Reasoning with Global Assumptions in Arithmetic Modal Logics

Clemens Kupke[1], Dirk Pattinson[2], and Lutz Schröder[3]

[1] University of Strathclyde
[2] Australian National University
[3] Friedrich-Alexander Universität Erlangen-Nürnberg

Abstract. We establish a generic upper bound ExpTime for reasoning with global assumptions in coalgebraic modal logics. Unlike earlier results of this kind, we do not require a tractable set of tableau rules for the instance logics, so that the result applies to wider classes of logics. Examples are Presburger modal logic, which extends graded modal logic with linear inequalities over numbers of successors, and probabilistic modal logic with polynomial inequalities over probabilities. We establish the theoretical upper bound using a type elimination algorithm. We also provide a global caching algorithm that offers potential for practical reasoning.

Arithmetic modal logics feature arithmetical constraints on the number or total weight of successors. The simplest logics of this type compare weights to constants, such as graded modal logic [12] or some variants of probabilistic modal logic [19, 17]. More involved examples are *Presburger modal logic* [7], which allows Presburger constraints on numbers of successors, and probabilistic modal logic with polynomial inequalities over probabilities. The former logic allows for statements like 'the majority of university students are female', or 'dance classes have an even number of participants', while probabilistic modal logic with polynomial inequalities can assert, for example, independence of events.

These logics are the main examples we address in a more general coalgebraic framework in this paper. Our main observation is that satisfiability for coalgebraic logics can be decided in a step-by-step fashion, peeling off one layer of operators at a time. We thus reduce the overall satisfiability problem to instances of a *one-step* satisfiability problem involving only immediate successor states, and hence no nesting of modalities [26, 21]. We define a *strict* variant of this problem, distinguished by a judicious redefinition of its input size; if strict one-step satisfiability is in ExpTime, we obtain a (typically optimal) ExpTime upper bound for satisfiability under global assumptions in the full logic. For our two main examples, the requisite complexity bounds (in fact, even PSpace) on strict one-step satisfiability follow in essence directly from known complexity results in integer programming and the existential theory of reals, respectively; in other words, even in fairly complex examples the complexity bound for the full logic is obtained with comparatively little effort once the generic result is in place.

Applied to Presburger constraints, our results complement a recent result [6, 7] showing that the complexity of Presburger modal logic without global assumptions is PSpace, the same as for the modal logic K (or equivalently the description logic $\mathcal{ALC}$). For polynomial inequalities on probabilities, our syntax generalizes propositional *polynomial weight* formulae [11] to a full modal logic allowing nesting of weights (and global assumptions).

In more detail, our first contribution is to show via a type elimination algorithm [24] that also in presence of global assumptions (and, hence, in presence of the universal modality [13]), the satisfiability problem for coalgebraic modal logics is no harder than for K, i.e. in ExpTime, provided strict one-step satisfiability is in ExpTime. We then refine the algorithm to use global caching in the spirit of Goré and Nguyen [15], i.e. bottom-up expansion of a tableau-like graph and propagation of satisfiability and unsatisfiability through the graph. We thus potentially avoid constructing the whole exponential-sized tableau, and provide maneuvering space for heuristic optimization. Global caching algorithms have been demonstrated to perform well in practice [16].

Related Work Our algorithms use a semantic method, and as such complement earlier results on global caching in coalgebraic description logics that rely on tractable sets of tableau rules [14], which are not currently available for our leading examples. (In fact, [18] gives tableau-style axiomatizations of various logics of linear inequalities over the reals and over the integers; however, over the integers the rules appear to be incomplete: if $\sharp(p)$ denotes the integer weight of successors satisfying p, the formula $2\sharp(p) < 1 \sqcup -2\sharp(p) < -1$ is clearly valid, but cannot be derived.)

Work related to XML query languages has shown that reasoning in Presburger fixpoint logic is ExpTime complete [30], and that a logic with Presburger constraints and nominals is in ExpTime [3], when these logics are interpreted *over finite trees*, thus not subsuming our ExpTime upper bound for Presburger modal logic with global assumptions. It will likely be possible to obtain this bound via looping tree automata like for graded modal logic [31]. However, this would mean translating the target formula into an exponential-sized automaton, making exponential runtime the typical rather than the worst case; contrastingly, the main goal of our global caching algorithm is to avoid building the full exponential-sized set of types. Description logics with explicit quantification over integer variables and number restrictions mentioning integer variables [2] appear to be incomparable to Presburger modal logic: they do not support general linear inequalities, but on the other hand allow integer variables to be used at different modal depths. Reasoning with polynomial inequalities over probabilities has been studied in propositional logics [11] and in many-dimensional modal logics [20], which work with a single distribution on worlds rather than with world-dependent probability distributions as in [19, 17, 10].

1 Coalgebraic Logic

We briefly describe the key concepts of coalgebraic logic, a general framework that allows us to treat structurally different modal logics, such as Presburger and probabilistic modal logics, in a uniform way. We parametrize modal logics in terms of their *syntax* and their coalgebraic *semantics*. Syntactically, we work with a modal similarity type Λ of modal operators with given finite arities. The set $\mathcal{F}(\Lambda)$ of *Λ-formulas* is then given by the grammar

$$\mathcal{F}(\Lambda) \ni \phi, \psi ::= \bot \mid \phi \wedge \psi \mid \neg\phi \mid \heartsuit(\phi_1, \dots, \phi_n) \qquad (\heartsuit \in \Lambda \text{ } n\text{-ary}).$$

We omit explicit propositional atoms; these can be regarded as nullary modalities. The operators $\top$, $\rightarrow$, $\vee$, $\leftrightarrow$ are assumed to be defined in the standard way.

