$, hence $(y_j, k) \in R_\ell^{-1}(X)$ and $k \in f_j(R_\ell^{-1}(X))$, for all $k \in \mathbb{N}$. Thus, $f_j(R_\ell^{-1}(X)) = \mathbb{N} \in u$, and so $R_\ell^{-1}(X) \in u_j$.

4) $\alpha, \beta \in V_z$ and $\langle \alpha, \beta \rangle_\ell \neq \mathsf{e}$. Let $\alpha = y_i$, $\beta = y_j$. To prove that $\langle u_i, u_j \rangle \in R_\ell^{\mathrm{ue}}$, take any $X \in u_j$. Then $f_j(X) \in u$. Let us show that $f_j(X) \subseteq f_i(R_\ell^{-1}(X))$, for then $f_i(R_\ell^{-1}(X)) \in u$ and thus $R_\ell^{-1}(X) \in u_i$. For every $k \in f_j(X)$, we have $(y_j, k) \in X$. We also have that $y_i \Pi_\ell y_j$, hence $\langle (y_i, k), (y_j, k) \rangle \in R_\ell^{(4)} \subseteq R_\ell$. Therefore, $(y_i, k) \in R_\ell^{-1}(X)$ and so $k \in f_i(R_\ell^{-1}(X))$.

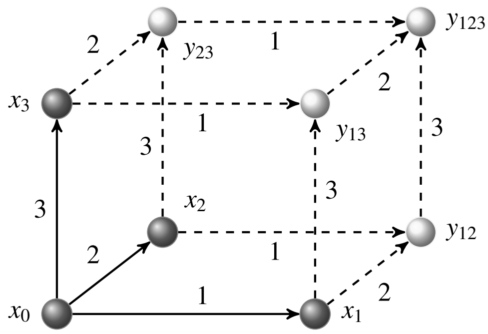
5) $\langle \alpha, \beta \rangle_\ell = \mathsf{e}$. So, $\alpha = y_i$, $\beta = y_j$, $\ell = \ell_0$. To prove that $\langle u_i, u_j \rangle \in R_{\ell_0}^{\mathrm{ue}}$, take any $X \in u_j$. Then $f_j(X) \in u$ and hence $f_j(X)$ is infinite. Since $y_i \Pi_{\ell_0} y_j$, we have $\langle (y_i, k), (y_j, k') \rangle \in R_{\ell_0}^{(5)} \subseteq R_{\ell_0}$ for every $k, k' \in \mathbb{N}$ with $k < k'$. As shown above, $(y_j, k') \in X$ for infinitely many k'. Hence $(y_i, k) \in R_{\ell_0}^{-1}(X)$ for all k. Therefore, $f_i(R_{\ell_0}^{-1}(X)) = \mathbb{N} \in u$ and so $R_{\ell_0}^{-1}(X) \in u_i$. □

6. Application to classical modal definability of ∀∃-formulas

Many (closed) first-order formulas considered in modal logic can be represented by pictures of a certain kind, which we will call ∀∃-*diagrams* (see Def. 6.2 below and examples in Fig. 4 and 5). Informally, these pictures contain black and white points (corresponding to the universally and existentially quantified variables, respectively) linked by solid and dashed arrows, and are read as follows: “if there exist black points connected by solid arrows, then there exist white points such that all points are connected by dashed arrows”.

Such diagrams are of particular use in multi-dimensional modal logics [14]. For example, quite complicated modifications of cubifying property were used in [27, 28] to prove that ≥3-dimensional products of modal logic are neither finitely axiomatizable, nor axiomatizable by any set of modal formulas using finitely many variables. The diagrams were also used in [21] for similar purposes. Moreover, [27] contains a first-order axiomatization for the class of frames for the n-dimensional modal logic $\mathbf{K}^n$, given in terms of such diagrams.

Since ∀∃-diagrams are a natural way of reasoning about many-dimensional structures, it seems interesting to study the question when they give rise to modally definable FO properties. It turns out that Theorem 4.1 together with a theorem of M. Kracht from [25] provide us with a sufficient condition for modal definability of ∀∃-diagrams, covering all cases from Figs. 4 and 5.



cubifying property

$$\forall x_0 \forall x_1 \forall x_2 \forall x_3 \big(x_0 R_1 x_1 \wedge x_0 R_2 x_2 \wedge x_0 R_3 x_3 \to \exists y_{12} \exists y_{13} \exists y_{23} \exists y_{123}$$
$$(x_1 R_2 y_{12} \wedge x_1 R_3 y_{13} \wedge x_2 R_1 y_{12} \wedge x_2 R_3 y_{23} \wedge x_3 R_1 y_{13} \wedge x_3 R_2 y_{23} \wedge y_{23} R_1 y_{123} \wedge y_{13} R_2 y_{123} \wedge y_{12} R_3 y_{123})\big)$$

Figure 5: The "cubifying" property as an $\forall\exists$-diagram.

Theorem 6.1 (Theorem 5.4.6 in [25])**.** *Every first-order formula with one free variable $A(x)$ obtained from modally definable formulas (with several free variables) using conjunction, disjunction, and restricted universal quantification $\forall y\,(x\mathsf{R}_\ell y \to \ldots)$ is modally definable.*

Definition 6.2. An $\forall\exists$-*diagram* is a tuple

$$\mathcal{D} = (V, V_\bullet, (S_\ell)_{\ell\in L}, V_\circ, (\Pi_\ell)_{\ell\in L}),$$

where $V = V_\bullet \cup V_\circ$, $V_\bullet = \{x_1, \ldots, x_n\} \neq \varnothing$ (*black* nodes), $V_\circ = \{y_1, \ldots, y_m\}$ (*white* nodes), $V_\bullet \cap V_\circ = \varnothing$, and $S_\ell \subseteq V_\bullet \times V_\bullet$ (*solid* arrows) and $\Pi_\ell \subseteq V\times V$ (*dashed* arrows) are binary relations, for each $\ell \in L$. By D we denote the $\exists$-diagram obtained from $\mathcal{D}$ by throwing away solid arrows $(S_\ell)_{\ell\in L}$ (in words, D is the existential part of $\mathcal{D}$). Recall from Sect. 3 that every $\exists$-diagram gives rise to an $\exists$&-formula $A_D(x_1, \ldots, x_n)$. Now, to every $\forall\exists$-diagram $\mathcal{D}$ we associate the following FO formula with one free variable x_1:

$$A_{\mathcal{D}}(x_1) = \forall x_2 \ldots \forall x_n \Big(\big(\textstyle\bigwedge_{\langle x,x'\rangle \in S_\ell} x\mathsf{R}_\ell x' \big) \to A_D(x_1, \ldots, x_n) \Big).$$

A closed FO formula A is said to be (*globally*) *modally definable* if there is a modal formula φ such that $F \models A$ iff $F \models \varphi$, for every frame F. It is easily seen that if $A(x)$ is (locally) modally definable, then $\forall x\, A(x)$ is globally modally definable. Here is the main result of this section.

