

Modal Logics are Coalgebraic¹

CORINA CÎRSTEĂ¹, ALEXANDER KURZ², DIRK PATTINSON³, LUTZ SCHRÖDER^{4,*}
AND YDE VENEMA⁵

¹ School of Electronics and Computer Science, University of Southampton,
Highfield, Southampton SO17 1BJ, UK

² Department of Computer Science, University of Leicester, University Road, Leicester, LE1 7RH, UK

³ Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK

⁴ DFKI Bremen and Department of Computer Science, Universität Bremen, Cartesium,
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

⁵ Institute for Logic, Language and Computation, University of Amsterdam, Plantage Muidergracht 24, 1018
TV Amsterdam, The Netherlands

*Corresponding author: lutz.schroeder@dfki.de

Applications of modal logics are abundant in computer science, and a large number of structurally different modal logics have been successfully employed in a diverse spectrum of application contexts. Coalgebraic semantics, on the other hand, provides a uniform and encompassing view on the large variety of specific logics used in particular domains. The coalgebraic approach is generic and compositional: tools and techniques simultaneously apply to a large class of application areas and can, moreover, be combined in a modular way. In particular, this facilitates a pick-and-choose approach to domain-specific formalisms, applicable across the entire scope of application areas, leading to generic software tools that are easier to design, to implement and to maintain. This paper substantiates the authors' firm belief that the systematic exploitation of the coalgebraic nature of modal logic will not only have impact on the field of modal logic itself but also lead to significant progress in a number of areas within computer science, such as knowledge representation and concurrency/mobility.

Keywords: modal logic; coalgebra; knowledge representation; concurrency; compositionality;
automata theory

Received 31 October 2008

Handling editor: Vladimiro Sassone

1. INTRODUCTION

Logics of all colours, shapes and sizes have traditionally played a central role in computer science, and in fact the standard design of the modern computer itself is based on a particular brand of logic—Boolean propositional logic. In a rough classification along a tradeoff between expressiveness and computational tractability, one finds simple logics such as propositional logic, which despite being NP-complete can nowadays be efficiently handled using modern SAT-solvers, at the one end, and very expressive higher-order logics, which, however, typically offer only a low degree of automation at the other end of the spectrum. Between the two extremes there is a large variety of logics that target application-specific sweet spots between the two conflicting goals of expressiveness and tractability. One large

class of such logics is the vast and growing family of *modal* logics, which are characterised by having operators that qualify formulas as holding in a certain way, e.g. ‘necessarily’, ‘in the future’, ‘everywhere’, ‘probably’, ‘as everyone knows’ or ‘normally’.

Applications of modal logics are abundant in computer science and related disciplines. A multitude of different formalisms, including hybrid and description logics that also axiomatise the notion of individuals, have been studied in a variety of application contexts. Apart from classical applications in the field of concurrent [1], mobile [2] and probabilistic systems [3], modal logics play a central role in artificial intelligence, e.g. in the context of reasoning with uncertainty [4], non-monotonic reasoning [5] and – in particular, in their description logic incarnation – in the field of knowledge representation and ontologies [6]. Modal logics are employed to reason about games [7] and coalitional power in multi-agent

¹A preliminary version of this paper was presented at the BCS08 Visions of Computer Science Conference, held on September 22–24, 2008.

systems [8]. In economics, they have been used to describe probabilistic information of economic agents [9], whereas, e.g. deontic logic, the logic of obligation and permission originally studied in philosophy [10], is being used to model contracts in multi-agent systems. While we do not pretend to work specifically on one of the UKCRC Grand Challenges [11], it is interesting to note that every one of the current challenges involves modal logic in some form or other, variously referring, e.g. to knowledge representation, logics of agents or logics for concurrency. Today, *Kripke frames* [12] constitute the most popular semantics of modal logics, and the class of logics that can be interpreted over Kripke frames forms the class of the so-called *normal* modal logics. However, many modal logics of interest, and indeed the majority of the logics listed above, fail to be normal and hence their semantics necessarily goes beyond Kripke frames as we illustrate below.

As an example in knowledge representation, imagine you want to connect two knowledge bases that describe different aspects of transport patterns. The first knowledge base describes travel activities in relation to individuals. The second knowledge base circumstantiates the volume of traffic on public and individual transport in temporal terms. Both knowledge bases will naturally use a plethora of different primitives to represent information. In a seemingly simple piece of knowledge such as ‘Normally, the likelihood of road congestion is smaller on weekends’, one implicitly makes use of default logics (‘normally’), probabilistic reasoning (‘the likelihood’) and temporal knowledge (‘weekends’) under a quantitative regime (‘smaller’). If we link this with both knowledge bases that we seek to combine, we will moreover encounter spatial reasoning (to cater for distances), epistemic principles (knowledge of individuals) and deontic constructs (obligations that arise, e.g. through social norms) adorned with constructs that formalise the joint behaviour of individual agents. Depending on the specifics of knowledge that we seek to combine, many more ways of logically expressing the relationship between the entities under scrutiny may be needed.

Central to the study of logics, in general, are a number of recurring questions, including completeness (‘are all valid statements derivable?’), decidability (‘is the logic amenable to automated reasoning?’) and complexity (‘what resources are required to mechanise the logic?’). Given the diversity of the modal logic family on the one hand and the uniformity of the problems arising in meta-theory and implementation on the other hand, it is clearly desirable to have a common framework that captures the syntax and semantics of the mentioned modal logics and many others, existing or yet to be developed, in a uniform way and at the same time allows for a common meta-theory and generic mechanised reasoning tools. The unifying ingredient that makes all this possible is found in the semantics, however varied shapes the latter assumes in specific cases. It turns out that the common denominator is a view of models as dynamic or reactive systems in a very general sense, which

is formally captured by regarding them as *coalgebras*. We illustrate this view by a quick glance at a few examples.

Kripke frames. The traditional textbook semantics of the modal logic K and its extensions is usually presented in relational form: a Kripke frame is a pair (W, R) where W is a set of worlds and $R \subseteq W \times W$ is an accessibility relation. The interpretation of the accessibility relation varies according to the application domain—e.g. in concurrency, worlds are seen as states of a system, and R as representing their potential temporal evolution, in knowledge representation, worlds are regarded as individuals and R is a relationship between individuals such as parthood, and in epistemic reasoning, R captures epistemic alternatives. In a Kripke frame, a world w satisfies a modal formula of the form $\Box\phi$, read, e.g. ‘necessarily ϕ ’ or ‘in all successor states, ϕ ’, if all worlds w' accessible from w , e.g. all successor states of w or all parts of w , satisfy ϕ . Now Kripke frames are easily seen to be in one-to-one correspondence with transition maps $\rho : W \rightarrow \mathcal{P}(W)$ that assign the set of successors $\{w' \in W \mid wRw'\}$ to each world w , where $\mathcal{P}(W)$ is the powerset of W . From this point of view, if ϕ is a modal formula with extension $\llbracket\phi\rrbracket \subseteq W$, then a world w satisfies $\Box\phi$ ($w \models \Box\phi$) if and only if the *successor set* of w is contained in $\llbracket\phi\rrbracket$. In other words,

$$w \models \Box\phi \iff \rho(w) \in \{B \in \mathcal{P}(W) \mid B \subseteq \llbracket\phi\rrbracket\}.$$

