PSPACE Bounds for Rank-1 Modal Logics

Lutz Schröder
Department of Computer Science
Universität Bremen

Dirk Pattinson
Department of Computer Science
University of Leicester

Abstract

For lack of general algorithmic methods that apply to wide classes of logics, establishing a complexity bound for a given modal logic is often a laborious task. The present work is a step towards a general theory of the complexity of modal logics. Our main result is that all rank-1 logics enjoy a shallow model property and thus are, under mild assumptions on the format of their axiomatization, in $PSPACE$. This leads not only to a unified derivation of (known) tight PSPACE-bounds for a number of logics including K, coalition logic, and graded modal logic (and to a new algorithm in the latter case), but also to a previously unknown tight PSPACE-bound for probabilistic modal logic, with rational probabilities coded in binary. This generality is made possible by a coalgebraic semantics, which conveniently abstracts from the details of a given model class and thus allows covering a broad range of logics in a uniform way.

1. Intoduction

Modal logics are attractive from a computational point view, as they often combine expressivity with decidability. For various modal logics, satisfiability is known to be in $PSPACE$. This is typically proved for one logic at a time, often by diligent modifications of the witness algorithm for the modal logic K [4], but also using markedly different methods such as in the constraint-based $PSPACE$-algorithm for graded modal logic [31]. A first glimpse of a generalisable method was given in [34], where various epistemic logics were shown to be in NP and $PSPACE$, respectively. Nevertheless, there is to date no generally applicable theorem that allows establishing $PSPACE$-bounds for large classes of modal logics in a uniform way.

Here, we generalise the methods of [34] to obtain $PSPACE$ bounds for rank-1 modal logics of essentially arbitrary nature in a systematic way, under mild assumptions on the format of rules and axioms. While requiring rank 1 is certainly a restriction, our approach nevertheless covers numerous relevant and non-trivial examples. E.g., our results recover (known) $PSPACE$ bounds for standard modal logics such as K and also for a range of non-normal modal logics such as graded modal logic [31] and coalition logic [23]. Moreover, our method goes beyond re-proving known results in a uniform fashion: we obtain a previously unknown $PSPACE$-bound for probabilistic modal logic [17, 13], with rational probabilities coded in binary. These logics are far from exotic: graded modal logic plays a role e.g. in decision support and knowledge representation [33, 19], while probabilistic modal logic has appeared both in connection with model checking [17] and in modelling economic behaviour [13].

The key to the generality is to parametrise the theory over the type of systems defining the semantics, using coalgebraic methods. Coalgebra conveniently abstracts from the details of a concrete class of models by encoding it as an endofunctor on the category of sets. As specific instances, one obtains e.g. Kripke frames, (monotone) neighbourhood frames [12], game frames [23], probabilistic transition systems and automata [24, 3], weighted automata, linear automata [6], and multigraphs [10]. Despite the broad range of systems covered by the coalgebraic approach, a substantial body of concepts and non-trivial results has emerged, encompassing e.g. generic notions of bisimilarity and coinduction [2], corecursion [32], duality, and ultrafilter extensions [15]. Coalgebraic modal logic features in actual specification languages such as the object oriented specification language CCSL [26] and CoCasl [18].

The coalgebraic treatment of computational aspects of modal logic was initiated in [30], where the finite model property and associated $NEXPTIME$-bounds were proved. Here, we push this further by giving a coalgebraic generalisation of the shallow model property. Our PSPACE-algorithm then traverses a shallow model, stripping off one layer of modalities in every step. This requires converting the axiomatisation of a given logic into a set of logical rules that obeys a specific closure condition, and a general construction to perform this conversion is provided. The algorithm runs in PSPACE, provided the induced set of rules has a polynomial bound on matchings, which is the case for all examples we are aware of.

2. Coalgebraic Modal Logic

We briefly recapitulate the basics of the coalgebraic interpretation of modal logic.

Definition 1. [27] Let $T : \mathbf{Set} \to \mathbf{Set}$ be a functor, referred to as the *signature functor*, where $\mathbf{Set}$ is the category of sets. A *T-coalgebra* $A = (X, \xi)$ is a pair (X, ξ) where X is a set (of *states*) and $\xi : X \to TX$ is a function called the *transition* function .

We view coalgebras as generalised transition systems: the transition function delivers a structured set of successors and observations for a state. Mutatis mutandis, we can in fact allow T to take proper classes as values, as we never iterate T or otherwise assume that T is a set; details are left implicit. This allows us to treat more examples, in particular Pauly's coalition logic (Example 5.7 below).

Assumption 2. We can assume w.l.o.g. that T preserves injective maps [1]. For convenience of notation, we will in fact sometimes assume that $TX \subseteq TY$ in case $X \subseteq Y$. Moreover, we assume w.l.o.g. that T is non-trivial, i.e. $TX = \emptyset \implies X = \emptyset$.

Modal logic in the form considered here has been introduced as a specification logic for coalgebraically modelled reactive systems in [22], generalising previous results [14, 25, 16, 20]. The coalgebraic semantics is based on predicate liftings, which abstract from the concrete interpretation of modal operators in the same way that the signature functor abstracts from a concrete class of models.

Definition 3. A *predicate lifting* for a functor T is a natural transformation $\lambda : 2 \to 2 \circ T$,where 2 denotes the contravariant powerset functor $\mathbf{Set}^{op} \to \mathbf{Set}$.

A coalgebraic semantics for a modal logic consists of a signature functor and an assignment of a predicate lifting to every modal operator; we write $[\lambda]$ for a modal operator that is interpreted using the lifting λ. Thus, a set Λ of predicate liftings for T determines the syntax of a modal logic $\mathcal{L}(\Lambda)$. Formulae $\phi, \psi \in \mathcal{L}(\Lambda)$ are defined by the grammar

$$\phi ::= \bot \mid \phi \wedge \psi \mid \neg \phi \mid [\lambda]\, \phi,$$

where λ ranges over Λ. Disjunctions $\phi \vee \psi$, truth $\top$, and other boolean operations are defined as usual.

The satisfaction relation $\models_C$ between states x of a T-coalgebra $C = (X, \xi)$ and $\mathcal{L}(\Lambda)$-formulae is defined inductively, with the usual clauses for the boolean operations. The clause for the modal operator $[\lambda]$ is

$$x \models_C [\lambda]\phi \iff \xi(x) \in \lambda[\![\phi]\!]_C,$$

where $[\![\phi]\!]_C = \{x \in X \mid x \models_C \phi\}$. We drop the subscripts C when C is clear from the context.

From a coalgebraic perspective, the logics $\mathcal{L}(\Lambda)$ exhibit a number of pleasant properties. Behaviourally equivalent states have the same theory, and we can – in case T is accessible – always find enough (polyadic) modal operators to distinguish non-equivalent states [29]. In the interest of readability, we restrict our attention to unary modalities for the purpose of this paper. However, we remark that our treatment extends to the polyadic case in a straightforward manner. Our main interest is in the (local) *satisfiability problem* for $\mathcal{L}(\Lambda)$:

Definition 4. An $\mathcal{L}(\Lambda)$-formula ϕ is *satisfiable* if there exist a T-coalgebra C and a state x in C such that $x \models_C \phi$.

For a more detailed discussion of global and local consequence and weak and strong completeness in a coalgebraic context see [30]. Many modal logics (including probabilistic modal logic and graded modal logic) fail to be compact and hence do not admit finitary *strongly* complete proof systems. The following examples show that the coalgebraic approach subsumes a large class of modal logics.

Example 5. [22, 9, 30]

1. Let $\mathcal{P}$ be the covariant powerset functor. Then $\mathcal{P}$-coalgebras are graphs, thought of as transition systems or indeed Kripke frames. The predicate lifting λ defined by

$$\lambda_X(A) = \{B \subset X \mid B \subset A\}$$

gives rise to the standard box modality $\Box = [\lambda]$. This translates verbatim to the finitely branching case, captured by the (accessible) finite powerset functor $\mathcal{P}_{fin}$.

2. Coalgebras for the functor $N = 2 \circ 2$ (composition of the contravariant powerset functor with itself) are neighbourhood frames, the canonical semantic domain of non-normal logics [8]. The coalgebraic semantics induced by the predicate lifting λ defined by

$$\lambda_X(A) = \{\alpha \in N(X) \mid A \in \alpha\}$$

is just the neighbourhood semantics for $\Box = [\lambda]$.

3. Similarly, coalgebras for the subfunctor $\mathsf{Up}\mathcal{P}$ of N obtained by restricting N to upwards closed subsets of 2^X are monotone neighbourhood frames [12]. Putting $\Box = [\lambda]$, with λ as above, gives the standard interpretation of the $\Box$-modality of monotone modal logic.

4. It is straightforward to extend a given coalgebraic modal logic for T with a set U of *propositional symbols*. This is captured by passing to the functor $T'X = TX \times \mathcal{P}(U)$ and extending the set of predicate liftings by the liftings λ^a, $a \in U$, defined by

$$\lambda^a_X(A) = \{(t, B) \in TX \times \mathcal{P}(U) \mid a \in B\}.$$

Since λ^a is independent of its argument, the induced modal 'operator' can be written as just the propositional symbol a, with the expected meaning.