Theorem 6.3. *Let $\mathcal{D}$ be an $\forall\exists$-diagram satisfying the following conditions:*

(*a*) *$(V_\bullet, (S_\ell)_{\ell\in L})$ is a tree with the root x_1, i.e., x_1 has no incoming arrows and every other node from $V_\bullet$ has exactly one incoming arrow;*

(*b*) *its existential part D is accessible and white-acyclic (see Def. 3.1).*

Then the formula $A_{\mathcal{D}}(x_1)$ is locally modally definable, and consequently, the formula $\forall x_1 A_{\mathcal{D}}(x_1)$ is globally modally definable. Moreover, the formula $A_{\mathcal{D}}(x_1)$ corresponds to a modal generalized Sahlqvist formula.

Proof. Condition (*a*) means that we can assume that x's are enumerated in such a way that for every $i \geq 2$ there is a number $p(i) \leq i$ and an index $\ell(i) \in L$ such that $x_{p(i)}$ is the only predecessor of x_i in $(V_\bullet, (S_\ell)_{\ell\in L})$, and $\langle x_{p(i)}, x_i\rangle \in S_{\ell(i)}$. Hence the formula $A_{\mathcal{D}}(x_1)$ can be equivalently rewritten as

$$\forall x_2\, (x_1 \mathsf{R}_{\ell(2)} x_2 \to \forall x_3\, (x_{p(3)} \mathsf{R}_{\ell(3)} x_3 \to \ldots \forall x_n\, (x_{p(n)} \mathsf{R}_{\ell(n)} x_n \to A_D(\vec{x})) \ldots)).$$

By (*b*) and Theorem 4.1, $A_D(\vec{x})$ is a conjunction of modally definable first-order formulas that correspond to the connected components of D. Therefore, by Theorem 6.1, $A_{\mathcal{D}}(x_1)$ is locally modally definable.

In addition, since D is accessible and white-acyclic, its $\exists$&-formula $A_D(\vec{x})$ is equivalent to a conjunction of connected such $\exists$&-formulas. By Theorem 4.12, each of these conjuncts corresponds to an expression of the form $x_i\colon t(\vec{x})$, for some safe term t. Since the above formula $A_{\mathcal{D}}(x_1)$ is built up from these formulas using conjunction and restricted universal quantification, it is a *generalized Kracht formula* introduced in [20, Def. 29]. Therefore, by Theorem 30 from [20], it corresponds to some modal generalized Sahlqvist formula. □

It is known from [22] that the cubifying property displayed in Fig. 5 does not correspond to any *ordinary* Sahlqvist modal formula. Thus, Theorem 6.3 gives us a family of modally definable FO formulas which is not covered by the classical Sahlqvist – Kracht correspondence theory.

It would be interesting to investigate whether our undefinability results for $\exists$-diagrams (see Sect. 5) can be generalized to $\forall\exists$-diagrams. For example, for many white-cyclic $\forall\exists$-diagrams we can prove modal undefinability using a modification of the construction from Theorem 5.15. This class of diagrams can be roughly described as "diagrams that do not have unexpected consequences", but formalization of this concept is rather cumbersome. At the same time, the problem of determining, given two $\forall\exists$-diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$, whether $A_{\mathcal{D}_1}$ implies $A_{\mathcal{D}_2}$ is undecidable [2]. This leaves little hope for an *algorithmic* criterion of modal definability of $\forall\exists$-diagrams.

7. Application to query answering

Description Logics (DLs) are knowledge representation formalisms. They provide, for example, the logical underpinning of the Web Ontology Language OWL [26, 1]. The problem of answering conjunctive queries in DL knowledge bases, which is a standard task in databases, has recently gained significant attention for expressive DLs (see, e.g., [31, 7, 13, 16, 29, 32, 23, 17, 15] and references therein). Here we show that the results obtained above can be used for answering efficiently a wide class of conjunctive queries.

In few words, the result obtained below can be presented as follows. Given a theory T and a first-order formula $q(\vec{x})$, called a *query* in this context, consider the task of finding all *answers* to it, i.e., tuples of constants $\vec{c}$ such that $T \models q(\vec{c})$. Assume that additionally we know that $q(\vec{x})$ is modally definable, i.e., it corresponds to a tuple of modal formulas $\langle \varphi_1, \ldots, \varphi_n \rangle$. Then the Reduction Theorem (see Theorem 7.2 below) guarantees that the query $q(\vec{x})$ has the same answers as the disjunction $\varphi_1^*(x_1) \vee \ldots \vee \varphi_n^*(x_n)$, where $\varphi^*(x)$ is the so-called *standard translation* of a modal formula φ into the FO language [5, Sect. 2.4]. This reduction was first obtained in [36] for ordinary modal formulas and in [37] for extended modal formulas, but in both cases only for queries $q(x)$ with one free variable. So, here we generalize these results to the case of several variables.

Why such a reduction is useful? The reason is that we are going to apply this method to theories T of a special kind (so-called *knowledge bases*). Only special types of axioms are admissible in these theories, in particular, those of the form $\varphi^*(c)$, i.e., obtained by substituting a constant in the standard translation of some modal formulas. For such theories it is known that the *consistency problem* ("Given a theory, determine whether it is consistent") is decidable, and efficient algorithms are already implemented. Now observe that, for a given tuple of constants $\vec{c}$, the entailment $T \models \varphi_1^*(c_1) \vee \ldots \vee \varphi_n^*(c_n)$ is equivalent to inconsistency of the theory $T \cup \{\neg\varphi_1^*(c_1), \ldots, \neg\varphi_n^*(c_n)\}$, which is itself a knowledge base, and hence whether it is (in)consistent can be verified by well-known algorithms.

We proceed as follows. In Sect. 7.1, we prove the Reduction Theorem in a general form that is suitable for applying to a wide spectrum of description logics. Sect. 7.2 introduces necessary notions from description logic and indicates their relationship to modal and first-order logic. Sect. 7.3 contains definitions related to query answering, as well some known results in the area. Finally, in Sect. 7.4, we prove the main theorem that allows us to "easily" answer a certain family of conjunctive queries using our modal definability results obtained in the first part of the paper (in Sect. 4).

7.1. *Query answering in theories*

Below, the term *an n-ary query* is a synonym for a FO formula with n free variables. Here we describe a method of reducing the problem of answering some n-ary queries (with respect to FO theories) to the problem of answering disjunctions of n unary queries of a simple kind (standard translations of modal formulas). To this end, we need two FO signatures:

- $\Sigma_{\mathsf{query}} = \{=\} \cup \{\mathsf{R}_\ell \mid \ell \in L\}$ is a binary relational signature in which queries are formulated;
- $\Sigma_{\mathsf{modal}} = \Sigma_{\mathsf{query}} \cup \{P_0, P_1, \ldots\}$ is the signature in which the standard translation of a modal formula is written; here P_i are unary predicate symbols.