The semantics of formulas is then parametrized over the choice of a *Λ-structure* consisting of a set functor $T : \mathsf{Set} \to \mathsf{Set}$ and the assignment of an n-ary predicate lifting $[\![\heartsuit]\!]$ to each modality $\heartsuit \in \Lambda$, of arity n; we briefly refer to this structure just by T. We recall that an *n-ary predicate lifting* for T is a natural transformation $\lambda : Q^n \to Q \circ T^{op}$ where $Q : \mathsf{Set}^{op} \to \mathsf{Set}$ is the contravariant powerset functor, $T^{op} : \mathsf{Set}^{op} \to \mathsf{Set}^{op}$ acts like T, and Q^n denotes the pointwise n-th Cartesian power, i.e. $Q^n(X) = Q(X)^n$. Naturality of λ then amounts to commutation with preimage, i.e. $\lambda_X(f^{-1}[A_1], \dots, f^{-1}[A_n]) = Tf^{-1}[\lambda_Y(A_1, \dots, A_n)]$ for $f : X \to Y$.

The idea here is that T determines the type of systems underlying the semantics, as the coalgebras of T: Recall that a T-coalgebra $C = (X, \gamma)$ consists of a set X of *states* and a map $\gamma : X \to TX$, which should be thought of as assigning to each state s a structured collection $\gamma(x)$ of successors. The basic example has $T = \mathcal{P}$, the powerset functor; in this case, $\gamma(x)$ is just a set of successors, so a $\mathcal{P}$-coalgebra is a Kripke frame. The predicate liftings then turn predicates on the set X of states into predicates on the set TX of structured collections of successors. A basic example is the predicate lifting for the usual diamond modality $\Diamond$, given by $[\![\Diamond]\!]_X(A) = \{B \in \mathcal{P}(X) \mid B \cap A \neq \emptyset\}$.

Satisfaction $x \models_C \phi$ of formulas $\phi \in \mathcal{F}(\Lambda)$ in states x of a coalgebra $C = (X, \gamma)$ is defined inductively by the expected clauses for Boolean operators, and

$$x \models_C \heartsuit(\phi_1, \dots, \phi_n) \quad \text{iff} \quad \gamma(x) \models \heartsuit([\![\phi_1]\!]_C, \dots, [\![\phi_n]\!]_C)$$

where we write $[\![\phi]\!]_C = \{x \in X \mid x \models_C \phi\}$, and for $t \in TX$, $t \models \heartsuit(A_1, \dots, A_n)$ is short for $t \in [\![\heartsuit]\!]_X(A_1, \dots, A_n)$. Continuing the above example, the predicate lifting $[\![\Diamond]\!]$ thus induces exactly the usual semantics of $\Diamond$, i.e. $x \models_C \Diamond\phi$ iff the set of successors of x intersects with $[\![\phi]\!]_C$.

We will be interested in satisfiability under global assumptions, or, in description logic terminology, reasoning with general TBoxes [1]: Given a formula ψ, the *global assumption* (or *TBox*), a coalgebra $C = (X, \gamma)$ is a *ψ-model* if $[\![\psi]\!]_C = X$; and a formula ϕ is *ψ-satisfiable* if there exists a ψ-model C such that $[\![\phi]\!]_C \neq \emptyset$. The *satisfiability problem with global assumptions* is to decide, given ψ and ϕ, whether ϕ is ψ-satisfiable. For the complexity analysis of these problems, we assume a suitable encoding of the modal operators in Λ that enters into the calculation of the *size* $|\phi|$ of formulas ϕ; in particular, we assume that numbers occurring in the description of modal operators are coded in binary.

Previous generic algorithms in coalgebraic logic did for the most part rely on complete rule sets for the given operators [27]. Our interest in the present paper is in cases for which suitable rule sets are not (currently) available. We proceed to present our leading examples of this kind, Presburger modal logic and a probabilistic modal logic with polynomial inequalities. For the sake of readability, we focus on the case with a single (weighted) transition relation, and omit propositional atoms. Both features are easily added, e.g. using compositionality results in coalgebraic logic [28], and in fact we use them freely in the examples.

1.1 Presburger Modal Logic

Presburger modal logic [7] admits statements in Presburger arithmetic over numbers $\sharp\phi$ of successors satisfying a formula ϕ, called *cardinalities*. Throughout, we let Rels denote the set $\{<,>,=\} \cup \{\equiv_k \mid k \in \mathbb{N}\}$ of *arithmetic relations*, with $\equiv_k$ read as congruence modulo k. Syntactically, Presburger modal logic is then defined by taking Λ to contain all modal operators of the form

$$L_{a_1,\dots,a_n;\sim b} = \textstyle\sum_{i=1}^n a_i \sharp(\cdot)_i \sim b$$

where $(\cdot)_i$ denotes the i-th argument of the operator, $\sim\, \in \mathsf{Rels}$, and $a_1,\dots,a_n, b \in \mathbb{Z}$. Weak inequalities can be coded as strict ones, replacing, e.g., $\geq k$ with $> k-1$. The numbers a_i and b, as well as the modulus k in $\equiv_k$, are referred to as the *coefficients* of a Presburger constraint. We also apply these terms to constraints $\sum_{i=1}^n a_i x_i \sim b$ in general, interpreted over the integers.

