Optimal Tableau Algorithms for Coalgebraic Logics

Rajeev Goré[1], Clemens Kupke[2], and Dirk Pattinson[2]

[1] Computer Science Laboratory, The Australian National University
[2] Department of Computing, Imperial College London

Abstract. Deciding whether a modal formula is satisfiable with respect to a given set of (global) assumptions is a question of fundamental importance in applications of logic in computer science. Tableau methods have proved extremely versatile for solving this problem for many different individual logics but they typically do not meet the known complexity bounds for the logics in question. Recently, it has been shown that optimality can be obtained for some logics while retaining practicality by using a technique called "global caching". Here, we show that global caching is applicable to all logics that can be equipped with coalgebraic semantics, for example, classical modal logic, graded modal logic, probabilistic modal logic and coalition logic. In particular, the coalgebraic approach also covers logics that combine these various features. We thus show that global caching is a widely applicable technique and also provide foundations for optimal tableau algorithms that uniformly apply to a large class of modal logics.

1 Introduction

Modal logics have many applications in computer science, and e.g. provide a rigorous foundation for reasoning about programs [15] and knowledge [7]. Typically, we are given a set formulas Δ that represents our assumptions (e.g. knowledge about a particular domain) and are faced with the task of deciding whether a formula A (that we may think of as a hypothesis) is logically consistent with Δ. From a model theoretic perspective, this means that there exists at least one model that validates Δ everywhere, but also makes A true in at least one point. The elements of Δ are usually referred to as global assumptions in modal logic, or as a TBox in description logic. Various automated theorem proving techniques have been developed to handle this task but it is fair to say that tableau methods have proved particularly versatile for solving this problem [1,9,26].

Tableau algorithms, however, often do not meet the known complexity bounds for the logics in question. For example, the traditional tableau algorithm for the modal logic K requires double exponential time in the worst case, even though the global satisfiability problem for this logic is known to be Exptime-complete [1]. The success of suboptimal tableau algorithms in practice, implemented in reasoners like Racer [14] and Fact++[27], lies in the vast array of optimisations that

have been developed for the underlying tableau methods [17]. In contrast, the (optimal) algorithms that underly typical complexity proofs either perform rather wholesale fixpoint computations or employ semantical means, which is infeasible in practice. Clearly the ideal situation is to have optimal tableau algorithms that remain amenable to proven optimisation techniques.

The main reason for the suboptimal behaviour of tableau algorithms is that they proceed by searching one branch at a time, using backtracking, and the same node can appear on multiple branches. The second occurrence of the node in a different branch will repeat the computations already performed by its previous incarnation, since the previous branch will have been reclaimed via backtracking. Although optimal tableau algorithms that avoid this behaviour are known [6], they are rarely used by practitioners because they are difficult to implement [1]. Recently, it has been shown that both optimality and ease of implementation can be reconciled while keeping the feasibility of tableau-based algorithms for the description logics $\mathcal{ALC}$ and $\mathcal{ALCI}$ [10,13] by employing so-called "global caching". The resulting tableau algorithms explore a graph of nodes, rather than a tree with distinct branches, since subsequent incarnations of a node lead to a "cache hit" to the first incarnation on a previous branch. It has been experimentally demonstrated that global caching compares very favourably with other caching techniques known in the literature [12].

Here, we show that global caching can be applied not only to logics with an underlying relational semantics, but also to a large class of logics that is amenable to coalgebraic semantics. This class contains many different logics such as classical modal logic, graded modal logic, probabilistic modal logic and coalition logic, as well as their various combinations. We first construct a complete tableau calculus for coalgebraic logics with global assumptions where all closed tableaux are finite trees, and then show that global caching is applicable to this type of calculus. Both results are self-contained, and completeness of global caching readily applies to any tableau calculus that can be encoded as reachability game. In summary, we derive a concrete algorithm to decide satisfiability of modal formulas in presence of global assumptions that uniformly applies to a large class of logics. We illustrate the technical development by instantiating the coalgebraic framework to three different logics: probabilistic modal logic, coalition logic, and coalition logic with probabilistic outcomes that arises as a combination of both. In summary, we not only extend the applicability of global caching by a large margin, but also obtain new and optimal tableau algorithms for a large class of logics, including e.g. probabilistic modal logic, for which no tableau-based decision procedure is so far known to exist.

Related Work. Global caching has so far been used for logics with relational semantics in [10,11]. The extension of global caching, given in this paper, to logics that do not have an underlying relational semantics is new. The complexity of coalgebraic logics has been studied previously in [24] without global assumptions, and [25] establishes an EXPTIME complexity bound in the presence of global assumptions. The tableau calculus given here is new, and unlike the

algorithm in *op.cit.* which is based on Hintikka sets, the resulting algorithm is easily implementable.

2   Preliminaries and Notation

To keep our treatment parametric in the underlying modal logic, we fix a *modal simlarity type* Λ consisting of modal operators with arities, and a denumerable set V of *propositional variables*. In the sequel, we will only consider formulas in negation normal form and abbreviate $\overline{\mathsf{V}} = \{\overline{p} \mid p \in \mathsf{V}\}$ and similary $\overline{\Lambda} = \{\overline{\heartsuit} \mid \heartsuit \in \Lambda\}$ where we consider $\overline{\heartsuit}$ as a modal operator with the same arity as $\heartsuit$. The set $\mathcal{F}(\Lambda)$ of Λ-*formulas* is given by the grammar below

$$\mathcal{F}(\Lambda) \ni A_1, \dots, A_n ::= p \mid \overline{p} \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \mid \heartsuit(A_1, \dots, A_n) \mid \overline{\heartsuit}(A_1, \dots, A_n)$$

where $\heartsuit \in \Lambda$ is an n-ary operator. The *rank* of a formula $A \in \mathcal{F}(\Lambda)$ is the maximal nesting depth of modal operators in A and is denoted by $\mathsf{rank}(A)$, and $\mathsf{subf}(A)$ denotes the set of subformulas of A. The *closure* $\mathsf{cl}(A)$ of A contains all subformulas of A and their negations, i.e. $\mathsf{cl}(A) = \mathsf{subf}(A) \cup \overline{\mathsf{subf}(A)}$. We write

$$(\Lambda \cup \overline{\Lambda})(F) = \{\heartsuit(A_1, \dots, A_n) \mid \heartsuit \in \Lambda \cup \overline{\Lambda} \ n\text{-ary}, A_1, \dots, A_n \in F\}$$

for the set of all formulas that can be constructed by applying a (possibly negated) modal operator to elements of a set F of formulas.