Observe that a FO Σ_{query} interpretation is essentially a Kripke frame, whereas a FO Σ_{modal} interpretation can be seen as a Kripke model, once we read the interpretation of P_i as the valuation $\theta(p_i)$. Furthermore, $\exists\&$-formulas defined in Sect. 3 are FO Σ_{query} formulas.

Given a modal formula φ, we denote by $\varphi^*(x)$ its *standard translation* [5, Sect. 2.4] defined as follows (Boolean cases are treated as usual, the variable y is fresh):

$$p_i^*(x) := P_i(x), \qquad (\Box_\ell\varphi)^*(x) := \forall y\,(\,x\mathsf{R}_\ell y \to \varphi^*(y)\,).$$

Note that $\varphi^*(x)$ is a FO Σ_{modal} formula. Since the standard translation "mimics" the semantics of modal formulas, we have $M \models \varphi^*(e)$ iff $M, e \models \varphi$, for every Kripke model M (i.e., a Σ_{modal} interpretation) and every point e in M. The standard translation extends naturally to modal expressions $\Phi(\vec{x})$. Below, we will sometimes write $x\colon\varphi$ as a shortcut for $\varphi^*(x)$, and so $\Phi(\vec{x})$ may stand for a modal expression or for its standard translation (semantically, they are equivalent).

Let $\Sigma = (\mathsf{Pred}, \mathsf{Const})$ be a countable FO signature consisting of some predicate symbols (of arbitrary arities) and constants. A signature Σ will be called *admissible* if it does not contain the symbols $\{P_0, P_1, \ldots\}$ (which are reserved for translating modal formulas into the FO language). Now comes the main notion of this section, first introduced in [36] for the case $n = 1$.

Definition 7.1 (Queries answered by modal formulas)**.** An n-ary query $q(x_1, \ldots, x_n)$ is *answered by* an n-tuple of modal formulas $\langle \varphi_1, \ldots, \varphi_n \rangle$ if, for every admissible signature Σ, every first-order theory T in Σ, and all constants $c_1, \ldots, c_n$ in Σ, the following equivalence holds:

$$T \models q(c_1, \ldots, c_n) \quad\Longleftrightarrow\quad T \models c_1\colon\varphi_1 \vee \ldots \vee c_n\colon\varphi_n.$$

In this case we use notation: $q(\vec{x}) \approx \langle \varphi_1, \ldots, \varphi_n \rangle$. Intuitively, this means that the query $q(\vec{x})$ has always the same answers as the query $x_1\colon\varphi_1 \vee \ldots \vee x_n\colon\varphi_n$.

Notice the difference with Def. 2.1. There, we quantified over arbitrary frames F and its worlds $\vec{e}$; here we quantify over any FO theories T (in admissible signatures) and constants $\vec{c}$ in them. In the definition of validity $F \models e_1\colon\varphi_1 \vee \ldots \vee e_n\colon\varphi_n$, the frame F interprets only modalities (i.e., binary relations), whereas we universally quantify over valuations of propositional variables that occur in φ_i. Similarly, here in the entailment $T \models c_1\colon\varphi_1 \vee \ldots \vee c_n\colon\varphi_n$, the theory T does not contain the predicate symbols $P_0, P_1, \ldots$ that occur to the right of $\models$, and hence they are actually universally quantified (although these monadic second-order quantifiers are not explicitly written).

Theorem 7.2 (Reduction Theorem)**.** *Let $q(x_1,\ldots,x_n)$ be an n-ary query in Σ_{query} and $\varphi_1,\ldots,\varphi_n$ be modal formulas. If $q(\vec{x})$ corresponds to $\langle\varphi_1,\ldots,\varphi_n\rangle$, then $q(\vec{x})$ is answered by $\langle\varphi_1,\ldots,\varphi_n\rangle$. In symbols:*

$$q(\vec{x}) \leftrightsquigarrow \langle\varphi_1,\ldots,\varphi_n\rangle \quad\Longrightarrow\quad q(\vec{x}) \approx \langle\varphi_1,\ldots,\varphi_n\rangle.$$

Proof. Let $\Phi(\vec{x})$ be a shortcut for $x_1\colon\varphi_1 \vee \ldots \vee x_n\colon\varphi_n$. Suppose that $q(\vec{x})$ corresponds to $\langle\varphi_1,\ldots,\varphi_n\rangle$, i.e., to $\Phi(\vec{x})$. Then for every frame F and all worlds $\vec{e}$ in F, we have

$$F \models q(\vec{e}) \quad\Longleftrightarrow\quad F \models \Phi(\vec{e}). \qquad (\leftrightsquigarrow)$$

We need to prove that, for every theory T and all constants $\vec{c}$ in every admissible signature Σ,

$$T \models q(\vec{c}) \quad\Longleftrightarrow\quad T \models \Phi(\vec{c}). \qquad (\approx)$$

($\Rightarrow$) Assume that $T \models q(\vec{c})$. In order to prove that $T \models \Phi(\vec{c})$, let us take any $\Sigma \cup \Sigma_{\mathsf{modal}}$ interpretation $\mathcal{I}$ (i.e., any interpretation of both T and Φ) such that $\mathcal{I} \models T$ and show that $\mathcal{I} \models \Phi(\vec{c})$. Let F be the frame underlying $\mathcal{I}$, and denote $\vec{e} := \vec{c}^{\mathcal{I}}$. From $\mathcal{I} \models T$ and $T \models q(\vec{c})$ we infer $\mathcal{I} \models q(\vec{c})$, or equivalently, $F \models q(\vec{e})$, since q is a Σ_{query} formula. By ($\leftrightsquigarrow$), we have $F \models \Phi(\vec{e})$. Hence $\mathcal{I} \models \Phi(\vec{e})$, because $\mathcal{I}$ contains a model based on F, or equivalently, $\mathcal{I} \models \Phi(\vec{c})$, as desired.

($\Leftarrow$) Assume that $T \models \Phi(\vec{c})$. In order to prove that $T \models q(\vec{c})$, let us take any $\Sigma \cup \Sigma_{\mathsf{query}}$ interpretation $\mathcal{I}$ (i.e., any interpretation of both T and q) such that $\mathcal{I} \models T$ and show that $\mathcal{I} \models q(\vec{c})$. Since q is a Σ_{query} formula, this is equivalent to showing that $F \models q(\vec{e})$, where F is the frame underlying $\mathcal{I}$ and $\vec{e} := \vec{c}^{\mathcal{I}}$. By ($\leftrightsquigarrow$), it suffices to prove that $F \models \Phi(\vec{e})$.