Probabilistic transition systems. One step up from Kripke frames, probabilistic transition systems extend the notion of transition with quantitative uncertainty. In its simplest form [3,9], probabilistic modal logic (PML) extends propositional logic with operators L_p (‘at least p ’) where $p \in [0, 1]$ is a rational number. PML is interpreted over probabilistic transition systems (W, P) where W is a set of worlds and $P = (P_w)_{w \in W}$ is a family of probability distributions on W , indexed by the set of worlds. Again, according to the application context P is variously interpreted as, e.g. governing the evolution of a black-box dynamic system or the beliefs of an agent. Correspondingly, the intended reading of $L_p\phi$ is ‘ ϕ holds with probability at least p in the next state’, or from the perspective of quantitative uncertainty ‘in their present state of belief, agents assign probability at least p to ϕ ’. We may view probabilistic transition systems as transition maps $\rho : W \rightarrow \mathcal{D}(W)$, where $\mathcal{D}(W)$ denotes the set of discrete probability distributions on W ; i.e. ρ assigns to each world w a *successor distribution* $\rho(w) \in \mathcal{D}(W)$. The main difference with Kripke frames lies in the fact that collections of successors are now *structured*: moving from frames to probabilistic models entails a shift from successor sets to distributions. The classical interpretation of probabilistic formulas, i.e.

$$w \models L_p\phi \iff P_w(\llbracket\phi\rrbracket) \geq p$$

can now be re-phrased in terms of *successor distributions*,

$$w \models L_p\phi \iff \rho(w) \in \{\mu \in \mathcal{D}(W) \mid \mu(\llbracket\phi\rrbracket) \geq p\},$$

i.e. a state w satisfies $L_p\phi$ if its successor distribution assigns probability at least p to the event $\llbracket\phi\rrbracket \subseteq W$. Again, the quintessential nature of a probabilistic modal operator manifests itself as providing a passage from properties of states to properties of successor distributions.

Conditional logic. The language of conditional logic [13] extends propositional logic with a binary connective that we write as \Rightarrow using infix notation. The operator \Rightarrow represents a non-monotonic conditional, whose intended readings include, e.g. default implication ‘if ϕ then normally ψ ’ and the conditional version ‘ ψ holds under the condition ϕ ’. The ensuing logics are often used in knowledge representation to deal with the non-monotonic nature of information. Note that the operator \Rightarrow is, in general, distinct from material implication \rightarrow . For example, the validity of $\phi \Rightarrow \psi$ does, in general, *not* imply that of $\phi \wedge \phi' \Rightarrow \psi$. Thus, conditional logic is a modalised version of default logic, where defaults may be nested. Conditional logic is usually interpreted in so-called (standard) conditional frames (or selection function frames), that is, pairs (W, f) where W is a set of worlds and $f : W \times \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ is a selection function that assigns a proposition $f(w, A) \subseteq W$ to each world w and condition $A \subseteq W$. Alternatively, we may view conditional frames as transition maps $\rho : W \rightarrow (\mathcal{P}(W) \rightarrow \mathcal{P}(W))$ that map each world $w \in W$ to a function $\rho(w) : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ from conditions to propositions, both formalised as subsets of W . That is, successor structures of worlds are now (selection) functions of type $\mathcal{P}(W) \rightarrow \mathcal{P}(W)$.

In a conditional frame (W, f) , the standard semantics of the conditional operator takes the form

$$w \models \phi \Rightarrow \psi \iff f(w, \llbracket\phi\rrbracket) \subseteq \llbracket\psi\rrbracket.$$

Again, the semantics of the conditional operator can be understood as specifying a property of successor structures, i.e. selection functions: we have

$$w \models \phi \Rightarrow \psi \iff \rho(w) \in \{f : \mathcal{P}(W) \rightarrow \mathcal{P}(W) \mid f(\llbracket\phi\rrbracket) \subseteq \llbracket\psi\rrbracket\},$$

where $\llbracket\phi\rrbracket$ and $\llbracket\psi\rrbracket$ are again the truth-sets of ϕ and ψ , respectively. We note that, as in the other examples above, the semantics of the conditional operator is embodied by an operation, in this case binary, that maps predicates on the set of worlds to predicates on the set of structured successors, in this case selection functions. Other examples include, e.g. spatial transition systems where binary modalities are used to decompose concurrent processes.

The pattern that becomes apparent in the above examples is that we may typically see the semantic transition structures over which modal logics are interpreted as maps of the type $W \rightarrow T(W)$, where T is some operator on sets, technically a *functor*, and to be thought of as a form of parametrised datatype, which determines the branching type of the transitions; and

moreover the interpretation of modal operators is embodied in terms of *predicate liftings* that transform predicates on the set W of worlds into predicates on the set $T(W)$ of successor structures. As a map $W \rightarrow T(W)$ is just what is technically termed a *coalgebra* for T , this is the starting point of *coalgebraic modal logic*: we can study modal logics at the right level of generality by parametrising their semantics in the choice of a functor T and a suitable set of predicate liftings.

Given that, as already illustrated by the above examples, the level of generality of the coalgebraic approach is quite high indeed, one may wonder whether it is actually the *right* level of generality as claimed, i.e. whether one can indeed develop a powerful generic theory rather than just gather lots of examples under a common umbrella. It is one of the aims of this paper to substantiate the claim that this is really the case by highlighting some of the achievements of coalgebraic modal logic to date. Indeed the scope of the established meta-theory of coalgebraic modal logic reaches surprisingly far. Besides basic meta-logical properties such as generic criteria for soundness and completeness [14], it includes, e.g. some new classical results of modal logic such as duality and ultrafilter extensions [15], but also computational aspects such as generic finite and shallow model constructions with ensuing decidability and complexity results [16,17], as well as proof-theoretic results such as cut elimination and interpolation [18]. The parametricity over system types afforded by the coalgebraic approach may be combined with reasoning principles of varying strength, leading to, e.g. generic fixed-point logics [19] and generic hybrid logics [20]. The coalgebraic framework is not only parametric, but also modular w.r.t. combinations of logics [21,22]. Moreover, the parametricity extends also to the underlying form of propositional logics and thus includes, e.g. logics over nominal frameworks [23].

Having thus emphasised the suitability of the coalgebraic approach as a universal framework for modal logic, we set out to develop our vision of a unified description logic with universally applicable automatic reasoning support. In this ideal future, workers in knowledge representation, verification of concurrent systems and many other areas will put together the domain-specific modal logic suited for their problem domain in a pick-and-choose approach, and immediately obtain efficient and scalable reasoning tools by instantiating the generic coalgebra-based reasoning framework.