The semantics of Presburger modal logic was originally defined over standard Kripke frames; in order to make sense of sums with arbitrary integer coefficients, one clearly needs to restrict to finitely branching frames. We consider an alternative more general semantics in terms of *multigraphs*, which have some key technical advantages [5]. Informally, a multigraph is a Kripke frame but with every transition edge annotated with an integer-valued multiplicity; ordinary finitely branching Kripke frames can be viewed as multigraphs by just taking edges to be transitions with multiplicity 1. Formally, a multigraph can be seen as a coalgebra for the *finite multiset functor* $\mathcal{B}$: For a set X, $\mathcal{B}(X)$ consists of the *finite multisets over* X, which are maps $\mu : X \to \mathbb{N}$ with finite support, i.e. $\mu(x) > 0$ for only finitely many x. We view μ as an $\mathbb{N}$-valued measure, and write $\mu(Y) = \sum_{x \in Y} \mu(x)$ for $Y \subseteq X$. Then, $\mathcal{B}(f)$, for maps f, acts as image measure formation. A coalgebra $\gamma : X \to \mathcal{B}(X)$ assigns to each state x a multiset $\gamma(x)$ of successor states, i.e. each successor state is assigned a transition multiplicity.

The semantics of the operators is then given by the predicate liftings

$$[\![L_{a_1,\dots,a_n;\sim b}]\!]_X(A_1,\dots,A_n) = \{\mu \in \mathcal{B}(X) \mid \textstyle\sum_{i=1}^n a_i \cdot \mu(A_i) \sim b\},$$

that is, a state x in a $\mathcal{B}$-coalgebra $C = (X,\gamma)$ satisfies $\sum_{i=1}^n a_i \cdot \sharp\phi_i \sim b$ iff $\sum_{i=1}^n a_i \cdot \gamma(x)([\![\phi_i]\!]_C) \sim b$. This setup generalises effortlessly to multiple (weighted) transition relations: If $\mathcal{R}$ is a set of roles, we take the modal operators to be

$$L_{a_1^{r_1},\dots,a_n^{r_n};\sim b} = \textstyle\sum_{i=1}^n a_i \sharp_{r_i}(\cdot)_i \sim b$$

where $r_i \in \mathcal{R}$ for every $1 \leq i \leq n$ and $\sharp_r(\cdot)$ is the number of successors along the (weighted) transition relation r. Logics with operators of this kind are then intrpreted by assigning $\mathcal{R}$-many multisets of successors to each world, i.e. as coalgebras of type $X \to \mathcal{B}(X)^{\mathcal{R}}$.

We note that satisfiability is the same over Kripke and over multigraphs:

Lemma 1. *[25] A formula ϕ is ψ-satisfiable over multigraphs iff ϕ is ψ-satisfiable over Kripke frames.*

(The proof of the non-trivial direction is by making copies of states according to their multiplicity.)

Expressiveness and Examples. Presburger modal logic subsumes graded modal logic [12]: the graded formula $\Diamond_k \phi$, read 'more than k successors satisfy ϕ, becomes $\sharp(\phi) > k$ in Presburger modal logic. Moreover, Presburger modal logic subsumes majority logic [22]: The *weak majority* formula $W\phi$ ('at least half the successors satisfy ϕ') is expressed in Presburger modal logic as $\sharp(\phi) - \sharp(\neg\phi) \geq 0$. Using propositional atoms as indicated above, we express the examples given in the abstract by the formulas

$$\mathsf{University} \to \sharp_{\mathsf{hasStudent}}(\mathsf{Female}) - \sharp_{\mathsf{hasStudent}}(\mathsf{Male}) > 0$$
$$\mathsf{DanceCourse} \to \sharp_{\mathsf{hasParticipant}}(\top) \equiv_2 0$$

where indices informally indicate the understanding of the successor relation, and the formulae are sensibly understood as global assumptions. As an example involving non-unit coefficients, a chamber of parliament in which a motion requiring a 2/3 majority has sufficient support is described by the concept

$$\sharp_{\mathsf{hasMember}}(\mathsf{SupportsMotion}) - 2\sharp_{\mathsf{hasMember}}(\neg\mathsf{SupportsMotion}) \geq 0.$$

1.2 Probabilistic Modal Logic with Polynomial Inequalities

Probabilistic logics of various forms have been studied in different contexts such as reactive systems [19] and uncertain knowledge [17, 10]. A typical feature of such logics is that they talk about probabilities $w(\phi)$ of formulas ϕ holding for the successors of a state; the concrete syntax then variously includes only inequalities of the form $w(\phi) \sim p$ for $\sim \in \{>, \geq, =, <, \leq\}$ and $p \in \mathbb{Q} \cap [0, 1]$ [19, 17], linear inequalities over terms $w(\phi)$ [10], or polynomial inequalities, with the latter so far treated only in either purely propositional settings [11] or in many-dimensional logics such as the probabilistic description logic Prob-$\mathcal{ALC}$ [20], which use a single global distribution over worlds. An important use of polynomial inequalities over probabilities is to express independence constraints [20]; e.g. two properties ϕ and ψ (of successors) are independent if $w(\phi \wedge \psi) = w(\phi)w(\psi)$.