A Λ-*tableau-sequent*, short Λ-*sequent* or just *sequent*, is a finite set of Λ-formulas that we read conjunctively, and we write $\mathcal{S}(\Lambda)$ for the set of Λ-sequents. The rank of a sequent Γ is the maximum of the ranks of the elements of Γ and we put $\mathsf{rank}(\emptyset) = 0$. The closure of a sequent is given by $\mathsf{cl}(\Gamma) = \bigcup\{\mathsf{cl}(A) \mid A \in \Gamma\}$. As usual, we identify a formula $A \in \mathcal{F}(\Lambda)$ with the singleton sequent $\{A\} \in \mathcal{S}(\Lambda)$ and write Γ, Δ for the union of Γ and Δ. We write $\mathsf{State}(\Lambda)$ for the set of Λ-sequents that neither contain a top-level propositional connective nor a pair $p, \overline{p}$ of complementary propositional variables. As we only deal with formulas in negation normal form, negation becomes a derived operation, and we write $\overline{A}$ for the negation of a formula $A \in \mathcal{F}(\Lambda)$ given by $\overline{\overline{p}} = p$, $\overline{(A \wedge B)} = \overline{A} \vee \overline{B}$, $\overline{A \vee B} = \overline{A} \wedge \overline{B}$, $\overline{\heartsuit(A_1, \dots A_n)} = \overline{\heartsuit}(\overline{A}_1, \dots, \overline{A}_n)$ and $\overline{\overline{\heartsuit}(A_1, \dots, A_n)} = \heartsuit(\overline{A}_1, \dots, \overline{A}_n)$. This notation extends to sequents so that $\overline{\Gamma} = \{\overline{A} \mid A \in \Gamma\}$. A *substitution* is a mapping $\sigma : \mathsf{V} \to \mathcal{F}(\Lambda)$, and the result of replacing every occurrence of $p \in \mathsf{V}$ in a formula $A \in \mathcal{F}(\Lambda)$ is denoted by $A\sigma$. Again, this extends to sequents, and $\Gamma\sigma = \{A\sigma \mid A \in \Gamma\}$ if $\Gamma \in \mathcal{S}(\Lambda)$.

On the semantical side, parametricity is achieved by adopting coalgebraic semantics [19]: formulas are interpreted over T-coalgebras, where T is an endofunctor on sets, and we recover the semantics of a large number of logics by specific choices for T (Example 1). To interpret the modal operators $\heartsuit \in \Lambda$, we require that T extends to a Λ-*structure*, i.e. T comes equipped with a predicate lifting (natural transformation) of type $[\![\heartsuit]\!] : 2^n \to 2 \circ T^{\mathrm{op}}$ for every n-ary modality $\heartsuit \in \Lambda$, where $2 : \mathsf{Set} \to \mathsf{Set}^{\mathrm{op}}$ is the contravariant powerset functor. In elementary terms, this amounts to assigning a set-indexed family

of functions $([\![\heartsuit]\!]_X : \mathcal{P}(X)^n \to \mathcal{P}(TX))_{X \in \mathsf{Set}}$ to every n-ary modal operator $\heartsuit \in \Lambda$ such that $(Tf)^{-1} \circ [\![\heartsuit]\!]_X(A_1, \ldots, A_n) = [\![\heartsuit]\!]_Y(f^{-1}(A_1), \ldots, f^{-1}(A_n))$ for all sets X, Y and all functions $f : Y \to X$. If $\heartsuit \in \Lambda$ is n-ary, we put $[\![\overline{\heartsuit}]\!]_X(A_1, \ldots, A_n) = (TX) \setminus [\![\heartsuit]\!]_X(X \setminus A_1, \ldots, X \setminus A_n)$. We often leave the predicate liftings implicit and refer to a Λ-structure just in terms of the underlying endofunctor T.

In the coalgebraic approach, the role of frames is played by T-*coalgebras*, i.e. pairs (C, γ) where C is a (state) set and $\gamma : C \to TC$ is a (transition) function. A T-*model* is a triple (C, γ, π) where (C, γ) is a T-coalgebra and $\pi : \mathsf{V} \to \mathcal{P}(C)$ is a valuation of the propositional variables. For a Λ-structure T and a T-model $M = (C, \gamma, \pi)$, the *truth set* $[\![A]\!]_M$ of a formula $A \in \mathcal{F}(\Lambda)$ w.r.t. M is given inductively by the following, where $\heartsuit \in \Lambda \cup \overline{\Lambda}$ is n-ary: $[\![p]\!]_M = \pi(p)$, $[\![\overline{p}]\!]_M = C \setminus \pi(p)$ and

$$[\![\heartsuit(A_1, \ldots, A_n)]\!]_M = \gamma^{-1} \circ [\![\heartsuit]\!]_C([\![A_1]\!]_M, \ldots, [\![A_n]\!]_M).$$

We write $M, c \models A$ if $c \in [\![A]\!]_M$ and $M \models A$ if $M, c \models A$ for all $c \in C$. Again, this extends to sequents under a conjunctive reading, and we put $[\![\Gamma]\!]_M = \bigcap \{[\![A]\!]_M \mid A \in \Gamma\}$ and write $M \models \Gamma$ if $M \models A$ for all $A \in \Gamma$. We denote the model class of a sequent $\Delta \in \mathcal{S}(\Lambda)$ by $\mathsf{Mod}(\Delta)$, which comprises the class of all T-models M with $M \models \Delta$, that is, M globally validates Δ. If $\Gamma, \Delta \in \mathcal{S}(\Lambda)$ are sequents, we say that Γ is *satisfiable in* $\mathsf{Mod}(\Delta)$ if there exists $M \in \mathsf{Mod}(\Delta)$ such that $[\![\Gamma]\!]_M \neq \emptyset$.

Our main interest in this paper is the *global satisfiability problem*, that is, to determine whether a sequent Γ is satisfiable in $\mathsf{Mod}(\Delta)$, for a set Δ of global assumptions. The generality of the coalgebraic approach allows us to treat this problem uniformly for a large class of structurally different modal logics that is moreover closed under composition, as the following example demonstrates.

Example 1. The generic approach of coalgebraic semantics specialises to a large class of different logics by instantiating the signature functor T appropriately. The class of these logics comprises classical and monotone modal logic in the sense of [2], the modal logic K, graded modal logic [8], probabilistic modal logic [16], coalition logic [21] and conditional logic [2]. We refer to [20,24] for details on their coalgebraic treatment. Here, we concentrate on probabilistic modal logic, coalition logic and a combination of both.

1. Coalition logic over a finite set N of agents has similarity type $\Lambda_{\mathsf{G}} = \{[C] \mid C \subseteq N\}$, and is interpreted over game frames, i.e. coalgebras for the functor

$$\mathsf{G}(X) = \{(f, (S_i)_{i \in N}) \mid \emptyset \neq S_i \subseteq \mathbb{N} \text{ finite for all } i \in \mathbb{N}, f : \prod_{i \in N} S_i \to X\}.$$

The S_i are the strategies of agent i and f is an outcome function. We read $[C]A$ as "coalition C can achieve A in the next round of the game", captured by

$$[\![C]\!]_X(A) = \{(f, (S_i)_{i \in N}) \in \mathsf{G}(X) \mid$$
$$\exists (s_i)_{i \in C} \in (S_i)_{i \in C}. \forall (s_i)_{i \in N \setminus C} \in (S_i)_{i \in N \setminus C}. f((s_i)_{i \in N}) \in A\}$$

that induces – up to the move to finite sets of strategies – the standard semantics of coalition logic [21].