2. A COOK’S TOUR OF COALGEBRAIC LOGICS

The above examples support the claim that modal logics can be interpreted over general coalgebraic models. The idea that underlies the whole body of research into coalgebraic logics is parametricity: the methods and tools of coalgebraic modal logic apply to coalgebras of *any type*. In other words, the abstract theory speaks about T -coalgebras $(C, \gamma : C \rightarrow TC)$ without ever assuming a concrete definition of T . Applications

to concrete logics then simply fall out by instantiating the type functor T accordingly. In this way, one obtains, e.g. algorithms for reasoning with coalgebraic logics that uniformly cover all the previously given examples. A surprisingly large body of results can be obtained at this high level of generality, indicating that coalgebras relate to modal logic at precisely the right level of abstraction. We illustrate this by the following overview of the main results, tools and techniques in the field of coalgebraic logics.

2.1. Compositionality of coalgebraic logics

As illustrated above, the coalgebraic paradigm is a very flexible means of describing dynamic behaviour. For example, we have seen that probabilistic transition systems can be described as coalgebras of type $C \rightarrow \mathcal{D}(C)$, where $\mathcal{D}(C)$ is the set of (discrete) probability distributions over C . Similarly, *game frames* [8] can be phrased coalgebraically: a game frame amounts to a coalgebra of type $C \rightarrow \mathcal{G}(C)$ for

$$\mathcal{G}(C) = \{(S_1, \dots, S_n, f) \mid S_i \neq \emptyset, i = 1, \dots, n; f : S_1 \times \dots \times S_n \rightarrow C\},$$

where informally the S_i are sets of strategies of the individual agents $1, \dots, n$ and f is an *outcome function* that produces a new position on the game board given the choice of individual strategies. Games with uncertainty can now be modelled by a simple combination: rather than a new position as in the game with certainty, the outcome of a choice of strategies is now an uncertain position, i.e. a probability distribution over positions on the game board. That is to say, a model for games with uncertainty is a coalgebra of type $C \rightarrow \mathcal{G}(\mathcal{D}(C))$ —a position on the game board and choice of strategies yields a distribution over successor states. In a similar way, simple Segala systems (called probabilistic transition systems in [24]) can be modelled as coalgebras of type

$$C \rightarrow (\mathcal{P}(\mathcal{D}(C)))^A,$$

where $(-)^A$ represents function space: to every state $c \in C$ and every label $a \in A$, one associates the non-deterministic choice of a probability distribution over successor states which accounts for a probabilistic interaction with the environment. Similar combinations abound in many areas of computing: in the area of knowledge representation, one may, for instance, consider combinations of non-monotonic conditionals and quantitative uncertainty, and logics for mobile systems are most conveniently addressed using logical primitives for communication and the generation of new channels at run time.

The power of the coalgebraic approach comes into play by associating a logical description to every component that can be composed to obtain a description of combined systems in the spirit of Abramsky's domain theory in logical form [25]. In more detail, we have for every type functor T on sets:

- a *one-step syntax*, consisting of a set of modal operators with arities that are used to describe possible next-state behaviours;
- a *one-step semantics* for each such modal operator, in the shape of a choice of predicate liftings as explained in the introduction; and
- a set of *one-step rules* axiomatising the one-step observable behaviours of states of T -coalgebras.

The great advantage of the above approach is its compositionality, which allows deriving composite modal logics in parallel with the *structure* of composite type functors such as $\mathcal{P}(\mathcal{D}(-))^A$ above: it suffices to identify a number of 'basic' logical features, such as non-determinism, strategies or probability, and a number of ways of combining such features (these two ingredients can even be seen as instances of the same concept [22]), and equip these constructions with the above-mentioned semantic and proof-theoretic structure. One thus obtains multi-sorted modal logics that mix the involved modal operators under a typing discipline that reflects the structure of the underlying systems, such as the PML for simple Segala systems advocated in [24], which distinguishes probabilistic and non-deterministic formulas. The compositional nature of this approach allows properties such as soundness, completeness, expressiveness (w.r.t. characterising similarity [21]) and decidability, as well as upper complexity bounds (Section 1.4), to be derived in a modular fashion: all one needs to show is that the basic building blocks satisfy certain conditions at the one-step level, such as one-step completeness or one-step expressiveness [21, 22]. Instantiated, e.g. with the PML of simple Segala systems [24], the coalgebraic approach provides in particular

- a modularised proof of the expressiveness of this logic w.r.t. the standard notion of bisimilarity,
- a modular way of deriving a *sound and complete* proof system, and
- a modular satisfiability algorithm (see Section 2.4) that witnesses a PSPACE upper bound.

While the first item is just an alternative proof of a known result in probabilistic process algebra, the latter two have, to our knowledge, first been stated and proved in a coalgebraic setting [21, 22]. So far, only a limited amount of interaction between the different components can be accounted for [26] but more general results are anticipated.

A variation of the main technique used to derive expressiveness results w.r.t. bisimilarity can be used to obtain logics that characterise (weaker) notions of similarity [27]. In addition to the three ingredients mentioned earlier, a fourth one identifies a particular notion of similarity between coalgebras by specifying a *one-step simulation relation* between one-step behaviours. Expressiveness of modal logics w.r.t. such notions of similarity is, as before, derived modularly using conditions at the one-step level, in this case relating the chosen one-step syntax and one-step semantics with the

one-step simulation relation under consideration. For simple Segala systems, for instance, this technique can be used to obtain a logic that characterises the *probabilistic similarity* of [24]—a similarity relation that accounts for the probabilistic nature of schedulers, by simulating (non-deterministic) transitions by convex combinations of (again non-deterministic) transitions.

2.2. Logics for nominal calculi

In the previous section, we have argued that the coalgebraic approach is parametric and compositional in the notion of *behaviour*, which is conveniently abstracted into a *type functor*. But what constitutes the semantical base whose properties we are to observe? It turns out that there is a plethora of different semantical structures which form the underlying basis over which we analyse behaviour.

This is reflected in the different ways of building logics and models. For example, on the logical side, the variation starts already at the level of the underlying propositional logic (classical, intuitionistic, positive, substructural, etc.). Semantically, we may start with basic entities that we call states and then add structure, beyond the coalgebraic transition structure, in many different ways, e.g. algebraic operations (reflecting the way process calculi allow us to construct new states from given ones), topological structure (capturing that not all but only ‘open’ sets of states are observable) and, most importantly, recursion. Like in the ‘plain’ coalgebraic setting, we wish to find a systematic and modular way of linking logics to structured models.

The semantics of the logics we want to use as the basic building blocks are best described via *Stone duality* [28] and its relatives. The idea is to describe a logic (syntax and proof system) as a category of algebras (such as Boolean algebras, Heyting algebras and distributive lattices) and the models as topological spaces (the topology corresponds to the fact that, e.g. finitary Boolean logic is not strong enough to reason about arbitrary infinite unions and intersections). Duality then amounts to a (dual) equivalence between a category of algebras and a category of topological spaces (such as Boolean algebras and Stone spaces). The two layers, duality and modular combination, can be brought together in a fruitful way to account for the computational structure over which observations are made. A famous example, and indeed the ancestor of this approach, is Abramsky’s programme of domain theory in logical form, which extends Stone dualities to the solution of recursive domain equations [25,29].

