Math. Struct. in Comp. Science (2011), *vol.* 21, *pp.* 235–266.
doi:10.1017/S0960129510000563

Modular algorithms for heterogeneous modal logics via multi-sorted coalgebra

LUTZ SCHRÖDER[†] and DIRK PATTINSON[‡]

[†]*DFKI Bremen and Dept. of Comput. Sci., Univ. Bremen, Cartesium, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany*
Email: Lutz.Schroeder@dfki.de
[‡]*Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK*
Email: dirk@doc.ic.ac.uk

Received 27 April 2010; revised 15 September 2010

State-based systems and modal logics for reasoning about them often heterogeneously combine a number of features such as non-determinism and probabilities. In this paper, we show that the combination of features can be reflected algorithmically, and we develop modular decision procedures for heterogeneous modal logics. The modularity is achieved by formalising the underlying state-based systems as multi-sorted coalgebras and associating both a logical and algorithmic description with a number of basic building blocks. Our main result is that logics arising as combinations of these building blocks can be decided in polynomial space provided this is also the case for the components. By instantiating the general framework to concrete cases, we obtain PSPACE decision procedures for a wide variety of structurally different logics, describing, for example, Segala systems and games with uncertain information.

1. Introduction

Modal logics appear in computer science in a variety of contexts. They are the formalism of choice for reasoning about reactive systems, and feature prominently in areas related to artificial intelligence such as knowledge representation and reasoning with uncertainty (Halpern 2003). The semantics of modal logics typically involves a notion of state and transition, which can take a number of different forms. Transitions can be probabilistic or weighted, as in probabilistic modal logic (Larsen and Skou 1991; Heifetz and Mongin 2001) and graded modal logic (Fine 1972; D'Agostino and Visser 2002), induced by joint actions of agents as in coalition logic or alternating-time logic (Pauly 2002; Alur *et al.* 2002), or non-monotonically conditioned as in conditional logic (Chellas 1980). An attractive aspect of many of these logics is that they are decidable in comparatively low complexity classes – for example, in the absence of fixpoint operators and global assumptions, typically in PSPACE (Tobies 2001; Pauly 2002; Schröder and Pattinson 2009),

[†] The work of Lutz Schröder was partially supported by BMBF grant FormalSafe (FKZ 01IW07002).
[‡] The work or Dirk Pattinson was supported by EPSRC grant EP/F031173/1.

which is the same as for the standard modal logic K (Blackburn *et al.* 2001), and not dramatically worse than propositional logic.

Features like non-determinism, probabilistic choice or joint actions are often combined, leading to systems that incorporate more than one type of transition. Moreover, features can be combined in different ways: for example, in the alternating model of probabilistic transition systems (Hansson and Jonsson 1990), states may have either non-deterministic or probabilistic transitions, whereas simple Segala systems (Segala 1995) have a two-layered structure where non-deterministic transitions lead to probability distributions over successor states. Bartels *et al.* (2003) discusses 12 different types of probabilistic transition systems arising in the literature that come about as such combinations of basic features.

In this paper, we introduce a simple calculus that formalises the combination of features, and show that combined logics typically inherit properties of their building blocks in a straightforward way; as an example, we show that shallow models and decidability in PSPACE according to the method of strictly one-step complete rule sets (Schröder and Pattinson 2009) transfer from the components to the combined logic. Our results and algorithms are *generic* and use the same algorithmic template to realise decision procedures at the level of each individual feature. This is achieved by formalising the combined logics in a multi-sorted extension of *coalgebraic modal logic* (Pattinson 2004) with a semantics that is parametric in a set functor T; models then appear as T-coalgebras. This pushes the generic PSPACE-decision procedure of Schröder and Pattinson (2009), which works uniformly for such diverse logics as Hennessy–Milner logic, coalition logic, graded modal logic, and probabilistic modal logic, to the level of combined logics that integrate several features.

Formally, a feature consists of a set of modal operators together with a set of associated proof rules. On the semantic level, a structure for a feature is an endofunctor of type $\mathbf{Set}^n \to \mathbf{Set}$, where n is the arity of the feature (for example, choice, fusion and conditionality are binary features). The notion of *gluing* formalises specific ways of combining given features. Syntactically, gluings define multi-sorted modal logics. Semantically, gluings induce endofunctors $T : \mathbf{Set}^n \to \mathbf{Set}^n$ such that T-coalgebras are models of the combined logic. The single-sorted case, $n = 1$, is of special interest since it captures the standard models of combined systems, including, for example, those presented in Bartels *et al.* (2003), which equip multi-sorted logics with a single-sorted semantics.

The central technical contribution of this work is the construction of a logically equivalent *flattening* of a given gluing, where flat gluings assign to each occurrence of a feature an individual sort in the semantics. Flat gluings are technically more tractable than general gluings, and, in fact, the extension of generic results in coalgebraic modal logic from the single-sorted case to flat gluings typically requires no more than accommodating some notational overhead for the sorts. We demonstrate this by establishing the shallow model property and a generic PSPACE algorithm for flat gluings as a generalisation of previous results for the single-sorted case (Schröder and Pattinson 2009). Together, these results imply PSPACE upper bounds for satisfiability over general gluings, including the standard single-sorted semantics of multi-sorted modal logics such as the logic of simple Segala systems (Jonsson *et al.* 2001).

1.1. *Related work*

Our work is closely related to several previous approaches (Cîrstea and Pattinson 2007; Jacobs 2001), which focus on completeness issues, with the main difference that our results make the multi-sorted nature of heterogeneous logics explicit by considering multi-sorted models. Our treatment of typed formulas resembles the concept of *ingredients* (Jacobs 2001), but the multi-sorted semantics avoids the use of the next-operator of *op.cit*. One advantage of the new framework is that constructions such as Cartesian product or disjoint union no longer have to be treated separately but are covered as special cases by the overall framework, which only needs to consider polyadic set functors and their predicate liftings. Moreover, generic results in coalgebraic modal logic such as the decision procedures of Schröder (2007) and Schröder and Pattinson (2009), and the many others listed in more detail in Section 6, generalise straightforwardly to the multi-sorted case. The multi-sorted approach to the complexity of composite modal logics complements transfer results obtained for the fusion of modal logics (Hemaspaandra 1994; Wolter 1998) in the sense that our framework currently covers logics with iterative axioms (Lewis 1975), that is, axioms that nest modalities, though only in a somewhat limited way through a multi-sorted extension of the results of Schröder and Pattinson (2008a). However, it also allows more flexible logic composition and, in particular, also applies to logics that are not amenable to Kripke semantics.

1.2. *Outline*

We begin with a brief informal discussion of a few examples of multi-sorted modal logics (Section 2) before we proceed to give a formal syntactic definition of our logic composition mechanism in Section 3. We then define the multi-sorted coalgebraic semantics of composite modal logics in Section 4, where we also present the main result of this work, *viz*. the flattening construction which reduces functor composition to multi-sortedness. Finally, we demonstrate the straightforward extension of single-sorted results to the multi-sorted setting in Section 5, using the above-mentioned generic PSPACE-algorithm as an example. This work is an extended and revised version of Schröder and Pattinson (2007).

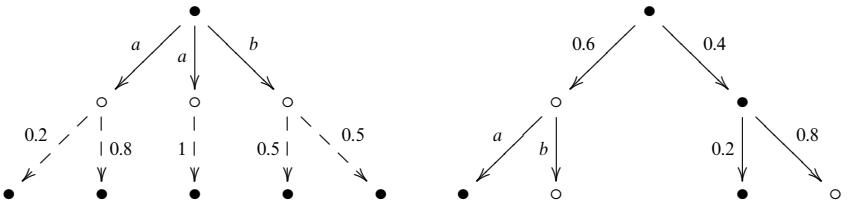
2. Multi-sorted modal logics by example

In this section we present some examples of logics that combine reasoning mechanisms in various ways as motivation for the technical definitions that follow. We discuss both the standard definitions of system types and their coalgebraic rephrasing. Readers unfamiliar with coalgebraic terminology can safely ignore this coalgebraic aspect for now, as we will recall the basic notions of coalgebra in Section 4. The coalgebraic treatment of probabilistic systems follows Bartels *et al.* (2003).

2.1. *Logics for probabilistic systems*

Segala systems (Segala 1995) and alternating systems (Hansson and Jonsson 1990) both combine probabilistic transitions and non-determinism. In Segala systems, each system

state can non-deterministically perform actions that lead to probability distributions over states. In contrast, alternating systems have two kinds of states engaging in purely probabilistic transitions and non-deterministic actions, respectively, that may end up in either kind of state. In the example below, the non-deterministic states are represented by solid circles • and the probabilistic ones by hollow circles ∘. Note that we have also represented the intermediate probability distributions occurring in Segala systems as hollow circles, although these do not as yet correspond to actual states of the system – the main message of the current work, which will be elaborated in the following sections, is that nothing is changed if we do eventually promote them to states in their own right.



Simple Segala systems Alternating systems

Formally, a simple Segala system over a set A of actions consists of a set X of states, with for each state $x \in X$ and each action $a \in A$, a set $\xi(x)(a)$ of probability distributions on X. In coalgebraic terms, this means that we have a coalgebra

$$\xi : X \to (A \to \mathscr{P}(\mathscr{D}(X)))$$

for the functor T defined by $TX = (A \to (\mathscr{P}(\mathscr{D}(X))))$, where $\to$ denotes function space, $\mathscr{P}$ denotes powerset, and $\mathscr{D}(X)$ is the set of discrete probability distributions on X. On the other hand, an alternating system consists of a set X of states and, for each state $x \in X$, either a probability distribution $\xi(x)$ on X or a map $\xi(x)$ assigning to each action $a \in A$ a set $\xi(x)(a)$ of successor states. In other words, we have a coalgebra

$$\xi : X \to \mathscr{D}(X) + (A \to \mathscr{P}(X))$$

where $+$ denotes disjoint sum.