We thus define the following *probabilistic modal logic with polynomial inequalities*: the system type is given by the *distribution functor* $\mathcal{D}$ that assigns to a set X the set $\mathcal{D}(X)$ of discrete probability distributions on X; again, for a map f, $\mathcal{D}(f)$ takes image measures. Then, a $\mathcal{D}$-coalgebra $\gamma : X \to \mathcal{D}(X)$ assigns to each state x a distribution $\gamma(x)$ over successor states. We can thus view γ as a Markov chain (interpreting $\gamma(x)$ as a distribution over possible future evolutions of the system), or as a (single-agent) type space in the sense of epistemic logic [17] (interpreting $\gamma(x)$ as the subjective probabilities assigned by the agent to possible alternative worlds in world x). We let the modal similarity type Λ consist of modalities L_p indexed over polynomials $p \in \mathbb{Q}[X_1, \dots, X_n]$, $n \geq 0$; L_p then has arity n. We denote the application of L_p to formulas $\phi_1, \dots, \phi_n$ by substituting each variable X_i in p with $w(\phi_i)$ and postulating the result to be non-negative; e.g., the formula $w(\phi \wedge \psi) - w(\phi)w(\psi) \geq 0$ denotes one half of the above-mentioned independence constraint. We correspondingly interpret L_p by the predicate lifting

$$[\![L_p]\!]_X(A_1, \dots, A_n) = \{\mu \in \mathcal{D}(X) \mid p(\mu(A_1), \dots, \mu(A_n)) \geq 0\}.$$

2 One-Step Satisfiability

The key to our approach is to deal with modalities level by level; the core concepts of the arising notion of one-step satisfiability checking go back to [25, 26, 21]. From now on, we mostly restrict the notation to unary operators although our central examples all have operators with higher arities, to avoid cumbersome notation; a fully general treatment requires no more than additional indexing. We fix a Λ-structure T throughout. Peeling off one level of modalities and abstracting from their arguments leads to the following notions.

Definition 2 (One-step pairs, one-step satisfiability). We assume a set $\mathcal{V}$ of (propositional) variables. We denote the set of Boolean formulas over a set Z of atoms by $\mathsf{Prop}(Z)$, and by $\Lambda(Z) = \{\heartsuit a \mid \heartsuit \in \Lambda, a \in Z\}$ the set of *modal atoms* over Z. As usual, a *literal* over Z is an element $z \in Z$ or a negation thereof, written ϵz where ϵ is either nothing or negation. A *modal literal* over Z is a literal over $\Lambda(Z)$. A *conjunctive clause* over Z is a finite set of literals over Z, read as a conjunction. A *one-step pair* (ϕ, η) *over* $V \subseteq \mathcal{V}$ consists of

- a conjunctive clause ϕ over $\Lambda(V)$ mentioning each variable at most once, and
- a Boolean formula $\eta \in \mathsf{Prop}(V)$ mentioning only variables occurring in ϕ.

A *one-step model* $M = (X, \tau, t)$ over V consists of

- a set X together with a $\mathcal{P}(X)$-valuation $\tau : V \to \mathcal{P}(X)$; and
- an element $t \in TX$, thought of as the successor structure of an anonymous state.

For $\eta \in \mathsf{Prop}(V)$, $\tau(\eta)$ is the interpretation of η in the Boolean algebra $\mathcal{P}(X)$ under the valuation τ. For a modal atom $\heartsuit a \in \Lambda(V)$, we put $\tau(\heartsuit a) = [\![\heartsuit]\!]_X(\tau(a)) \subseteq TX$. Via the Boolean algebra structure on $\mathcal{P}(TX)$, this extends to an assignment of $\tau(\phi) \in \mathcal{P}(TX)$ to each $\phi \in \mathsf{Prop}(\Lambda(V))$. We say that the one-step model $M = (X, \tau, t)$ *satisfies* the one step pair (ϕ, η), and write $M \models (\phi, \eta)$, if

$$\tau(\eta) = X \qquad \text{and} \qquad t \in \tau(\phi).$$

Then, (ϕ, η) is *(one-step) satisfiable* if there exists a one-step model M such that $M \models (\phi, \eta)$. The *one-step satisfiability problem* is to decide whether a given (ϕ, η) is one-step satisfiable, with η given as a DNF consisting of conjunctive clauses each mentioning every variable occurring in ϕ. The *strict one-step satisfiability problem* is the same problem, but with the *input size* defined to be just the size of ϕ; the representation of η is, then, irrelevant. We say that Λ has the *one-step* *small model property* if there is a polynomial p such that every one-step satisfiable (ϕ, η) has a one-step model (X, τ, t) with $|X| \leq p(|\phi|)$ (no bound is assumed on the representation of t).

The intuition behind these definitions is that propositional variables are placeholders for argument concepts; their valuation τ in a one-step model represents the extensions of these argument concepts; and the second component η of a one-step pair captures the Boolean constraints on the argument concepts that are globally satisfied in a given model. One-step satisfiability is precisely what will allow us to construct satisfying models later on. Note that most of a one-step pair (ϕ, η) is disregarded for purposes of determining the input size of the *strict* one-step satisfiability problem, as η, a propositional formula, can be exponentially larger than the conjunctive clause ϕ.