2. The syntax of probabilistic modal logic is induced by the similarity type $\Lambda_D = \{\langle p \rangle \mid p \in [0,1] \cap \mathbb{Q}\}$ and we put $[p] = \overline{\langle p \rangle}$. The formula $\langle p \rangle A$ reads as "A holds with probability at least p in the next state". The semantics of the probabilistic modal logic is given by the structure

$$D(X) = \{\mu \colon X \to_f [0,1] \mid \mu(X) = 1\} \quad [[\langle p \rangle]]_X(A) = \{\mu \in D(X) \mid \mu(A) \geq p\}$$

where $X \to_f [0,1]$ is the set of all functions $f : X \to [0,1]$ with finite support, i.e. $f(x) \neq 0$ for only finitely many $x \in X$, and $\mu(A) = \sum_{x \in A} \mu(x)$. Coalgebras for D are precisely image-finite Markov chains.

3. A combination of probabilistic modal logic and coalition logic over a set N of agents arises by considering the (combined) similarity type

$$\Lambda_{D \circ G} = \{\langle p \rangle[C] \mid p \in [0,1] \cap \mathbb{Q}, C \subseteq N\}$$

and we read the formula $\langle p \rangle[C]A$ as "with probability p coalition C has a collaborative strategy to achieve A in the next round of the game". Formulas are interpreted over coalgebras for the (combined) endofunctor $D \circ G$ by the (combined) predicate lifting

$$[[\langle p \rangle[C]]]_X = [[\langle p \rangle]]_{GX} \circ [[[C]]]_X : \mathcal{P}(X) \to \mathcal{P}(D \circ G(X))$$

where the interpretation of the individual modalities $[[\langle p \rangle]]$ and $[[[C]]]$ is as above. In a $D \circ G$-coalgebra (C, γ), the transition function γ delivers a probability distribution over possible outcomes of a strategic game. The predicate lifting $[[\langle p \rangle[C]]]$ singles out all those distributions that assign probability $\geq p$ to the set of those outcomes for which coalition C can achieve A.

Note that this is just one possible combination that naturally finds its place in the coalgebraic framework and refer the reader to [3,5,23] for details.

3   Tableaux and Games for Global Consequence

The first goal of this paper is to set up a sound and complete tableau system for global satisfiability in coalgebraic modal logics. Completeness is established via winning strategies in the associated reachability games. We begin by introducing a generic version of both that we later specialise to coalgebraic logics.

Definition 2. A *tableau system* is a pair (S, R) where S is a set (of sequents) and R is a set of rules of the form Γ/Ψ where $\Gamma \in S$ and $\Psi \subseteq S$ is finite.

A sequent $\Gamma \in S$ *has a closed tableau* in the system (S, R) if Γ is an element of the least set closed under the rules in R, that is, an element of the least fixpoint of the (evidently monotone) operator

$$M : \mathcal{P}(S) \to \mathcal{P}(S), M(X) = \{\Gamma \in S \mid \exists \Psi \subseteq X.(\Gamma, \Psi) \in R\}.$$

We say that (S, R) is *finite*, if both S and R are finite.

We understand axioms as rules $\Gamma/\emptyset$ with no conclusions so that the least fixpoint of M will contain all sequents Γ for which we can construct a closed tableau, i.e. a tree with root Γ constructed according to the rules in R whose leaves are all axioms. Tableau systems can be described in terms of reachability games:

Definition 3. A *reachability game* played by the two players $\exists$ (Éloise) and $\forall$ (Abelard) is a tuple $\mathsf{G} = (B_\exists, B_\forall, E)$ with $B_\exists \cap B_\forall = \emptyset$, where

- $B_\exists$ and $B_\forall$ are the *positions* owned by the players $\exists$ and $\forall$, respectively
- $E \subseteq (B_\exists \cup B_\forall)^2$ is a binary relation that indicates the allowed moves.

The *board* B of a reachability game $(B_\exists, B_\forall, E)$ is the disjoint union of positions, i.e. $B = B_\exists \cup B_\forall$. A *play* in G is a finite or infinite sequence of positions $(b_0, b_1, \dots)$ with the property that $(b_i, b_{i+1}) \in E$ for all i, i.e. all moves are legal, and b_0 is the *initial position* of the play. A *full play* is either infinite, or a finite play ending in a position b_n where $E[b_n] = \{b \in B \mid (b_n, b) \in E\} = \emptyset$, i.e. no more moves are possible. A finite play is lost by the player who cannot move, and infinite plays are lost by $\forall$. A *history-free strategy* for a player $P \in \{\exists, \forall\}$ is a partial function $f : B_P \to B$ such that $f(b)$ is defined whenever $E[b] \neq \emptyset$ and $(b, f(b)) \in E$ in this case. A play $(b_0, b_1, \dots)$ is *played according to* f if $b_{i+1} = f(b_i)$ for all i with $b_i \in B_P$, and f is a *history-free winning strategy* from position $b \in B$ if P wins all plays with initial position b that are played according to f. A position $b' \in B$ is called f-*reachable* from $b \in B$ if there is a play $(b_0, b_1, \dots, b_k)$ that is played according to f and such that $b_0 = b$ and $b_k = b'$.

Reachability games are history-free determined, i.e. from every position b of the game board, one of the players has a history-free winning strategy (this holds for the more general class of parity games [18]). To every tableau system we associate the following reachability game.

Definition 4. The *tableau game* induced by a tableau system (S, R) is reachability game $(B_\exists, B_\forall, E)$ where

- $B_\exists = \{\Psi \subseteq S \mid \Psi \text{ finite}\}$ and $B_\forall = S$
- $E = \{(\Psi, \Gamma) \in B_\exists \times B_\forall \mid \Gamma \in \Psi\} \cup \{(\Gamma, \Psi) \in B_\forall \times B_\exists \mid \Gamma/\Psi \in R\}$.

In other words, $\forall$ plays a tableau rule, and $\exists$ selects one of its conclusions. Note that $\exists$ wins all infinite plays, which correspond to infinite paths in a tableau. As a consequence, $\forall$ has a winning strategy from position Γ in a tableau game, if he can select a tableau rule applicable to Γ so that every conclusion that $\exists$ can possibly choose eventually leads to a tableau axiom, at which point $\forall$ wins.