It has been shown that a suitable variant of *probabilistic modal logic* (Larsen and Skou 1991) over a set A of actions characterises states of image-finite Segala systems up to bisimilarity (Jonsson *et al.* 2001, Theorem 8). This logic has two sorts n and u of non-deterministic and probabilistic ('uncertain') formulas, respectively, and two families of modal operators

$$\Box_a : u \to n \quad (a \in A) \qquad \text{and} \qquad L_p : n \to u \quad (p \in [0,1] \cap \mathbb{Q}),$$

where L_p can be read as 'with probability at least p'. The sets $\mathscr{L}_n$ and $\mathscr{L}_u$ of non-deterministic and probabilistic formulas, respectively, are thus defined by the grammar

$$\mathscr{L}_n \ni \phi ::= \top \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \Box_a\psi \quad (\psi \in \mathscr{L}_u, a \in A)$$
$$\mathscr{L}_u \ni \psi ::= \top \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid L_p\phi \quad (\phi \in \mathscr{L}_n, p \in [0,1] \cap \mathbb{Q}).$$

Given a Segala system (X, ξ) as above, the semantics of the two sorts of formulas is given by satisfaction relations $\models$ between distributions P on X and probabilistic formulas, and between states $x \in X$ and non-deterministic formulas, respectively, defined by mutual recursion with the obvious clauses for Boolean operators and

$$\begin{aligned} x &\models \Box_a \psi \quad \text{iff} \quad \forall P \in \xi(x)(a).\, P \models \psi \\ P &\models L_p \phi \quad \text{iff} \quad P(\{x \mid x \models \phi\}) \geqslant p \end{aligned}$$

for the modal operators. Alternating systems, on the other hand, can be captured by a logic comprising three sorts n, u and o of non-deterministic, probabilistic and alternating formulas, respectively, and modal operators

$$+ : u, n \to o \qquad L_p : o \to u \qquad \Box_a : o \to n$$

that induce the three-sorted grammar

$$\begin{aligned} \mathscr{L}_u \ni \phi &::= \top \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid L_p \chi \quad (\chi \in \mathscr{L}_a,\ p \in [0,1] \cap \mathbb{Q}) \\ \mathscr{L}_n \ni \psi &::= \top \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \Box_a \chi \quad (\chi \in \mathscr{L}_a,\ a \in A) \\ \mathscr{L}_a \ni \chi &::= \top \mid \chi_1 \wedge \chi_2 \mid \neg\chi \mid \phi + \psi \quad (\phi \in \mathscr{L}_u,\ \psi \in \mathscr{L}_n), \end{aligned}$$

which defines the sets $\mathscr{L}_n$, $\mathscr{L}_u$ and $\mathscr{L}_a$ of non-deterministic, probabilistic and alternating formulas, respectively. The binary modal operator $+$ implements the choice between probabilistic and non-deterministic transitions, and is a case statement: $\phi + \psi$ requires that ϕ holds if the present state is probabilistic whereas ψ holds if it is non-deterministic. Formally, the semantics of the logic over an alternating system (X, ξ) is given by satisfaction relations between maps $f : A \to \mathscr{P}(X)$ and non-deterministic formulas, between distributions $P \in \mathscr{D}(X)$ and probabilistic formulas, and between states $x \in X$ and alternating formulas, respectively, defined by mutual recursion, with the clauses for the modal operators being

$$\begin{aligned} P &\models L_p \chi \quad \text{iff} \quad P(\{x \mid x \models \chi\}) \geqslant p \\ f &\models \Box_a \psi \quad \text{iff} \quad \forall x \in f(a).\, x \models \psi \\ x &\models \phi + \psi \quad \text{iff} \quad \begin{cases} \xi(x) \models \phi & \text{if } \xi(x) \in \mathscr{D}(X) \\ \xi(x) \models \psi & \text{if } \xi(x) \in (A \to \mathscr{P}(X)). \end{cases} \end{aligned}$$

For example, a state x in an alternating system satisfies the formula

$$L_{1/2}(\bot + \Box_a \bot) + \Box_a \bot$$

(where $\bot \equiv \neg\top$) if either x behaves probabilistically, and with probability at least $1/2$ leads to a non-deterministic state where a is blocked, or x behaves non-deterministically and blocks a.

2.2. *Fusion of modal logics*

Both logics described above wire up the component logics in a restricted way by imposing layering and choice, respectively. The unrestricted combination of logics $\mathscr{L}_a$ and $\mathscr{L}_b$ can

be modelled by a logic with sorts a, b and f, and modal operators with associated source and target sorts

$$[\pi_1] : a \to f \qquad [\pi_2] : b \to f \qquad \Box : f \to a \qquad \heartsuit : f \to b$$

where $\Box$ ranges over the modal operators of $\mathscr{L}_a$ and $\heartsuit$ over those of $\mathscr{L}_b$. This gives rise to an obvious three-sorted grammar. The semantics of $[\pi_1]$ and $[\pi_2]$ (which will be given in Section 4) will ensure that both operators commute with all Boolean operations. Taking into account the typing information for the modal operators, this means that formulas of sort f are essentially just the well-known *fusion* $\mathscr{L}_a \otimes \mathscr{L}_b$ (see, for example, Kurucz (2006)). Recall that fusion disjointly combines the axioms and modalities of $\mathscr{L}_a$ and $\mathscr{L}_b$, giving rise to a one-sorted grammar. One can translate back and forth between the fusion and formulas of sort f by translating the operator $\Box$ of the fusion to the composite operator $[\pi_1]\Box$, and the operator $\heartsuit$ to $[\pi_2]\heartsuit$; a converse translation works by first distributing $[\pi_1]$ and $[\pi_2]$ over Boolean connectives and then replacing $[\pi_2]\heartsuit$ with $\heartsuit$ and $[\pi_1]\Box$ with $\Box$. Thus, fusion is an instance of the multi-sorted combination of modal logics.

As fusion does not impose any well-typedness constraints on formulas, it can be regarded as the maximally permissive way of combining two modal logics. However, as shown by the previous example, formulas of the fusion do not in general have an interpretation over the intended type of systems, so, for many purposes, it is preferable to work with the more restrictive well-typed combinations considered here. Moreover, although the theoretical complexity of the fusion typically does not exceed that of well-typed fragments, excluding ill-typed formulas still reduces the search space for the purposes of practical implementation.

2.3. *Conditional logic*

The standard conditional logic CK (Chellas 1980) has a binary modal operator $\Rightarrow$, where $\phi \Rightarrow \psi$ is read as a non-monotonic conditional 'if ... then normally ...'. In the right-hand argument, $\Rightarrow$ behaves essentially like the $\Box$ of the normal modal logic K, and, in particular, obeys the usual K-axiom when the left-hand argument is fixed. Indeed we can embed CK into a two-sorted *extended conditional logic* with sorts c, k and modal operators

$$\overset{\bullet}{\Rightarrow} : c, k \to c$$
$$\Box : c \to k$$

by translating $\alpha \Rightarrow \beta$ to $\alpha \overset{\bullet}{\Rightarrow} \Box\beta$. Here, $\overset{\bullet}{\Rightarrow}$ represents a rudimentary conditional and $\Box$ is the standard box modality of K. The semantic and syntactic details of this combination will be made explicit in Sections 3 and 4. This shows how a given complex logic can be broken down into simpler building blocks.

3. Compositional syntax of multi-sorted modal logic

For our purposes, it is convenient to present the syntax of multi-sorted modal logic in a way that provides explicitly for a decomposition into building blocks. The building

blocks, which we call *features*, are collections of (possibly polyadic) modal operators and associated proof rules that capture specific properties of a logic, such as the ability to describe choice, non-determinism or uncertainty.

Definition 3.1. An n-ary *feature* is a pair $\mathsf{F} = (\Lambda, \mathscr{R})$ consisting of a *(modal) similarity type* Λ, that is, a set of modal operators $\heartsuit$ with profiles $\heartsuit : i_1, \ldots, i_k \to *$, where $1 \leqslant i_1, \ldots, i_k \leqslant n$ are formal argument sorts and $*$ is a formal target sort, and a set $\mathscr{R}$ of *one-step rules* of the form

$$R = (\phi_1; \ldots; \phi_n)/\psi,$$

where ϕ_i, for $i = 1, \ldots, n$, is a propositional formula over a set V_i of propositional variables, and ψ is a clause over atoms of the form $\heartsuit(a_1, \ldots, a_k)$ with $\heartsuit : i_1, \ldots, i_k \to *$ in Λ and $a_j \in V_{i_j}$, $j = 1, \ldots, k$. Here we use the standard terminology for propositional logic: a *literal* over a set Z of atoms is either an element of Z or the negation of such an element; a *clause* over Z is a finite disjunction of literals over Z; and a *conjunctive clause* over Z is a finite conjunction of literals over Z. Given a substitution σ of the variables in R by elements of Z, we refer to the substituted rule $R\sigma = (\phi_1\sigma; \ldots; \phi_n\sigma)/\psi\sigma$ as a *Z-instance* of R. Finally, we use phrases such as *propositional reasoning*, *propositional entailment*, *propositional equivalence*, and so on, to refer to reasoning using propositional tautologies and *modus ponens*.

Although it does not play a crucial role in the present development, we also impose the technical condition needed in other contexts that rules be *injective*, which means that every propositional variable can occur at most once in the conclusion of a rule (Schröder and Pattinson 2009). This is not a real restriction, since different variables a, b can be made to carry the same interpretation by introducing a premise $a \leftrightarrow b$, but it keeps the rules in synchrony with the presentation elsewhere.

Note that the rule format forbids nested modalities in the conclusion, so rules describe the one-step behaviour of a system. As in the single-sorted case (Schröder 2007), this format always suffices to completely axiomatise the features of interest, as long as no global conditions (such as transitivity) are imposed on the coalgebraic models.

Example 3.2. We will now describe the features used in the Examples of Section 2, as well as the standard features that implicitly occur in previous treatments of compositional coalgebraic logics (Jacobs 2001; Cîrstea and Pattinson 2007). The similarity types are as follows (with intuitive meanings supported by the respective semantics to be given in Section 4):

Non-determinism: Given a set A of actions, the unary feature N_A has modal operators $\Box_a : 1 \to *$ for $a \in A$, which can be read as 'all a-successors satisfy ...'. If A is a singleton, we write K instead of N_A, and $\Box$ instead of $\Box_a$.

Uncertainty: The unary feature U has modal operators $L_p : 1 \to *$ for $p \in [0, 1] \cap \mathbb{Q}$, which can be read as 'with probability at least p, the next state satisfies ...'.

Coalition: Given a finite set N of *agents*, the unary feature G_N has modal operators $[C] : 1 \to *$ for all *coalitions* C, that is, all subsets of N, which can be read as 'coalition C has a collaborative strategy to enforce ...'.

Choice: The binary feature S has a single binary modal operator $+ : 1,2 \to *$, where $\phi + \psi$ can be read as 'if the next state is in the left alternative, then it satisfies ϕ, and if the next state is in the right alternative, then it satisfies ψ'.

Fusion: The binary feature P has two modal operators $[\pi_i] : i \to *$, $i = 1,2$, which can be read as 'the ith component satisfies ...'.

Conditionality: The binary feature C has a binary modal operator $\overset{\bullet}{\Rightarrow} : 1,2 \to *$, which can be read as 'if ... then normally ...'.

Input: Given a set I of inputs, the unary feature I_I has modal operators $(i) : 1 \to *$ for $i \in I$, which can be read as 'upon input i'.

Output: Given a finite set O of outputs (infinite sets can also be handled, but require a slightly different rule set), the nullary feature O_O has nullary modal operators (that is, flexible constants) $o :\to *$ for $o \in O$, which can be read as 'the current state outputs o'. For the special case of O_1, where 1 is a singleton output set, we write $\mathsf{1}$ for O_1.

Figure 1 shows the associated proof rules – they are written in a special format that will be needed in Section 5. The rules for non-determinism and uncertainty are taken from Schröder and Pattinson (2009), where we use an obviously equivalent simplification of the rule for uncertainty; the others are obtained by the same principles. The sum expression in the uncertainty rule refers to the arithmetic of characteristic functions (Schröder and Pattinson 2009); explicitly,

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

where $r(J) = \sum_{j\in J} r_j$. Moreover, $sgn(r)\phi \equiv \phi$ when $r > 0$, and $sgn(r)\phi \equiv \neg\phi$ otherwise.