5. The *finite multiset* (or *bag*) functor $\mathcal{B}$ maps a set X to the set of maps $b : X \to \mathbb{N}$ with finite support. The action on morphisms $f : X \to Y$ is given by $\mathcal{B}f : \mathcal{B}X \to \mathcal{B}Y, b \mapsto \lambda y. \sum_{f(x)=y} b(x)$. Coalgebras for $\mathcal{B}$ are directed graphs with $\mathbb{N}$-weighted edges, often referred to as *multigraphs* [10], and provide a coalgebraic semantics for *graded modal logic* (GML): One defines a set of predicate liftings $\{\lambda^k \mid k \in \mathbb{N}\}$ by

$$\lambda^k_X(A) = \{b : X \to \mathbb{N} \in \mathcal{B}(X) \mid \textstyle\sum_{a \in A} b(a) > k\}.$$

The arising modal operators are precisely the modalities $\Diamond_k$ of GML [10], i.e. $x \models \Diamond_k \phi$ iff ϕ holds for more than k successor states of x, taking into account multiplicities. (GML is more standardly interpreted over Kripke frames, where $\Diamond_k \phi$ reads 'there are more than k successors satisfying ϕ'. One easily checks that both interpretations induce the same notion of satisfiability [30]). Note that $\Box_k$, defined as $\neg \Diamond_k \neg$, is monotone, but fails to be normal unless $k = 0$. (Recall that a modal operator $\Box$ is called *monotone* if it satisfies $\Box(p \wedge q) \to \Box p$, and *normal* if it satisfies $\Box p \to \Box q \to \Box(p \wedge q)$). A non-monotone variation of GML arises when negative multiplicities are admitted.

6. The *finite distribution functor* D_ω maps a set X to the set of probability distributions on X with finite support. Coalgebras for the functor $T = D_\omega \times \mathcal{P}(U)$, where U is a set of propositional symbols, are probabilistic transition systems (also called *probabilistic type spaces* [13]) with finite branching degree.
The natural predicate liftings for T consists of the propositional symbols (Item 4 above) together with the liftings λ^P defined by

$$\lambda^p(A) = \{P \in D_\omega X \mid PA \geq p\}$$

where $p \in [0, 1] \cap \mathbb{Q}$. The induced operators are the modalities $L_p = [\lambda^p]$ of *probabilistic modal logic (PML)* [17, 13], where $L_p \phi$ reads 'ϕ holds in the next step with probability at least p'. (In general [13], probabilistic type spaces can have arbitrary branching degree, but since PML has the finite model property, this has no bearing on satisfiability.) Note that PML is non-normal (e.g. $L_p a \vee L_p b \to L_p(a \vee b)$ is not valid for $p > 0$).

7. We describe a coalgebraic semantics for *coalition logic* [23] using the class-valued signature functor T defined by

$$TX = \{(S_1, \dots, S_n, f) \mid \emptyset \neq S_i \in \mathbf{Set}, f : \prod_{i \in N} S_i \to S\}$$

where $N = \{1, \dots, n\}$ is a fixed set of *agents*. Thus, the elements of TX are *strategic games* with set X of states, i.e. tuples consisting of nonempty sets S_i of *strategies* for all agents i, and an *outcome function* $(\prod S_i) \to X$. Then, a T-coalgebra is a *game frame* [23]. Subsets of N are called *coalitions*. For a coalition C, we denote the set $\prod_{i \in C} S_i$ by S_C, and for $\sigma_C \in S_C, \sigma_{\bar{C}} \in S_{\bar{C}}$, where $\bar{C} = N - C$, $(\sigma_C, \sigma_{\bar{C}})$ denotes the obvious element of $\prod_{i=1}^n S_i$. The modalities $[C]$, $C \subset N$, of coalition logic are captured as $[C] = [\lambda^C]$ by the set of predicate liftings λ^C, given by

$$\lambda^C_X(A) = \{(S_1, \dots, S_n, f) \in TX \mid \exists \sigma_C \in S_C. \forall \sigma_{\bar{C}} \in S_{\bar{C}}. f(\sigma_C, \sigma_{\bar{C}}) \in A\}.$$

Intuitively, $[C]\phi$ means that coalition C can force ϕ.

All the above examples can be canonically extended to systems that process inputs from a set I by passing from the signature functor T to the functor T^I. Similarly, output of elements of O is modelled by extending the functor T to the assignment $X \mapsto O \times TX$. We refer to [9] for a detailed account of the induced logics.

3. Proof Systems For Coalgebraic Modal Logic

Our decision procedure for rank-1 logics relies on a complete axiomatisation in a certain format. Deduction for modal logics with coalgebraic semantics has been considered in [21, 9, 15, 30]. It has turned out that every such logic can be axiomatised in rank 1, essentially because functors, as opposed to comonads, only encode the one-step behaviour of systems [30]. Here, we use a slight variation and focus on rules leading from rank 0 to rank 1, rather than axioms in rank 1. This leads to novel notions of *resolution closure* and *strict one-step completeness*, which are crucial for the shallow model construction and the ensuing $PSPACE$ algorithm.

For the remainder of the paper, we assume given a functor T and a set Λ of predicate liftings for T. We recall a few basic notions from propositional logic, as well as notation for coalgebraic modal logic introduced in [21, 9]:

Definition 6. Let V be a set. We denote the set of propositional formulae over V by $\mathsf{Prop}(V)$. Here, we regard $\neg$ and $\wedge$ as the basic connectives, with all other connectives defined in the standard way. A *literal* over V is either an element of V or the negation of such an element. A *clause* is a finite, possibly empty, disjunction of literals. The set of all clauses over V is denoted by $\mathsf{Cl}(V)$. Although we regard clauses as formulae rather than sets of literals, we shall sometimes use terminology such as 'a literal is contained in a clause' or 'a clause contains another', with the obvious meaning. We occasionally use variables ϵ etc. to denote either nothing or negation, so that a general clause over atoms $a_i \in V$ has the form $\bigvee_i \epsilon_i a_i$. We denote by $\mathsf{Up}(V)$ the set $\{[\lambda]a \mid \lambda \in \Lambda, a \in V\}$.

If the elements of V are, or have an interpretation as, subsets of a given set X, then $\phi \in \mathsf{Prop}(V)$ can be interpreted as a subset $[\![\phi]\!]_X$ of X; we say that ϕ *holds in* X and write $X \models \phi$ if $[\![\phi]\!]_X = X$, and we say that ϕ is *satisfiable in* X if $[\![\phi]\!]_X \neq \emptyset$. Similarly, if $a \in V$ is interpreted as a subset A of X, then we interpret $[\lambda]a \in \mathsf{Up}(V)$ as the subset $[\![[\lambda]a]\!] = \lambda_X(A)$ of TX. This can be iterated, leading to interpretations $[\![\phi]\!] \subset TX$ of $\phi \in \mathsf{Up}(\mathsf{Prop}(V))$ etc.

In case the elements of V are formulae in $\mathcal{L}(\Lambda)$, we also regard propositional formulae over V as formulae in $\mathcal{L}(\Lambda)$. We sometimes explicitly designate V as consisting of *propositional variables*; propositional variables retain their status across further applications of Up and Prop (e.g. if V is a set of propositional variables, then V and not $\mathsf{Prop}(V)$ is the set of propositional variables for $\mathsf{Up}(\mathsf{Prop}(V))$). Given a set L, an *L-substitution* is a substitution σ of the propositional variables by elements of L and $\phi\sigma$ is called an *instance* of ϕ over L. If $L \subset \mathcal{P}(X)$ for some X, then we also refer to σ as an *L-valuation* or a *$\mathcal{P}(X)$-valuation*. If L is the set $\{\bot, \top\}$ of truth values, then we just speak of a *valuation*.

Definition 7. A *(one-step) rule* R over a set V of propositional variables is a rule ϕ/ψ, where $\phi \in \mathsf{Prop}(V)$ and $\psi \in \mathsf{Cl}(\mathsf{Up}(V))$. We silently identify rules under α-equivalence. The rule R is *sound* if, whenever $\phi\sigma$ is valid for an $\mathcal{L}(\Lambda)$-substitution σ, then $\psi\sigma$ is valid. Moreover, R is *one-step sound* if for each set X and each $\mathcal{P}(X)$-valuation τ, $X \models \phi\tau$ implies $TX \models \psi\tau$.

Remark 8. We can always assume that every propositional variable a appearing in the premise ϕ of a one-step rule appears also in the conclusion: otherwise, we can eliminate a by passing from ϕ to $\phi[\top/a] \vee \phi[\bot/a]$.

Proposition 9. *Every one-step sound rule is sound.*

The converse holds under additional assumptions [29]; note however that the obviously sound rule $\bot/\bot$ is one-step sound iff $T\emptyset = \emptyset$ (as is the case e.g. for PML).

A given set $\mathcal{R}$ of one-step sound rules induces a proof system for $\mathcal{L}(\Lambda)$ as follows.

Definition 10. Let $\mathcal{R}_C$ denote the set of rules obtained by extending $\mathcal{R}$ with the *congruence rule*

$$(C)\quad \frac{a \leftrightarrow b}{[\lambda]a \rightarrow [\lambda]b}.$$

(This rule of course implies a rule where $\rightarrow$ is replaced by $\leftrightarrow$, which however does not fit the format for one-step rules.) The set of *derivable* formulae is the smallest set closed under propositional entailment and the rules in $\mathcal{R}_C$, with propositional variables instantiated to formulae in $\mathcal{L}(\Lambda)$.

It is easy to see that this proof system is sound. Completeness requires 'enough' rules in the following sense.