Proposition 5. *Suppose (S, R) is a tableau system. Then $\Gamma \in S$ has a closed tableau if and only if $\forall$ has a winning strategy in the associated tableau game starting from position Γ.*

We will come back to this general formulation of tableaux in Section 5 and now introduce tableau systems for coalgebraic logics with global assumptions. These are most conveniently formulated in terms of *one-step rules*.

Definition 6. A *one-step tableau rule* over Λ is a tuple $(\Gamma_0, \Gamma_1, \ldots, \Gamma_n)$, written as $\Gamma_0 / \Gamma_1 \ldots \Gamma_n$, where $\Gamma_0 \subseteq (\Lambda \cup \overline{\Lambda})(\mathsf{V} \cup \overline{\mathsf{V}})$ and $\Gamma_i \subseteq \mathsf{V} \cup \overline{\mathsf{V}}$ so that every variable that occurs in the conclusion $\Gamma_1 \ldots \Gamma_n$ also occurs in the premise Γ_0, and every propositional variable occurs at most once in the premise Γ_0.

We can think of one-step rules as a syntactic representation of the inverse image $\gamma^{-1} : \mathcal{P}(TC) \to \mathcal{P}(C)$ of a generic coalgebra map $\gamma : C \to TC$ in that the premise describes a property of successors, whereas the conclusion describes states. The requirement that propositional variables do not occur twice in the premise is for technical convenience, as it later allows us to speak of injective substitutions, rather than substitutions that do not identify elements of the premise. While this rigid format of one-step rules suffices to completely axiomatise all coalgebraic logics [22], they do not accommodate frame conditions like transitivity ($\Box p \to \Box \Box p$) which require separate consideration.

Example 7. One-step rules that axiomatise the logics in Example 1 can be found (in the form of proof rules) in [20,24]. Continuing Example 1, we single out coalition logic, probabilistic modal logic and their combination.

1. A tableau system for coalition logic is induced by the set $\mathsf{R_G}$ that comprises

$$(C_1)\frac{[C_1]p_1, \ldots, [C_n]p_n}{p_1, \ldots, p_n} \qquad (C_2)\frac{[C_1]p_1, \ldots, [C_n]p_n, \overline{[D]}q, \overline{[N]}r_1, \ldots, \overline{[N]}r_m}{p_1, \ldots, p_n, q, r_1, \ldots, r_m}$$

for $n, m \geq 0$ provided that the $C_i \subseteq N$ are pairwise disjoint sets of coalitions, and additionally $C_i \subseteq D$ in (C_2) for all $i = 1, \ldots, n$.

2. The rules $\mathsf{R_D}$ for probabilistic modal logic contain

$$(P)\frac{\langle a_1 \rangle p_1, \ldots, \langle a_n \rangle p_n, [b_1]q_1, \ldots, [b_m]q_m}{\sum_{j=1}^{m} s_j \overline{q}_j - \sum_{i=1}^{n} r_i p_i < k}$$

where $n, m \in \mathbb{N}$ and $r_i, s_j \in \mathbb{N} \setminus \{0\}$ satisfy the side condition $\sum_{i=1}^{n} r_i a_i - \sum_{j=1}^{m} s_j b_j \leq k$ if $n > 0$ and $-\sum_{j=1}^{m} s_j b_j < k$ if $n = 0$. The conclusion of (P) contains all clauses in the disjunctive normal form of the associated $\{0, 1\}$-valued predicate.

3. Games with quantitative uncertainty are described by the rule set $\mathsf{R_{D \circ G}}$

$$\frac{\Gamma_0 \sigma}{\Sigma_1^1 \sigma_1 \quad \ldots \quad \Sigma_{k_1}^1 \sigma_1 \quad \ldots \quad \Sigma_1^n \sigma_n \quad \ldots \quad \Sigma_{k_n}^n \sigma_n}$$

that can be constructed from rules $\Gamma_0 / \Gamma_1 \ldots \Gamma_n \in \mathsf{R_D}$ and $\Sigma_0^i / \Sigma_1^i \ldots \Sigma_{k_i}^i \in \mathsf{R_G}$ ($1 \leq i \leq n$) by injective substitutions $\sigma : V \to (\Lambda_{\mathsf{G}} \cup \overline{\Lambda}_{\mathsf{G}})(\mathsf{V} \cup \overline{\mathsf{V}})$ and $\sigma_1, \ldots, \sigma_n : \mathsf{V} \to \mathsf{V}$ satisfying $\Gamma_i \sigma = \Sigma_0^i \sigma_i$. That is, rules for the combined logic first deconstruct the top-level probabilistic modal operators by means of a probabilistic rule in $\mathsf{R_D}$, and then apply a rule of coalition logic (R_G) to each conclusion.

Given a sequent Δ that represents the global assumptions, every set of one-step rules induces a tableau system that arises by adding Δ to each of the conclusions of modal rules. To reduce the bureaucracy of dealing with propositional rules, we use *skeletal tableaux* where they are subsumed into a single rule schema.

Definition 8. Suppose $\Delta \in \mathcal{S}(\Lambda)$ is a set of global assumptions and R is a set of one-step tableau rules over Λ. The *skeletal system* over R with global assumptions Δ is the tableau system $(\mathcal{S}(\Lambda), \mathsf{S}(\mathsf{R}))$ where $\mathsf{S}(\mathsf{R})$ contains $\Gamma/\mathsf{sat}(\Gamma)$ for all $\Gamma \in \mathcal{S}(\Lambda)$ and all rules $\Gamma_0\sigma, \Gamma'/\Gamma_1\sigma, \Delta \quad \ldots \quad \Gamma_n\sigma, \Delta$ where $\Gamma_0/\Gamma_1 \ldots \Gamma_n \in$ R, $\sigma : \mathsf{V} \to \mathcal{F}(\Lambda)$ is an injective substitution and $\Gamma' \in \mathcal{S}(\Lambda)$ is arbitrary. The operation $\mathsf{sat} : \mathcal{S}(\Gamma) \to \mathcal{S}(\Gamma)$ is called *saturation* and is inductively given by

$$\mathsf{sat}(\Delta') = \{\Delta'\} \qquad \mathsf{sat}(A \vee B, \Gamma) = \mathsf{sat}(A, \Gamma) \cup \mathsf{sat}(B, \Gamma)$$
$$\mathsf{sat}(p, \overline{p}, \Gamma) = \emptyset \qquad \mathsf{sat}(A \wedge B, \Gamma) = \mathsf{sat}(A, B, \Gamma)$$

where $A, B \in \mathcal{F}(\Lambda)$ are formulas, $\Gamma \in \mathcal{S}(\Lambda)$ is a sequent, $p \in \mathsf{V}$ is a propositional variable and $\Delta' \in \mathsf{State}(\Lambda)$ is a state, i.e. contains neither complementary propositional variables nor top-level propositional connectives.