Note that the above list of features is by no means exhaustive. In particular, we can see every modal logic that is axiomatised by a set of one-step rules (Schröder and Pattinson 2010, Definition 2) as a unary feature. Other natural features that arise in this way are, for instance, monotonicity, neighbourhoods and integer weights, where the associated proof rules can be found in Schröder and Pattinson (2009, Examples 3.19 and 6.2).

Remark 3.3. In the non-determinism feature, we have opted not to decompose the feature further into K and the input feature. This is both to illustrate the fact that we are not forced to use the finest possible granularity in our choice of basic features, and to keep the running examples in tune with the introduction to Section 2.

The examples from Section 2 demonstrate that features can be combined in different ways. This is formalised by the notion of gluing.

Definition 3.4. Let Φ be a set of features and $\mathscr{S}$ be a set of sorts. *Feature expressions* t (over $\mathscr{S}$) are terms over the set $\mathscr{S}$ of variables, where the features appear as function symbols, that is,

$$t ::= a \mid \mathsf{F}(t_1,\dots,t_n) \qquad a \in \mathscr{S},\ \mathsf{F} \in \Phi\ n\text{-ary}.$$

A *gluing* of Φ over $\mathscr{S}$ is a family $\mathsf{G} = (t_a)_{a\in\mathscr{S}}$ of feature expressions, denoted by

$$(a_1 \to t_{a_1},\dots,a_n \to t_{a_n})$$

for $\mathscr{S} = \{a_1,\dots,a_n\}$; in this case we also write $a_i \to t_{a_i} \in \mathsf{G}$.

Non-determinism: $\dfrac{\bigwedge_{j=1}^{n} \alpha_j \to \beta}{\bigwedge_{j=1}^{n} \Box_a \alpha_j \to \Box_a \beta} \quad (n \geq 0, a \in A)$

Uncertainty: $\dfrac{\sum_{i=0}^{n} r_i a_i \sqsupset \sum_{i=0}^{n} r_i p_i}{\bigvee_{0 \leq i \leq n} sgn(r_i) L_{p_i} a_i}$ $\left(\begin{array}{l} n \geq 0; r_0, \ldots, r_n \in \mathbb{Z} - \{0\}; \sqsupset \in \{>, \geq\}; \\ \sqsupset \text{ is } > \text{ if } r_j < 0 \text{ for all } j, \\ \text{and } \geq \text{ otherwise.} \end{array} \right)$

Coalition: $\dfrac{\bigvee_{i=1}^{n} \neg \alpha_i}{\bigvee_{i=1}^{n} \neg [C_i] \alpha_i} \quad \dfrac{\bigwedge_{i=1}^{n} \alpha_i \to (\beta \vee \bigvee_{j=1}^{m} c_j)}{\bigwedge_{i=1}^{n} [C_i] \alpha_i \to ([D] \beta \vee \bigvee_{j=1}^{m} [N] c_j)}$

$(m, n \geq 0; C_j \subseteq D \text{ for } j = 1, \ldots, m; C_i \cap C_j = \emptyset \text{ for } i \neq j)$

Choice: $\dfrac{(\bigwedge_{j=1}^{m} \alpha_j \to \bigvee_{k=1}^{n} \beta_k) : 1 \quad (\bigwedge_{j=1}^{m} \gamma_j \to \bigvee_{k=1}^{n} \delta_k) : 2}{\bigwedge_{j=1}^{m} (\alpha_j + \gamma_j) \to \bigvee_{k=1}^{n} (\beta_k + \delta_k)} \; (m, n \geq 0)$

Fusion: $\dfrac{(\bigwedge_{j=1}^{m} \alpha_j \to \bigvee_{k=1}^{n} \beta_k) : i}{\bigwedge_{j=1}^{m} [\pi_i] \alpha_j \to \bigvee_{k=1}^{n} [\pi_i] \beta_k} \quad (i = 1, 2; m, n \geq 0)$

Conditionality: $\dfrac{(\gamma_i \leftrightarrow \gamma_j) : 1 \; (i, j \in \{1, \ldots, m+n\}) \quad (\bigwedge_{j=1}^{m} \alpha_j \to \bigvee_{k=1}^{n} \beta_k) : 2}{\bigwedge_{j=1}^{m} (\gamma_i \stackrel{\bullet}{\Rightarrow} \alpha_j) \to \bigvee_{k=1}^{n} (\gamma_{k+m} \stackrel{\bullet}{\Rightarrow} \beta_k)} \quad (m, n \geq 0)$

Input: $\dfrac{(\bigwedge_{j=1}^{m} \alpha_j \to \bigvee_{k=1}^{n} \beta_k) : i}{\bigwedge_{j=1}^{m} (i) \alpha_j \to \bigvee_{k=1}^{n} (i) \beta_k} \quad (i \in I; m, n \geq 0)$

Output: $\dfrac{}{o_1 \wedge o_2 \to \bot} \; (o_1, o_2 \in O; o_1 \neq o_2) \quad \dfrac{}{\bigvee_{o \in O} o}$

Fig. 1. Proof rules for the features of Example 3.2

A gluing $\mathsf{G} = (t_a)_{a \in \mathscr{S}}$ induces a **multi-sorted modal logic**, as follows. The set $\mathsf{Types(G)}$ of *types* consists of the proper subterms of the t_a, where the sorts $a \in \mathscr{S}$ are called *base types* and the expressions $t \in \mathsf{Types(G)} \setminus \mathscr{S}$ are the *composite types*. The reason the t_a themselves are not in general regarded as types is that we can instead represent them by their index sort a; this is related to the semantic interpretation of the association $a \to t_a$ as a component of a multi-sorted coalgebra map, which will be defined in Section 4. The types are related to the *ingredients* of Jacobs (2001), and will serve to type modal formulas in our logic. In some sense, they also correspond to types of states, an intuition that will be made explicit in our flattening construction. We say a gluing is *flat* if $\mathscr{S} = \mathsf{Types(G)}$, that is, there are no composite types, which is the case if every term t_a is of the form $\mathsf{F}(a_1, \ldots, a_n)$; intuitively, a gluing is flat if all types of states are already explicit in the sort set $\mathscr{S}$.

Typed G-*formulas* $\phi : s$, where $s \in \mathsf{Types(G)}$, are inductively generated by closure under Boolean operators $\bot$, $\neg$, $\wedge$ at each type (with further Boolean operators $\vee$, $\top$, $\to$, $\leftrightarrow$ defined in the standard way), and by the following typing rules for:

— Composite types:

$$\frac{\phi_1 : s_1 \quad \ldots \quad \phi_n : s_n}{\heartsuit(\phi_{i_1}, \ldots, \phi_{i_n}) : \mathsf{F}(s_1, \ldots, s_n)}$$

with the side condition $\mathsf{F}(s_1, \ldots, s_n) \in \mathsf{Types(G)}$

— Base types:

$$\frac{\phi_1 : s_1 \quad \dots \quad \phi_n : s_n}{\heartsuit(\phi_{i_1}, \dots, \phi_{i_n}) : a},$$

with side condition $a \to \mathsf{F}(s_1, \dots, s_n) \in \mathsf{G}$

and in both cases $\heartsuit : i_1, \dots, i_n \to *$ in F.

The special status of base types is related to the fact that for $a \to \mathsf{F}(s_1, \dots, s_n) \in \mathsf{G}$, the feature expression $\mathsf{F}(s_1, \dots, s_n)$ is not, in general, a type, being instead thought of as represented by the base type a. We write $\mathscr{F}_s(\mathsf{G})$ for the set of G-formulas of type s and use $\mathscr{F}(\mathsf{G})$ to denote the family $(\mathscr{F}_s(\mathsf{G}))_{s \in \mathsf{Types}(\mathsf{G})}$.

Similarly, a gluing induces a typed **proof system**, described in terms of a $\mathsf{Types}(\mathsf{G})$-indexed family of derivability predicates $\vdash_s \subseteq \mathscr{F}_s(\mathsf{G})$. These predicates are defined inductively by closure under propositional reasoning at each type and the following deduction rules, which are only distinguished by the type discipline in the same way as the above typing rules for formulas:

— Composite types:

$$\frac{\vdash_{s_1} \phi_1\sigma \quad \dots \quad \vdash_{s_n} \phi_n\sigma}{\vdash_{\mathsf{F}(s_1,\dots,s_n)} \psi\sigma}$$

where $\mathsf{F}(s_1, \dots, s_n) \in \mathsf{Types}(\mathsf{G})$

— Base types:

$$\frac{\vdash_{s_1} \phi_1\sigma \quad \dots \quad \vdash_{s_n} \phi_n\sigma}{\vdash_a \psi\sigma}$$

where $a \to \mathsf{F}(s_1, \dots, s_n) \in \mathsf{G}$

where in both rules, $(\phi_1; \dots; \phi_n)/\psi$ is a rule of F and σ is a substitution mapping variables $a \in V_i$ to formulas $\sigma(a) : s_i$.

Remark 3.5. Formulas and proofs induced by a gluing $\mathsf{G} = (t_a)_{a \in \mathscr{S}}$ need not have unique types – specifically, when t_a is a subterm of t_b for some $a, b \in \mathscr{S}$. We will therefore disambiguate formulas explicitly by their type when we define the semantics in Section 4.

A given logic can be syntactically generated by different gluings, which typically include both flat and non-flat ones, determining different classes of semantic structures (see Section 4). One of the main contributions of this paper is the proof of logical equivalence for the respective semantics. Flat gluings are technically more tractable, while logics occurring in the literature, including the ones described in Section 2, are typically non-flat.

Example 3.6. From the features S, U and N_A (Example 3.2), we can form gluings

$$\mathsf{G}_1 \equiv (o \to \mathsf{S}(\mathsf{U}(o), \mathsf{N}_A(o))) \text{ and } \mathsf{G}_2 \equiv (o \to \mathsf{S}(u, n), u \to \mathsf{U}(o), n \to \mathsf{N}_A(o)).$$

Here, G_1 has types o, $\mathsf{N}_A(o)$, $\mathsf{U}(o)$, whereas G_2 is flat with types o, n, u. Modulo identifications $\mathsf{N}_A(o) = n$ and $\mathsf{U}(o) = u$, both gluings give rise to the (typed) formulas describing alternating systems (Section 2.1). To illustrate the typing system, we will now

briefly show how the formula $L_p\alpha + \Box_a\beta$ is typed in the non-flat gluing G_1, assuming $\alpha : o$, $\beta : o$:

$$\dfrac{\dfrac{\alpha : o}{L_p\alpha : \mathsf{U}(o)} \quad \dfrac{\beta : o}{\Box_a\beta : \mathsf{N}_A(o)}}{L_p\alpha + \Box_a\beta : o}$$

Here, the last derivation step uses the typing rule for base types, and the other derivation steps use the typing rule for composite types.

The remaining example logics from Section 2 are captured by the following gluings:

Probabilistic modal logic of Segala Systems: This is the single-sorted gluing $n \to \mathsf{N}_A(\mathsf{U}(n))$. It has types n and $\mathsf{U}(n)$, the latter corresponding to the sort u in the discussion in Section 2. We can generate the same logic from the flat gluing $(n \to \mathsf{N}_A(p), p \to \mathsf{U}(n))$.

Fusion: The fusion of logics $\mathscr{L}_a$ and $\mathscr{L}_b$ as in Section 2.2, regarded as features, is $f \to \mathsf{P}(\mathscr{L}_a(f), \mathscr{L}_b(f))$.