Definition 11. The set $\mathcal{R}$ is *(strictly) one-step complete* if, for each set X and each finite $\mathfrak{A} \subset \mathcal{P}(X)$, whenever $TX \models \chi$ for $\chi \in \mathsf{Cl}(\mathsf{Up}(\mathfrak{A}))$, then χ is *(strictly) derivable*; i.e. χ is propositionally entailed by clauses $\psi\tau$, where ϕ/ψ is in $\mathcal{R}$ and τ is a $\mathsf{Prop}(\mathfrak{A})$-valuation (an $\mathfrak{A}$-valuation) such that $X \models \phi\tau$.

The distinctive feature of *strict* one-step completeness is that strict derivation precludes intermediate reasoning over $\mathsf{Up}(\mathsf{Prop}(\mathfrak{A}))$. This plays a central role in the shallow model construction to be presented in Section 4.

Remark 12. It is easy to see that in the definition of one-step completeness, it does not matter whether elements of $\mathsf{Prop}(\mathfrak{A})$ are regarded as formulae or as subsets of X.

Lemma and Definition 13. *If $\mathcal{R}$ is strictly one-step complete, then for each set X, each $\phi \in \mathsf{Prop}(\mathsf{Up}(V))$, and each $\mathcal{P}(X)$-valuation τ such that $TX \models \phi\tau$, ϕ is* strictly congruence derivable*, i.e. propositionally entailed by clauses $\psi\sigma$, where ϕ/ψ is in $\mathcal{R}_C$ (Definition 10) and σ is a V-substitution such that $X \models \phi\sigma\tau$.*

Remark 14. It is implicitly shown in [30] that the set of all one-step sound rules is always strictly one-step complete and that the proof system induced by a one-step complete set of rules is *weakly complete*, i.e. proves all valid formulae.

Proposition 15. *A set $\mathcal{R}$ of one-step rules is (strictly) one-step complete iff for all* finite *sets X and all subsets $\mathfrak{A} \subset \mathcal{P}X$ that generate $\mathcal{P}(X)$ as a boolean algebra, χ is (strictly) derivable under $\mathcal{R}$ whenever $TX \models \chi$ for $\chi \in \mathsf{Cl}(\mathsf{Up}(\mathfrak{A}))$.*

Strictly one-step complete sets of rules are generally more complicated than one-step complete sets of rules or axioms [21, 30]. In our terminology, part of the effort of [34] and [23] is devoted to finding strictly one-step complete sets of rules. We now develop a systematic procedure for turning one-step complete rule sets into strictly one-step complete ones.

Definition 16. A set $\mathcal{R}$ of one-step rules is *resolution closed* if it satisfies the following requirement. Let $R_1, R_2 \in \mathcal{R}$, where $R_1 = \phi_1/\psi_1$ and $R_2 = \phi_2/\psi_2$. We can assume that R_1 and R_2 have disjoint sets of propositional variables. Let $[\lambda]a$ be in ψ_1, and let $\neg[\lambda]b$ be in ψ_2 for some $\lambda \in \Lambda$, so that we have a resolvent ψ of ψ_1 and $\psi_2[a/b]$, obtained from $\psi_1 \vee \psi_2[a/b]$ by removing $\neg[\lambda]a$ and $[\lambda]a$. Then $\mathcal{R}$ is required to contain a rule $R = \phi/\psi$ such that ϕ is propositionally entailed by $\phi_1 \wedge \phi_2[a/b]$; in this case, R is called a *resolvent* of R_1 and R_2.

Remark 17. One can construct resolution closed sets by iterated addition of missing resolvents. Here, an obvious choice for a resolvent ϕ/ψ as above is to take ϕ as $\phi_1 \wedge \phi_2[a/b]$, with a eliminated according to Remark 8 if a is not contained in ψ; it is clear that $\phi_1 \wedge \phi_2[a/b]/\psi$ is one-step sound if R_1 and R_2 are one-step sound.

Remark 18. One should not confuse the terminology introduced above with existing resolution-based approaches to decision procedures for modal logic (e.g. [11]), which rely on translating modal formulae into first-order logic by making their semantics explicit.

Lemma 19. *Let V be a set of propositional variables, let $\psi \in \mathsf{Cl}(V)$, and let $\Phi \subset \mathsf{Cl}(V)$ be closed under resolution. Then Φ propositionally entails ψ iff ψ contains one of the clauses in Φ.*

Theorem 20. *If $\mathcal{R}$ is one-step complete and resolution closed, then $\mathcal{R}$ is strictly one-step complete.*

Proof. Let $\mathfrak{A} \subset \mathcal{P}X$, and let $\gamma \in \mathsf{Cl}(\mathsf{Up}(\mathfrak{A}))$ such that $TX \models \gamma$. By one-step completeness, γ is propositionally entailed by the subset

$$\Psi = \{\psi\sigma \mid \phi/\psi \in \mathcal{R}, \sigma \text{ a } \mathsf{Prop}(\mathfrak{A})\text{-valuation}, X \models \phi\sigma\}$$

of $\mathsf{Cl}(\mathsf{Up}(\mathsf{Prop}(\mathfrak{A})))$. This set is resolution closed: let ρ be a resolvent of $\psi_1\sigma_1, \psi_2\sigma_2 \in \Psi$; i.e. we have rules $R_i = \phi_i/\psi_i \in \mathcal{R}$, $i = 1, 2$, literals $[\lambda]a$ in ψ_1 and $\neg[\lambda]b$ in ψ_2, and $\mathsf{Prop}(\mathfrak{A})$-valuations σ_1, σ_2 such that $\sigma_1(a) = \sigma_2(b) =: \chi$, where $X \models \phi_1\sigma_1 \wedge \phi_2\sigma_2$, and ρ is obtained by removing $[\lambda]\chi$ and $\neg[\lambda]\chi$ from $\psi_1\sigma_1 \vee \psi_2\sigma_2$. We can assume that R_1 and R_2 have disjoint propositional variables. By resolution closedness, $\mathcal{R}$ contains a resolvent ϕ/ψ of R_1 and R_2 obtained by matching $\neg[\lambda]b$ with $[\lambda]a$ as in Definition 16. Let σ act like σ_1 on the variables of R_1 and like σ_2 on the variables of R_2. Then $X \models \phi_1\sigma \wedge \phi_2\sigma$ and $\phi_2\sigma \equiv \phi_2[a/b]\sigma$. Thus, $X \models \phi\sigma$ and hence $\rho \equiv \psi\sigma \in \Psi$.

By Lemma 19, it follows that γ contains, and hence is propositionally entailed by, a clause in Ψ. This clause is necessarily of the form $\psi\sigma$, where $\phi/\psi \in \mathcal{R}$, $X \models \phi\sigma$, and σ is an $\mathfrak{A}$-valuation. This proves that $\mathcal{R}$ is strictly one-step complete. □

In summary, strictly one-step complete rule sets can be constructed by resolving the rules of a one-step complete axiomatisation against each other. Below, we give examples of strictly one-step complete systems obtained in this way. In order to simplify the presentation for the case of graded modal logic and probabilistic modal logic, we use the following notation. If ϕ_i is a formula, $r_i \in \mathbb{Z}$ for all $i \in I$ and $k \in \mathbb{Z}$, we abbreviate

$$\sum_{i \in I} r_i\phi_i \geq k \equiv \bigwedge_{r(J) < k} \Big(\bigwedge_{j \in J} \phi_j \to \bigvee_{j \notin J} \phi_j\Big),$$

where $r(J) = \sum_{j \in J} r_j$. Moreover, if $r \in \mathbb{Z} - \{0\}$ and ϕ is a formula, then we put

$$sgn(r)\phi = \begin{cases} \phi & r > 0 \\ \neg\phi & r < 0. \end{cases}$$

The formula $\sum_{i \in I} r_i a_i \geq k$ translates into the arithmetic of characteristic functions as suggested by the notation:

Lemma 21. *An element $x \in X$ belongs to the interpretation of $\sum_{i \in I} r_i a_i \geq k$ under a $\mathcal{P}(X)$-valuation σ iff*

$$\sum_{i \in I} r_i \mathbb{1}_{\sigma(a_i)}(x) \geq k,$$

where $\mathbb{1}_A : X \to \{0, 1\}$ is the characteristic function of $A \subseteq X$.

In all examples, the resolution process, applied to known one-step complete rule sets, can be kept under control; by Theorem 20, the resulting rule sets are strictly one-step complete.

Example 22. 1. The empty set of rules is one-step complete for neighbourhood frame semantics (Example 5.2). This set is trivially resolution closed.

2. (Monotone modal logic) The one-step rule

$$(M) \quad \frac{a \to b}{\Box a \to \Box b}$$

is one-step complete for monotone neighbourhood frame semantics (Example 5.2), and clearly resolution closed.

3. (Standard modal logic K) The one-step rules

$$\frac{a}{\Box a} \qquad \frac{a \wedge b \to c}{\Box a \wedge \Box b \to \Box c}$$

are one-step complete for Kripke semantics (Example 5.1), i.e. for the modal logic K [21]. The resolution closure $\mathcal{R}$ of this rule consists of the rules

$$\frac{\bigwedge_{i=1}^n a_i \to b}{\bigwedge_{i=1}^n \Box a_i \to \Box b}$$

for all $n \in \mathbb{N}$ (here, strict one-step completeness is also easily seen directly).