To this end, let us take any Kripke model (i.e., a Σ_{modal} interpretation) M based on F and show that $M \models \Phi(\vec{e})$. Let $\mathcal{J}$ be a $\Sigma \cup \Sigma_{\mathsf{modal}}$ interpretation whose restriction to Σ is $\mathcal{I}$ and the restriction to Σ_{modal} is M. Such an interpretation exists (and is even unique), since $\mathcal{I}$ and M agree on their common signature Σ_{query} and Σ does not contain symbols from $\Sigma_{\mathsf{modal}} \setminus \Sigma_{\mathsf{query}}$. Then we have $\mathcal{J} \models T$. Now recall that $T \models \Phi(\vec{c})$. Hence $\mathcal{J} \models \Phi(\vec{c})$, or equivalently, $M \models \Phi(\vec{e})$, since Φ is a Σ_{modal} formula, thus we are done. □

Remark 7.3. The converse of this theorem does not hold in general; however, if we exclude a simple reason for its failure, then the problem is open. More exactly, let $\overset{\omega}{\leftrightsquigarrow}$ stand for the *countable correspondence* of first-order and modal formulas, i.e., instead of saying 'for every frame F' in Def. 2.1, now we say 'for every at most countable frame F'. Then the picture looks as follows:

$$q(\vec{x}) \leftrightsquigarrow \langle\varphi_1,\ldots,\varphi_n\rangle \quad\Longrightarrow\quad q(\vec{x}) \overset{\omega}{\leftrightsquigarrow} \langle\varphi_1,\ldots,\varphi_n\rangle \quad\Longrightarrow\quad q(\vec{x}) \approx \langle\varphi_1,\ldots,\varphi_n\rangle$$

The implication from $\leftrightsquigarrow$ to $\overset{\omega}{\leftrightsquigarrow}$ is trivial. The converse implication fails even in case $n = 1$, due to a counterexample by Doets (see [10, p. 186]). The implication from $\overset{\omega}{\leftrightsquigarrow}$ to $\approx$ can be proved by a simple modification of the proof of the Reduction Theorem 7.2, since first-order entailment $T \models A$ is equivalent to the entailment over countable structures, by Löwenheim–Skolem theorem. Finally, whether the implication from $\approx$ to $\overset{\omega}{\leftrightsquigarrow}$ holds is an open problem.

7.2. Description logic knowledge bases

Syntax. The basic DL $\mathcal{ALC}$ is introduced as follows. Its vocabulary $\Sigma = (\mathsf{CN}, \mathsf{RN}, \mathsf{IN})$ consists of finite sets of *concept names* CN, *role names* RN, and *individual names* IN (also called *constants*). *Concepts* (analogues of formulas) of $\mathcal{ALC}$ are built up according to the following syntax:

$$C, D ::= \quad A \mid \neg C \mid C \sqcap D \mid \forall R.C,$$

where A is a concept name, R a role name, C and D concepts. Other connectives are taken as shortcuts, e.g. $(C \sqcup D) := \neg(\neg C \sqcap \neg D)$, $\exists R.C := \neg\forall R.\neg C$.

A *TBox* (or a *terminology*) is a finite set of *axioms* of the form $C \sqsubseteq D$, where C, D are arbitrary concepts. An *ABox* is a finite set of *assertions* of the form $a\colon C$ and aRb, where C is a concept, R a role name, and $a, b \in \mathsf{IN}$. Finally, a *knowledge base* (KB) $\mathcal{K} = \langle\mathcal{T}, \mathcal{A}\rangle$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

Semantics. An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty set called a *domain*, and $\cdot^{\mathcal{I}}$ is an interpretation function that maps:

- each constant $a \in \mathsf{IN}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$,
- each concept name $A \in \mathsf{CN}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- each role name $R \in \mathsf{RN}$ to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$,

and is extended to all concepts (so that $C^{\mathcal{I}}$ is always a subset of $\Delta^{\mathcal{I}}$) as follows:

$$\begin{aligned}(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}},\\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}},\\ (\forall R.C)^{\mathcal{I}} &= \{e \in \Delta^{\mathcal{I}} \mid \forall d \in \Delta^{\mathcal{I}}.\, \langle e, d\rangle \in R^{\mathcal{I}} \Rightarrow d \in C^{\mathcal{I}}\}.\end{aligned}$$

The *satisfaction* relation is defined as follows: $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; $\mathcal{I} \models a\colon C$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$; $\mathcal{I} \models aRb$ iff $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$. Furthermore, $\mathcal{I}$ is called a *model* of a knowledge base $\mathcal{K}$ (notation: $\mathcal{I} \models \mathcal{K}$) if it satisfies all its TBox axioms and ABox assertions. A knowledge base $\mathcal{K}$ is called *consistent* if it has a model.

Relationship to modal logic. The DL $\mathcal{ALC}$ is known (cf. [34]) to be a notational variant of the modal logic discussed above (see Sect. 2 for its syntax). Namely, any modal formula φ can be rewritten into a DL concept C_φ by replacing p_i with A_i, $\wedge$ with $\sqcap$, $\Box_\ell$ with $\forall R_\ell$ (and hence $\vee$ with $\sqcup$ and $\Diamond_\ell$ with $\exists R_\ell$). Conversely, one can translate any DL concept into a modal formula.

Moreover, this translation respect semantics. A DL interpretation $\mathcal{I}$ can be seen as a Kripke model, where the interpretations of role names $R_\ell^{\mathcal{I}}$ serve as accessibility relations and interpretations of concept names $A_i^{\mathcal{I}}$ as valuations of propositional variables p_i. Conversely, any Kripke model yields a DL interpretation. In this setting, for any model $\mathcal{I}$ and its world e, for any modal formula φ, we have: $\mathcal{I}, e \models \varphi$ iff $e \in C_\varphi^{\mathcal{I}}$.

We leave the discussion on the modal counterpart of knowledge bases beyond this paper.

Relationship to first-order logic. The *standard translation* maps any $\mathcal{ALC}$ concept C into a FO formula $C^*(x)$ with one free variable in the signature $\Sigma = (\mathsf{Pred}, \mathsf{Const})$, where Pred consists of unary predicates from CN and binary predicates from RN, and $\mathsf{Const} = \mathsf{IN}$. It is defined by induction on the syntax of concepts: a concept name $A \in \mathsf{CN}$ turns into $A(x)$, Boolean connectives $\neg$ and $\sqcap$ map into $\neg$ and $\wedge$, and finally $(\forall R_\ell.C)^*(x) = \forall y\,(\mathsf{R}_\ell(x, y) \to C^*(y))$, where y is a fresh variable.