We now proceed to illustrate how coalgebraic techniques and a suitable choice of base category can be combined to derive in a systematic way a logic for the π -calculus [30], which characterises strong late bisimilarity and accounts for name binding; see [23] for a full treatment. This showcases yet another orthogonal aspect of genericity in the coalgebraic framework. To capture the semantics of the π -calculus, we need to interpret processes not in the standard set-theoretic universe

(as before) but in a variation where each set is equipped with the action of a group of renamings that affect the so-called free names of a process. More formally, we use the presheaf category \mathbf{N} of functors from finite sets (of channel names) with injective renamings to a certain category of domains (representing observable behaviour in the presence of recursion). As with the coalgebraic approach, in general, the type of possible one-step behaviours of processes is captured by a type functor, but now taking sets with names, i.e. the category \mathbf{N} as the semantic base. For the π -calculus, the following functor on \mathbf{N} was introduced independently in [31,32]:

$$Pi(X) = \mathcal{P}(X + N \times X^N + N \times (N \times X) + N \times \delta X).$$

As before \mathcal{P} is for non-determinism, $+$ is binary choice; N is a constant for the set of names, δ allows creating a fresh name and $(\cdot)^N$ inputs a (possibly fresh) name. We read Pi as follows. The possible one-step behaviours of a process are non-deterministic (due to \mathcal{P}) and may be one of the following alternatives: A silent step (the X component), an input of a name (X^N) over a channel (N), the output of a free name over a channel (due to $N \times N \times X$) or the allocation and sending of a fresh name (the $N \times \delta(X)$ -part).

In the same way in that modularity was invoked to obtain sound, complete and fully abstract logics for set-based models, the same machinery also applies uniformly in this different, more complex setting: One has to describe the logical structure of the semantical base category, in this case \mathbf{N} , and of the functor Pi . This is not hard: The case of the basic functors \mathcal{P} , $+$, \times has been treated in Abramsky [25,33] and the axiomatisations can be reused without further modification (providing, e.g. the usual modalities \Box and \Diamond); allocation of new names gives rise to a modal operator $[\nu b]$ for name creation. The result is a new fully abstract, sound and complete modal logic for the π -calculus. Again, the power lies in the modularity: the same techniques give rise to out-of-the-box logics both for other calculi and other forms of equivalences, in particular, including ones yet to be developed.

2.3. Automata and fixpoint logics

We now proceed to give another example of the unifying power of the coalgebraic perspective, now in a classical area of computer science: *automata theory*. More specifically, we consider the theory of finite automata as devices for classifying infinite, or possibly infinite, objects. This branch of theoretical computer science has found important applications in areas of computer science where one investigates the ongoing behaviour of non-terminating programs such as operating systems. As an example we mention the automata-based verification method of *model checking* [34].

This research also has a long and strong theoretical tradition, in which an extensive body of knowledge has been developed, with a number of landmark results. Many of these link the field to neighbouring areas such as logic and game theory, see [35]

for an overview. The outstanding example here is of course Rabin's decidability theorem [36] for the monadic second-order logic of trees; to mention a more recent result, Janin and Walukiewicz [37] identified the modal μ -calculus as the bisimulation invariant fragment of the monadic second-order logic of labelled transition systems.

The automata that we refer to come in many kinds and shapes, and can be classified according to a number of criteria, including

- (i) the *kind of objects* the automata operate on: words, trees, transition systems, ...
- (ii) the *degree of interaction* encoded in the automaton transition map: is it deterministic, non-deterministic or alternating?
- (iii) the *acceptance condition* of the automaton: using a Büchi, Muller or parity condition?

Note that the objects that automata operate on very often *are* coalgebras, so it should come as no surprise that coalgebraic notions will play a role here. Interestingly, many of the key results in automata theory *involve* a comparison of automata that fall in different classes according to the second and third criterion above, but *apply* separately to each class of automata as given by the first criterion. This applies for instance to various closure properties of the class of recognisable languages, and to the fact that alternating parity automata can be transformed into equivalent non-deterministic ones: these results hold for word, tree and graph automata alike. We claim that it will increase our understanding of automata theory if we see these results as manifestations of a more general, 'universal' automata theory which is essentially *coalgebraic* in nature.

The key idea underlying the coalgebraic perspective on automata theory is that *acceptance* (of an object by an automaton) generalises *bisimilarity* (of two objects). As a simple example of this, we consider tree automata.

Here, by a *tree* we mean an infinite binary tree whose nodes are labelled with elements of a set C of colours. In other words, a tree is a map $\tau : \{0, 1\}^* \rightarrow C$ that labels each finite bitstring w with a colour $\tau(w) \in C$. These maps may be viewed as *coalgebras* of type \mathbf{B}_C , where \mathbf{B}_C is the functor taking a set S to the set $\mathbf{B}_C(S) := C \times S \times S$. Arbitrary coalgebras for this functor are pairs (S, σ) , where $\sigma : S \rightarrow \mathbf{B}_C S$ associates with each state s a colour $\sigma_C(s)$ and a left and right successor $\sigma_0(s)$ and $\sigma_1(s)$, respectively. It is easy to see that C -coloured binary trees are of this shape.

As mentioned earlier, each type of coalgebra comes with a notion of bisimilarity or behavioural indistinguishability. This notion can be nicely captured by an infinite *bisimilarity game* of two players that we call \exists (Éloise) and \forall (Abélard). The bisimilarity game is played on *pointed* \mathbf{B}_C -coalgebras (X, ξ, x_0) , that is, coalgebras (X, ξ) augmented with an initial state $x_0 \in X$. Given two pointed \mathbf{B}_C -coalgebras $\mathbb{A} = (A, \alpha, a_0)$ and $\mathbb{S} = (S, \sigma, s_0)$, matches of the game start at the position (a_0, s_0) . Whenever a match arrives at some basic position

$(a, s) \in A \times S$, \exists *looses* if a and s have a different colour. If on the other hand $\alpha_C(a) = \sigma_C(s)$, then the match continues with opponent \forall choosing a direction (left or right). According to \forall 's choice, the next position of the match is either $(\alpha_0(a), \sigma_0(s))$ or $(\alpha_1(a), \sigma_1(s))$. \exists may have no opportunity to win finite matches of this game, and is declared the winner of every infinite match. This game (which is essentially a one-player game due to the simplicity of the functor \mathbf{B}_C) characterises bisimilarity in the sense that a_0 and s_0 are bisimilar if and only if every match of the game is won by \exists .