4. (Coalition logic) In Lemma 6.1 of [23], the following set of one-step rules for coalition logic (Example 5.7), numbered as in loc. cit., is implicit:

$$(1) \ \frac{\bigvee_{i=1}^n \neg a_i}{\bigvee_{i=1}^n \neg [C_i] a_i} \quad (2) \ \frac{a}{[C]a} \quad (3) \ \frac{a \vee b}{[0]a \vee [N]b}$$

$$(4) \ \frac{\bigwedge_{i=1}^n a_i \to b}{\bigwedge_{i=1}^n [C_i] a_i \to [\bigcup C_i] b}$$

where $n \geq 0$, and rules (1) and (4) are subject to the side condition that the C_i are pairwise disjoint. This set of rules extends the axiomatisation of coalition logic, which one easily proves to be one-step complete given the results of [23]. The rules are moreover 'nearly' resolution closed (full resolution closure is not needed in [23] due to a slightly different notion of closed rule sets). Resolving rule (4) with rules (2) and (3), one obtains the rule schema

$$(4') \quad \frac{\bigwedge_{i=1}^{n} a_i \to b \vee \bigvee_{j=1}^{m} c_j}{\bigwedge_{i=1}^{n} [C_i] a_i \to [D] b \vee \bigvee_{j=1}^{m} [N] c_j}$$

where $m, n \geq 0$, subject to the side condition that the C_i are pairwise disjoint subsets of D, which subsumes rules (2)–(4) above. The set consisting of the rules (1) and (4') is easily seen to be resolution closed.

5. (Graded modal logic) Using Proposition 15, one shows directly that the one-step rules

$$(W) \ \Diamond_{k+1} a \to \Diamond_k a \qquad (A_1) \ \frac{b \to \bigvee_{i=1}^{m} a_i}{\Diamond_{\sum_{i=1}^{m} k_i} b \to \bigvee_{i=1}^{m} \Diamond_{k_i} a_i}$$

$$(A_2) \ \frac{\bigwedge_{\substack{1 \leq i,j \leq n \\ i \neq j}} (\neg b_i \vee \neg b_j) \quad \bigwedge_{j=1}^{n} (b_j \to a)}{\bigwedge_{j=1}^{n} \Diamond_{k_j} b_j \to \Diamond_k a} \ (\textstyle\sum_{j=1}^{n} (k_j + 1) = k + 1),$$

where $m \geq 0$, $n \geq 1$, are one-step complete for GML (Example 5.5). All these rules are subsumed by the rule schema

$$(G) \ \frac{\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \geq 0}{\bigwedge_{j=1}^{n} \Diamond_{l_j} b_j \to \bigvee_{i=1}^{m} \Diamond_{k_i} a_i},$$

where $n, m \geq 0$ and $n + m \geq 1$, subject to the side condition $\sum_{j=1}^{n} (l_i + 1) \geq 1 + \sum_{i=1}^{m} k_i$. Soundness of this rule is seen analogously as for similar rules in probabilistic modal logic [13]. It is easy to see that the rule schema is closed under resolution.

6. (Probabilistic modal logic) By reformulating the one-step complete set of axioms for probabilistic modal logic given in [9] as one-step rules and subsequently applying resolution, one obtains the rules

$$(P_k) \ \frac{\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \geq k}{\bigwedge_{j=1}^{n} L_{q_j} b_j \to \bigvee_{i=1}^{m} L_{p_i} a_i},$$

where $m, n \geq 0$, $m + n \geq 1$, and $k \in \mathbb{Z}$, subject to the side condition

$$\textstyle\sum_{i=1}^{m} p_i - \sum_{j=1}^{n} q_j \leq k, \text{ and}$$
$$\text{if } m = 0 \text{ then } -\textstyle\sum_{j=1}^{n} q_j < k.$$

This rule schema subsumes the axiomatisation in loc. cit. and hence is one-step complete. Using Lemma 21, one can show directly that (P_k) is one-step sound in the same way as for the axiomatisation in [9]. Moreover, it is easy to see that the rule schema is resolution closed: resolving an instance of P_k against an instance of P_l gives an instance of P_{k+l}.

4. The Shallow Model Construction

We now present the announced generic shallow model construction. The construction is performed along with the proof of a recursive characterisation of satisfiable formulae which generalises results from [34].

Definition 23. A set Σ of formulae is called *closed* if it is closed under subformulae and under *normalised negation* $\sim$, where $\sim\phi$ is defined to be ψ in case ϕ is of the form $\neg\psi$, and $\neg\phi$ otherwise. The smallest closed set containing a given formula ϕ is denoted $\Sigma(\phi)$. A subset H of Σ is called a *Σ-Hintikka set* if $\bot \notin H$ and, for $\phi \wedge \psi \in \Sigma$, $\phi \wedge \psi \in H$ iff $\phi, \psi \in H$, and, for $\neg\phi \in \Sigma$, $\neg\phi \in H$ iff $\phi \notin H$.

For a formula $\chi \in \mathsf{Prop}(V)$ and a Σ-substitution σ, we define satisfaction of $\chi\sigma$ in H ($H \models \chi\sigma$) inductively by

$$\begin{aligned} H &\models (\chi_1 \wedge \chi_2)\sigma &&:\iff H \models \chi_1\sigma \text{ and } H \models \chi_2\sigma \\ H &\models (\neg\chi)\sigma &&:\iff H \not\models \chi\sigma \\ H &\models a\sigma &&:\iff \sigma(a) \in H \\ H &\not\models \bot. \end{aligned}$$

This is well-defined because H is Hintikka.

Lemma 24. *Let Σ be closed, let H be a Σ-Hintikka set, and let $\phi, \psi \in \mathsf{Prop}(V)$. Then $H \models \phi \vee \psi$ iff $H \models \phi$ or $H \models \psi$.*

Lemma 25 (Soundness of propositional reasoning for Hintikka sets)**.** *Let Σ be closed, and let H be a Σ-Hintikka set. Let $\phi, \psi \in \mathsf{Prop}(V)$, and let σ be a Σ-substitution. If $H \models \phi\sigma$ and ϕ propositionally entails ψ, then $H \models \psi\sigma$.*

The following result generalises Propositions 3.2, 3.5, 3.8, 3.13, and 3.16 (but not 3.10 and 3.18) of [34] and Lemma 6.1 of [23].

Theorem 26. *Let $\mathcal{R}$ be strictly one-step complete. Then $\phi \in \mathcal{L}(\Lambda)$ is satisfiable iff $\phi \in H$ for some Hintikka set $H \subset \Sigma(\phi)$ such that, for every clause $\rho = \bigvee_{i=1}^{n} \epsilon_i[\lambda_i]\rho_i$ over $\Sigma(\phi)$ with $H \not\models \rho$ and for each rule $\psi / \bigvee_{i=1}^{n} (\epsilon_i[\lambda_i] a_i)$ in $\mathcal{R}_C$, the formula $\neg\psi[\rho_i/a_i]_{i=1,\dots,n}$ is satisfiable.*

Proof. *'Only if':* Take H to be the intersection of $\Sigma(\phi)$ with the theory of a state satisfying ϕ.

'If': For each formula $\chi \equiv \neg\psi[\rho_i/a_i]_{i=1,\dots,n}$ as in the statement, there exists a coalgebra $C_\chi = (X_\chi, \xi_\chi)$ and a state x_χ in C_χ such that $x_\chi \models_{C_\chi} \chi$; we can assume that the X_χ are pairwise disjoint. Define the sets X and $\hat{\rho}$ by

$$X = \{x_0\} \cup \bigcup_{\chi} X_\chi \quad \text{and} \quad \hat{\rho} = A_\rho \cup \bigcup_{\chi} [\![\rho]\!]_{C_\chi},$$

where x_0 is a fresh element, $\rho \in \Sigma(\phi)$, and $A_\rho = \{x_0\}$ if $\rho \in H$, $A_\rho = \emptyset$ otherwise. We define a coalgebra structure

ξ on X as follows. For $x \in X_\chi$, we put $\xi(x) = \xi_\chi(x) \in TX_\chi \subset TX$ (cf. Assumption 2). Then for each $[\lambda]\rho \in \Sigma(\phi)$,

$$\xi(x) \in \lambda\hat{\rho} \iff x \models_{C_\chi} [\lambda]\rho, \qquad (1)$$

because by naturality $(\lambda\hat{\rho}) \cap TX_\chi = \lambda(\hat{\rho} \cap X_\chi) = \lambda[\![\rho]\!]_{C_\chi}$. Moreover, we will show that there exists $\xi(x_0) \in TY \subset TX$, where Y is the set of all x_χ, such that for $[\lambda]\rho \in \Sigma(\phi)$,

$$\xi(x_0) \in \lambda\hat{\rho} \iff [\lambda]\rho \in H. \qquad (2)$$

By structural induction, Equations (1) and (2) then imply

$$x \models_C \rho \iff x \models_{C_\chi} \rho \text{ for each } x \in X_\chi, \text{ and}$$
$$x_0 \models_C \rho \iff \rho \in H$$

for all $\rho \in \Sigma(\phi)$. In particular, $x_0 \models \phi$, and we are done.