Furthermore, a TBox axiom $C \sqsubseteq D$ is translated into the closed formula $\forall x\,(C^*(x) \to D^*(x))$; ABox assertions $a\colon C$ and aRb are translated into the closed formulas $C^*(a)$ and $\mathsf{R}(a, b)$, respectively. Thus, a knowledge base $\mathcal{K}$ is translated into a collection of closed formulas, i.e., a first-order theory $\mathcal{K}^*$.

This translation respects semantics. Any DL interpretation $\mathcal{I}$ can be seen as a first-order $(\mathsf{Pred}, \mathsf{Const})$-structure. Then, for any concept C and any element e in $\mathcal{I}$, we have: $e \in C^{\mathcal{I}}$ iff $\mathcal{I} \models C^*(e)$. Similarly, for any TBox axiom or ABox assertion E, we have $\mathcal{I} \models E$ (in the DL sense) iff $\mathcal{I} \models E^*$ (in the FO sense).

Expressive description logics. The DL $\mathcal{ALC}$ has been extended in various ways to meet the needs of practical applications; let us recall some of them. According to the tradition of naming these extensions, the letters $\mathcal{I}, \mathcal{O}, \mathcal{Q}, \mathcal{H}, \mathcal{S}$ in the name of a logic refer to the presence of the following features:

- $\mathcal{I}$: *inverse roles*: if R is a role, the R^- is a role; semantics: $(R^-)^{\mathcal{I}} = \{\langle e, d\rangle \mid \langle d, e\rangle \in R^{\mathcal{I}}\}$;
- $\mathcal{O}$: *nominals*: if $a \in \mathsf{IN}$, then $\{a\}$ is a concept; semantics: $\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$;
- $\mathcal{Q}$: *qualified number restrictions*: if R is a role, C a concept, and $n > 0$, then the $(\geq n\, R.C)$ is a concept; semantics:

$$(\geq n\, R.C)^{\mathcal{I}} = \left\{ e \in \Delta^{\mathcal{I}} \;\middle|\; \begin{array}{c}\text{there are at least } n \text{ elements } d \in \Delta^{\mathcal{I}} \\ \text{such that } \langle e, d\rangle \in R^{\mathcal{I}} \text{ and } d \in C^{\mathcal{I}}\end{array} \right\}.$$

- $\mathcal{H}$: *role hierarchy*: axioms of the form $R \sqsubseteq S$ are allowed in a special part of a TBox called an *RBox*; semantics: $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$;
- $\mathcal{S}$: *transitivity axioms*: axioms of the form $\mathsf{Trans(R)}$ are allowed in an RBox; semantics: $\mathcal{I} \models \mathsf{Trans(R)}$ iff $R^{\mathcal{I}}$ is a transitive relation.

To be more precise, the letters $\mathcal{I}, \mathcal{O}, \mathcal{Q}, \mathcal{H}$ are appended to the name of the logic $\mathcal{ALC}$, whereas the letter $\mathcal{S}$ replaces the three letters $\mathcal{ALC}$. To maintain decidability, in presence of $\mathcal{S}$ and $\mathcal{Q}$ together, the restriction is imposed on the syntax of concepts: expressions of the form $(\geq n\, R.C)$ are regarded as well-formed concepts only if the role R is *simple* (i.e., has no transitive subroles) with respect to a given RBox. For instance, $\mathcal{SHIQ}$ extends $\mathcal{ALC}$ with inverse roles, qualified number restrictions, role transitivity and role inclusion axioms, and has the above mentioned restriction on the syntax of concepts.

Reasoning. For DLs extending $\mathcal{ALC}$ the following is the central decision problem:

Knowledge base consistency problem:
given a knowledge base $\mathcal{K}$, decide whether it is consistent.

For all DLs described above, this problem is decidable, and its complexity was extensively studied. Let us summarize some known results.

Theorem 7.4 ([33, 34, 35])**.** *The knowledge base consistency problem is*
- ExpTime*-complete for any logic between $\mathcal{ALC}$ and $\mathcal{SHIQ}$;*
- NExpTime*-complete for any logic between $\mathcal{ALCOIQ}$ and $\mathcal{SHOIQ}$.*

7.3. Query answering in knowledge bases

To formulate queries, we need individual variables Var (the same that are used in first-order formulas, see Sect. 2).

Definition 7.5. A *(conjunctive) query* $q(\vec{x})$ is an expression of the form $\exists \vec{y}\, Q(\vec{x}, \vec{y})$, where $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_m)$ are its *free* and *bound* variables from Var, Q is a conjunction of atomic formulas of the form $z\colon C$ and uRv, with C a concept, R a role name, z a variable from Var, and u, v either variables from Var or constants from IN. The number of free variables n is called the *arity* of the query. Queries of arity 0 are called *Boolean*.

A query can be seen as a FO formula, if we equivalently rewrite conjuncts of the form $z\colon C$ as $C^*(z)$; and if the query contains neither constants nor conjuncts of the form $z\colon C$, it is an $\exists\&$-formula (see Sect. 3). Therefore, given a knowledge base $\mathcal{K}$ (which can be seen as a FO theory) and a tuple of constants $\vec{a} = (a_1, \dots, a_n)$ from IN, the entailment $\mathcal{K} \models q(\vec{a})$ is well-defined. If it holds, we say that $\vec{a}$ is an *answer* to the query $q(\vec{x})$ in $\mathcal{K}$. The following algorithmic problems are central in this area.

Query entailment problem:
given a knowledge base $\mathcal{K}$, a conjunctive query $q(\vec{x})$, and a tuple of constants $\vec{a}$ from IN, *decide whether $\vec{a}$ is an answer to $q(\vec{x})$ in $\mathcal{K}$.*

Query answering problem:
given a knowledge base $\mathcal{K}$ and a conjunctive query $q(\vec{x})$, find all tuples of constants $\vec{a}$ from IN *that are answers to $q(\vec{x})$ in $\mathcal{K}$.*

The query answering problem is trivially reduced to the query entailment problem by searching through all tuples of constants. However, in many cases, even the query entailment problem is computationally harder than the problem of knowledge base consistency, as the following recent results illustrate.

Theorem 7.6 ([29, 15])**.** *Query entailment problem is*

- ExpTime-*complete for any logic between $\mathcal{ALC}$ and $\mathcal{SHQ}$;*
- 2ExpTime-*complete for any logic between $\mathcal{ALCI}$ and $\mathcal{SHIQ}$;*
- co-N2ExpTime-*hard (but decidable) for the logic $\mathcal{ALCOIQ}$.*

In these results for logics that allow for transitivity axioms, atomic formulas of the form uRv in conjunctive queries may only involve simple roles R.