In order to bring automata into the picture, we think of one structure, \mathbb{A} , from now on called the *automaton*, as *classifying* the other structure, \mathbb{S} , from now on called the *coalgebra*. This conceptual breach of the symmetry between the two structures allows us to make some modifications to the structure of the automaton. Most importantly, we give \exists a bigger role in the game by replacing the transition map $\alpha : A \rightarrow \mathbf{B}_C A$ with a *non-deterministic* variant $\Delta : A \rightarrow \mathcal{P}(\mathbf{B}_C A)$ where \mathcal{P} is powerset. The game is modified accordingly: Instead of fixing the coalgebraic reading of a state $a \in A$ as the element $\alpha(a) \in \mathbf{B}_C A$, we allow \exists to dynamically *pick* such a reading from the set $\Delta(a) \subseteq \mathbf{B}_C A$, whenever the state a pops up during the match. Similar modifications of the bisimilarity game allow us to add an acceptance condition that allows \forall to win some infinite matches, and a requirement that the automaton be finite.

The key point is that we have turned the bisimilarity game into an *acceptance game*, and that when we take the coalgebra \mathbb{S} to be a binary tree, then this acceptance game is exactly the standard one that we know from classical automata theory. For readers eager to check this: our presentation of the transition structure $\Delta : A \rightarrow \mathcal{P}(C \times A \times A)$ of an automaton is *isomorphic* to the more standard presentation $\Delta : A \times C \rightarrow \mathcal{P}(A \times A)$.

The coalgebraic perspective on automata theory that we just described applies to set coalgebras of *arbitrary* type T [38]. In addition, in order to specify and reason about *ongoing* coalgebraic behaviour, one may extend the coalgebraic logics mentioned in the earlier parts of this note with *fixpoint* operators, obtaining coalgebraic generalisations of the modal μ -calculus [37]. Perhaps of more significance, under some mild condition on the functor T (namely, that it preserves so-called weak pullbacks), most of the important results in the theory of tree and graph automata can in fact be proved at this level of generality [39]. As examples we mention the following results:

- reduction of alternating to non-deterministic automata
- various closure properties of recognisable languages
- decidability and finite model property of an associated coalgebraic fixpoint logic.

We believe that the coalgebraic perspective has a lot to offer in the area of automata theory and its applications.

To start with the latter, within the framework sketched in this paper every (new) type of coalgebra comes equipped with a modular arsenal of concepts (such as bisimilarity) and tools (such as coalgebraic modal logic) that are well understood,

backed up by a strong mathematical theory and ready to be applied. The point that we argue in this section is that as an integral part of this ‘coalgebraic package’ we obtain a natural notion of automaton and a corresponding modal fixpoint language for specifying *ongoing* coalgebraic behaviour, both with nice computational properties.

Second, on the more theoretical side, coalgebra as an organisational principle helps to understand some of the non-trivial parts of automata theory through identifying general principles shared by automata of various coalgebraic type. For instance, in the analysis of [39] the transformation of a non-deterministic equivalent of an alternating automaton is the combination of a coalgebraic automaton construction which works for arbitrary types and the classical Safra construction which works for stream automata (ω -automata) only. The specific coalgebraic contribution is that it separates the dynamics (inside the coalgebra functor) from the combinatorics (encoded in the acceptance condition of automata).

It should be noted that all the above constructions are naturally of a modular nature [40] and so truly faithful to the coalgebraic paradigm.

2.4. Generic algorithms and reasoners for modal logics

Given the extremely broad scope of coalgebraic modal logic, one of the most important and also the most surprising aspects of its emerging meta-theory is that it allows for a generic algorithmic treatment, including both the proof of tight generic complexity bounds and actual implementations. The corresponding theoretical results take the shape of well-defined and easily verified criteria that a logic must satisfy in order to have a decidable satisfiability problem or even to be of low computational complexity. Technically, these criteria reduce properties of the logic to much simpler properties of the underlying coalgebraic structure as given by the choice of a functor and associated predicate liftings; generic complexity bounds are then witnessed by generic algorithms that are parametrised by subroutines dealing with logic-specific aspects of *local* satisfiability.

Early results of this type have been limited to the so-called rank-1 logics, characterised semantically as imposing only local restrictions (such as seriality) rather than global restrictions (e.g. transitivity) on the underlying models. Advances into the generic study of logics outside rank 1 are being made currently [26,41]. The first widely applicable criterion [16] made use of a filtration-based finite model construction to obtain a generic decidability criterion and ensuing EXPTIME or NEXPTIME upper bounds, establishing a (universally valid) finite model property for coalgebraic logics along the way. A strongly improved version of this result [41] applies to various logics outside rank 1, and in particular has led to the design of a description logic with qualified number restrictions that is able to handle parthood across several layers of decomposition while keeping decidability; this contrasts sharply with existing

approaches using transitive parthood, which quickly lead to undecidability [42].

Within the realm of rank-1 logics (unlike in the case of higher-rank logics as treated in [41]), the generic exponential time bounds of [16] generally do not match the actual complexity of individual logics, which is typically PSPACE. It does, however, turn out that these bounds *can* be matched by generic algorithms, and indeed the latter have been used to determine the exact complexity of a number of logics where this was previously unknown. The generic algorithms available can be broadly grouped into two classes: syntax-oriented algorithms that connect proof search with shallow models, and semantically minded algorithms that reduce global satisfiability to local satisfiability using a direct construction of shallow models. An algorithm of the former type, based on the central notion of resolution closed rule sets [17], captures the known tight PSPACE upper bounds for such diverse logics as K (or KD), coalition logic, graded modal logic and PML, and moreover has led (simultaneously with [43]) to a new PSPACE upper bound for majority logic [44]. This algorithm, in modularised form, is implemented in the prototypic coalgebraic logic satisfiability solver CoLoSS [45]. Alternative semantics-based algorithms [46] cover, e.g. complex logics such as Presburger modal logic [43] and the modal logic of probability of [47], which both feature linear inequalities between formula weights and whose axiomatisations have so far evaded attempts at harnessing their resolution closure, a prerequisite for the application of the syntax-oriented approach. Moreover, the semantics-based approach has been used to establish new PSPACE upper bounds, e.g. for Elgesem’s logic of agency [48]. Not only does this already cover an impressive collection of modal logics. What is more is that—faithful to the coalgebraic paradigm—the reasoning principles for individual logics can be combined and induce reasoners for composite logics [22].

Ongoing externally funded research projects are aimed at extending the generic algorithmic framework, in particular, to more general logics including fixpoints. Recent results moreover include a generic algorithmic treatment of hybrid logics, which provide the necessary facilities for reasoning about individual states in the shape of nominals, as featured in many description logics, and satisfaction operators. These generic results yield new tight upper complexity bounds PSPACE for a wide variety of hybrid logics including probabilistic hybrid logic, conditional hybrid logic, hybrid coalition logic and (an extension of) the description logic $\mathcal{ALCH}OQ$ [20]. One important goal is to provide optimised implementations of the generic algorithms, thus paving the way for efficient, universal and modular coalgebraic reasoners.