Example 3. In Presburger modal logic, let $\phi = \sharp(a) \geq 1 \wedge \sharp(b) \geq 1$. Then (ϕ, η) is one-step satisfiable as long as η does not force the interpretation of either a or b to be empty, i.e. both $\eta \wedge a$ and $\eta \wedge b$ need to be (propositionally) satisfiable. Thus, the strongest possible η are $a \wedge b$ and $(a \wedge \neg b) \vee (\neg a \wedge b)$.

Lemma 4. *A one-step pair (ϕ, η) over V is satisfiable iff it is satisfiable by a one-step model of the form (X, τ, t) where X is the set of valuations $V \to 2$ satisfying η (where $2 = \{\top, \bot\}$ is the set of Booleans) and $\tau(a) = \{\kappa \in X \mid \kappa(a) = \top\}$ for $a \in V$.*

Under the one-step small model property, the two versions of the one-step satisfiability problem coincide for our purposes:

Lemma 5. *Let T have the one-step small model property. Then for any complexity class $\mathcal{C}$ containing* PSpace*, strict one-step satisfiability is in $\mathcal{C}$ iff one-step satisfiability is in $\mathcal{C}$.*

Although not phrased in these terms, the complexity analysis of (TBox-free) Presburger modal logic by Demri and Lugiez [7] is based on showing that the strict one-step satisfiability problem is in PSpace [26], without using the one-step small model property – in fact, the latter is based on more recent results from integer programming:

Lemma 6. *[8] Every system of d linear inequalities over the integers with coefficients of binary length at most s has a solution with at most polynomially many non-zero components in d and s.*

The corresponding statement over the rationals (where in fact one has at most d non-zero components) is well-known, and features centrally in the analysis of probabilistic logics [11]. From these observations, we obtain sufficient tractability of one-step satisfiability in our key examples:

Example 7. 1. Presburger modal logic has the one-step small model property. To see this, let (ϕ, η) be satisfied by $M = (X, \tau, \mu)$, where by Lemma 4 we can assume that X consists of satisfying valuations of η, hence of at most exponential size in $|\phi|$. Let $V = \{a_1, \dots, a_n\}$, and put $q_i = \mu(\tau(a_i))$. By standard estimates in integer programming [23] we can assume that the $\mu(x)$ and, hence, the q_i (being sums of at most exponentially many $\mu(x)$) have polynomial binary length in $|\phi|$. Now all we need to know about τ to guarantee that M satisfies ϕ is that

$$\textstyle\sum_{x \in \tau(a_i)} \mu(x) = q_i.$$

We can see this as a system of linear constraints on the $\mu(x)$, which by Lemma 6 has a solution with only m nonzero components where m is polynomially bounded in n and the binary length s of the largest q_i, and hence in $|\phi|$; from this solution, we immediately obtain a one-step model of (ϕ, η) with m states.
Moreover, again using Lemma 4, one-step satisfiability in Presburger modal logic easily reduces to checking solvability of Presburger constraints over the integers, which can be done in NP and hence in PSpace; by Lemma 5, we obtain that *strict one-step satisfiability in Presburger modal logic is in* PSpace.

2. By a completely analogous (slightly easier) argument as for Presburger modal logic, probabilistic modal logic with polynomial inequalities has the one-step small model property. In this case, one-step satisfiability reduces to solvability of systems of polynomial inequalities over the reals, which can be checked in PSpace [4] (this argument can essentially be found in [11]). Again, we obtain that *strict one-step satisfiability in probabilistic modal logic with polynomial inequalities is in* PSpace.

By [26], these observations imply decidability in PSpace of the *plain* satisfiability problem. We show below that one obtains an optimal upper bound ExpTime for satisfiability under global assumptions. One should note that the proof of the one-step small model property will in both cases work for any coalgebraic modal logic over integer- or real-weighted systems whose modalities depend only on the measures of their arguments.

Remark 8. Most previous generic complexity results in coalgebraic logic have relied on tractable sets of tableau rules, e.g. [27, 29, 14]. These rules are of the shape ϕ/η where ϕ is a conjunctive clause over $\Lambda(V)$ and $\eta \in \mathsf{Prop}(V)$, to be read, within a system including also the standard propositional rules, as 'in order to establish that ψ is satisfiable, show that the conclusions of all rule matches to ψ are satisfiable'. E.g. PSpace-tractability [27] of a rule set essentially amounts to the rules being codable in such a way that it suffices, for each ψ, to consider only rules with polynomial-sized codes. In terms of one-step pairs, this means essentially that there are certificates (in the shape of rule codes) for *un*satisfiability of a one-step pair (ϕ, η) that are of polynomial size in $|\phi|$ and can be checked in polynomial space in $|\phi|$ (in particular comparing η with the conclusion of the encoded rule), so that the strict one-step satisfiability problem is in PSpace. Summing up, complexity bounds obtained by our current semantic approach subsume earlier tableau-based ones.

3 Type Elimination

We now describe a type elimination algorithm that realizes an ExpTime upper bound for reasoning with global assumptions in coalgebraic logics. Like all type elimination algorithms, it is not suited for practical use, as it begins by constructing the full exponential-sized set of types. We therefore refine the algorithm to a global caching algorithm in Section 4.