It remains to prove that $\xi(x_0)$ satisfying (2) exists. Assume the contrary. Let V be the set of propositional variables b_ρ, where $[\lambda]\rho \in \Sigma(\phi)$ for some λ. Let $\theta \in \mathsf{Cl}(\mathsf{Up}(V))$ consist of the literals $\neg[\lambda]b_\rho$ for $[\lambda]\rho \in H$ and $[\lambda]b_\rho$ for $\neg[\lambda]\rho \in H$. By assumption, $TY \models \theta\tau^Y$, where τ^Y is the $\mathcal{P}(Y)$-valuation taking b_ρ to $\hat{\rho} \cap Y = \{x_\chi \mid x_\chi \models_{C_\chi} \rho\}$. By Lemma 13, it follows that θ is strictly congruence derivable from those $\zeta \in \mathsf{Prop}(V)$ such that $Y \models \zeta\tau^Y$.

From the derivation of θ, it now follows that $H \models \theta\sigma$, where σ is the $\Sigma(\phi)$-substitution taking b_ρ to ρ (note that $\theta\sigma$ is a propositional formula over atoms $[\lambda]\rho \in \Sigma(\phi)$), by Lemma 24 a contradiction to the construction of θ: by Lemma 25, the propositional steps are sound over H; it remains to be shown that if the derivation of θ uses a rule $R \equiv \psi / \bigvee_{i=1}^n (\epsilon_i[\lambda_i]a_i)$ in $\mathcal{R}_C$, instantiated for a V-substitution η, then the conclusion of $R\eta\sigma$ is satisfied over H. Assume the contrary. By Lemma 24, it follows that $\epsilon_i[\lambda_i]\sigma(\eta(a_i)) \notin H$ for all i. By construction, we have $x_\chi \models_{C_\chi} \chi$ for $\chi \equiv \neg\psi\eta\sigma$. But since $R\eta$ appears in the derivation of θ, $Y \models \psi\eta\tau^Y$ and hence $x_\chi \in [\![\psi\eta\tau^\chi]\!]$, where τ^χ is the $\mathcal{P}(X_\chi)$-valuation taking b_ρ to $[\![\rho]\!]_{C_\chi}$. Since $[\![\psi\eta\tau^\chi]\!] = [\![\psi\eta\sigma]\!]_{C_\chi}$, we have arrived at a contradiction. □

As a corollary to the above proof, we obtain that coalgebraic modal logic has the shallow model property. The formulation of this property requires the following notion.

Definition 27. A *supporting Kripke frame* of a T-coalgebra (X, ξ) is a Kripke frame (X, R) such that for each $x \in X$,

$$\xi(x) \in T\{y \mid xRy\} \subset TX.$$

As clauses suffice for satisfiability checking, we obtain

Corollary 28 (Shallow model property)**.** *Every satisfiable $\mathcal{L}(\Lambda)$-formula ϕ is satisfiable in a* shallow model*, i.e. in a T-coalgebra (X, ξ) that has a supporting Kripke frame which consists of a tree of depth at most the rank of ϕ and of branching degree at most 2^n, where n is the number of subformulae of ϕ, and possibly an additional* final *state $x_\top$, i.e. for all x, $xRx_\top$, and $x_\top Rx$ implies $x = x_\top$.*

(The state $x_\top$ may arise from the rule $\bot/\bot$, cf. Sect. 3.)

5. A Generic PSPACE Algorithm

The shallow model result (Theorem 26) will be exploited to design a decision procedure in the spirit of [34]. Since resolution closed rule sets are in general infinite, this requires ensuring that we never need to instantiate literals in the conclusions of rules with identical formulae: otherwise, an infinite number of rules could match a single given clause over a Hintikka set. This is formally captured as follows.

Definition 29. We call a clause over L *reduced* if all its literals are distinct. An L-instance $\phi\sigma/\psi\sigma$ of a rule $\phi/\psi \in \mathcal{R}$ is *reduced* if the clause $\psi\sigma$ is reduced. Finally, $\mathcal{R}$ is *closed under reduction* if for every V-instance $\phi\sigma/\psi\sigma$ of a rule ϕ/ψ over V in $\mathcal{R}$, there exists a reduced V-instance $\phi'\sigma'/\psi'\sigma'$ of a rule $\phi'/\psi' \in \mathcal{R}$ such that $\psi\sigma$ and $\psi'\sigma'$ are propositionally equivalent and and $\phi'\sigma'$ is propositionally entailed by $\phi\sigma$.

I.e. a rule set is reduced if every instance of a rule that duplicates literals in the conclusion can be replaced by a reduced instance of a different rule. Not all the rule sets discussed in Example 22 satisfy this property, but they can easily be extended to reduction closed sets: just add a rule ϕ'/ψ' for every rule ϕ/ψ over V in $\mathcal{R}$ and every V-substitution σ, where ϕ' is some suitably chosen propositional equivalent of $\phi\sigma$ and ψ' is obtained from $\psi\sigma$ by removing duplicate literals. It is clear that the new rules remain one-step sound. Note that there is no need to preserve closure under resolution when passing to a reduced rule set, as Theorem 26 requires only strict one-step completeness, which is preserved under extending the rule set.

Example 30. 1. The strictly one-step complete rule sets of Examples 22.1–4 (including monotone modal logic, K, and coalition logic) are easily seen to be closed under reduction, essentially because in all relevant rule schemas, the premise is a clause of the same general format as the conclusion.

2. (Graded modal logic) The rule schema (G) of Example 22.5 fails to be closed under reduction, as duplicating literals in the conclusion substantially affects both the premise and the side condition. We can close (G) under reduction as described above; this results in the rule schema

$$(G') \ \frac{\sum_{i=1}^n r_i a_i \geq 0}{\bigvee_{i=1}^n sgn(r_i) \Diamond_{k_i} a_i},$$

where $n \geq 1$ and $r_1, \dots, r_n \in \mathbb{Z} - \{0\}$, subject to the side condition $\sum_{r_i<0} r_i(k_i + 1) \geq 1 + \sum_{r_i>0} r_i k_i$.

3. (Probabilistic modal logic) Similarly, the rule schema of Example 22.6 fails to be closed under reduction. Closure under reduction as described above leads to the rule schema

$$(P'_k) \quad \frac{\sum_{i=1}^{n} r_i a_i \geq k}{\bigvee_{1 \leq i \leq n} sgn(r_i) L_{p_i} a_i}$$

where $n \geq 1$, $r_1, \dots, r_n \in \mathbb{Z} - \{0\}$, subject to the side condition

$$\sum_{i=1}^{n} r_i p_i \leq k, \text{ and}$$
$$\text{if } \forall i.\, r_i < 0 \text{ then } \sum_{i=1}^{n} r_i p_i < k.$$

As instances of the congruence rule never contain duplicate literals, we have the following trivial fact.

Lemma 31. *If $\mathcal{R}$ is closed under reduction, then so is $\mathcal{R}_C$.*

Thus the following is immediate from Theorem 26.

Corollary 32. *Let $\mathcal{R}$ be strictly one-step complete and closed under reduction. Then $\phi \in \mathcal{L}(\Lambda)$ is satisfiable iff $\phi \in H$ for some Hintikka set $H \subset \Sigma(\phi)$ such that, for every* reduced *clause $\rho = \bigvee_{i=1}^{n} \epsilon_i[\lambda_i]\rho_i$ over $\Sigma(\phi)$ with $H \not\models \rho$ and for each rule $\psi / \bigvee_{i=1}^{n}(\epsilon_i[\lambda_i]a_i)$ in $\mathcal{R}_C$, the formula $\neg\psi[\rho_i/a_i]_{i=1,\dots,n}$ is satisfiable.*

In the implementation of the algorithm suggested by Corollary 32, we need to pass around matches of rules with given clauses. Since rules, in particular their premises, are generally too large to pass around directly, we assume that every rule (i.e. every instance of a rule scheme) is given by a *code*, i.e. a string over some alphabet which identifies the rule; when rules appear as data, they are always represented by their code. Moreover, we assume that propositional variables a_i in rules are uniformly represented by indices that point to literals $\epsilon_i[\lambda_i]a_i$ of the conclusion.

Definition 33. We say that a rule $R \in \mathcal{R}$ *matches* a reduced clause $\rho \equiv \bigvee_{i=1}^{n} \epsilon_i[\lambda_i]\phi_i$ if the conclusion of R is of the form $\bigvee_{i=1}^{n} \epsilon_i[\lambda_i]a_i$. By the above variable convention, the instantiation $\psi[\phi_i/a_i]_{i=1,\dots,n}$ of a conjunct ψ of the premise of R can be computed in polynomial time from ψ and ρ; we denote the result by $\psi[\rho]$. Two matching rules are *equivalent* if their premises are propositionally equivalent; equivalence classes $[R]$ are called *$\mathcal{R}$-matchings*. The code of R is also a *code* for $[R]$.

We fix some size measures for complexity purposes:

Definition 34. The size $\mathit{size}(a)$ of an integer a is $\lceil \log_2(|a| + 1) \rceil$, where $\lceil r \rceil = \min\{z \in \mathbb{Z} \mid z \geq r\}$ as usual. The size $\mathit{size}(p)$ of a rational number $p = a/b$, with a, b relatively prime, is $1 + \mathit{size}(a) + \mathit{size}(b)$. The *size* $|\phi|$ of a formula ϕ over V is defined by counting 1 for each propositional variable, boolean operator, or modal operator, and additionally the size of each index of a modal operator. (In the examples, indices are either numbers, with sizes as above, or subsets of $\{1, \dots, n\}$, assumed to be of size n.)

In particular, indices of graded or probabilistic modal operators are coded in binary.