We often leave the underlying set of one-step rules implicit and say that Γ has a closed skeletal tableau with global assumptions Δ, and refer to the induced tableau game as the skeletal game with global assumptions Δ. An easy confluence argument shows that sat is well-defined, i.e. the sequence of steps when computing $\mathsf{sat}(\Gamma)$ is immaterial. Given $\Gamma \in \mathcal{S}(\Lambda)$, the restriction to injective substitutions avoids a possible source of infinity when computing rules that can be applied to Γ. Conclusions are always contained in the closure of its premise and the global assumptions:

Lemma 9. *Suppose $\Gamma, \Delta \in \mathcal{S}(\Lambda)$ and $\Sigma \subseteq \mathsf{cl}(\Gamma, \Delta)$. Then $\Sigma_i \subseteq \mathsf{cl}(\Gamma, \Delta)$ for all $i = 1, \ldots, n$ if $\Sigma/\Sigma_1 \ldots \Sigma_n \in \mathsf{S}(\mathsf{R})$. Moreover, $\mathsf{cl}(\Gamma, \Delta)$ is finite.*

4   Soundness and Completeness

It is evidently impossible to prove even as much as soundness of skeletal tableaux unless the underlying set of one-step rules is suitably linked to the intended (coalgebraic) semantics. This is achieved by imposing coherence conditions that relate premise and conclusions of one-step rules to the underlying (coalgebraic) semantics that can be checked locally, i.e. without reference to models.

Definition 10. Suppose that T is a Λ-structure, X is a set and $\tau : \mathsf{V} \to \mathcal{P}(X)$ is a valuation. The interpretation of a propositional sequent $\Gamma \subseteq \mathsf{V} \cup \overline{\mathsf{V}}$ over X, τ is given by $[\![\Gamma]\!]_{X,\tau} = \bigcap\{\tau(p) \mid p \in \Gamma\} \cap \bigcap\{X \setminus \tau(p) \mid \overline{p} \in \Gamma\} \subseteq X$. Modalised sequents $\Gamma \subseteq (\Lambda \cup \overline{\Lambda})(\mathsf{V} \cup \overline{\mathsf{V}})$ are interpreted as subsets of TX by

$$[\![\Gamma]\!]_{TX,\tau} = \bigcap\{[\![\heartsuit]\!]_X([\![p_1]\!]_{X,\tau}, \ldots, [\![p_n]\!]_{X,\tau}) \mid \heartsuit(p_1, \ldots, p_n) \in \Gamma\}$$

where $p_1, \ldots, p_n \in \mathsf{V} \cup \overline{\mathsf{V}}$ and $\heartsuit \in \Lambda \cup \overline{\Lambda}$.

The announced coherence conditions now take the following form:

Definition 11. Suppose that T is a Λ-structure and R is a set of one-step tableau rules. We say that R is *one-step tableau sound* (resp. *one-step tableau*

complete) with respect to T if, for all $\Gamma \in \mathcal{S}((\Lambda \cup \overline{\Lambda})(V \cup \overline{V}))$, all sets X and valuations $\tau : V \to \mathcal{P}(X)$:

$\llbracket \Gamma \rrbracket_{TX,\tau} \neq \emptyset$ only if (if) for all rules $\Gamma_0/\Gamma_1 \ldots \Gamma_n \in \mathsf{R}$ and all renamings $\sigma : V \to V$ with $\Gamma_0\sigma \subseteq \Gamma$, we have that $\llbracket \Gamma_i\sigma \rrbracket_{X,\tau} \neq \emptyset$ for some $1 \leq i \leq n$.

This means that a rule set is both sound and complete if a modalised sequent is satisfiable iff every one-step rule applicable to it has at least one satisfiable conclusion. Soundness follows immediately from one-step soundness.

Proposition 12 (Soundness). *Suppose $\Gamma, \Delta \in \mathcal{S}(\Lambda)$ and Γ, Δ has a closed tableau in the skeletal system given by R with global assumptions Δ. Then Γ is unsatisfiable in $\mathsf{Mod}(\Delta)$.*

For completeness, we show that the existence of a winning strategy for $\exists$ from Γ, Δ implies that Γ is satisfiable in $\mathsf{Mod}(\Delta)$ via suitable truth and existence lemmas that account for possibly non-monotone modal operators.

Definition 13. If $A \in \mathcal{F}(\Lambda)$ then $\mathsf{spec}(A) = \{\Sigma \in \mathsf{State}(\Lambda) \mid \exists \Sigma' \in \mathsf{sat}(A).\Sigma' \subseteq \Sigma\}$ are the *specified states* of A.

If we think of of $\mathsf{sat}(A)$ as the disjunctive normal form of A, a state $\Sigma \in \mathsf{State}(\Lambda)$ satisfies A if Σ contains all formulas of an element of $\mathsf{sat}(A)$. Thus $\mathsf{spec}(A)$ is the collection of states where A is required to hold, and non-monotonicity forces us to sandwich the interpretation of A between $\mathsf{spec}(A)$ and the complement of $\mathsf{spec}(\overline{A})$ in a syntactic model based on Λ-states. This will be a consequence of coherence, introduced next.

Definition 14 (Coherence). Suppose $W \subseteq \mathsf{State}(\Lambda)$. A coalgebra structure $w : W \to TW$ is *coherent*, if

$$w(\Gamma) \in \llbracket \heartsuit \rrbracket_W(X_1, \ldots, X_n)$$

whenever $\heartsuit(A_1, \ldots, A_n) \in \Gamma$ and $W \cap \mathsf{spec}(A_i) \subseteq X_i \subseteq W \setminus \mathsf{spec}(A_i)$ for all $i = 1, \ldots, n$. A valuation $\pi : V \to \mathcal{P}(W)$ is *coherent* if $\mathsf{spec}(p) \cap W \subseteq \pi(p) \subseteq W \setminus \mathsf{spec}(\overline{p})$ for all $p \in V$. Finally, a T-model (W, w, π) is coherent, if both (W, w) and π are coherent.

Given that $\exists$ has a winning strategy in the skeletal system, the next lemma asserts the existence of a coherent structure, that we will use later to prove satisfiability, given that the underlying set of one-step rules admits contraction:

Definition 15. A set R of one-step rules *admits contraction* if, for all rules $\Gamma_0/\Gamma_1, \ldots, \Gamma_n \in \mathsf{R}$ and all renamings $\sigma : V \to V$ we can find a rule $\Sigma_0/\Sigma_1 \ldots \Sigma_k \in \mathsf{R}$ and an injective renaming $\rho : V \to V$ such that $\Sigma_0\rho \subseteq \Gamma_0\sigma$ and, for all $j = 1, \ldots, k$ there exists $1 \leq i \leq n$ such that $\Sigma_j\rho \supseteq \Gamma_i\sigma$.