As usual, we rely on defining a scope of relevant concepts:

Definition 9. A set Σ of concepts is *closed* if Σ is closed under subconcepts and single negations.

We fix from now on a global assumption ψ and a formula ϕ_0 to be checked for ψ-satisfiability. We denote the closure of $\{\psi, \phi_0\}$ in the above sense by Σ.

Definition 10. A ψ*-type* is a subset $\mathcal{T} \subseteq \Sigma$ such that

- $\psi \in \mathcal{T} \not\ni \bot$;
- whenever $\neg\phi \in \Sigma$, then $\neg\phi \in \mathcal{T}$ iff $\phi \notin \mathcal{T}$;
- whenever $\phi \wedge \chi \in \Sigma$, then $\phi \sqcap \chi \in \mathcal{T}$ iff $\phi, \chi \in \mathcal{T}$.

The design of the algorithm relies on one-step satisfiability as an abstraction: We denote the set of all ψ-types by S_0. We take V to be the set of propositional variables $a_{\heartsuit\phi}$ for all modal atoms $\heartsuit\phi \in \Sigma$; we then define a substitution σ by $\sigma(a_{\heartsuit\phi}) = \phi$ for all $a_{\heartsuit\phi} \in V$. For $S \subseteq S_0$ and $\mathcal{T} \in S$, we construct a one-step pair $(\phi_\mathcal{T}, \eta_S)$ over V by taking $\phi_\mathcal{T}$ to be the set of all modal literals $\epsilon\heartsuit a$ over V such that $\epsilon\heartsuit\sigma(a) \in \mathcal{T}$, and η_S a DNF consisting of all conjunctive clauses ϑ (seen as sets of literals L) over V such that $\{L\sigma \mid L \in \vartheta\} \subseteq \mathcal{T}$ for some $\mathcal{T} \in S$. Then we define a functional $\mathcal{E} : \mathcal{P}(S_0) \to \mathcal{P}(S_0)$ by $S \mapsto \{\mathcal{T} \in S_0 \mid (\phi_\mathcal{T}, \eta_S) \text{ one-step satisfiable}\}$.

Lemma 11. *$\mathcal{E}$ is monotone w.r.t. set inclusion.*

We can thus compute the greatest postfixpoint $\nu\mathcal{E}$ of $\mathcal{E}$ by just iterating $\mathcal{E}$:

Algorithm 12. (Decide by type elimination whether ϕ_0 is satisfiable over ψ)

1. Set $S := S_0$.
2. Compute $S' = \mathcal{E}(S)$; if $S' \neq S$ then put $S := S'$ and repeat.
3. Return 'yes' if $\phi_0 \in \mathcal{T}$ for some $\mathcal{T} \in S$, and 'no' otherwise.

If strict (!) one-step satisfiability is in ExpTime, then this algorithm has at most exponential run time. We analyse correctness:

Definition 13. A type $\mathcal{T}$ is *realized* in a ψ-model $C = (X, \gamma)$ if there exists $x \in X$ such that $x \models \phi$ for all $\phi \in \mathcal{T}$.

Lemma 14. *The set of types realized in a given ψ-model is a postfixpoint of $\mathcal{E}$.*

By Lemma 14, all ψ-satisfiable types are in $\nu\mathcal{E}$. Thus, the algorithm is sound, i.e. answers 'yes' on ψ-satisfiable concepts. To see completeness, we show

Lemma 15. *Let S be a postfixpoint of $\mathcal{E}$. Then there exists a T-coalgebra $C = (S, \gamma)$ such that for each $\phi \in \Sigma$, $[\![\phi]\!]_C = \{\mathcal{T} \in S \mid \phi \in \mathcal{T}\}$.*

An interpretation as in Lemma 15 is clearly a ψ-model, so that Algorithm 12 is complete, i.e. answers 'yes' *only* on ψ-satisfiable concepts.

Theorem 16. *If strict one-step satisfiability in T is in* EXPTIME*, then satisfiability with global assumptions is in* EXPTIME.

Example 17. By the results of the previous section and by inheriting lower bounds from reasoning with global assumptions in K, we obtain that reasoning with global assumptions in Presburger modal logic and in probabilistic modal logic with polynomial inequalities is EXPTIME-complete.

4 Global Caching

We now develop the type elimination algorithm from the preceding section into a global caching algorithm. Existing global caching algorithms work with systems of tableau rules (satisfiability is guaranteed if every applicable rule has at least one satisfiable conclusion) [14]. The fact that we work with a semantics-based decision procedure impacts on the design of the algorithm in two ways:

- In a tableaux setting, node generation is driven by the tableau rules, and a global caching algorithm generates successor nodes by applying tableau rules. In principle, however, successor nodes can be generated at will, with the rules just pointing to relevant nodes. In our setting, we make the relevant nodes explicit using the concept of *children*.
- The rules govern the propagation of satisfiability and unsatisfiability among the nodes. Semantic propagation of satisfiability is straightforward, but propagation of unsatisfiability again needs the concept of children: a node can only be marked as unsatisfiable once all its children have been generated (and too many of them are unsatisfiable).

We continue to work with a closed set Σ as in Section 3 (generated by the global assumption ψ and the target formula ϕ_0) but replace types with *(tableau) sequents*, i.e. arbitrary subsets $\Gamma, \Theta \subseteq \Sigma$, understood conjunctively; in particular, a sequent need not mention every formula in Σ. We write $\mathsf{Seqs} = \mathcal{P}(\Sigma)$. A *state* is a sequent consisting of modal literals only (recall that we take atomic propositions as nullary operators). We denote the set of states by States.