Example 35. For the rules of Examples 22 and 30, we just take the parameters of a rule as its code in the obvious way. E.g. the code of an instance of (P'_k) as displayed in Example 30.3 consists of n, k, the r_i, and the p_i. The size of the code is determined by the sizes of these numbers plus separating letters, say, $\sum(1 + \mathit{size}(a_i)) + \sum(1 + \mathit{size}(p_i)) + \mathit{size}(n) + \mathit{size}(k) + 1$. Note that not all such codes represent valid rules.

The following decision procedure on an alternating Turing machine generalises the algorithms in [34], given a strictly one-step complete and reduction closed set $\mathcal{R}$.

Algorithm 1. (Decide satisfiability of $\phi \in \mathcal{L}(\Lambda)$)

1. (Initialise) Construct the set $\Sigma(\phi)$.
2. (Existential) Guess a Hintikka set $H \subset \Sigma(\phi)$ with $\phi \in H$.
3. (Universal) Guess a reduced clause $\bot \neq \rho \in \Sigma(\phi)$ with $H \not\models \rho$ and an $\mathcal{R}_C$-matching $[R]$ of ρ.
4. (Existential) Guess a clause γ from the conjunctive normal form (CNF) of the premise of R and recursively check that $\neg\gamma[\rho]$ is satisfiable.

The algorithm succeeds if all possible choices at steps marked *universal* lead to successful termination, and for all steps marked *existential*, there exists a choice leading to successful termination.

Correctness of the algorithm is guaranteed by Corollary 32. Note that the algorithm terminates successfully in Step 3 if there are no rules matching clauses over H. In particular, the algorithm terminates either in Step 2 or in Step 3 if ϕ has rank 0. We emphasise that in Step 3, it suffices to guess one code for each matching.

The crucial requirement for the effectivity of Algorithm 1 is that Steps 3 and 4 can be performed in polynomial time, i.e. by suitable nondeterministic polynomial-time multivalued functions (NPMV) [5]. We recall that a function $f : \Sigma^* \to \mathcal{P}(\Delta^*)$, where Σ and Δ are alphabets, is NPMV iff

1. there exists a polynomial p such that $|y| \leq p(|x|)$ for all $y \in f(x)$, where $|\cdot|$ denotes size, and
2. the graph $\{(x, y) \mid y \in f(x)\}$ of f is in *NP*.

Thus, the following conditions guarantee that Algorithm 1 has polynomial running time:

Definition 36. A set $\mathcal{R}$ of rules is called *PSPACE-tractable* if there exists a polynomial p such that all $\mathcal{R}$-matchings of a reduced clause ρ over $\mathcal{L}(\Lambda)$ have a code of size at most $p(|\rho|)$, and it can be decided in NP

1. whether a given code is the code of some rule in $\mathcal{R}$;
2. whether a rule matches a given reduced clause; and
3. whether a clause belongs to the CNF of the premise of a given rule.

Theorem 37 (Space Complexity)**.** *Let $\mathcal{R}$ be strictly one-step complete, closed under reduction, and PSPACE-tractable. Then the satisfiability problem for $\mathcal{L}(\Lambda)$ is in $PSPACE$.*

Remark 38. A more careful analysis of Algorithm 1 reveals that it suffices to require the decision problems in Definition 36 to be in PH, the polynomial time hierarchy. In our examples, however, the complexity is in fact P rather than NP, and we expect that this situation is typical, with the actually crucial condition for $PSPACE$-tractability being the polynomial bound on $\mathcal{R}$-matchings.

The next lemma, which follows directly from size estimates in linear integer programming [28], is crucial for establishing $PSPACE$-tractability in the examples. Following usual practice, we take the *size* $|W|$ of a rational inequality $W \equiv (\sum_{i=1}^{n} u_i x_i \text{ op } u_0)$, op $\in \{<, \leq, >, \geq\}$, to be $1 + n + \sum_{i=1}^{n} size(u_i)$.

Lemma 39. *There exists a polynomial p such that for every rational linear inequality W and every solution $r_0, \dots, r_n \in \mathbb{Z}$ of W, there exists a solution $r'_0, \dots, r'_n \in \mathbb{Z}$ of W such that $size(r'_i) \leq p(|W|)$ for all i, and the formulae $\sum_{i=1}^{n} r_i a_i \geq r_0$ and $\sum_{i=1}^{n} r'_i a_i \geq r'_0$ are propositionally equivalent.*

We now illustrate how Theorem 37 allows us to establish $PSPACE$ bounds for many modal logics in a uniform way. In particular, we obtain a new (tight) $PSPACE$ bound for probabilistic modal logic.

Example 40. Conditions 1 and 2 of Definition 36 are immediate for all the rule sets of Example 22 — the decision problems in question involve no more than checking computationally harmless side conditions in the case of Condition 1 (disjointness and containment of finite sets, linear inequalities), and comparing clauses of polynomial (in fact, linear) size in the case of Condition 2. Moreover, Condition 3 is immediate in those cases where the premises of rules are just single clauses. This leaves only GML and PML; but the definition of $\sum_{i \in I} r_i a_i \geq k$ is already in CNF, and checking whether a given clause belongs to this CNF is clearly in P.

It remains to establish the polynomial bound on the matchings. For GML and PML, this is guaranteed by Lemma 39. In all other cases, every reduced clause ρ matches at most one rule, whose code has size linear in the size of ρ.

We thus have obtained $PSPACE$-tractability and hence decidability in $PSPACE$ for all logics in Example 22. The logic of neighbourhood frames and monotone modal logic are of lesser interest here, as the corresponding modal logics are in NP [34]. We briefly comment on the algorithms and bounds for the other cases.

1. For the standard modal logic K (Example 22.3), Algorithm 1 is essentially the witness algorithm [34, 4], with reduced clauses violated by H corresponding to *demands*.

2. For coalition logic (Example 22.4), we arrive, due to minor differences of the rule sets, at a slight variant of Pauly's $PSPACE$-algorithm [23].

3. For graded modal logic, we obtain a new algorithm which confirms the known $PSPACE$ bound [31]. One might claim that the new algorithm is not only nicely embedded into a unified framework, but also conceptually simpler than the constraint-based algorithm of [31] (which corrects an erroneous algorithm previously given elsewhere).

4. For probabilistic modal logic, we obtain a new algorithm which yields a previously unknown $PSPACE$-bound (to our knowledge, the best previously published bound for PML is $EXPTIME$ [30]). The bound is tight, as PML contains the $PSPACE$-complete logic KD as a fragment (embedded by mapping $\Box$ to L_1).

6. Conclusion

Generalising results of [34], we have shown that coalgebraic modal logic has the shallow model property, and we have presented a generic $PSPACE$ algorithm for satisfiability based on depth-first exploration of shallow models. We have thus

- reproduced the *witness algorithm* for K [4]
- obtained a slight variant of the known $PSPACE$ algorithm for coalition logic [23]
- obtained a novel $PSPACE$ algorithm for graded modal logic, thus recovering the known $PSPACE$ bound [31]
- obtained a novel $PSPACE$ bound for probabilistic modal logic [17, 13].

In all these cases, the bound obtained is tight.

The crucial prerequisite for the generic algorithm is an axiomatisation by so-called one-step rules (going from rank 0 to rank 1) obeying two closedness conditions: closedness under resolution and under removal of duplicate literals. In the examples, it has not only turned out that is it feasible to keep this closure process under control, but also that the axiomatisations obtained have pleasingly compact presentations — often, one ends up with a single rule schema. Nevertheless, it remains desirable to prove a $PSPACE$ bound relying on purely semantic conditions such as the ones appearing in [30]; this is the subject of further research, as is the extension of the theory beyond rank 1 by means of comonads.

Ongoing work indicates that every modal logic can be equipped with a coalgebraic semantics, provided it is axiomatisable in rank 1 and satisfies the congruence rule. This would mean in particular that the method employed here applies to every such modal logic, i.e. one obtains a hard-to-miss purely syntactic criterion for modal logics in rank 1 to be decidable in $PSPACE$.

Acknowledgements. The authors wish to thank Alexander Kurz for useful discussions, and the Department of Computer Science at the University of Bremen for funding a visit of the second author.

References

[1] M. Barr. Terminal coalgebras in well-founded set theory. *Theoret. Comput. Sci.*, 114:299–315, 1993.

[2] F. Bartels. Generalised coinduction. *Math. Struct. Comput. Sci.*, 13:321–348, 2003.

[3] F. Bartels, A. Sokolova, and E. de Vink. A hierarchy of probabilistic system types. In *Coalgebraic Methods in Computer Science*, volume 82 of *ENTCS*. Elsevier, 2003.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge, 2001.

[5] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SIAM J. Computing*, 13:461–487, 1984.

[6] J. W. Carlyle and A. Paz. Realizations by stochastic finite automata. *J. Comput. System Sci.*, 5:26–40, 1971.

[7] A. Chandra and L. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

[8] B. Chellas. *Modal Logic*. Cambridge, 1980.

[9] C. Cîrstea and D. Pattinson. Modular construction of modal logics. In *Concurrency Theory*, volume 3170 of *LNCS*, pages 258–275. Springer, 2004.

[10] G. D'Agostino and A. Visser. Finality regained: A coalgebraic study of Scott-sets and multisets. *Arch. Math. Logic*, 41:267–298, 2002.

[11] H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8:265–292, 2000.

[12] H. H. Hansen and C. Kupke. A coalgebraic perspective on monotone modal logic. In *Coalgebraic Methods in Computer Science*, volume 106 of *ENTCS*, pages 121–143. Elsevier, 2004.