3. VISIONS

We believe that, in the medium term, coalgebraic logics will contribute to many areas, mainly because of their flexibility and

compositional nature. One of the areas is modal logic itself, but the potential is far greater. We outline some visions in the areas of knowledge representation and reasoning about concurrent and mobile systems.

3.1. Modal logic

Imagine you are exploring a newly designed or discovered modal logic. Chances are that the logic is not normal, and therefore not amenable to Kripke semantics. The questions on the agenda could be computational (is the logic decidable? how difficult is the decision problem?), of a modelling character (what is the natural semantic domain for the logic? is it complete? is it expressive?) or possibly pertaining to its meta-theory (does it have the interpolation property? is it canonical?).

The standard approach to questions of the above type is to set up a semantics for the logic under scrutiny and to try and adapt known constructions from other settings to the new semantic domain in order to shed light on the properties of interest—a laborious process that leads to results that are specific to the logic under consideration.

Or, why not simply consult the rich and expanding literature on coalgebraic semantics and instantiate off-the-shelf results to obtain the properties in question? And even in case ready-made results do not fit (yet), investigating your logic in the generic and abstract coalgebraic framework has benefits that go far beyond the concrete logic at hand and will later help those who ask similar questions about their own favourite logics.

In summary, we expect the future of modal logics to be coalgebraic. Judging from the rapid growth of the body of literature in coalgebraic logics, and the already impressive number of logics that fall within the coalgebraic paradigm, coalgebras will be the standard semantics of (non-normal) modal logics in years to come. We believe that the coalgebraic view brings about a number of significant advantages:

Genericity. Both theoretical results and practical tools based on the coalgebraic framework are *by construction* applicable to a large class of modal logics.

Compositionality. Not only do different logics naturally co-exist in the coalgebraic framework, but the coalgebraic approach moreover allows for a natural and seamless integration of logics. Coalgebra thus provides a convenient setting for compositionality results, and under suitable assumptions caters for the modular combination of reasoning principles [22].

Adaptability. Application areas are dynamic rather than static, and the generic and compositional approach of coalgebraic logics allows for an easy integration of new requirements in particular application domains. Coalgebraic modal logic caters for both semantics-centred approaches, where one needs to design a logic to describe given semantic phenomena,

and syntax-centred ones [49], where one needs semantic underpinnings for the analysis of given means of expression.

As new and domain-specific modal logics emerge steadily, for example, in the fields of knowledge representation and concurrency/mobility as outlined below, the time is ripe for a more unified approach that will largely eliminate the need for tinkering with the particulars of specifically given logics. What we see before us in the medium term is a unitised coalgebraic foundation that covers not only the logics of today, but also all those logics that will be developed tomorrow to harness the ever increasing complexity of the modern digital society.

3.2. AI and knowledge representation

Recall the example from the introduction involving transport patterns. We have seen that the expressive means potentially required in traffic-related formal knowledge bases can be cast as instances of coalgebraic modal logic. What next? Of course we would like to reason about the information that is represented by the amalgamation of both knowledge bases. This involves modularity: we need to combine reasoning principles (to capture the interaction between different logical constructs), we need to combine knowledge and data and we need to synthesise algorithms that allow deriving valid conclusions from the amalgamated knowledge base automatically.

But reasoning may not be enough. We might want to employ mechanisms of knowledge discovery, e.g. with the aim of supporting transport planning or to provide decision support for network managers. This leads us into the area of machine learning, and we would like to employ mechanisms for knowledge discovery in this specific setting.

Of course, reasoning about transport patterns is only one example, and we are faced with similar tasks, most prominently in the area of medicine (with a comparatively large body of knowledge formalised, e.g. in the GALEN ontology [50]), but also in other areas like civil engineering, law and life sciences. The diversity of the form of knowledge to be formalised, which stems from the different application areas, calls for modular and compositional systems that allow representing and reasoning about combinations of many different facets of knowledge—and if we were granted a wish, they should moreover allow for induction of hypotheses.

Can such systems be achieved? We think that coalgebraic techniques will have both a natural and a central place in the field of knowledge representation in the years to come. The pick-and-choose approach to modal, hybrid and description logics allows us to combine logical features and reasoning principles in a modular fashion. The modularity goes beyond the blueprint stage as coalgebraic techniques also facilitate the automated construction of reasoning engines based on a combination of logical features. Of course, more research is needed, and the application of coalgebraic techniques in

knowledge representation, in particular, calls for progress in two specific areas. The first concern is the development of a generic theory of learning, or induction of hypotheses, to be able tap into today's distributed knowledge bases. Equally important is the creation of distributed reasoning engines that support the modular paradigm to harness the generally large volume of data computationally.

In summary, coalgebraic techniques have a lot to offer for knowledge representation, first and foremost a dramatic increase in expressive power that stems from incorporating and combining different logics and reasoning principles that are relevant for representing knowledge. We envisage that this potential will be realised in the medium term in the form of tangible tool support for (coalgebraic) reasoning about knowledge: In a few years time, we will be able to specify the relations of distributed knowledge bases and employ distributed, compositional reasoning to provide, e.g. decision support for traffic network planning, based on the integration of a large body of knowledge over the web.

3.3. Concurrency and mobility

Imagine you are to design the IT infrastructure of a security-relevant operation, say an airport, that is currently being planned. Users will want to attach to the infrastructure using a plethora of mobile devices, from handheld computers to mobile phones. Clearly the overall architecture will have to be location aware up to the point of distances between individuals, reflect different security clearances and cater for availability of finite resources in terms of personnel and hardware. It will need to accommodate quantitative uncertainty to cope with e.g. hardware failure or human error. One extremely important factor is time, and the ability to incorporate both soft deadlines (related e.g. to average baggage throughput) and hard deadlines, dictated by security requirements.

Part of the challenge of this task is to provide quality assurances regarding both functional and non-functional requirements while the requirements, and consequently also the layout of the system, are still subject to change. Of course, our first task in this enterprise is to build a model that caters for all the aspects indicated above, quite possibly in the form of a dedicated calculus, followed by the design of a formalism that allows expressing the varied requirements of the overall architecture. In other words, we are to provide both a *model* and a *specification* of a system, whose design and layout may still change. The task of validating this specification against the model then takes the form of logical reasoning, and will be achieved by a combination of both automated theorem proving and model checking. Modal logics in general, and coalgebraic logics in particular, are well positioned to meet this challenge, as they combine a high degree of expressive power with good computational properties, and the coalgebraic framework in particular provides the common ground that allows integrating the heterogeneous and varied

aspects of the model. Clearly, compositionality is the key ingredient without which an endeavour like the above would not be feasible. One needs to combine both reasoning principles and their underlying semantics. We need a modular way to combine location aware and spatial logics with probabilistic aspects, notions of resource, security, concurrency and mobility, all in a framework that honours time. The emphasis needs to be placed on flexibility, as new requirements may emerge and the model is subject to constant change.