Extended conditional logic: This may be captured by the single-sorted gluing $c \to \mathsf{C}(c, \mathsf{K}(c))$. Note, in particular, that in the induced proof system, we can derive the standard rule

$$(RCK)\ \dfrac{\bigwedge_{i=1}^n \alpha_i \to \beta}{\bigwedge_{i=1}^n (\gamma \Rightarrow \alpha_i) \to (\gamma \Rightarrow \beta)}\ (n \geqslant 0)$$

of the conditional logic CK (Chellas 1980) (presented for the sake of readability in non-injective form, that is, with repeated occurrences of γ), where $\gamma \Rightarrow \alpha$ abbreviates $\gamma \stackrel{\bullet}{\Rightarrow} \Box\alpha$, as follows:

$$\dfrac{\vdash_c \gamma \leftrightarrow \gamma \qquad \dfrac{\vdash_c \bigwedge_{i=1}^n \alpha_i \to \beta}{\vdash_{\mathsf{K}(c)} \bigwedge_{i=1}^n \Box\alpha_i \to \Box\beta}}{\vdash_c \bigwedge_{i=1}^n \gamma \stackrel{\bullet}{\Rightarrow} \Box\alpha_i \to \gamma \stackrel{\bullet}{\Rightarrow} \Box\beta}$$

Here, the first step in the right-hand branch applies the rule for K following the typing discipline for composite types, and the second step applies the rule for C following the typing discipline for base types.

The conditional logic $CKCEM$ (see, for example, Olivetti *et al.* (2007)), which additionally obeys the axiom of *conditional excluded middle*

$$(\alpha \Rightarrow \beta) \vee (\alpha \Rightarrow \neg\beta),$$

can be captured (in extended form) as the gluing

$$c \to \mathsf{C}(c, \mathsf{S}(c, 1)).$$

Here, we encode the standard conditional arrow $\phi \Rightarrow \psi$ as $\phi \stackrel{\bullet}{\Rightarrow} (\psi + \top)$.

Probabilistic Conditional Logic: A probabilistic variant of conditional logic is given by the gluing

$$c \to \mathsf{C}(c, \mathsf{U}(c)),$$

in which formulas such as $\phi \stackrel{\bullet}{\Rightarrow} L_p\psi$ can be read as 'if ϕ, then we can normally expect ψ with confidence at least p'.

Probabilistic Coalition Logic: A combination of quantitative uncertainty and coalition logic is captured by the gluing

$$c \to \mathsf{G}_N(\mathsf{U}(c)),$$

which involves the Uncertainty and Coalition features described in Example 3.2. It allows us to describe games with elements of chance; for example, the formula $[C]L_p\phi$ expresses the fact that coalition C has a joint strategy to bring about ϕ with probability at least p.

4. Multi-sorted versus single-sorted coalgebraic semantics

In this section we generalise the coalgebraic interpretation of modal logic, which was introduced in Pattinson (2004), to the multi-sorted case. Crucially, we interpret multi-sorted logics over multi-sorted coalgebras. The parametricity over signature functors for coalgebras is the key feature of our framework that allows for uniform results that can be instantiated to a large number of structurally different systems and logics. We will begin by recalling some basic notions of multi-sorted coalgebra (see, for example, Mossakowski *et al.* (2006)), which generalises the single-sorted setting (Rutten 2000).

Definition 4.1. We write **Set** for the category of sets and functions. Let $\mathbf{Set}^{\mathscr{S}}$ denote the category of $\mathscr{S}$-sorted sets and $\mathscr{S}$-sorted functions, with objects being families $X = (X_a)_{a\in\mathscr{S}}$ (or just (X_a)) of sets X_a, and morphisms $f : (X_a) \to (Y_a)$ being families $f = (f_a)_{a\in\mathscr{S}}$ of maps $f_a : X_a \to Y_a$. We shall often introduce $\mathscr{S}$-sorted sets and maps using just one letter, say X, and then implicitly understand X_a as a notation for the ath component of X. We write $\mathbf{Set}^n$ for $\mathbf{Set}^{\{1,\dots,n\}}$. A functor $T : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set}^{\mathscr{S}}$ may be regarded as a family $T = (T_a)_{a\in\mathscr{S}}$ of functors $T_a : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set}$. A *$T$-coalgebra* $A = (X, \xi)$ is a pair (X, ξ) where X is an $\mathscr{S}$-sorted set and $\xi = (\xi_a) : X \to TX$ is an $\mathscr{S}$-sorted function (that is, $\xi_a : X_a \to T_aX$) called the *transition* function. A *morphism* between T-coalgebras (X, ξ) and (Y, ζ) is an $\mathscr{S}$-sorted function $f : X \to Y$ such that $(Tf)\xi = \zeta f$ in $\mathbf{Set}^{\mathscr{S}}$, that is, at every sort $a \in \mathscr{S}$, the diagram

$$\begin{array}{ccc} X_a & \xrightarrow{f_a} & Y_a \\ {\scriptstyle \xi_a}\downarrow & & \downarrow{\scriptstyle \zeta_a} \\ T_aX & \xrightarrow[T_af]{} & T_aY \end{array}$$

commutes.

We view coalgebras as generalised transition systems: the transition function maps states to structured sets of observations and successor states, the latter taken from the available sorts as specified by T.

Assumption 4.2. We can assume without loss of generality that T preserves tuples of injective maps by an easy extension of a construction from Barr (1993) from single-sorted to the multi-sorted maps. For convenience of notation, we will in fact assume that (multi-sorted) subset inclusions $X \hookrightarrow Y$ are mapped to subset inclusions $TX \hookrightarrow TY$.

The interpretation of modal operators is based on predicate liftings (Pattinson 2004; Schröder 2008); in the multi-sorted setting, this takes the following shape.

Definition 4.3. A *predicate lifting* λ of profile $\lambda : i_1, \dots, i_k \stackrel{\bullet}{\to} *$ for a functor $T : \mathbf{Set}^n \to \mathbf{Set}$, where $i_1, \dots, i_k \leqslant n$, is a natural transformation

$$\lambda : (\mathcal{Q} \circ P_{i_1}^{op}) \times \cdots \times (\mathcal{Q} \circ P_{i_k}^{op}) \to \mathcal{Q} \circ T^{op}$$

between functors $(\mathbf{Set}^n)^{op} \to \mathbf{Set}$, where $\mathcal{Q}$ denotes the contravariant powerset functor $\mathbf{Set}^{op} \to \mathbf{Set}$ (that is, $\mathcal{Q}X = \mathcal{P}X$, and $\mathcal{Q}(f)(A) = f^{-1}[A]$) and $P_i : \mathbf{Set}^n \to \mathbf{Set}$ is the ith projection.

We will now construct a compositional coalgebraic semantics of the logic $\mathscr{F}(\mathsf{G})$ induced by a gluing G from structures associated with the features combined by G. We first describe a notion of structure associated with a single feature, and then the combination of such structures along a gluing.

Definition 4.4. Let $\mathsf{F} = (\Lambda, \mathscr{R})$ be an n-ary feature. A *structure* for F consists of a functor $[\![\mathsf{F}]\!] : \mathbf{Set}^n \to \mathbf{Set}$ and an assignment of a predicate lifting $[\![\heartsuit]\!] : i_1, \dots, i_k \stackrel{\bullet}{\to} *$ for T to every modal operator $\heartsuit : i_1, \dots, i_k \to *$ in Λ, subject to the condition that every rule $R = \phi_1; \dots; \phi_n/\psi$ over V in $\mathscr{R}$ is *one-step sound*: for every n-sorted set X and every assignment τ of subsets $\tau(a) \subseteq X_i$ to the variables $a \in V_i$, if $[\![\phi_i]\!]_{X,\tau} = X_i$ for all i, then $[\![\psi]\!]_{TX,\tau} = TX$, where $[\![\phi_i]\!]_{X,\tau} \subseteq X_i$ and $[\![\psi]\!]_{TX,\tau} \subseteq TX$ are defined by the usual clauses for Boolean operators and

$$[\![\heartsuit(a_1, \dots, a_k)]\!]_{TX,\tau} = [\![\heartsuit]\!]_X(\tau(a_1), \dots, \tau(a_k)).$$

When features are equipped with structures, every feature expression t over the set $\mathscr{S}$ of sorts defines a functor $[\![t]\!] : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set}$ by

$$[\![a]\!] = P_a : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set} \quad (a \in \mathscr{S})$$
$$[\![\mathsf{F}(t_1, \dots, t_n)]\!] = [\![\mathsf{F}]\!] \circ \langle [\![t_1]\!], \dots, [\![t_n]\!] \rangle,$$

where P_a is the projection to the ath component and $\langle \cdot \rangle$ represents tupling. Thus, a gluing $\mathsf{G} = (t_a)_{a \in \mathscr{S}}$ induces a functor

$$[\![\mathsf{G}]\!] = \langle [\![t_a]\!] \rangle_{a \in \mathscr{S}} : \mathbf{Set}^S \to \mathbf{Set}^S.$$

We will refer to $[\![\mathsf{G}]\!]$-coalgebras as G-*coalgebras* for short. Strictly speaking, the semantics

$$[\![t]\!] : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set}$$

of a feature expression t depends on the sort context $\mathscr{S}$, and, when necessary, we will make this explicit by writing $[\![\mathscr{S} \rhd t]\!]$ in place of $[\![t]\!]$. There is an obvious *substitution lemma* stating that substitution of feature expressions corresponds to composition of functors.

The coalgebraic semantics of $\mathscr{F}(\mathsf{G})$ is now given with respect to G-coalgebras $C = (X, \xi)$. For a type $s \in \mathsf{Types}(\mathsf{G})$, an *s-state* of C is an element $x \in [\![s]\!]X$ (recall that $[\![s]\!] : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set}$ denotes a functor). The semantics of a formula $\phi : s$ is a set $[\![\phi]\!]_C \subseteq [\![s]\!]X$ of s-states. We have the usual clauses for propositional connectives, and the semantics of modal

operators is given by the following clauses:

— Composite types:

$$[\![\heartsuit(\phi_1,\dots,\phi_n) : \mathsf{F}(s_1,\dots,s_n)]\!]_C = [\![\heartsuit]\!]([\![\phi_1]\!]_C,\dots,[\![\phi_n]\!]_C)$$

assuming $\mathsf{F}(s_1,\dots,s_n) \in \mathsf{Types}(\mathsf{G})$)

— Base types:

$$[\![\heartsuit(\phi_1,\dots,\phi_n) : a]\!]_C = \xi_a^{-1} \circ [\![\heartsuit]\!]([\![\phi_1]\!]_C,\dots,[\![\phi_n]\!]_C)$$

for $a \to \mathsf{F}(s_1,\dots,s_n) \in \mathsf{G}$

where, in both cases, $\heartsuit : i_1,\dots,i_n \to *$ in F.

We write $x \models^s_C \phi$ if $\phi : s$ and $x \in [\![\phi]\!]_C$, omitting sub- and superscripts whenever they are clear from the context.

Note that the requirement that rules are one-step sound immediately yields soundness of the logic with respect to the semantics described above; this is as in Cîrstea and Pattinson (2007).