Notably, the KB consistency problem for the DLs mentioned above can be handled by numerous modern reasoners, while query answering algorithms are still waiting for their efficient implementations. In this context the cases when the query answering problem is polynomially reducible to the KB (in)consistency problem are especially interesting, because the query answering algorithm can be implemented easier than in general. Such a reduction is almost trivial for acyclic queries (and known as the *rolling up* technique). In [36] a reduction based on classical modal corresponding theory was suggested for unary queries $q(x)$ without cycles that involve only bound variables.

Here we extend this technique to queries of arbitrary arity. Before we present it, we need a lemma that allows us to consider only queries that are essentially $\exists\&$-formulas (with constants).

Lemma 7.7. *The conjunctive query answering problem is linearly reducible to the same problem for queries that do not involve conjuncts of the form $z\colon C$.*

Proof. Suppose that we are given a knowledge base $\mathcal{K}$ and a conjunctive query $q(\vec{x})$ of the form:

$$q(\vec{x}) = \exists\vec{y}\ (Q(\vec{x},\vec{y}) \wedge x_1\colon C_1 \wedge \dots \wedge x_n\colon C_n \wedge y_1\colon D_1 \wedge \dots \wedge y_m\colon D_m),$$

where Q consists only of conjuncts of the form uRv. Let $q'(\vec{x})$ be a query obtained from $q(\vec{x})$ by replacing each $x_i\colon C_i$ with $x_i\colon \exists R_i.\top$ and each $y_j\colon D_j$ with $y_j\colon \exists S_j.\top$, where R_i and S_j are fresh role names. Consider a knowledge base

$$\mathcal{K}' \;=\; \mathcal{K} \cup \{\, C_i \equiv \exists R_i.\top \mid 1 \le i \le m \,\} \cup \{\, D_j \equiv \exists S_j.\top \mid 1 \le j \le m \,\}.$$

Then one can easily prove that the query q has the same answers in $\mathcal{K}$ as q' in $\mathcal{K}'$; that is, $\mathcal{K} \models q(\vec{a}) \Leftrightarrow \mathcal{K}' \models q'(\vec{a})$, for any tuple of constants $\vec{a}$. Now observe that $x_i\colon \exists R_i.\top$ is equivalent to $\exists z_i\,(x_i R_i z_i)$, and similarly for $y_j\colon \exists S_j.\top$. Therefore, the query $q'(\vec{x})$ is equivalent to the query

$$q''(\vec{x}) = \exists\vec{y}\,\exists\vec{z}\,\exists\vec{v}\ (Q(\vec{x},\vec{y}) \wedge x_1R_1z_1 \wedge \dots \wedge x_nR_nz_n \wedge y_1S_1v_1 \wedge \dots \wedge y_mS_mv_m),$$

which has the desired form. □

7.4. Main result for query entailment

Now we are ready to apply the Reduction Theorem 7.2 to our modal definability result (Theorem 4.1) and thus obtain the following result that enables us to answer "easily" a wide family of conjunctive queries.

Theorem 7.8. **(a)** *Suppose that $q(\vec{a})$ is a Boolean conjunctive query, where $\vec{a}$ is an n-tuple of constants and $q(\vec{x})$ is a query without constants whose graph satisfies the following conditions:*

(1) *it is connected;*
(2) *each bound variable is reachable from some free variable via a directed path;*
(3) *it has no cycles (even undirected) containing only bound variables.*

Then, in any description logic extending $\mathcal{ALC}$, the query entailment problem for queries of this kind is reducible, in time polynomial in the size of a query, to the problem of knowledge base inconsistency.

(b) *If only conditions* (1) *and* (3) *are satisfied, then a similar reduction is available in any description logic extending $\mathcal{ALCI}$.*

Proof. **(a)** By Lemma 7.7, without loss of generality we can assume that $q(\vec{x})$ contains no concept names. Note that the application of Lemma 7.7 preserves conditions (1–3). Then $q(\vec{x})$ is an ∃&-formula (see Sect. 3) and is connected, accessible, and white-acyclic (see Def. 3.1).

Therefore, by Theorem 4.1, the first-order formula $q(\vec{x})$ *corresponds* to a tuple of modal formulas $\langle\varphi_1, \ldots, \varphi_n\rangle$ with $n = |\vec{x}|$, which can be built in time polynomial (in fact, at most quadratic) in the size of the query q, according to the procedure described in Sect. 4.3–4.4.

Let the $\mathcal{ALC}$-concepts $C_1, \ldots, C_n$ be the notational variants of the modal formulas $\varphi_1, \ldots, \varphi_n$, where we assume that propositional letters p_i are translated into *fresh* concept names P_i, i.e., which cannot occur in any KB against which we answer queries. The concept names P_i serve as internal (local) variables in our query answering algorithm.

Now we apply the Reduction Theorem 7.2, which implies that $q(\vec{x})$ is *answered by* the tuple of modal formulas $\langle\varphi_1, \ldots, \varphi_n\rangle$. This means that, for any first-order theory, in particular, for any knowledge base $\mathcal{K}$ not involving the concept names P_i, the equivalence holds:

$$\begin{aligned} \mathcal{K} \models q(a_1, \ldots, a_n) &\iff \mathcal{K} \models a_1\!:\varphi_1 \vee \ldots \vee a_n\!:\varphi_n \\ &\iff \mathcal{K} \models a_1\!:C_1 \vee \ldots \vee a_n\!:C_n \end{aligned}$$

Using notation similar to that from Def. 7.1, we can write this fact as $q(\vec{x}) \approx x_1\!:C_1 \vee \ldots \vee x_n\!:C_n$. It remains to note that the following equivalence holds:

$$\mathcal{K} \models a_1\!:C_1 \vee \ldots \vee a_n\!:C_n \iff \mathcal{K} \cup \{a_1\!:\neg C_1, \ldots, a_n\!:\neg C_n\} \text{ is inconsistent.}$$

Thus we reduced query entailment to the KB inconsistency problem:

$$\mathcal{K} \models q(a_1, \ldots, a_n) \iff \mathcal{K} \cup \{a_1\!:\neg C_1, \ldots, a_n\!:\neg C_n\} \text{ is inconsistent.}$$

(b) Intuitively, "forgetting" the direction of edges in the graph of a query makes condition (2) redundant, as in this case it follows from (1). Repeating the argument from **(a)** yields $\mathcal{ALC}$-concepts C_j in which some roles are inverted, i.e., $\mathcal{ALCI}$-concepts.

More precisely, it is easily seen that a query $q(\vec{x})$ satisfying (1) and (3) can be transformed into a query satisfying (1–3) by inverting some of its edges. Simply, for every white node we can find a minimal undirected path to it from black nodes and turn it into a directed path (note that different paths will not conflict).