That is to say, an application of contraction to the premise of a modal rule (via a substitution that identifies propositional variables) can always be replaced by a different rule for which this is not the case, and moreover the conclusions of this rule are even harder to satisfy. The existence lemma now takes the following form:

Lemma 16 (Existence Lemma). *Suppose that* R *is one-step tableau complete and admits contraction. If* ∃ *has a winning strategy* f *in the game induced by the skeletal tableau system with global assumptions* Δ, *then there exists a coherent coalgebra structure* $w : W \to TW$ *on the set of states that are* f-*reachable from* Γ, Δ.

Given coherence, we can now prove:

Lemma 17 (Truth Lemma). *Suppose that* $M = (W, w, \pi)$ *is coherent. Then* $\mathsf{spec}(A) \cap W \subseteq [\![A]\!]_M \subseteq W \setminus \mathsf{spec}(\overline{A})$ *for all* $A \in \mathcal{F}(\Lambda)$.

Completeness is now an immediate consequence of the Truth Lemma and the Existence Lemma.

Proposition 18. *If* ∃ *has a winning strategy from* $\Gamma, \Delta \in \mathsf{State}(\Lambda)$ *in the skeletal game with global assumptions* Δ, *then* Γ *is satisfiable in* $\mathsf{Mod}(\Delta)$.

In summary, we have the following result that lays the semantical foundation of the algorithms in the following section.

Theorem 19. *Suppose that* R *is one-step sound and complete with respect to a* Λ-*structure* T. *The following are equivalent for* $\Gamma, \Delta \in \mathcal{S}(\Lambda)$:

1. Γ *is satisfiable in* $\mathsf{Mod}(\Delta)$
2. Γ *does not have a closed skeletal tableau with global assumptions* Δ
3. ∃ *has a winning strategy in the skeletal tableau game with global assumptions* Δ *from position* Γ, Δ.

As a by-product of this theorem, we obtain admissibility of cut (via semantical completeness) and the small model property, which is implicit in the proof of Proposition 18. We remark that the rules given in Example 7 are both one-step sound and complete [24].

5   Global Caching

In this section, we show that global caching [10] is applicable to coalgebraic logics, and give a feasible algorithm to decide satisfiability of a sequent Γ over a set Δ of global assumptions. The idea behind global caching is very simple: every sequent is expanded at most once, and sequents are not expanded unnecessarily. We begin our discussion of global caching in the context of a generic tableau system that we then subsequently specialise to coalgebraic logics to prove optimality.

Definition 20. Suppose that (S, R) is a tableau system. A *caching graph* for (S, R) is a quintuple $G = (A, U, E, X, L)$ where $A, U, E, X \subseteq S$ and $L \subseteq S \times \mathcal{P}(S) \cup \mathcal{P}(S) \times S$. The set $\mathsf{supp}(G) = A \cup U \cup E$ is called the *support* of G. A caching graph $G = (A, U, E, X, L)$ is *expanded* if

$$L = \bigcup_{\Gamma \in \mathsf{supp}(G)} \{(\Gamma, \Psi) \mid \Gamma/\Psi \in R\} \cup \{(\Psi, \Sigma) \mid \Gamma/\Psi \in R, \Sigma \in \Psi\}$$

and $\Psi \subseteq \mathsf{supp}(G) \cup X$ for all $\Gamma/\Psi \in R$ with $\Gamma \in \mathsf{supp}(G)$.

In other words, a caching graph is a concrete data structure that not only stores sequents, but also links every sequent in its support to each conclusion of a rule applicable to it, and every conclusion to each of its elements. We think of A as the set of winning positions of $\forall$ in the associated tableau game, and similarly E represents $\exists$'s winning positions. The set L (of links) represents the collection of all rules that can be applied to a sequent in the support of a caching graph. The status of sequents in U is undecided, but they are expanded, in the sense that L contains all rules that are applicable to elements in U. The conclusions of such rules that are not already contained in the support of a caching graph are collected in the set X, the set of sequents that are still unexpanded.

Definition 21. We define two transition relations $\rightarrow_E$ ("expand") and $\rightarrow_P$ ("propagate") on caching graphs. We put $(A, U, E, X, L) \rightarrow_E (A', U', E', X', L')$ if $A' = A$, $E' = E$ and there exists $\Gamma \in X$ such that

$$U' = U \cup \{\Gamma\} \qquad X' = X \cup \left(\bigcup\{\Psi \mid \Gamma/\Psi \in R\}\right) \setminus (A' \cup E' \cup U')$$
$$L' = L \cup \{(\Psi, \Sigma) \mid \Sigma \in \Psi, \Gamma/\Psi \in \mathsf{R}\} \cup \{(\Gamma, \Psi) \mid \Gamma/\Psi \in \mathsf{R}\}.$$

Moreover, $(A, U, E, X, L) \rightarrow_P (A', U', E', X', L')$ in case $X' = X$, $L = L'$ and

$$A' = A \cup \mu M^L \qquad E' = E \cup \nu W^L \qquad U' = U \setminus (A' \cup E')$$

where μM^L and νW^L are, respectively, the least and greatest fixpoints of the operators $W^L : \mathcal{P}(U) \rightarrow \mathcal{P}(U)$ and $M^L : \mathcal{P}(U) \rightarrow \mathcal{P}(U)$ given by

$$W^L(X) = \{\Gamma \in U \mid \forall(\Gamma, \Psi) \in L.\, \exists(\Psi, \Sigma) \in L.\, \Sigma \in X \cup E\}$$
$$M^L(X) = \{\Gamma \in U \mid \exists(\Gamma, \Psi) \in L.\, \forall(\Psi, \Sigma) \in L.\, \Sigma \in X \cup A\}$$

for $X \subseteq U$. We write $\rightarrow_{PE}$ for the union of $\rightarrow_P$ and $\rightarrow_E$ and $\rightarrow_{PE}^*$ for its reflexive-transitive closure.

In an expansion step, an unexpanded sequent $\Gamma \in X$ is chosen, all rules that are applicable to Γ are recorded in L and Γ is moved to U. To ensure that the ensuing caching graph is expanded, new conclusions that arise from expanding Γ that are not yet contained in the support are added to X, the set of unexpanded sequents. The (deterministic) propagation steps update the set of winning positions of $\forall$ and $\exists$ in the tableau game. For $\forall$, this amounts to recursively adding a sequent to A if we can apply a tableau rule whose conclusions are contained in A – this is achieved by the least fixpoint construction above. The winning positions of $\exists$ are computed by means of a greatest fixpoint, and we extend E by the largest set Ψ of sequents so that for every tableau rule applied to Ψ, at least one conclusion is contained in $\Psi \cup E$, in other words, the construction of a closed tableau for elements of $\Psi \cup E$ is impossible, provided that E already enjoys this property.

   If we interleave expansion and propagation steps until all sequents are expanded ($X = \emptyset$), we update the winning positions in a final propagation step, and all elements not known to be either satisfiable or unsatisfiable (the elements

of U) are declared to be satisfiable, since any tableau rule applied to a sequent in U necessarily has at least one conclusion in $E \cup U$, since this sequent would otherwise have been moved to A by propagation.