Example 4.5. The standard semantics for the features of Example 3.2 is induced by the following structures:

Non-determinism: A structure for N_A is given by:

$$[\![\mathsf{N}_A]\!] = \mathscr{P}(A \times \_)$$
$$[\![\Box_a]\!]_X(C) = \{B \in \mathscr{P}(A \times X) \mid \{y \in X \mid (a,y) \in B\} \subseteq C\}.$$

Note that (single-sorted) coalgebras for $\mathscr{P}(A \times \_)$ are labelled transition systems since they associate to each state x a set of pairs (a,y) consisting of an action a and a successor state y, which we can read as 'there is an a-transition from x to y'. The lifting associated with $\Box_a$ gives rise to the usual semantics of Hennessy–Milner logic (Pattinson 2004): a state x satisfies a formula $\Box_a\phi$ if and only if all states y to which x has an a-transition satisfy ϕ.

Uncertainty: We put $[\![\mathsf{U}]\!] = \mathscr{D}$, where $\mathscr{D}$ is the *discrete distribution functor* $\mathscr{D}$ that maps a set X to the set of discrete probability distributions on X; that is, $P \in \mathscr{D}(X)$ is given as $P(A) = \sum_{x \in A} p(x)$ by a map $p : X \to [0,1]$ such that $\sum_{x \in X} p(x) = 1$ (which implies that $p(x) > 0$ for at most countably many x). The modal operators L_p are interpreted by

$$[\![L_p]\!]_X(A) = \{P \in \mathscr{D}X \mid PA \geqslant p\}.$$

(Single-sorted) $\mathscr{D}$-coalgebras are probabilistic transition systems. For $\mathsf{G} = (n \to \mathsf{N}_A(\mathsf{U}(n)))$ (Example 3.6), we have $[\![\mathsf{G}]\!] = \mathscr{P} \circ \mathscr{D}$, so G-coalgebras are precisely Segala systems, while coalgebras for the corresponding flat gluing $(n \to \mathsf{N}_A(p), p \to \mathsf{U}(n))$ have

an explicit separation between non-deterministic and probabilistic states, and thus correspond to strictly alternating systems in the sense of Hansson (1994).

Coalition: We opt for a semantics with finite strategy sets, following the standard semantics of alternating-time temporal logic (Alur *et al.* 2002). Thus, we put

$$[\![\mathsf{G}_N]\!]X = \left\{(d,f) \mid d : N \to \mathbb{N}, f : \left(\prod_{a\in N}[d(a)]\right) \to X\right\},$$

where $[d(a)] = \{0,\ldots,d(a)\}$ is the set of *strategies* available to agent a, and interpret $[C]$ for a coalition $C \subseteq N$ by

$$[\![[C]]\!]_X A = \Big\{(d,f) \in [\![\mathsf{G}_N]\!]X \mid \exists \sigma_C \in \textstyle\prod_{a\in C}[d(a)].\ \forall \sigma_{N-C} \in \prod_{a\in N-C}[d(a)].\, f(\sigma_C,\sigma_{N-C}) \in A\Big\}$$

where (σ_C,σ_{N-C}) denotes the obvious element of $\prod_{a\in N}[d(a)]$. That is, the informal reading of $[C]\phi$, 'C can enforce ϕ', translates into the requirement that there exists a collaborative strategy for C such that whatever collaborative strategy is chosen by the opponents $N - C$, the outcome satisfies ϕ.

Choice: We let $[\![\mathsf{S}]\!]$ be the disjoint sum functor $[\![\mathsf{S}]\!](X,Y) = X + Y$, and interpret the modality $+$ by

$$[\![+]\!]_{X,Y}(A,B) = A + B \subseteq X + Y.$$

Thus the condition A applies in the left-hand alternative of the sum, and the condition B applies in the right-hand alternative, in accordance with the informal reading of the $+$ operator given in Example 3.2.

Fusion: We let $[\![\mathsf{P}]\!]$ be the binary product functor $[\![\mathsf{P}]\!](X,Y) = X \times Y$, and put

$$[\![\pi_1]\!]_{X,Y}A = \{(x,y) \mid x \in A\} \text{ and } [\![\pi_2]\!]_{X,Y}B = \{(x,y) \mid y \in B\}.$$

Again, this definition is in agreement with the informal description of the operators $[\pi_1]$, $[\pi_2]$ given in Example 3.2.

Conditionality: We define the functor $[\![\mathsf{C}]\!]$ by $[\![\mathsf{C}]\!](X,Y) = \mathcal{Q}X \to Y$, with $\mathcal{Q}$ denoting contravariant powerset and $\to$ denoting function space, and put

$$[\![\overset{\bullet}{\Rightarrow}]\!]_{X,Y}(A,B) = \{f : \mathcal{Q}X \to Y \mid f(A) \in B\}.$$

For $\mathsf{G} = (c \to \mathsf{C}(c,K(c)))$ (Example 3.6), we have $[\![\mathsf{G}]\!]X = \mathcal{Q}X \to \mathcal{P}X$, and G-coalgebras are conditional frames (Chellas 1980), that is, we associate with each state x and each proposition A a set of A-successor states of x.

Input: We put $[\![\mathsf{I}_I]\!]X = I \to X$, and $[\![(i)]\!]_X A = \{f : I \to X \mid f(i) \in A\}$. Thus, $[\![\mathsf{I}_I]\!]$-coalgebras associate with each state x and each input i a successor state y, and then x satisfies $(i)\phi$ if and only if y satisfies ϕ.

Output: We put $[\![\mathsf{O}_O]\!] = O$, and $[\![o]\!]_X = \{o\}$. Thus, for $\mathsf{G} = (a \to \mathsf{O}_O)$, a G-coalgebra ξ just maps each state x to an output $\xi(x) \in O$, and then x satisfies o if and only if $\xi(x) = o$.

Modal logic only considers the observable behaviour of states, and this is formally expressed as the invariance of the logic under morphisms.

Proposition 4.6. Let $f : C \to D$ be a morphism of G-coalgebras. Then for each G-formula $\phi : s$ and each s-state x in C, we have $x \models^s_C \phi$ if and only if $([\![s]\!]f)(x) \models^s_D \phi$.

Proof. The proof is by induction over the formula structure, with the case of modal operators taken care of by the naturality of predicate liftings. □

We can now state the (local) *satisfiability problem* for multi-sorted modal logics.

Definition 4.7. A G-formula $\phi : s$ is G-*satisfiable* if there exist a G-coalgebra C and an s-state x in C such that $x \models^s_C \phi$.

The central contribution of this paper is to show that for every gluing, we can construct a flat gluing with an equivalent satisfiability problem. For flat gluings, we can easily generalise existing model constructions and complexity results for coalgebraic modal logic (see, for example, Schröder (2007), Schröder and Pattinson (2008b; 2009)), essentially, by introducing additional indices for the sorts, and the relevant criteria reduce to the component logics; by way of an example, this is discussed in more detail in Section 5 for the shallow-model-based PSPACE algorithm of Schröder and Pattinson (2009). In this way, we also obtain compositional algorithmic methods for the standard single-sorted semantics present in the literature.

The core of the flattening result is the following construction, which converts a feature expression s that occurs in a gluing into a new sort. Let $\mathsf{G} = (t_a)_{a\in\mathscr{S}}$ be a gluing over the set $\mathscr{S}$ in which the feature expression s occurs, and pick a fresh sort $\overline{s}$ (the sort into which the feature expression s is converted). That is, we assume that

$$t_a = t'_a[s/\overline{s}]$$

for all $a \in \mathscr{S}$, where $t[s/\overline{s}]$ denotes the result of replacing all occurrences of (the fresh sort) $\overline{s}$ by (the feature expression) s, and the t'_a are feature expressions over $\mathscr{S}' := \mathscr{S} \cup \{\overline{s}\}$. We then construct a gluing $\mathsf{G}' = (t'_a)_{a\in\mathscr{S}'}$ over $\mathscr{S}'$ by extending the family $(t'_a)_{a\in\mathscr{S}}$ of feature expressions to a gluing $(t'_a)_{a\in\mathscr{S}'}$ by putting

$$t'_{\overline{s}} = s.$$

To simplify the notation, we regard $\overline{s}$ as the 'first' sort in $\mathscr{S}'$, and, correspondingly, we use (y, x) to denote, for any $\mathscr{S}$-indexed tuple $x = (x_a)_{a\in\mathscr{S}}$ (for example, an $\mathscr{S}$-sorted set or a $\mathscr{S}$-sorted map), the $\mathscr{S}'$-sorted tuple whose a-component is x_a for $a \in \mathscr{S}$ and whose $\overline{s}$-component is y. Conversely, given an $\mathscr{S}'$-indexed tuple $x = (x_a)_{a\in\mathscr{S}'}$, we use $x|_{\mathscr{S}}$ to denote the $\mathscr{S}$-indexed tuple $(x_a)_{a\in\mathscr{S}}$.

Example 4.8. Consider the gluing

$$\mathsf{G} = (a \to F(G(b,c), H(a)), b \to G(b,c), c \to F(a,c))$$

and let $s = G(b,c)$. This allows us to write

$$\mathsf{G} = (a \to F(\overline{s}, H(a))[s/\overline{s}], b \to \overline{s}[s/\overline{s}], c \to F(a,c)[s/\overline{s}]),$$

and we obtain a new gluing

$$\mathsf{G}' = (\overline{s} \to s, a \to F(\overline{s}, H(a)), b \to \overline{s}, c \to F(a,c))$$

over the set $\mathscr{S}' = \{a, b, c, \overline{s}\}$ where we have given the $\overline{s}$-component first according to the convention above.

We have a functor Pad from G-coalgebras to G'-coalgebras, which extends a G-coalgebra (X, ξ) to the G'-coalgebra

$$\mathsf{Pad}(X, \xi) = ((\llbracket \mathscr{S} \rhd s \rrbracket X, X), (id_{\llbracket \mathscr{S} \rhd s \rrbracket X}, \xi))$$

(where, for $a \in \mathscr{S}$, the maps ξ_a have the correct type for a G'-coalgebra due to the choice of the $\overline{s}$-component). Similarly, Pad extends a morphism f of G-coalgebras to the morphism

$$\mathsf{Pad}(f) = (\llbracket \mathscr{S} \rhd s \rrbracket f, f)$$

of G'-coalgebras. In short, Pad adds an identity $\overline{s}$-component to (X, ξ).