Having this in mind, take every edge (i.e., conjunct) $z_i\, R_{\ell_i} v_i$ in $q(\vec{x})$ involved in the above transformation and replace it with the edge $v_i\, S_i\, z_i$, where S_i is a fresh role name. This way we obtain a new query $p(\vec{x})$ that satisfies conditions (1–3) and involves the original role names $\{R_\ell \mid \ell \in L\}$ and the fresh ones $\{S_i \mid i \in I\}$.

By **(a)**, we have $p(\vec{x}) \approx x_1\!:C_1 \vee \ldots \vee x_n\!:C_n$, for some $\mathcal{ALC}$-concepts C_j involving role names R_ℓ and S_i. Now let us replace here S_i with $R^-_{\ell_i}$, for all $i \in I$, and then rewrite every conjunct of the form $v\, R^-_\ell\, z$ into an equivalent one $z\, R_\ell\, v$. On the left-hand side of $\approx$, we will obtain the original query $q(\vec{x})$. On the right-hand side of $\approx$, the $\mathcal{ALC}$-concepts C_j will turn into $\mathcal{ALCI}$-concepts D_j. Since $\approx$ is preserved under substituting (inverse) roles for roles (an easy exercise), we obtain that $q(\vec{x}) \approx x_1\!: D_1 \vee \ldots \vee x_n\!: D_n$, as desired. □

Note that conditions (1–3) can be verified in polynomial time. Thus, given a query, we can check in polynomial time whether our technique is applicable (or we give up), and if so, we have a polynomial reduction of the problem of answering this query to the problem of knowledge base inconsistency.

It is important to emphasize the uniformity of this technique: concepts $C_1, \ldots, C_n$ used for answering a query are not only independent of a KB against which the query is answered, but also independent of a DL in which a KB is formulated (and does not impose restrictions such as those at the end of Theorem 7.6). Therefore, extending the expressive power of a DL does not destroy this query answering algorithm, in contrast to other approaches.

Remark 7.9. The connectedness condition (1) is not an obstacle, since answering an unconnected query is equivalent to answering its connected components independently and then intersecting the results: $\vec{a}$ is an answer to $q_1(\vec{x}) \wedge q_2(\vec{x})$ iff it is an answer to both $q_1(\vec{x})$ and $q_2(\vec{x})$.

7.5. Discussion

Let us summarize the essence of our query answering method. We start with a conjuctive Boolean query $q(a_1, \ldots, a_n)$, which is essentially a *first-order* formula. Then, if we are lucky and the query has the required form, we replace it with a *second-order* formula $\forall P_1 \ldots \forall P_m (C^*_1(a_1) \vee \ldots \vee C^*_n(a_n))$ and, curiously enough, win in complexity, because, given a KB $\mathcal{K}$, the search for a model in which $\mathcal{K}$ holds and this second-order formula fails is equivalent to checking the consistency of the knowledge base $\mathcal{K} \cup \{a_1\!: \neg C_1, \ldots, a_n\!: \neg C_n\}$.

We believe that, for deciding entailment of modally definable CQs, this method is perfect in practice, because

- it can be easily implemented using any off-the-shelf reasoner that supports KB consistency checking;
- implemented once, it will work for any description logic extending $\mathcal{ALC}$;
- it works as fast as a KB consistency check; the latter can be done, as numerous experiments with modern reasoners show, in a fraction of a second.

However, if we are interested in query answering, i.e., in *retrieving* all tuples of constants $\vec{d}$ that satisfy a query $q(\vec{x})$, the only way to do this is to search through all tuples of constants and check if they are answers to $q(\vec{x})$, which is still hard for modern computers. So, a possible direction for future research is to look for ways of reducing this search. For some light-weight DLs (in particular, for $\mathcal{EL}$ [30] and dialects of DL-Lite [7, 23]) this search can be delegated to the Relational Database Management Systems using the technique called "query rewriting". It would be interesting to study how far these two methods can be combined for expressive DLs.

8. Conclusion and future work

In this paper, we presented an algorithm for checking modal definability of first-order ∃&-formulas with several free variables and producing the corresponding modal formulas. We used this criterion for describing a large family of first-order ∀∃-formulas with one free variable that are modally definable, some of which lie beyond the so-called Kracht's fragment. As an application of these theoretical investigations, we used our definability results to obtain an efficient algorithm for answering a certain family of conjunctive queries, by providing a polynomial-time reduction of the problem of answering these queries to the problem of knowledge base consistency.

The research induced further questions, some of which we formulate below.

1. Can our criterion (Sect. 3.1) of modal definability of ∃&-formulas be extended to ∀∃-formulas (Def. 6.2)? Will it be algorithmically efficient?
2. More generally, can we classify fragments of the first-order logic with respect to the decidability of the modal definability problem for FO formulas? This question is inspired by the well-known classification of fragments of FOL with respect to the decidability of the satisfiability problem for FO formulas [6].
3. Which ∀∃-formulas (Def. 6.2) are equivalent to formulas from Kracht's fragment?
4. Safe terms are modally definable (Theorem 4.13). What other terms are modally definable? Is it decidable to check whether a term is modally definable? We conjecture that a term is modally definable iff it is equivalent to a safe term. Equivalence is not avoidable here: the terms $\Diamond\diamondminus\Diamond\top$ and $\Diamond(x \wedge \diamondminus\Diamond x)$ are not safe, but they are equivalent to the safe terms $\Diamond\top$ and $\Diamond x$, respectively, and so are modally definable.
5. Is there a reasonable explanation to the choice of the operators $\wedge, \vee, \Diamond, \Box, \diamondminus$ in the syntax of terms (Def. 4.6)? Why are simple terms defined with the help of $\diamondminus\wedge$-paths (Def. 4.9)? If we add $\boxminus$ to the syntax of terms, how can the notions of simple and safe terms be generalized to this case so that simple terms are still modally definable?
6. Terms are temporal hybrid formulas of a special kind (see Def. 4.6). The notion of modal definability of terms introduced in Def. 4.11 can be naturally extended to arbitrary (temporal) hybrid formulas (without propositional variables), once we regard nominals occurring in a formula as first-order free variables. Then the question is: which (temporal) hybrid formulas are modally definable in this sense?
7. Does the converse of the Reduction Theorem 7.2 hold? More precisely, does $q(x) \approx \varphi$ always imply $q(x) \stackrel{\omega}{\leftrightsquigarrow} \varphi$? See Remark 7.3 in Sect. 7.1 for details.
8. Can we improve the efficiency of answering conjunctive queries (at least for modally definable queries for which we formulated Theorem 7.8) so that to reduce the search through all tuples of constants? See Sect. 7.5 for details.

Acknowledgements

The research is partially supported by RFBR grants 11-01-00281, 11-01-00958, RFBR-CNRS grant 11-01-93107, and by EPSRC grant EP/H05099X/1. The authors would like to thank the anonymous reviewers for their careful reading of the paper and useful suggestions.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, 2nd edition, 2007.