To convert sequents into states, we emply the usual *propositional rules*

$$\frac{\Gamma, \phi_1 \sqcap \phi_2}{\Gamma, \phi_1, \phi_2} \qquad \frac{\Gamma, \neg(\phi_1 \sqcap \phi_2)}{\Gamma, \neg\phi_1 \mid \Gamma, \neg\phi_2} \qquad \frac{\Gamma, \neg\neg\phi}{\Gamma, \phi} \qquad \frac{\Gamma, \bot}{}$$

where $\mid$ denotes alternative conclusions. (As usual, a rule $\Gamma, \phi, \neg\phi/$ with no conclusions is admissible.)

Definition 18. The *children* of a state Γ are the sequents consisting of ψ and, for each modal literal $\epsilon\heartsuit\phi \in \Gamma$, a choice of either ϕ or $\neg\phi$. The *children* of a non-state sequent are its conclusions under the propositional rules.

We modify the functional $\mathcal{E}$ defined in the previous section to work also with sequents and depend on a set $G \subseteq \mathsf{Seqs}$ of sequents already generated: we define $\mathcal{E}_G : \mathcal{P}(G) \to \mathcal{P}(G)$ by taking $\mathcal{E}_G(S)$ to contain

- a non-state sequent $\Gamma \in G - \mathsf{States}$ iff every propositional rule that applies to Γ has a satisfiable conclusion that is contained S, and
- a state $\Gamma \in G \cap \mathsf{States}$ iff, for C the set of children of Γ, the one-step pair $(\phi_\Gamma, \eta_{S\cap C})$ over V_Γ is one-step satisfiable where V_Γ contains a variable $a_{\epsilon\heartsuit\phi}$ for each modal literal $\epsilon\heartsuit\phi \in \Gamma$, and ϕ_Γ, $\eta_{S\cap C}$ are defined like $\phi_\mathcal{T}$, η_S in the previous section, using the substitution $\sigma_\Gamma(a_{\epsilon\heartsuit\phi}) = \phi$ in place of σ.

To propagate unsatisfiability, we introduce a second functional $\mathcal{A}_G : \mathcal{P}(G) \to \mathcal{P}(G)$, where we take $\mathcal{A}_G(S)$ to contain

- a non-state sequent $\Gamma \in G - \mathsf{States}$ iff there is a propositional rule applying to Γ all whose conclusions are in S, and
- a state $\Gamma \in G \cap \mathsf{States}$ iff, for C the set of children of Γ, we have $C \subseteq G$ and the one-step pair $(\phi_\Gamma, \eta_{C\setminus S})$ is one-step unsatisfiable.

The global caching algorithm maintains, as global variables, a set G of sequents with subsets E and A of sequents already decided as satisfiable or unsatisfiable, respectively.

Algorithm 19. (Decide $\mathcal{T}$-satisfiability of ϕ_0 by global caching.)

1. Initialize $G = \{\Gamma_0\}$ with $\Gamma_0 = \{\phi_0, \psi\}$, and $E = A = \emptyset$.
2. (Expand) Select a sequent $\Gamma \in G$ that has children that are not in G, and add any number of these children to G. If no sequents with missing children are found, go to Step 5
3. (Propagate) Optionally recalculate E as the greatest fixed point $\nu S.\, \mathcal{E}_G(S \cup E)$, and A as $\mu S.\, \mathcal{A}_G(S \cup A)$. If $\Gamma_0 \in E$, return 'yes'; if $\Gamma_0 \in A$, return 'no'.
4. Go to Step 2.
5. Recalculate E as $\nu S.\, \mathcal{E}_G(S \cup E)$; return 'yes' if $\Gamma_0 \in E$, and 'no' otherwise.

Theorem 20. *If the strict one-step satisfiability problem of T is in* ExpTime *then the global caching algorithm decides satisfiability under global assumptions in* ExpTime.

The key feature of the algorithm is that it avoids generating the full set of types by detecting satisfiability or unsatisfiability on the fly in the intermediate propagation step. The non-determinism in the formulation of the algorithm can be resolved arbitrarily, i.e. any choice (e.g. of which sequents to add in the expansion step and whether or not to trigger propagation) leads to correct results; thus, it affords room for heuristic optimization. Detecting *un*satisfiability (but not satisfiability) in Step 3 requires previous generation of all, in principle exponentially many, children of a sequent. This is presumably not necessarily prohibitive in practice, as the exponential dependence is only in the number of *top-level* modalities in a sequent. As an extreme example, if we encode $\Diamond\phi$ as $\sharp(\phi) > 0$, then the sequent $\{\Diamond^n\top\}$ (n successive diamonds) induces 2^n types but has only two children, $\{\Diamond^{n-1}\top\}$ and $\{\neg\Diamond^{n-1}\top\}$.

5 Conclusions

We have provided a generic upper bound ExpTime for reasoning with global assumptions in coalgebraic modal logics, based on a generic semantic approach centered around *one-step satisfiability checking*. This approach is particularly suitable for logics for which no tractable sets of modal tableau rules are known; our core examples of this type are Presburger modal logic and probabilistic modal logic with polynomial inequalities. (Another example is Elgesem's logic of agency [9], which also satisfies the conditions of our generic result [26].) The upper complexity bounds we obtain for these logics by instantiating our generic results appear to be new. The upper bound is based on a type elimination algorithm; additionally, we have designed a more practical global caching algorithm that offers a perspective for efficient reasoning in practice.