Example 4.9. We will now continue Example 4.8 and assume that each feature is equipped with a structure, denoted by $\llbracket \cdot \rrbracket$. The gluing G induces the functor

$$\llbracket \mathsf{G} \rrbracket = (\llbracket F \rrbracket \circ \langle \llbracket G \rrbracket \circ \langle P_b, P_c \rangle, \llbracket H \rrbracket \circ P_a \rangle, \llbracket G \rrbracket \circ \langle P_b, P_c \rangle, \llbracket F \rrbracket \circ \langle P_a, P_c \rangle) : \mathbf{Set}^{\mathscr{S}} \to \mathbf{Set}^{\mathscr{S}},$$

whereas $\llbracket \mathsf{G}' \rrbracket$ is the four-sorted functor

$$\llbracket \mathsf{G}' \rrbracket = (\llbracket G \rrbracket \circ \langle P_b, P_c \rangle, \llbracket F \rrbracket \circ \langle P_{\overline{s}}, \llbracket H \rrbracket \circ P_a \rangle, P_{\overline{s}}, \llbracket F \rrbracket \circ \langle P_a, P_c \rangle) : \mathbf{Set}^{\mathscr{S}'} \to \mathbf{Set}^{\mathscr{S}'}$$

where again the $\overline{s}$-component appears in the first position. The induced functor Pad then acts on G-coalgebras by

$$((X_a, X_b, X_c), (\xi_a, \xi_b, \xi_c)) \mapsto (\llbracket s \rrbracket (X_a, X_b, X_c), X_a, X_b, X_c), (id, \xi_a, \xi_b, \xi_c)),$$

that is, by inserting the identity on $\llbracket s \rrbracket (X_a, X_b, X_c)$ in the first component.

Our next goal is to establish that the semantics of modal logics is invariant under Pad, and we will begin by collecting together some simple properties.

Lemma 4.10. The functor Pad is full, faithful and injective on objects.

Proof. The second and third claim are clear. To see that Pad is full, let

$$f : \mathsf{Pad}(X, \xi) \to \mathsf{Pad}(Y, \zeta)$$

be a morphism of G'-coalgebras. Then $f = \mathsf{Pad}(f|_{\mathscr{S}})$, since, by the morphism property in the $\overline{s}$-component, the diagram

$$\begin{array}{ccccccc}
\llbracket s \rrbracket X & = & X_{\overline{s}} & \xrightarrow{f_{\overline{s}}} & Y_{\overline{s}} & = & \llbracket s \rrbracket Y \\
 & & \downarrow{\scriptstyle id} & & \downarrow{\scriptstyle id} & & \\
 & & \llbracket s \rrbracket X & \underset{= \llbracket \mathscr{S}' \rhd s \rrbracket f}{\xrightarrow{\llbracket \mathscr{S} \rhd s \rrbracket (f|_{\mathscr{S}})}} & \llbracket s \rrbracket Y & &
\end{array}$$

commutes. □

In other words, the category of G-coalgebras can be identified with the full subcategory of G'-coalgebras with an identity as the $\overline{s}$-component. We will show that this subcategory is reflective, which will give rise to a reflection functor Comp (*compose*) from G'-coalgebras

to G-coalgebras. For a G'-coalgebra (X,ξ), $\mathsf{Comp}(X,\xi)$ is defined as the G-coalgebra $(X|_{\mathscr{S}},\bar{\xi})$ where

$$\bar{\xi}_a = [\![t'_a]\!](\xi_{\bar{s}}, id_{X|_{\mathscr{S}}}) \circ \xi_a \text{ for } a \in \mathscr{S},$$

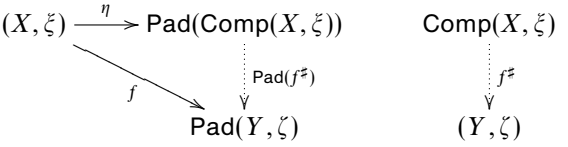
with type information visualised by

$$X_a \xrightarrow{\xi_a} [\![t'_a]\!]X \xrightarrow{[\![t'_a]\!](\xi_{\bar{s}}, id_{X|_{\mathscr{S}}})} [\![t'_a]\!]([\![s]\!](X|_{\mathscr{S}}), X|_{\mathscr{S}}) = [\![t_a]\!](X|_{\mathscr{S}}),$$

where the last equality is by the substitution lemma. We have a canonical map

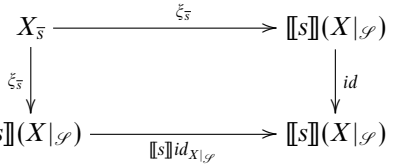
$$\eta = (\xi_{\bar{s}}, id_{X|_{\mathscr{S}}}) : (X,\xi) \to \mathsf{Pad}(\mathsf{Comp}(X,\xi)).$$

Lemma 4.11. The map η is a morphism of G'-coalgebras, and as such a reflective arrow. That is, every morphism of the form $f : \xi \to \mathsf{Pad}(Y,\zeta)$ factors uniquely as $f = \mathsf{Pad}(f^\sharp)\circ\eta$ for some $f^\sharp : \mathsf{Comp}(X,\xi) \to (Y,\zeta)$:

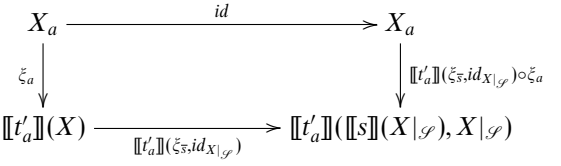


Specifically, $f^\sharp = f|_{\mathscr{S}}$.

Proof. We begin by checking that η is a coalgebra morphism. In the $\bar{s}$-component, the morphism property amounts to the commutation of



Moreover, in the a-component for $a \in \mathscr{S}$, the morphism property is the evident commutation of



Secondly, we have to verify the universal property. Let $f : (X,\xi) \to \mathsf{Pad}(Y,\zeta)$ for a G-coalgebra (Y,ζ). Then the only candidate for a morphism $f^\sharp : \mathsf{Comp}(X,\xi) \to (Y,\zeta)$ such that $\mathsf{Pad}(f^\#)\circ\eta = f$ is $f^\sharp = f|_{\mathscr{S}}$, as in the claim. To see that indeed $\mathsf{Pad}(f^\#)\circ\eta = f$, it remains only to check commutation in the $\bar{s}$-component, that is, $[\![s]\!](f|_{\mathscr{S}}) \circ \xi_{\bar{s}} = f_{\bar{s}}$. This follows from the morphism property of f, which in the $\bar{s}$-component amounts to

commutation of

$$\begin{array}{ccc} X_{\overline{s}} & \xrightarrow{\quad f_{\overline{s}} \quad} & [\![s]\!]Y \\ {\scriptstyle \xi_{\overline{s}}}\downarrow & & \downarrow{\scriptstyle id} \\ [\![s]\!](X|_{\mathscr{S}}) & \xrightarrow[[\![s]\!](f|_{\mathscr{S}})]{} & [\![s]\!]Y \end{array}$$

Moreover, we have to verify that $f^\sharp$ is a morphism $\mathsf{Comp}(X,\xi) \to (Y,\zeta)$. For $a \in \mathscr{S}$, we have to prove the commutation of

$$\begin{array}{ccc} X_a & \xrightarrow{\quad f_a \quad} & Y_a \\ {\scriptstyle [\![t'_a]\!](\xi_{\overline{s}}, id_{X|_{\mathscr{S}}})\circ\xi_a}\downarrow & & \downarrow{\scriptstyle \zeta_a} \\ [\![t'_a]\!]([\![s]\!](X|_{\mathscr{S}}), X|_{\mathscr{S}}) & \xrightarrow[=[\![t_a]\!](f|_{\mathscr{S}})]{[\![t'_a]\!]([\![s]\!](f|_{\mathscr{S}}), f|_{\mathscr{S}})} & [\![t'_a]\!]([\![s]\!]Y, Y) \end{array}$$

where the equality on the bottom arrow is by the substitution lemma. Now, by the previous diagram, the lower-left path in this diagram equals $[\![t'_a]\!](f) \circ \xi_a$, so commutation follows from the morphism property of $f : (X,\xi) \to \mathsf{Pad}(Y,\zeta)$ in the a-component. □

The previous lemma implies that Comp *extends to a functor from* G′*-coalgebras to* G*-coalgebras, which is a left adjoint of* Pad (see, for example, Adámek *et al.* (1990)). Since Pad is full and faithful, the arising co-unit $\epsilon : \mathsf{Comp} \circ \mathsf{Pad} \to id$ is necessarily an isomorphism; in fact, we have $\mathsf{Comp} \circ \mathsf{Pad} = id$, and ϵ is the identity.

Morally, the above means, in particular, that G-coalgebras are equivalent to G′-coalgebras with respect to the behaviour of states: G-coalgebras can, through Pad, be regarded as G′-coalgebras, and every G′-coalgebra has a behaviour-preserving map into a G-coalgebra. This will mean, in particular, that G′-coalgebras and G-coalgebras are equivalent with respect to satisfiability of modal formulae, which we will now make more precise.

Note that the types of G′ are essentially the same as the types of G, except that we have introduced a new base type $\overline{s}$ as a synonym for s, and removed some or all occurrences of the composite type s. The following lemma, which can be proved by a straightforward induction over typing derivations, states that G and G′ induce essentially the same modal syntax.

Lemma 4.12. The formulas of type $\overline{s}$ in G′ are precisely the formulas of type s in G; moreover, for $r \in \mathsf{Types}(\mathsf{G}') - \{\overline{s}\}$, the formulas of type r in G and G′ coincide.

In short, the formulas of G and G′ are the same up to interchange of the types s and $\overline{s}$, and we shall silently identify the two sets of formulas altogether. The same holds, *mutatis mutandis*, for proofs in G and G′. By the adjoint situation between Comp and Pad described above, together with the invariance of modal formulas under coalgebra morphisms (Proposition 4.6), it is immediate that the transition from G to G′ leaves modal semantics invariant.

Corollary 4.13. A formula is G′-satisfiable if and only if it is G-satisfiable. □

It is clear that the above-described process of promoting a type of a gluing to a new sort can be iterated. In particular, we can turn *all* composite types of a gluing into sorts, and thus obtain, given a gluing G, a flat gluing $\mathsf{G}^\flat$, which we call the *flattening* of G. Explicitly, $\mathsf{G}^\flat$ is described as follows. The set of sorts is $\mathscr{S}^\flat = \{\overline{s} \mid s \in \mathsf{Types}(\mathsf{G})\}$. For a feature expression over $\mathscr{S}$ of the form $t = F(s_1, \ldots, s_n)$, we write $t^\flat = F(\overline{s_1}, \ldots, \overline{s_n})$, and for $a \in \mathscr{S}$, we write $a^\flat = \overline{a}$. Then

$$\mathsf{G}^\flat = (u_{\overline{s}})_{\overline{s} \in \mathscr{S}^\flat}, \qquad \text{where } u_{\overline{s}} = \begin{cases} (t_a)^\flat & \text{for } s = a \in \mathscr{S} \\ s^\flat & \text{otherwise.} \end{cases}$$

Example 4.14. Given the gluings

$$G_1 \equiv (o \to \mathsf{S}(\mathsf{U}(o), \mathsf{N}_A(o))$$
$$G_2 \equiv (o \to \mathsf{S}(u, n), u \to \mathsf{U}(o), n \to \mathsf{N}_A(o))$$

from Example 3.6, G_2 is the flattening of G_1, up to renaming the sorts $\overline{o}$, $\overline{\mathsf{U}(o)}$ and $\overline{\mathsf{N}_A(o)}$ of the flattening into o, u and n, respectively.

By an iterated application of Lemma 4.12 (and the following comment), *the flattening* $\mathsf{G}^\flat$ *syntactically induces the same logic as* G, that is, the types, formulas and proof systems coincide (up to renaming s into $\overline{s}$ for $s \in \mathsf{Types}(\mathsf{G})$). Moreover, iterated application of Corollary 4.13 yields our main result.