Lemma 22. *Suppose that (S, R) is a tableau system and (A, U, E, X, L) is an expanded caching graph for (S, R) for which all $\Gamma \in A$ but none of the $\Gamma \in E$, have a closed tableau. If $(A, U, E, X, L) \to_{PE}^* (A', U', E', \emptyset, L) \to_P (\hat{A}, \hat{U}, \hat{E}, \emptyset, \hat{L})$ then all $\Gamma \in \hat{A}$, but none of the $\Gamma \in \hat{U} \cup \hat{E}$ have a closed tableau.*

Correctness of global caching induces the following (nondeterministic) algorithm.

Algorithm 23. Decide whether $\Gamma \in S$ has a closed tableau.

1. initialise: Put $G = (A, U, E, X, L)$ where
   - $A = E = \emptyset$, $U = \{\Gamma\}$ and $X = \bigcup\{\Psi \mid \Gamma/\Psi \in R\}$
   - $L = \{(\Gamma, \Psi) \mid \Gamma/\Psi \in R\} \cup \{(\Psi, \Sigma) \mid \Sigma \in \Psi, \Gamma/\Psi \in R\}$
2. while $(X \neq \emptyset)$ do
   (a) choose G' with $G \to_E G'$ and put $G := G'$
   (b) (optional)
       - find G' with $G \to_P G'$ and put $G := G'$
       - return "yes" if $\Gamma \in A$ and "no" if $\Gamma \in E$
3. find G' with $G \to_P G'$ and put $G := G'$
4. return "yes" if $\Gamma \in E$ and "no" otherwise.

Correctness of this algorithm follows from Lemma 22, and termination is clear as every expand-transition adds one sequent to the support of a caching graph, as long as (S, R) is finite. Since transitions between caching graphs preserve the property that all elements in A, but none of the elements in E, have a closed tableau, we may in fact terminate earlier if we find the initial sequent in $E \cup A$.

Theorem 24. *Suppose that (S, R) is a finite tableau system and $\Gamma \in S$. Then every execution terminates in at most $3 \cdot |S| + 1$ steps and returns "yes" if and only if Γ has a closed tableau.*

We remark that – although Algorithm 23 is non-deterministic – we just need to check *one* particular execution, i.e. there is no inherent non-determinism, but room for heuristics. We now specialise Algorithm 23 to the case of coalgebraic logics and establish an (optimal) EXPTIME bound. In the general (coalgebraic) setting, we can not expect that satisfiability of a sequent Γ under global assumptions Δ is even decidable unless we make additional assumptions about the underlying set of one-step rules (which may in general be non-recursive). However, all rule sets that we are aware of, in particular the rule sets that completely axiomatise the logics introduced in Example 1 satisfy an additional assumption: the set of conclusions of a rule can be polynomially encoded in terms of the size of the premise. This was used in [24] and applied to fixpoint logics in [4].

To be precise, we assume that the underlying similarity type Λ is equipped with a size measure $s : \Lambda \to \mathbb{N}$ and measure the size of a formula A in terms of the number of subformulas of A adding $s(\heartsuit)$ for every occurrence of a modal operator $\heartsuit$ or $\overline{\heartsuit}$ in A. For the logics in our running example, we code numbers

in binary, that is $\langle p/q \rangle = [p/q] = \lceil \log_2 p \rceil + \lceil \log_2 q \rceil$ for probabilistic modal logic and $s([C]) = 1$ for operators of coalition logic, as the overall set of agents is fixed. The definition of size is extended to sequents by $\mathsf{size}(\Gamma) = \sum_{A \in \Gamma} \mathsf{size}(A)$ for $\Gamma \in \mathcal{S}(\Lambda)$. In particular, the size of the closure of a sequent is polynomially bounded.

Lemma 25. *Suppose $\Gamma \in \mathcal{S}(\Lambda)$. Then $\mathsf{size}(\Sigma) \leq 2\mathsf{size}(\Gamma)^2$ for all $\Sigma \subseteq \mathsf{cl}(\Gamma)$.*

The notion of size allows us to formulate polynomial encodings.

Definition 26. A set R of tableau rules is *exponentially tractable*, if there exists an alphabet Σ and two functions $f : \mathcal{S}(\Lambda) \to \mathcal{P}(\Sigma^*)$ and $g : \Sigma^* \to \mathcal{P}(\mathcal{S}(\Lambda))$ together with a polynomial p such that $|x| \leq p(\mathsf{size}(\Gamma))$ for all $x \in f(\Gamma)$, $\mathsf{size}(\Delta) \leq p(|y|)$ for all $\Delta \in g(y)$, so that, for $\Gamma \in \mathcal{S}(\Lambda)$,

$$\{g(x) \mid x \in f(\Gamma_0)\} = \{\{\Gamma_1, \ldots, \Gamma_n\} \mid \Gamma_0/\Gamma_1, \ldots, \Gamma_n \in R\}$$

and both relations $x \in f(\Gamma)$ and $\Gamma \in g(x)$ are decidable in EXPTIME.

Tractability of the set $\mathsf{S}(R)$ of tableau rules follows from tractability of the substitution instances of rules in R, as both propositional rules and saturation can be encoded easily. At this point, we use the fact that the modal rules in the skeletal system are defined in terms of injective substitutions as otherwise a rule can be generated through infinitely many substitution instances.

Lemma 27. *Suppose R is a set of one-step rules. Then $\mathsf{S}(R)$ is exponentially tractable iff the set $\{\Gamma_0\sigma/\Gamma_1\sigma, \ldots, \Gamma_n\sigma \mid \Gamma_0/\Gamma_1, \ldots, \Gamma_n \in R, \sigma : V \to \mathcal{F}(\Lambda)$ injective$\}$ of substituted one-step rules is exponentially tractable.*

Tractability ensures that we can encode the data on which Algorithm 23 operates as strings of at most polynomial length in terms of the initial sequent and the global assumptions.

Lemma 28. *Suppose that R is exponentially tractable and $\Gamma, \Delta \in \mathcal{S}(\Lambda)$. Then every $\Psi \subseteq \mathsf{cl}(\Gamma, \Delta)$ that appears as the conclusion of a rule $\Sigma/\Psi \in \mathsf{S}(R)$ for which $\Sigma \in \mathsf{cl}(\Gamma, \Delta)$ can can be encoded as a string of polynomial length (in $\mathsf{size}(\Gamma, \Delta)$). Under this coding, the relations $\{(\Sigma_0, \{\Sigma_1, \ldots, \Sigma_n\}) \mid \Sigma_0/\Sigma_1 \ldots \Sigma_n \in \mathsf{S}(R), \Sigma_0 \subseteq \mathsf{cl}(\Gamma, \Delta)$ and $\{((\{\Sigma_1, \ldots, \Sigma_n\}, \Sigma_i) \mid 1 \leq i \leq n, \exists \Sigma_0 \subseteq \mathsf{cl}(\Gamma/\Delta). \Sigma_0/\Sigma_1 \ldots \Sigma_n \in R\}$ are decidable in exponential time.*

Tractability of rule sets guarantees that Algorithm 23 runs in EXPTIME.