The above scenario discusses just one (very concrete) example of tasks that lie ahead of us in the future. Very similar problems manifest themselves in three of the nine grand challenges in computing research: we mention global and ubiquitous computing, dependable systems evolution and scalable ubiquitous computing systems. We believe that coalgebraic modelling, and associated coalgebraic logics, are very well positioned to bring about significant advances in global computing at large. First, *the coalgebraic model is flexible*. That is, it can incorporate many different types of behaviour and interaction, e.g. location awareness, mobility and quantitative uncertainty, to name but a few. Second, *the coalgebraic model is compositional*: both on the logical and the semantical level it allows us to combine computational features and reason about their interaction. Third, *the coalgebraic model is uniform*, i.e. all computational aspects of the model share the same meta-theory. This, in particular, leads to software tools that are easier to design, to maintain and to implement. Finally, *the coalgebraic model is compatible* in the sense that it subsumes nearly all existing formal notions of state-based system as special cases.

It is precisely the large number and diverse nature of networked devices as well as the possibly disastrous consequences of failure that call for an integrated and compositional approach to modelling and verification as provided by the coalgebraic paradigm. As it stands, this presents two research challenges. To fully maximise the benefit of the coalgebraic approach, more investment both at the theoretical and practical level is needed. On the theoretical side, a more compartmentalised analysis of mobility primitives and their interactions needs to be provided, together with a compositionality layer that specifically addresses the needs of ubiquitous computing. On the practical side, this needs to be matched with adequate and modular tool support, specifically concerning automated reasoning and model checking.

In summary, we envisage that the coalgebraic approach will play a leading role in the area of formal models of ubiquitous computation in the medium and long term. In a world where we rely on increasingly complex and self-managing networks to an unprecedented level, quality assurance in the sense of mathematical proof will be indispensable soon in a large number of areas, ranging from intelligent sensor networks to medical smartcards that store and encode highly confidential information.

4. CONCLUSIONS

Coalgebraic logic is an immensely rich field with a multitude of applications, of which we have just described a tiny fraction. We believe that the coalgebraic approach views computational phenomena at precisely the right level of abstraction: the modelling language is extremely flexible, while many of the associated logics are still decidable in reasonable complexity classes. In conjunction with the built-in compositionality of the method at large, one obtains an extremely powerful framework for the analysis of phenomena in Computer Science and Artificial Intelligence. Apart from foundational research that extends the arsenal of logical machinery, the most important challenge is the further development of the tool support to create optimised and efficient reasoners. Our philosophy here is pick-and-choose, and the genericity of the coalgebraic approach will manifest itself in a modular and compositional reasoning framework that will be used in areas ranging from the verification of mobile systems to knowledge representation and artificial intelligence.

FUNDING

Work of the third author was partially supported by EPSRC grant number EP/F031173/1. Work of the fourth author forms part of the DFG project *Generic Algorithms and Complexity Bounds in Coalgebraic Modal Logic* (SCHR 1118/5-1). Research of the fifth author has been made possible by VICI grant 639.073.501 of the Netherlands Organization for Scientific Research (NWO).

REFERENCES

- [1] Hennessy, M. and Milner, R. (1985) Algebraic laws for non-determinism and concurrency. *J. ACM*, **32**, 137–161.
- [2] Milner, R., Parrow, J. and Walker, D. (1993) Modal logics for mobile processes. *Theoret. Comput. Sci.*, **114**, 149–171.
- [3] Larsen, K. and Skou, A. (1991) Bisimulation through probabilistic testing. *Inform. Comput.*, **94**, 1–28.
- [4] Halpern, J. (2003) *Reasoning About Uncertainty*. MIT Press, Cambridge, MA.
- [5] Friedman, N. and Halpern, J.Y. (1994) Conditional Logics for Belief Change. In Hayes-Roth, B. and Korf, R. (eds), *Artificial Intelligence, AAAI 1994*, pp. 915–921. AAAI Press, Menlo Park, CA.
- [6] Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P. (eds) (2003) *The Description Logic Handbook*. Cambridge University Press.
- [7] Parikh, R. (1985) The Logic of Games and Its Applications. In Karpinski, M. and van Leeuwen, J. (eds), *Topics in the Theory of Computation*, Annals of Discrete Mathematics, Vol. 24. Elsevier, Amsterdam.
- [8] Pauly, M. (2002) A modal logic for coalitional power in games. *J. Logic Comput.*, **12**, 149–166.
- [9] Heifetz, A. and Mongin, P. (2001) Probabilistic logic for type spaces. *Games Econom. Behav.*, **35**, 31–53.
- [10] von Wright, G.H. (1951) Deontic logic. *Mind*, **60**, 1–15.
- [11] Hoare, T. and Milner, R. (eds) (2004) *Grand Challenges in Computing—Research*. The British Computer Society.
- [12] Kripke, S. (1963) Semantical considerations on modal logic. *Acta Philos. Fenn.*, **24**, 83–94.
- [13] Chellas, B. (1980) *Modal Logic*. Cambridge University Press.
- [14] Pattinson, D. (2003) Coalgebraic modal logic: soundness, completeness and decidability of local consequence. *Theoret. Comput. Sci.*, **309**, 177–193.
- [15] Kupke, C., Kurz, A. and Pattinson, D. (2005) Ultrafilter Extensions for Coalgebras. In Fiadeiro, J.L., Harman, N., Roggenbach, M. and Rutten, J. (eds), *Algebra and Coalgebra in Computer Science, CALCO 2005*, Lecture Notes in Computer Science, Vol. 3629, pp. 263–277. Springer, Berlin.
- [16] Schröder, L. (2007) A finite model construction for coalgebraic modal logic. *J. Log. Algebr. Program.*, **73**, 97–110.
- [17] Schröder, L. and Pattinson, D. (2006) PSPACE Bounds for Rank-1 Modal Logics. In Alur, R. (ed), *Logic in Computer Science, LICS 2006*, pp. 231–240. IEEE Press, Los Alamitos, CA. Extended version to appear in *ACM Trans. Comput. Logic*; preprint available as e-print arXiv:0706.4044.
- [18] Pattinson, D. and Schröder, L. (2008) Admissibility of Cut in Coalgebraic Logics. In Adamek, J. (ed), *Coalgebraic Methods in Computer Science, CMCS 2008*, Electronic Notes in Theoretical Computer Science, Vol. 203. Elsevier, Amsterdam.
- [19] Kupke, C. and Venema, Y. (2005) Closure Properties of Coalgebra Automata. In Panangaden, P. (ed), *Logic in Computer Science, LICS 2005*, pp. 199–208. IEEE Press, Los Alamitos, CA.
- [20] Myers, R., Pattinson, D. and Schröder, L. (2009) Coalgebraic Hybrid Logic. In Foundations of Software Science and Computation Structures, FOSSACS 2009, Lecture Notes in Computer Science. Springer, Berlin (in press).
- [21] Cîrstea, C. and Pattinson, D. (2007) Modular construction of complete coalgebraic logics. *Theoret. Comput. Sci.*, **388**, 83–108.
- [22] Schröder, L. and Pattinson, D. (2007) Modular Algorithms for Heterogeneous Modal Logics. In Arge, L., Tarlecki, A., and Cachin, C. (eds), *Automata, Languages and Programming, ICALP 07*, Lecture Notes in Computer Science, Vol. 4596, pp. 459–471. Springer, Berlin.
- [23] Bonsangue, M. and Kurz, A. (2007) π -Calculus in Logical Form. In Ong, L. (ed), *Logic in Computer Science, LICS 2007*, pp. 303–312. IEEE Press, Los Alamitos, CA.
- [24] Jonsson, B., Larsen, K. and Yi, W. (2001) Probabilistic Extensions of Process Algebras. In Bergstra, J., Ponse, A., and Smolka, S. (eds), *Handbook of Process Algebra*, pp. 685–710. Elsevier, Amsterdam.
- [25] Abramsky, S. (1991) Domain theory in logical form. *Ann. Pure Appl. Logic*, **51**, 1–77.
- [26] Pattinson, D. and Schröder, L. (2008) Beyond Rank 1: Algebraic Semantics and Finite Models for Coalgebraic Logics. In Amadio, R. (ed), *Foundations of Software Science and Computation Structures, FOSSACS 2008*, Lecture Notes in Computer Science, Vol. 4962, pp. 66–80. Springer, Berlin.
- [27] Cîrstea, C. (2006) A modular approach to defining and characterising notions of simulation. *Inform. Comput.*, **204**, 468–502.