Theorem 4.15. A formula is G-satisfiable if and only if it is $\mathsf{G}^\flat$-satisfiable.

Note that the order in which composite types are promoted to sorts in the formation of the flattening of a gluing G does not matter – not only is it clear that the resulting flat gluing $\mathsf{G}^\flat$ is independent of this order, but also that the resulting embedding functor from G-coalgebras into $\mathsf{G}^\flat$-coalgebras is insensitive to the order of steps: the embedding is composed of the relevant Pad functors, and hence, in any case, just adds identity components for all new sorts. The uniqueness of left adjoints then implies that the reflection functor from $\mathsf{G}^\flat$-coalgebras to G-coalgebras that arises from the relevant Comp functors is independent of the order of steps. We therefore use Pad and Comp again to denote the embedding and reflection functors, respectively, between G-coalgebras and $\mathsf{G}^\flat$-coalgebras; note that in this setting, Comp involves recursive composition of coalgebra maps. The following theorem records this explicitly.

Theorem 4.16. Via Pad, the G-coalgebras form a full reflective subcategory of the $\mathsf{G}^\flat$-coalgebras, with Comp as reflector.

In our running example, the situation is as follows.

Example 4.17. Consider the gluings G_1 and G_2 over $\mathscr{S} = \{o, u, n\}$ from Example 3.6 and recall from Example 4.14 that $\mathsf{G}_1^\flat = \mathsf{G}_2$. Let

$$C = (Y, \zeta : X \to \mathscr{D}X + \mathscr{P}(A \times X))$$

be a G_1-coalgebra. Then C can be converted into a G_2-coalgebra $\mathsf{Pad}(C) = (X, \xi)$ where

$$\begin{array}{ll} X_o = Y & \xi_o = \zeta \\ X_u = \mathscr{D}Y & \xi_u = id_{X_u} \\ X_n = \mathscr{P}(A \times Y) & \xi_n = id_{X_n}. \end{array}$$

That is, Pad just adds identities at the new sorts u, n.

Conversely, given a G_2-coalgebra $D = (X, \xi)$, we construct a G_1-coalgebra $\mathsf{Comp}(D) = (Y, \zeta)$ (which has a single sort, o) by putting $Y_o = X_o$ and $\zeta_o = (\xi_u + \xi_n) \circ \xi_o$. The triple (id_{X_o}, ξ_u, ξ_s) is a coalgebra morphism $D \to \mathsf{Pad}(\mathsf{Comp}(D))$, the unit of the adjunction between Pad and Comp.

Using general properties of reflective subcategories, we obtain the following from Theorem 4.16.

Corollary 4.18. If C is a final G-coalgebra, then $\mathsf{Pad}(C)$ is a final $\mathsf{G}^\flat$-coalgebra. Conversely, if D is a final $\mathsf{G}^\flat$-algebra, then D is isomorphic to $\mathsf{Pad}(\mathsf{Comp}(D))$, and $\mathsf{Comp}(D)$ is a final G-coalgebra.

Remark 4.19. In the case of polynomial functors, the dual of the previous corollary is semi-folklore: nested recursive datatypes can be flattened into mutually recursive datatypes, as done in the theorem prover Isabelle/HOL (see, for example, Berghofer and Wenzel (1999), and references therein). We do not know of a formal treatment of this fact at the level of generality we employ here, nor of a statement of the dual of Theorem 4.16 in the literature.

Remark 4.20. If a final $\mathsf{G}^\flat$-coalgebra exists, one can conclude Theorem 4.15 from Corollary 4.18 alone. However, in many cases of interest, final coalgebras do not exist (for example, even in the basic case of unbounded non-determinism).

Remark 4.21. Although, principally with readability in mind, we have restricted the current development to purely modal logics, we should point out that Theorem 4.16 also implies a general invariance result for coalgebraic *hybrid* logics, that is, logics that include additional expressive means for individual states (Myers *et al.* 2009; Schröder *et al.* 2009; Goré *et al.* 2010b), as long as these are restricted to states that live in the base types of the original gluing G: while such hybrid logics are no longer invariant under arbitrary coalgebra morphisms, they are certainly invariant under coalgebra homomorphisms that are carried by bijective maps at the base types, as is the case for the unit and the counit of the adjunction of Theorem 4.16.

Remark 4.22. Our definition of flat gluings admits *unguarded* sorts, that is, in a flat gluing $\mathsf{G} = (t_a)_{a \in \mathscr{S}}$ we may have $t_b = c$ for some $b, c \in \mathscr{S}$. One can turn such gluings into *guarded* flat gluings (where all sorts are associated to the composite feature expressions), essentially getting rid of the sort b, as follows. We have to distinguish two subcases:

— If b is distinct from c, we can form a new gluing $\mathsf{G}^{-b} = (t_a[c/b])_{a \in \mathscr{S}-\{b\}}$, and the equivalence of G and G^{-b} is then just a special case of Corollary 4.13, applied to $s = c$ and with $\overline{c}$ renamed to b.
— The slightly more complicated case is where $b = c$, for which we can form the new gluing

$$\mathsf{G}^{-b} = (t_a[\mathsf{1}/b])_{a \in \mathscr{S}-\{b\}},$$

where $\mathsf{1}$ is the (trivial) output feature for the singleton output set 1. It is easy to check the equivalence of G^{-b} with G. The gluing G^{-b} is not in general flat, but can, of course, be flattened into a guarded flat gluing, which in the end means that we replace all sorts b such that $b \to b \in \mathsf{G}$ with a single sort $\overline{\mathsf{1}}$ that we associate with the feature expression $\mathsf{1}$ (which by the definition given in Section 3 is composite).

The intuition behind these constructions is that if $b \to c \in \mathsf{G}$ for $b \neq c$, then the observations on b are precisely those that arise by moving to c and then making observations on c, so b may be identified with c. If, on the other hand, $b \to b \in \mathsf{G}$, no observations at all can be made on b, so b may be identified with $\mathsf{1}$.

Remark 4.23. It is evident that the preceding development is entirely independent of the choice of **Set** as the base category, except that the definition of predicates and predicate liftings as presented here is somewhat set-oriented. A generalisation to arbitrary base categories would, for example, work with fibrations of predicates. Indeed one does not even need to assume that all sorts live in the same base category. The core message of our flattening construction for a gluing G is that G-coalgebras can be regarded as particular $\mathsf{G}^\flat$-coalgebras, and every behaviour that is realised in a G-coalgebra is, through the reflective arrow into the corresponding $\mathsf{G}^\flat$-coalgebra, realised also in a $\mathsf{G}^\flat$-coalgebra. Summarising, G-*coalgebras and* $\mathsf{G}^\flat$-*coalgebras realise the same behaviours*. From the perspective of coalgebraic modal logics, this will mean that over any base category (or base categories), modal semantics will be invariant under the transition from gluings to their flattenings as long as the logic is *adequate*, that is, invariant under coalgebra morphisms.

As indicated above, the main benefit of the flattening result is that functor composition vanishes from the picture, and is replaced by multi-sortedness. This means that results that have earlier been phrased as compositionality results, that is, results that reduce properties of composite functors to properties of their components, now just become generalisations of single-sorted results to the multi-sorted setting, which typically require no more than the annotation of previous results and proofs with sort indices. Roughly speaking, any result on coalgebraic modal logic (and more generally any property of coalgebras that is invariant under behavioural equivalence) will generalise to multi-sorted coalgebras provided it uses only such properties of the base category that are stable under products of categories.

We conclude this section with a further brief comment on the relationship between Segala systems and strictly-alternating systems.

Example 4.24. The flattening of the gluing $n \to \mathsf{N}_A(\mathsf{U}(n))$ is the gluing $(n \to \mathsf{N}_A(p), p \to \mathsf{U}(n))$ (Example 3.6). As discussed in Example 4.5, the corresponding coalgebras are Segala systems and strictly alternating systems, respectively. The relationship between Segala systems and strictly alternating systems has received some recent attention (Segala and Turrini 2005; Segala 2006). Our results show that both types of systems admit an interpretation of the same variant of probabilistic modal logic (Section 2) and validate the same formulas of this logic. Moreover, the notion of behavioural equivalence on the sort of non-deterministic states (which unlike the sort of probabilistic states is explicitly present in both types of systems) is the same in both cases, and since the distribution functor preserves weak pullbacks (Moss 1999; de Vink and Rutten 1999; Sokolova *et al.* 2009), this equivalence extends to coalgebraic bisimulation, which captures standard notions of strong bisimulation. In particular, the notions of strong bisimulation coincide for Segala systems and strictly alternating systems, which is a result that appears not to be claimed in Segala and Turrini (2005).

5. Applications to model construction and complexity

We have seen in Section 3 that the same multi-sorted logic can arise from different gluings of given features, where the difference only manifests itself on a semantic level. The different interpretations of the logic are related by Theorem 4.15, which shows that the satisfiability problem for a given gluing is equivalent to that of its *flattening*. We now show that the generic shallow model construction and the ensuing PSPACE decision procedure from Schröder and Pattinson (2009) generalise to flat gluings, which enables us to derive upper PSPACE bounds for arbitrary gluings, and, in particular, for heterogeneous logics equipped with their standard single-sorted semantics as in Section 2. The entire point of this exercise is to show that the generalisation from the single-sorted case to the multi-sorted case amounts to no more than some additional bookkeeping for the sorts; the minor differences that do exist between the proof presented below and the one given in Schröder and Pattinson (2009) are entirely due to presentational choices made in the context of *op. cit.*

The shallow model construction requires that the structures involved are strictly one-step complete in the following sense, where the notation $[\![\_]\!]_{X,\tau}$ and $[\![\_]\!]_{TX,\tau}$ is as in Definition 4.4. Note that (strict) one-step completeness implies weak completeness of the rule system (Pattinson 2004; Schröder 2007).

Definition 5.1. An n-ary feature F is *strictly one-step complete* for a structure

$$T = [\![\mathsf{F}]\!] : \mathbf{Set}^n \to \mathbf{Set}$$

if, whenever

$$[\![\chi]\!]_{TX,\tau} = T(X_1, \ldots, X_n)$$

for a sorted set $V = (V_1, \ldots, V_n)$ of variables, an assignment τ of subsets $\tau(a) \subseteq X_i$ to variables $a \in V_i$, and a clause χ over atoms of the form $\heartsuit(a_{i_1}, \ldots, a_{i_k})$, where $\heartsuit : i_1, \ldots, i_k \to *$ in F and $a_{i_j} \in V_{i_j}$, then χ is provable from propositional formulas valid under τ using a single instance of a rule of F. Formally, χ is propositionally entailed by a clause $\psi\sigma$ (that

is, either χ is a propositional tautology, or χ contains all literals of $\psi\sigma$), where $(\phi_i)/\psi$ is a rule of F and σ is a $(V_1,\dots,V_n)$-substitution (that is, $\sigma(a) \in V_i$ for $a \in V_i$) such that $[\![\phi_i\sigma]\!]_{X,\tau} = X_i$ for all i.

Remark 5.2. We can assume in the above definition that all V_i are finite and that χ is not a propositional tautology.

One can show, analogously to the single-sorted case (Schröder 2007), that the set of all one-step sound rules for a given F-structure is strictly one-step complete, so that strictly one-step complete axiomatisations always exist. Formally, we have the following proposition.