[2] C. Beeri, M.Y. Vardi, The implication problem for data dependencies, in: S. Even, O. Kariv (Eds.), Automata, Languages and Programming, 8th Colloquium (ICALP 1981), volume 115 of *Lecture Notes in Computer Science*, Springer, 1981, pp. 73–85.

[3] J. van Benthem, Modal Logic and Classical Logic, volume 3 of *Monographs in Philosophical Logic and Formal Linguistics*, Bibliopolis, 1983.

[4] P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, volume 3 of *Studies in Logic and Practical Reasoning*, Elsevier, 2006.

[5] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, volume 53 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 2002.

[6] E. Börger, E. Grädel, Y. Gurevich, The Classical Decision Problem, Perspectives in Mathematical Logic, Springer, 1997.

[7] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, Journal of Automated Reasoning 39 (2007) 385–429.

[8] B. ten Cate, Model theory for extended modal languages, Ph.D. thesis, University of Amsterdam, 2005. ILLC Dissertation Series DS-2005-01.

[9] A. Chagrov, L. Chagrova, The truth about algorithmic problems in correspondence theory, in: G. Governatori, I. Hodkinson, Y. Venema (Eds.), Advances in Modal Logic 6, College Publications, 2006, pp. 121–138.

[10] A. Chagrov, M. Zakharyaschev, Modal Logic, volume 35 of *Oxford Logic Guides*, Oxford University Press, 1997.

[11] A. Chandra, P. Merlin, Optimal implementation of conjunctive queries in relational data bases, in: J.E. Hopcroft, E.P. Friedman, M.A. Harrison (Eds.), Proc. of the 9th Annual ACM Symposium on Theory of Computing (STOC'77), ACM, 1977, pp. 77–90.

[12] B. Dunin-Keplicz, R. Verbrugge, Teamwork in multi-agent systems: A formal approach, Wiley, 2010.

[13] T. Eiter, G. Gottlob, M. Ortiz, M. Šimkus, Query answering in the description logic Horn-$\mathcal{SHIQ}$, in: S. Hölldobler, C. Lutz, H. Wansing (Eds.), Proc. of the 11th Eur. Conf. on Logics in Artificial Intelligence (JELIA 2008), volume 5293 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 166–179.

[14] D. Gabbay, A. Kurucz, F. Wolter, M. Zakharyaschev, Many-dimensional modal logics: Theory and applications, volume 148 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 2003.

[15] B. Glimm, Y. Kazakov, C. Lutz, Status $\mathcal{QIO}$: An update, in: R. Rosati, S. Rudolph, M. Zakharyaschev (Eds.), Proc. of the 24th Int. Workshop on Description Logics (DL 2011), volume 745 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2011.

[16] B. Glimm, C. Lutz, I. Horrocks, U. Sattler, Conjunctive query answering for the description logic $\mathcal{SHIQ}$, Journal of Artificial Intelligence Research 31 (2008) 157–204.

[17] B. Glimm, S. Rudolph, Status $\mathcal{QIO}$: Conjunctive query entailment is decidable, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010), AAAI Press, 2010.

[18] V. Goranko, D. Vakarelov, Elementary canonical formulae: extending Sahlqvist's theorem, Annals of Pure and Applied Logic 141 (2006) 180–217.

[19] G. Hughes, Every world can see a reflexive world, Studia Logica 49 (1990) 175–181.

[20] S. Kikot, An extension of Kracht's theorem to generalized Sahlqvist formulas, Journal of Applied Non-Classical Logics 19 (2009) 227–251.

[21] S. Kikot, Axiomatization of modal logic squares with distinguished diagonal, Mathematical Notes 88 (2010) 238–250.

[22] S. Kikot, Semantic characterization of Kracht's formulas, in: L. Beklemishev, V. Goranko, V. Shehtman (Eds.), Advances in Modal Logic 8, College Publications, 2010, pp. 218–234.

[23] R. Kontchakov, C. Lutz, D. Toman, F. Wolter, M. Zakharyaschev, The combined approach to query answering in DL-Lite, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010), AAAI Press, 2010.

[24] M. Kracht, How completeness and correspondence theory got married, in: M. de Rijke (Ed.), Diamonds and Defaults, volume 229 of *Synthese Library*, Kluwer Academic Publishers, 1993, pp. 175–214.

[25] M. Kracht, Tools and Techniques in Modal Logic, volume 142 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 1999.

[26] M. Krötzsch, F. Simancik, I. Horrocks, A description logic primer, Computing Research Repository (CoRR) arXiv:1201.4089 (2012).

[27] A. Kurucz, On axiomatising products of Kripke frames, Journal of Symbolic Logic 65 (2000) 923–945.

[28] A. Kurucz, On axiomatising products of Kripke frames, part II, in: C. Areces, R. Goldblatt (Eds.), Advances in Modal Logic 7, College Publications, 2008, pp. 219–230.

[29] C. Lutz, The complexity of conjunctive query answering in expressive description logics, in: A. Armando, P. Baumgartner, G. Dowek (Eds.), Proc. of the 4th Int. Joint Conf. on Automated Reasoning (IJCAR 2008), volume 5195 of *Lecture Notes in Artificial Intelligence*, Springer, 2008, pp. 179–193.

[30] C. Lutz, F. Wolter, D. Toman, Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system, in: C. Boutilier (Ed.), Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009), AAAI Press, 2009, pp. 2070–2075.

[31] M.M. Ortiz, D. Calvanese, T. Eiter, Characterizing data complexity for conjunctive query answering in expressive description logics, in: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006), AAAI Press, 2006, pp. 275–280.

[32] H. Pérez-Urbina, B. Motik, I. Horrocks, A comparison of query rewriting techniques for DL-Lite, in: B.C. Grau, I. Horrocks, B. Motik, U. Sattler (Eds.), Proc. of the 22nd Int. Workshop on Description Logics (DL 2009), volume 477 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2009.

[33] A. Schaerf, Reasoning with individuals in concept languages, Data and Knowledge Engineering 13 (1994) 141–176.

[34] K. Schild, A correspondence theory for terminological logics: Preliminary report, in: J. Mylopoulos, R. Reiter (Eds.), Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI 1991), Morgan Kaufmann, 1991, pp. 466–471.

[35] S. Tobies, Complexity results and practical algorithms for logics in Knowledge Representation, Ph.D. thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

[36] E. Zolin, Query answering based on modal correspondence theory, in: Proc. of the 4th "Methods for modalities" Workshop (M4M-4), m4m.loria.fr, 2005, pp. 21–37.

[37] E. Zolin, Modal logic applied to query answering and the case for variable modalities, in: D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A.Y. Turhan, S. Tessaris (Eds.), Proc. of the 20th Int. Workshop on Description Logics (DL 2007), volume 250 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2007.