Acknowledgements We wish to thank Erwin R. Catesbeiana for remarks on unsatisfiability. Work of the first author forms part of the DFG project GenMod2 (SCHR 1118/5-2).

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, eds. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. F. Baader and U. Sattler. Description logics with symbolic number restrictions. In *European Conf. Artificial Intelligence, ECAI 1996*, pp. 283–287. Wiley, 1996.
3. E. Bárcenas and J. Lavalle. Expressive reasoning on tree structures: Recursion, inverse programs, Presburger constraints and nominals. In *Advances in Artificial Intelligence and its Applications, MICAI 2013*, vol. 8265 of *LNCS*, pp. 80–91. Springer, 2013.
4. J. Canny. Some algebraic and geometric computations in PSPACE. In *Symposium on Theory of Computing, STOC 1988*, pp. 460–467. ACM, 1988.
5. G. D'Agostino and A. Visser. Finality regained: A coalgebraic study of Scott-sets and multisets. *Arch. Math. Logic*, 41:267–298, 2002.
6. S. Demri and D. Lugiez. Presburger modal logic is pspace-complete. In *Automated Reasoning, IJCAR 2006*, vol. 4130 of *LNCS*, pp. 541–556. Springer, 2006.
7. S. Demri and D. Lugiez. Complexity of modal logics with Presburger constraints. *J. Applied Logic*, 8:233–252, 2010.
8. F. Eisenbrand and G. Shmonin. Carathéodory bounds for integer cones. *Oper. Res. Lett.*, pp. 564–568, 2006.
9. D. Elgesem. The modal logic of agency. *Nordic J. Philos. Logic*, 2:1–46, 1997.
10. R. Fagin and J. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41:340–367, 1994.

11. R. Fagin, J. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inform. Comput.*, 87:78–128, 1990.
12. K. Fine. In so many possible worlds. *Notre Dame J. Form. Log.*, 13:516–520, 1972.
13. V. Goranko and S. Passy. Using the universal modality: Gains and questions. *J. Log. Comput.*, 2:5–30, 1992.
14. R. Goré, C. Kupke, and D. Pattinson. Optimal tableau algorithms for coalgebraic logics. In *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2010*, vol. 6015 of *LNCS*, pp. 114–128. Springer, 2010.
15. R. Goré and L. Nguyen. EXPTIME tableaux for $\mathcal{ALC}$ using sound global caching. In *Description Logics, DL 2007*, vol. 250 of *CEUR Workshop Proc.*, 2007.
16. R. Goré and L. Postniece. An experimental evaluation of global caching for $\mathcal{ALC}$ (system description). In *Automated Reasoning, IJCAR 2008*, vol. 5195 of *LNCS*, pp. 299–305. Springer, 2008.
17. A. Heifetz and P. Mongin. Probabilistic logic for type spaces. *Games and Economic Behavior*, 35:31–53, 2001.
18. C. Kupke and D. Pattinson. On modal logics of linear inequalities. In *Advances in Modal Logic, AiML 2010*, pp. 235–255. College Publications, 2010.
19. K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94:1–28, 1991.
20. C. Lutz and L. Schröder. Probabilistic description logics for subjective uncertainty. In *Principles of Knowledge Representation and Reasoning, KR 2010*, pp. 393–403. AAAI, 2010.
21. R. Myers, D. Pattinson, and L. Schröder. Coalgebraic hybrid logic. In *Foundations of Software Science and Computation Structures, FoSSaCS 2009*, vol. 5504 of *LNCS*, pp. 137–151. Springer, 2009.
22. E. Pacuit and S. Salame. Majority logic. In *Principles of Knowledge Representation and Reasoning, KR 2004*, pp. 598–605. AAAI Press, 2004.
23. C. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28:765–768, 1981.
24. V. Pratt. Models of program logics. In *Foundations of Computer Science, FOCS 1979*, pp. 115–122. IEEE Comp. Soc., 1979.
25. L. Schröder. A finite model construction for coalgebraic modal logic. *J. Log. Algebr. Prog.*, 73:97–110, 2007.
26. L. Schröder and D. Pattinson. Shallow models for non-iterative modal logics. In *Advances in Artificial Intelligence, KI 2008*, vol. 5243 of *LNAI*, pp. 324–331. Springer, 2008.
27. L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. *ACM Trans. Comput. Log.*, 10:13:1–13:33, 2009.
28. L. Schröder and D. Pattinson. Modular algorithms for heterogeneous modal logics via multi-sorted coalgebra. *Math. Struct. Comput. Sci.*, 21:235–266, 2011.
29. L. Schröder, D. Pattinson, and C. Kupke. Nominals for everyone. In *Int. Joint Conf. Artificial Intelligence, IJCAI 2009*, pp. 917–922, 2009.
30. H. Seidl, T. Schwentick, and A. Muscholl. Counting in trees. In *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, pp. 575–612. Amsterdam Univ. Press, 2008.
31. S. Tobies. *Complexity results and practical algorithms for logics in Knowledge Representation.* PhD thesis, RWTH Aachen, 2001.