Theorem 29. *Suppose that Λ is a modal similarity type and T is a Λ-structure. If R is a one-step tableau sound and complete set of one-step rules that admits contraction, then Algorithm 23 decides satisfiability of Γ in $\mathsf{Mod}(\Delta)$ in at most exponential time (w.r.t. $\mathsf{size}(\Gamma, \Delta)$), if R is exponentially tractable.*

In our examples, the situation is as follows:

Example 30. All logics mentioned in Example 1 can be captured by a one-step sound and complete rule set that is exponentially tractable [20,24], and we briefly discuss the case for those logics that we have singled out in Example 1 and Example 7.

1. For coalition logic, no coding is needed at all, as the size of sequents decreases when we move from the premise to the conclusion of a rule, and EXPTIME decidability of the rule set is clear.

2. For the conclusions of the rule schema that axiomatises probabilistic modal logic, we take the linear inequality $\sum_{j=1}^{m} s_j \overline{q}_j - \sum_{i=1}^{n} r_i p_i < k$ itself as a code for the associated set of conclusions. Tractability was shown in [24] using the fact that the (binary) size of the coefficients r_i can be polynomially bounded.

3. For rule sets that arise as combinations, tractability follows from tractability of the individual components, which is most conveniently made explicit in a multi-sorted setting [23].

As the modal logic K can be encoded into all logics mentioned in Example 1 with the exception of classical and monotone modal logic, global satisfiability for these logics is EXPTIME hard, and hence optimality of Algorithm 23.

6   Conclusions

We have given a sound and complete tableau calculus for coalgebraic modal logics in the presence of global assumptions. Based on the completeness of the tableau calculus, we have then described a concrete tableau algorithm to decide satisfiability in presence of global assumptions, based on global caching. In particular, this algorithm meets the (in nearly all cases optimal) EXPTIME bound, while avoiding the unnecessary overhead of computing least fixpoints naively. This showcases not only the wide applicability of global caching, but also demonstrates that automated reasoning with coalgebraic logics in the presence of global assumptions is also in practice not (much) harder than for modal logics with an underlying relational semantics. We have demonstrated by means of examples, that the general (coalgebraic) framework specialises to a large class of modal logics, and have thus not only described the first tableau algorithm for deciding e.g. probabilistic modal logic, but an algorithm that is also worst-case optimal that we plan to implement and evaluate experimentally in the future.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
2. Chellas, B.: Modal Logic, Cambridge (1980)
3. Cîrstea, C.: A compositional approach to defining logics for coalgebras. Theoret. Comput. Sci. 327, 45–69 (2004)
4. Cîrstea, C., Kupke, C., Pattinson, D.: EXPTIME tableaux for the coalgebraic μ-calculus. In: Grädel, E., Kahle, R. (eds.) CSL 2009. LNCS, vol. 5771, pp. 179–193. Springer, Heidelberg (2009)
5. Cîrstea, C., Pattinson, D.: Modular proof systems for coalgebraic logics. Theor. Comp. Sci. 388, 83–108 (2007)
6. Donini, F., Massacci, F.: Exptime tableaux for $\mathcal{ALC}$. Artif. Intell. 124(1), 87–138 (2000)

7. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. The MIT Press, Cambridge (1995)
8. Fine, K.: In so many possible worlds. Notre Dame J. Formal Logic 13, 516–520 (1972)
9. Goré, R.: Tableau methods for modal and temporal logics. In: D'Agostino, et al. (eds.) Handbook of Tableau Methods, pp. 297–396. Kluwer, Dordrecht (1999)
10. Goré, R., Nguyen, L.: EXPTIME tableaux for $\mathcal{ALC}$ using sound global caching. In: Calvanese, D., Franconi, E., Haarslev, V., Lembo, D., Motik, B., Turhan, A., Tessaris, S. (eds.) Proc. Description Logics 2007. CEUR Workshop Proceedings. CEUR-WS.org, vol. 250 (2007)
11. Goré, R., Nguyen, L.: EXPTIME tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In: Olivetti, N. (ed.) TABLEAUX 2007. LNCS (LNAI), vol. 4548, pp. 133–148. Springer, Heidelberg (2007)
12. Goré, R., Postniece, L.: An experimental evaluation of global caching for $\mathcal{ALC}$ (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 299–305. Springer, Heidelberg (2008)
13. Goré, R., Widmann, F.: Sound global state caching for $\mathcal{ALC}$ with inverse roles. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 205–219. Springer, Heidelberg (2009)
14. Haarslev, V., Möller, R.: RACER system description. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 701–705. Springer, Heidelberg (2001)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. The MIT Press, Cambridge (2000)
16. Heifetz, A., Mongin, P.: Probabilistic logic for type spaces. Games and Economic Behavior 35, 31–53 (2001)
17. Horrocks, I., Patel-Schneider, P.: Optimising description logic subsumption. J. Logic Comput. 9, 267–293 (1999)
18. Mazala, R.: Infinite games. In: Grädel, E., Thomas, W., Wilke, T. (eds.) Automata, Logics, and Infinite Games. LNCS, vol. 2500, pp. 23–42. Springer, Heidelberg (2002)
19. Pattinson, D.: Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. Theoret. Comput. Sci. 309, 177–193 (2003)
20. Pattinson, D., Schröder, L.: Admissibility of cut in coalgebraic logics. Electr. Notes Theor. Comput. Sci. 203(5), 221–241 (2008)
21. Pauly, M.: A modal logic for coalitional power in games. J. Logic Comput. 12(1), 149–166 (2002)
22. Schröder, L.: A finite model construction for coalgebraic modal logic. J. Log. Algebr. Program. 73(1-2), 97–110 (2007)
23. Schröder, L., Pattinson, D.: Compositional algorithms for heterogeneous modal logics. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 459–471. Springer, Heidelberg (2007)
24. Schröder, L., Pattinson, D.: PSPACE bounds for rank-1 modal logics. ACM Transactions on Computational Logics 10(2) (2009)
25. Schröder, L., Pattinson, D., Kupke, C.: Nominals for everyone. In: Boutilier, C. (ed.) Proc. IJCAI 2009, pp. 917–922 (2009)
26. Stirling, C.: Modal and temporal logics. In: Handbook of logic in computer science. Background: computational structures, vol. 2, pp. 477–563. Oxford University Press, Inc., Oxford (1992)
27. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 292–297. Springer, Heidelberg (2006)