- [28] Johnstone, P. (1993) *Stone Spaces*, Cambridge Studies in Advanced Mathematics, Vol. 3. Cambridge University Press.
- [29] Abramsky, S. and Jung, A. (1994) Domain Theory. In Abramsky, S., Gabbay, D. and Maibaum, T. (eds), *Handbook of Logic in Computer Science*. Oxford University Press.
- [30] Milner, R. (1999) *Communicating and Mobile Systems: The π -Calculus*. Cambridge University Press.
- [31] Fiore, M., Moggi, E. and Sangiorgi, D. (1996) A Fully-Abstract Model for the π -Calculus (extended abstract). In Clarke, E. (ed), *Logic in Computer Science, LICS 1996*, pp. 43–54. IEEE Press, Los Alamitos, CA.
- [32] Stark, I. (1996) A Fully Abstract Domain Model for the π -Calculus. In Clarke, E. (ed), *Logic in Computer Science, LICS 1996*, pp. 36–42. IEEE Press, Los Alamitos, CA.
- [33] Abramsky, S. (1991) A domain equation for bisimulation. *Inform. Comput.*, **92**, 161–218.
- [34] Clarke, E., Grumberg, O. and Peled, D. (eds) (2000) *Model Checking*. MIT Press, Cambridge, MA.
- [35] Grädel, E., Thomas, W. and Wilke, T. (eds) (2002) *Automata, Logic, and Infinite Games*, Lecture Notes in Computer Science, Vol. 2500. Springer, Berlin.
- [36] Rabin, M. (1969) Decidability of second-order theories and automata on infinite trees. *Trans. AMS*, **141**, 1–35.
- [37] Janin, D. and Walukiewicz, I. (1995) Automata for the Modal μ -Calculus and Related Results. Wiedermann, J. and Hájek, P. (eds), *Mathematical Foundations of Computer Science, MFCS 1995*, Lecture Notes in Computer Science, Vol. 969, pp. 552–562. Springer.
- [38] Venema, Y. (2006) Automata and fixed point logic: a coalgebraic perspective. *Inform. Comput.*, **204**, 637–678.
- [39] Kupke, C. and Venema, Y. (2008) Coalgebraic automata theory: basic results. *Log. Methods Comput. Sci.*, **4**(4:10), pp. 1–43.
- [40] Cîrstea, C. and Sadrzadeh, M. (2008) Modular Games for Coalgebraic Fixed Point Logics. In Adámek, J. and Kupke, C. (eds), *Coalgebraic Methods in Computer Science, CMCS 2008*, Electronic Notes in Theoretical Computer Science, Vol. 203, pp. 65–86. Elsevier, Amsterdam.
- [41] Schröder, L. and Pattinson, D. (2008) How Many Toes Do I Have? Parthood and Number Restrictions in Description Logics. In Brewka, G. and Lang, J. (eds), *Principles of Knowledge Representation and Reasoning, KR 2008*, pp. 307–218. AAAI Press, Menlo Park, CA.
- [42] Horrocks, I., Sattler, U. and Tobies, S. (1999) Practical Reasoning for Expressive Description Logics. In Ganzinger, H., McAllester, D., and Voronkov, A. (eds), *Logic for Programming and Automated Reasoning, LPAR 1999*, Lecture Notes in Computer Science, Vol. 1705, pp. 161–180. Springer, Berlin.
- [43] Demri, S. and Lugiez, D. (2006) Presburger Modal Logic is Only PSPACE-Complete. In Furbach, U. and Shankar, N. (eds), *Automated Reasoning, IJCAR 06*, Lecture Notes in Artificial Intelligence, Vol. 4130, pp. 541–556. Springer, Berlin.
- [44] Pacuit, E. and Salame, S. (2004) Majority Logic. In Dubois, D., Welty, C. and Williams, M.-A. (eds), *Principles of Knowledge Representation and Reasoning, KR 2004*, pp. 598–605. AAAI Press, Menlo Park, CA.
- [45] Calin, G., Myers, R., Pattinson, D. and Schröder, L. (2008) CoLoSS: The Coalgebraic Logic Satisfiability Solver (System Description). In Areces, C. and Demri, S. (eds), *Methods for Modalities, M4M-5* Electronic Notes in Theoretical Computer Science, Elsevier, Amsterdam, to appear.
- [46] Schröder, L. and Pattinson, D. (2008) Shallow Models for Non-Iterative Modal Logics. In Dengel, A., Berns, K., Breuel, T., Bomarius, F. and Roth-Berghofer, T. (eds), *Advances in Artificial Intelligence, KI 2008*, Lecture Notes in Artificial Intelligence, Vol. 5243, pp. 324–331. Springer, Berlin. Extended version available as e-print arXiv:0802.0116.
- [47] Fagin, R. and Halpern, J.Y. (1994) Reasoning about knowledge and probability. *J. ACM*, **41**, 340–367.
- [48] Elgesem, D. (1997) The modal logic of agency. *Nordic J. Philos. Logic*, **2**, 1–46.
- [49] Schröder, L. and Pattinson, D. (2007) Rank-1 Modal Logics are Coalgebraic. In Thomas, W. and Weil, P. (eds), *Theoretical Aspects of Computer Science, STACS 2007*, Lecture Notes in Computer Science, Vol. 4393, pp. 573–585. Springer, Berlin.
- [50] Rector, A. and Horrocks, I. (1997) Experience Building a Large, Re-Usable Medical Ontology Using a Description Logic with Transitivity and Concept Inclusions. In Farquhar, A. (ed), *AAAI Spring Symposium on Ontological Engineering, AAAI 1997*. AAAI Press, Menlo Park, CA.