[13] A. Heifetz and P. Mongin. Probabilistic logic for type spaces. *Games and Economic Behavior*, 35:31–53, 2001.

[14] B. Jacobs. Towards a duality result in the modal logic of coalgebras. In *Coalgebraic Methods in Computer Science*, volume 33 of *ENTCS*. Elsevier, 2000.

[15] C. Kupke, A. Kurz, and D. Pattinson. Ultrafilter extensions for coalgebras. In *Algebra and Coalgebra in Computer Science*, volume 3629 of *LNCS*, pages 263–277. Springer, 2005.

[16] A. Kurz. Specifying coalgebras with modal logic. *Theoret. Comput. Sci.*, 260:119–138, 2001.

[17] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inform. Comput.*, 94:1–28, 1991.

[18] T. Mossakowski, L. Schröder, M. Roggenbach, and H. Reichel. Algebraic-co-algebraic specification in CoCasl. *J. Logic Algebraic Programming*. To appear.

[19] H. J. Ohlbach and J. Koehler. Modal logics, description logics and arithmetic reasoning. *Artif. Intell.*, 109:1–31, 1999.

[20] D. Pattinson. Semantical principles in the modal logic of coalgebras. In *Symposium on Theoretical Aspects of Computer Science*, volume 2010 of *LNCS*, pages 514–526. Springer, 2001.

[21] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoret. Comput. Sci.*, 309:177–193, 2003.

[22] D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame J. Formal Logic*, 45:19–33, 2004.

[23] M. Pauly. A modal logic for coalitional power in games. *J. Logic Comput.*, 12:149–166, 2002.

[24] M. Rabin. Probabilistic automata. *Inform. Control*, 6:220–245, 1963.

[25] M. Rößiger. Coalgebras and modal logic. In *Coalgebraic Methods in Computer Science*, volume 33 of *ENTCS*. Elsevier, 2000.

[26] J. Rothe, H. Tews, and B. Jacobs. The Coalgebraic Class Specification Language CCSL. *J. Universal Comput. Sci.*, 7:175–193, 2001.

[27] J. Rutten. Universal coalgebra: A theory of systems. *Theoret. Comput. Sci.*, 249:3–80, 2000.

[28] A. Schrijver. *Theory of linear and integer programming*. Wiley Interscience, 1986.

[29] L. Schröder. Expressivity of coalgebraic modal logic: the limits and beyond. In *Foundations of Software Science And Computation Structures*, volume 3441 of *LNCS*, pages 440–454. Springer, 2005.

[30] L. Schröder. A finite model construction for coalgebraic modal logic. In *Foundations of Software Science And Computation Structures*, LNCS. Springer, 2006. To appear.

[31] S. Tobies. $PSPACE$ reasoning for graded modal logics. *J. Logic Comput.*, 11:85–106, 2001.

[32] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Logic in Computer Science*, pages 280–291. IEEE, 1997.

[33] W. van der Hoek and J.-J. Meyer. Graded modalities in epistemic logic. In *Logical foundations of computer science*, volume 620 of *LNCS*, pages 503–514. Springer, 1992.

[34] M. Vardi. On the complexity of epistemic reasoning. In *Logic in Computer Science*, pages 243–251. IEEE, 1989.

A. Appendix: Omitted Proofs

(Not for inclusion in the final version.)

Lemma 13

Proof. Let $\mathfrak{A} = \{\tau(a) \mid a \in V\}$, and choose for each $A \in \mathfrak{A}$ a variable $a_A \in V$ such that $\tau(a_A) = A$. By reflexivity, $\phi\tau$ is strictly derivable in the sense of Definition 11. This derivation can be copied to obtain a strict derivation of $\phi\sigma$, where σ substitutes b with $a_{\tau(b)}$. The further derivation of ϕ is then by propositional reasoning with V-instances of the congruence rule. □

Proposition 15

Proof. The 'only if' direction is trivial; we prove the 'if' direction. Let $\mathfrak{A} \subset \mathcal{P}(Y)$ be finite, and let the clause χ over $\mathsf{Up}(\mathfrak{A})$ hold in TY. Define the map $f : Y \to 2^{\mathfrak{A}}$, where 2 is the set of truth values, by $f(x)(A) = (x \in A)$. Put $X = f[Y]$, and, abusing $\mathfrak{A}$ as a set of propositional variables, let σ be the $\mathcal{P}(X)$-valuation for $\mathfrak{A}$ defined by $\sigma(A) = \{\tau \in X \mid \tau(A) = \top\}$. Then the $\sigma(A)$ generate $\mathcal{P}(X)$ as a boolean algebra. We have

$$A = f^{-1}[\sigma(A)] \quad \text{for all } A \in \mathfrak{A}. \qquad (*)$$

By naturality of predicate liftings, it follows that $(Tf)^{-1}[\chi\sigma] = [\![\chi]\!] = TY$; since $Tf : TY \to TX$ is surjective (all set functors preserve surjective maps), this implies $TX \models \chi\sigma$. Since X is finite, we obtain that $\chi\sigma$ is (strictly) derivable in TX. By $(*)$, $Y \models \phi$ if $X \models \phi\sigma$ for all $\phi \in \mathsf{Prop}(\mathfrak{A})$; we can thus copy the derivation of $\chi\sigma$ to obtain a (strict) derivation of χ. □

Lemma 19

Proof. The 'if' direction is clear. 'Only if': Let $\Phi \vdash \psi$. Let Φ'_0 denote the union of Φ and $\neg\psi$, the latter regarded as a set of singleton clauses. Let Φ' consist of the clauses in $\neg\psi$ and of those clauses that can be obtained from clauses in Φ by repeated resolution with the clauses in $\neg\psi$. One easily checks that Φ' is resolution closed, hence is the resolution closure of Φ'_0. It follows that Φ' contains the empty clause, which is thus obtained as described from a clause $\phi \in \Phi$; then ϕ is contained in ψ. □

Lemma 21

Proof. The element x satisfies the negation of $\sum_{i \in I} r_i a_i \geq k$ iff $r(J) < k$ for $J = \{i \in I \mid x \in \sigma(a_i)\}$ iff $\sum_{i \in I} r_i \mathbb{1}_{\sigma(a_i)}(x) < k$. □

Example 22

4. *One-step completeness* of rules (1)–(4): Let $\mathfrak{A} \subset \mathcal{P}X$; by Proposition 15, we can assume that X is finite and $\mathfrak{A}$ generates $\mathcal{P}(X)$ as a boolean algebra. Then it suffices to show that every consistent formula ϕ in $\mathsf{Prop}(\mathsf{Up}(\mathcal{P}))$ is satisfiable in X. We can extend ϕ to a maximal consistent subset S of $\mathsf{Up}(\mathcal{P}(X))$. Since S obeys the given axioms, it determines an effectivity frame in the sense of [23] and hence is, by results of loc. cit., realized by a game frame over X.

Resolution closedness of rules (1) and $(4')$: We discuss only the case of resolving $(4')$ against itself; the other case is similar. Let one instance of $(4')$ be denoted as in the rule schema, and another instance with all entities primed (a'_i etc.). The two instances can be resolved in two essentially different ways. The subcase where matching is with $[D]b$ is straightforward. Thus assume w.l.o.g. that matching is via $[N]c_1 \equiv [C'_1]a'_1$. Then by the side conditions, $D' = N$ and $C'_i = \emptyset$ for $i = 2, \dots, n'$. Thus, the resolvent has the conclusion

$$\bigwedge_{i=1}^{n}[C_i]a_i \wedge \bigwedge_{i=2}^{n'}[C'_i]a'_i \to$$
$$[D]b \vee [N]b' \vee \bigvee_{j=2}^{m}[N]c_j \vee \bigvee_{j=1}^{m'}[N]c'_j,$$

which fits the format of the rule scheme $(4')$. It is easy to check that the combined premises imply the required premise for the resolved conclusion, and similarly for the side conditions.

5. *One-step completeness* of rules (W), (A_1), (A_2): Let X be a set, and let $\mathfrak{A} \subset \mathcal{P}(X)$. By Proposition 15, we can assume that X is finite and $\mathfrak{A}$ generates $\mathcal{P}(X)$ as a boolean algebra. Then one-step completeness reduces to showing that every consistent formula ϕ over $\mathsf{Up}(\mathcal{P}(X))$ is satisfiable in X. We can extend ϕ to a maximally consistent subset S of the set $\Sigma(\phi) \subset \mathsf{Prop}(\mathsf{Up}(\mathcal{P}(X))$ consisting of those formulae whose modal operators $\Diamond_l$ have degree $l \leq k_{\max}$, where $k_{\max}$ is the maximal degree of a modal operator occurring in ϕ. We show that S is satisfiable in X. For $x \in X$, we put

$$n(x) = \max\{k + 1 \mid \Diamond_k\{x\} \in S\},$$

with the convention $\max\emptyset = 0$, and $t = \sum_{x \in X} n(x)x$. We claim that for $A \subset X$,

$$t \models \Diamond_k A \quad \text{iff} \quad \Diamond_k A \in S, \qquad (*)$$

and hence $t \models \phi$. It remains to prove $(*)$:

'Only if': By (W), we have $B = \{b_1, \dots, b_n\} \subset A$ and $\Diamond_{k_i}\{b_i\} \in S$ such that

$$\sum_{i=1}^{n}(k_i + 1) = k + 1.$$

Then $\Diamond_k A \in S$ by maximal consistency and rule (A_2).
'If': Let $A = \{a_1, \dots, a_m\}$, and let $t \not\models \Diamond_k A$. Then $\Diamond_{k_{\max}}\{a_i\} \notin S$ for all i, and hence by (W)

$$n(a_i) = \min\{l \mid \neg\Diamond_l\{a_i\} \in S\}.$$

Thus we have $k_i \leq k_{\max}$ such that $\Diamond_{k_i}\{a_i\} \notin S$ for all i, and $\sum k_i \leq k$. By (A_1), $\Diamond_{\sum_{i=1}^m k_i} A \notin S$, and by (W), $\Diamond_k \notin S$.
One-step soundness of (G)*:* Let σ be a $\mathcal{P}(X)$-substitution such that $X \models (\sum_{i=1}^m a_i - \sum_{j=1}^n b_j \geq 0)\sigma$. For $A \subset X$, put $s_A(\sum n_i x_i) = \sum_{x_i \in A} n_i$. From Lemma 21, we conclude

$$\sum_{j=1}^n s_{\sigma(b_j)}(t) \leq \sum_{i=1}^m s_{\sigma(a_i)}(t)$$

for all $t \in \mathcal{B}(X)$. It follows that t is in the interpretation of the conclusion of (G) under σ.
Resolution closure of (G)*:* Take two instances of (G), one denoted like in the general form of the rule and one with all entities primed (a_i' etc.), with the resolution taking place w.l.o.g. by matching $\Diamond_{k_1} a_1 \equiv \Diamond_{l_1'} b_1'$. The conclusion of the arising resolvent is

$$\bigwedge_{j=1}^n \Diamond_{l_j} b_j \wedge \bigwedge_{j=2}^{m'} \Diamond_{l_j'} b_j' \to \bigvee_{i=2}^m \Diamond_{k_i} a_i \bigvee_{i=1}^{m'} \Diamond_{k_i'} a_i'.$$

Since $a_1 \equiv b_1'$, the combined premises

$$\sum_{i=1}^m a_i - \sum_{j=1}^n b_j \geq 0 \wedge \sum_{i=1}^{m'} a_i' - \sum_{j=1}^n b_j' \geq 0$$

imply

$$\sum_{i=2}^m a_i + \sum_{i=1}^{m'} a_i' - \sum_{j=1}^n b_j - \sum_{j=2}^n b_j' \geq 0,$$

and since $k_1 = l_1'$, the combined side conditions

$$\sum_{j=1}^n (l_i + 1) \geq 1 + \sum_{i=1}^m k_i \wedge \sum_{j=1}^{n'} (l_i' + 1) \geq 1 + \sum_{i=1}^{m'} k_i'$$

imply

$$\sum_{j=1}^n (l_i + 1) + \sum_{j=2}^{n'} (l_i' + 1) \geq 1 + \sum_{i=2}^m k_i + \sum_{i=1}^{m'} k_i',$$

so that we arrive again at an instance of (G).
6. *One-step soundness of* (P_k)*:* Analogous to rule (G) in the previous example, noting additionally that probabilities are bounded by 1.
Resolution closedness: Analogous to the previous example.

One-step completeness: It has been shown in [9] that the following set of rules (slightly rewritten) is one-step complete for PML:

$$(0)\ \ L_0 a \qquad (\top)\ \ \frac{a}{L_p a} \qquad (>1)\ \ \frac{\neg a \vee \neg b}{\neg L_p a \vee \neg L_q b}\ (p+q>1)$$

$$(1)\ \ \frac{a \vee b}{L_p a \vee L_q b}\ (p+q=1)$$

$$(\mathbb{1})\ \ \frac{\sum_{i=1}^r c_i = \sum_{j=1}^s \bar{d}_j}{\bigwedge_{i=1}^r L_{u_i} c_i \wedge \bigwedge_{j=2}^s L_{(1-v_j)} d_j \to L_{v_1} d_1},$$

where $\bar{d}_1 = d_1$ and $\bar{d}_j = \neg d_j$ for $j \geq 2$, and rule $(\mathbb{1})$ is subject to the side condition

$$\sum_{j=1}^s v_j = \sum_{i=1}^r u_i.$$

All these rules are subsumed by the rule schema (P_k), as follows. Rule (0): take $m = 1$, $n = 0$, $k = 0$, $p_1 = 0$. Rule $(\top)$: take $m = 1$, $n = 0$, $k = 1$. Rule > 1: take $n = 2$, $m = 0$, $k = -1$. Rule (1): take $n = 0$, $m = 2$, $k = -1$. Rule $(\mathbb{1})$: take $m = 1$, $n = r + s - 1$, $k = 1 - s$, and instantiate b_i to c_i for $i = 1, \dots, r$, b_i to d_{i-r+1} for $i = r+1, \dots, r+s-1$, a_1 to d_1, q_i to u_i for $i = 1, \dots, r$, q_i to $1 - v_{i-r+1}$ for $i = r+1, \dots, r+s-1$, and p_1 to v_1.

Lemma 24

Proof. Recall that $\vee$ is coded in terms of $\wedge$ and $\neg$. □

Lemma 25

Proof. Recall that we assume that all propositional connectives are coded in terms of $\wedge$, $\neg$, and $\bot$. Soundness of the usual introduction and elimination rules for these connectives is immediate from the above definition of satisfaction in H. □

Corollary 28

Proof. The model is constructed in the same way as in the proof of Theorem 26, however with branches constructed only for negated *clauses* χ over H — this is sufficient, as the negation of a formula is satisfiable iff the negation of one of the clauses in its CNF is satisfiable. Moreover, a satisfying state $x_\top$ for $\top$, arising from the rule $\bot/\bot$ which is present iff $T\emptyset = \emptyset$, is constructed only once and then reused; this is the origin of the mentioned additional final state. The supporting Kripke frame (X, R) is constructed recursively along with the model by putting, in the notation of the said proof, $x_0 R x_\chi$ for all relevant χ. By Remark 14, the construction is always applicable. □

Theorem 37 (Space complexity)

Proof. Since $\mathcal{R}$ is $PSPACE$-tractable, so is $\mathcal{R}_C$. Thus, the functions mapping a clause ρ to the set of its $\mathcal{R}_C$-matchings and a rule to the CNF of its premise, respectively, are NPMV: in the former case, the polynomial bound is ensured by the definition of $PSPACE$-tractability, as we only need to produce one code for each matching, and in the latter case, the polynomial bound holds universally, as clauses are of polynomial size. Therefore, Steps 3 and 4 in Algorithm 1 can be performed in polynomial time. Since the depth of recursion is bounded by the rank of ϕ, it follows that the algorithm runs in $APTIME = PSPACE$ [7]. $\square$

Lemma 39

Proof. Let $V = \{a_1, \dots, a_n\}$. We note that the formulas $\sum_{i=1}^n r_i a_i \geq a_0$ and $\sum_{i=1}^n s_i a_i \geq s_0$ are propositionally equivalent iff for all valuations $\sigma : V \to \{0,1\}$, one has $\sum_{i=1}^n r_i \sigma(a_i) \geq r_0$ if and only if $\sum_{i=1}^n s_i \sigma(a_i) \geq s_0$, read as linear inequalities. Hence every formula $\phi \equiv \sum_{i=1}^n r_i a_i \geq r_0$ gives rise to the system of linear inequalities $I(\phi)$ that contains, for every valuation $\sigma : V \to \{0,1\}$, the linear inequality

$$E\sigma = \begin{cases} \sum_{i=1}^n r_i x_i \geq x_0 & \text{if } \sum_{i=1}^n r_i \sigma(a_i) \geq r_0 \\ \sum_{i=1}^n r_i x_i < x_0 & \text{if } \sum_{i=1}^n r_i \sigma(a_i) < r_0. \end{cases}$$

The formula $\sum_{i=1}^n s_i a_i \geq s_0$ is propositionally equivalent to $\phi \equiv \sum_{i=1}^n r_i a_i \geq r_0$ iff $(s_0, \dots, s_n)$ is a solution of $I\sigma(\phi)$.

Now suppose that $(r_0, \dots, r_n)$ solves the linear inequality $W = \sum_{i=1}^n u_i x_i$ op x_0, with op $\in \{<, \leq, >, \geq\}$, and let $\phi \equiv \sum_{i=1}^n r_i a_i \geq r_0$. Then $(r_0, \dots, r_n)$ solves all inequalities contained in $I(\phi) \cup \{W\}$. It follows from [28, Corollary 17.1b] that the set $I(\phi) \cup \{W\}$ has a solution $s_0, \dots, s_n$ whose logarithmic size is bounded by $6c(n+1)^3$, where c is the facet complexity of the system, i.e. an upper bound of the size of the largest inequality. As the coefficients of inequalities contained in $I(\phi)$ are either 0 or 1, we have $c \leq |W| + n + 1 \leq 2|W|$. Now let $p(x) = 12|i|^4$. Then $p(x) \geq 6c(n+1)^3$, therefore the system $I(\phi) \cup \{W\}$ has a solution $(s_0, \dots, s_n)$ such that $\lceil \log_2 |s_i| \rceil \leq p(|W|)$. As $(s_0, \dots, s_n)$ in particular solves $I(\phi)$, we have that the formulae $\sum_{i=1}^n r_i a_i \geq k$ and $\sum_{i=1}^n s_i a_i \geq s_0$ are propositionally equivalent. By construction, $(s_0, \dots, s_n)$ also solves W. $\square$