Proposition 5.3. Suppose $\mathsf{F} = (\Lambda, \mathscr{R})$ is an n-ary feature with an associated structure over $[\![\mathsf{F}]\!] : \mathbf{Set}^n \to \mathbf{Set}$. If $\mathscr{R}$ consists of all one-step rules that are one-step sound for $[\![\mathsf{F}]\!]$, then F is strictly one-step complete for $[\![\mathsf{F}]\!]$.

Proof. Let $T = [\![\mathsf{F}]\!]$ be the underlying functor of the given structure for F. We assume $[\![\chi]\!]_{X,\tau} = X$ with X, τ, χ, V as in Definition 4.4, and that V is finite in each sort. We let ϕ_i, for $1 \leqslant i \leqslant n$, be the propositional theory of τ over i, that is, the finite conjunction of all clauses ρ over V_i such that $[\![\rho]\!]_{X,\tau} = X_i$. We will now show that the one-step rule $R \equiv (\phi_1;\dots;\phi_n)/\chi$ over V is one-step sound. It will then follow that χ is derivable, as required.

So we let $Y \in \mathbf{Set}^n$ and σ be a $\mathscr{P}Y$-valuation such that $[\![\phi_i]\!]_\sigma = Y_i$ for all $i = 1,\dots,n$. We have to show $[\![\chi]\!]_\sigma = TY$. We claim that for each i and each $y \in Y_i$, there exists $x \in X_i$ such that

$$x \in \tau(v) \iff y \in \sigma(v)$$

for all $v \in V_i$. In order to show a contradiction, we assume that $i \in \{1,\dots,n\}$ violates this property. Take ρ to be the propositional theory of y under σ, that is, the (finite) conjunctive clause

$$\rho = \bigwedge_{\{v \in V_i \mid y \in \sigma(v)\}} v \wedge \bigwedge_{\{v \in V_i \mid y \notin \sigma(v)\}} \neg v.$$

Then $[\![\neg\rho]\!]_{X,\tau} = X_i$ by assumption, and $y \in [\![\rho]\!]\sigma$ by construction, which contradicts $[\![\phi_i]\!]_\sigma = Y_i$.

Thus, we have $f_i : Y_i \to X_i$ such that

$$\sigma(v) = f_i^{-1}[\tau(v)] \quad \text{for all } v \text{ in } V_i.$$

By the naturality of predicate liftings, and since preimages commute with intersections and complements, we now have

$$[\![\chi]\!]_\sigma = (Tf)^{-1}[[\![\chi]\!]\tau],$$

where $f = (f_i)_{1 \leqslant i \leqslant n}$, and since $[\![\chi]\!]_{TX,\tau} = TX$, we can conclude that $[\![\chi]\!]_\sigma = TY$, as required. □

Schröder and Pattinson (2009) described a systematic procedure for obtaining strictly one-step complete rule sets, which was called *rule resolution*, and can be straightforwardly generalised to the multi-sorted setting.

Throughout this section, we fix a gluing G of a set Φ of features over a set $\mathscr{S}$ of sorts, and we assume that every feature is equipped with a structure.

Definition 5.4. The set $MA(\phi)$ of *modal atoms* of an G-formula ϕ is defined recursively by

$$MA(\phi \wedge \psi) = MA(\phi) \cup MA(\psi)$$
$$MA(\neg\phi) = MA(\phi)$$
$$MA(\heartsuit(\rho_1, \ldots, \rho_n)) = \{\heartsuit(\rho_1, \ldots, \rho_n)\}.$$

A *pseudovaluation* for ϕ is a subset H of $MA(\phi)$. We define satisfaction of propositional formulas χ over $MA(\phi)$ by H ($H \models \chi$) inductively in the obvious way, with $H \models \chi \iff \chi \in H$ for $\chi \in MA(\phi)$.

Assuming

$$s = \mathsf{F}(s_1, \ldots, s_n) \in \mathsf{Types}(\mathsf{G})$$

if s is composite and

$$a \to \mathsf{F}(s_1, \ldots, s_n) \in \mathsf{G}$$

if $s = a$ is a base type, we say that a rule $R = (\phi_1; \ldots; \phi_n)/\psi$ associated with the feature F *matches* a pseudovaluation H for $\phi : s$ if there is a substitution σ such that $\psi\sigma$ is a clause over $MA(\phi)$ with $H \not\models \psi\sigma$. In this case, the pair (R, σ) is said to be a *matching* of H.

The significance of rule matching arises in relation to satisfiability checking. Essentially, a matching rule for H is one that threatens to prove *unsatisfiability* of H. In more detail, given a pseudovaluation H and a rule $R = (\phi_1; \ldots; \phi_n)/\psi$ that matches H through a substitution σ, H can only be satisfiable if $\neg\psi\sigma$ is satisfiable, which by soundness of the rule can only be the case if one of the premises of R, instantiated by σ, is not valid, that is, if one of the $\neg\phi_i\sigma$ is satisfiable. This observation is the basis of both the shallow model construction and the ensuing decision procedure. The completeness of this principle is guaranteed by strict one-step completeness.

More explicitly, our shallow model theorem now takes the following form.

Theorem 5.5. If every feature in G is strictly one-step complete, then a formula $\phi : s$ is satisfiable in a G-model if and only if $H \models \phi$ for some pseudovaluation H for ϕ such that for every matching $((\phi_1; \ldots; \phi_n)/\psi, \sigma)$ of H, one of the formulas $\neg\phi_i\sigma$ is satisfiable.

Proof. By Theorem 4.15 and Remark 4.22, we can assume that $\mathsf{G} = (t_a)_{a \in \mathscr{S}}$ is a guarded flat gluing, that is, $\mathscr{S} = \mathsf{Types}(\mathsf{G})$, and, in particular, $s = b$ for some sort b and

$$b \to \mathsf{F}(a_1, \ldots, a_n) \in \mathsf{G}$$

for some n-ary feature F and sorts $a_1, \ldots, a_n$.

— **Only if:**
If $x \models^b_C \phi$, let

$$H = \{\chi \in MA(\phi) \mid x \models^b_C \chi\}.$$

Then use the soundness of the rules.

— **If:**
Let Π be a set of formulas consisting of one satisfiable formula $\neg\phi_i\sigma : a_i$ for each

matching $((\phi_1;\ldots;\phi_n)/\psi,\sigma)$ of H, which is guaranteed to exist by assumption. For each $\pi \in \Pi$, there exists a G-coalgebra

$$C_\pi = (X^\pi, \xi^\pi) = ((X^\pi_a), (\xi^\pi_a)_{a\in\mathscr{S}})$$

and $x^\pi \in X^\pi_{a_i}$ such that $x^\pi \models^{a_i}_{C^\pi} \pi$. We can assume that for all a, the X^π_a are pairwise disjoint.

Now put

$$X_a = \bigcup_{\pi\in\Pi} X^\pi_a \text{ for } a \neq b$$
$$X_b = \{x_0\} \cup \bigcup_{\pi\in\Pi} X^\pi_b,$$

where x_0 is a fresh element. Let Y be the $\mathscr{S}$-sorted set of the x^π, $\pi \in \Pi$. For $\chi : a$, put

$$\hat{\chi} = [\![\chi]\!]_{C^\pi} \cap Y_a.$$

We define a G-coalgebra

$$C = (X, \xi) = ((X_a), (\xi_a))$$

as follows. For $x \in X^\pi_a$, we put

$$\xi_a(x) = \xi^\pi_a(x) \in [\![t_a]\!]X^\pi \subseteq [\![t_a]\!]X.$$

Then (independently of the pending definition of $\xi_b(x_0)$) for $\chi : a$ and $x \in X^\pi_a$, we have

$$x \models^a_C \chi \iff x \models^a_{C^\pi} \chi \tag{1}$$

as C^π is a subcoalgebra of C. In particular, $\hat{\chi} = [\![\chi]\!]_C \cap Y_a$ for all $\chi : a$.

Crucially, we will show that there exists $r \in [\![\mathsf{F}]\!]Y$ such that for $\heartsuit(\rho_1,\ldots,\rho_k)$ in $MA(\phi)$,

$$r \in [\![\heartsuit]\!](\hat{\rho}_1,\ldots,\hat{\rho}_k) \iff \heartsuit(\rho_1,\ldots,\rho_k) \in H. \tag{2}$$

Putting $\xi_b(x_0) = r$, we then obtain

$$x_0 \models_C \rho \iff H \models \rho \tag{3}$$

for all propositional formulas ρ over $MA(\phi)$: this follows immediately from (1) and (2) for $\rho \in MA(\phi)$, and then extends easily to $\mathsf{Prop}(MA(\phi))$. By (3), we have, in particular, $x_0 \models_C \phi$, and we are done.

We still need to prove that r satisfying (2) exists. To derive a contradiction, we assume the contrary. Let V be the set of propositional variables b_ρ indexed over all formulas ρ that occur as arguments of modal atoms in ϕ. That is, $\rho = \rho_i$ for some $\heartsuit(\rho_1,\ldots,\rho_k) \in MA(\phi)$ and some $i = 1,\ldots,k$. Let the clause θ over $MA(\phi)$ consist of the literals $\neg\heartsuit(b_{\rho_1},\ldots,b_{\rho_k})$ for $\heartsuit(\rho_1,\ldots,\rho_k) \in H$ and $\heartsuit(b_{\rho_1},\ldots,b_{\rho_k})$ for $\heartsuit(\rho_1,\ldots,\rho_k) \in MA(\phi) - H$. That is, using σ to denote the substitution given by

$$\sigma(b_\rho) = \rho \text{ for all } b_\rho \in V,$$

we have

$$H \models \neg\theta\sigma. \tag{4}$$

Now, by our assumption of the non-existence of r satisfying (2),

$$[\![\theta]\!]_{[\![\mathsf{F}]\!]Y,\tau_Y} = [\![\mathsf{F}]\!]Y,$$

where τ_Y is the $\mathscr{P}Y$-valuation given by

$$\tau_Y(b_\rho) = \hat{\rho}.$$

By strict one-step completeness of F, it follows that θ is derivable using a single rule of F from propositional formulas over V that are valid under τ_Y. Explicitly, there exists a rule $(\phi_1;\ldots;\phi_n)/\psi$ in F and a substitution η such that

$$[\![\phi_i\eta]\!]_{Y,\tau_Y} = Y_{a_i} \text{ for } i = 1,\ldots,n \tag{5}$$

and $\psi\eta$ propositionally entails θ, that is, assuming, without loss of generality, that θ is not a propositional tautology, θ contains $\psi\eta$. In particular, $\psi\eta$ is a clause over $MA(\phi)$. We now prove that

$$H \models \psi\eta\sigma, \tag{6}$$

where σ is the substitution taking b_ρ to ρ. This will complete the proof as it implies $H \models \theta\sigma$, in contradiction to (4).

We proceed to prove (6) by contradiction, so we assume that $H \not\models \psi\eta\sigma$. By construction of Π, it follows that $\pi :\equiv \neg\phi_i\eta\sigma \in \Pi$ for some i; recall that $\pi : a_i$ and $x^\pi \models^{a_i}_{C^\pi} \pi$. Let τ^π denote the $\mathscr{P}(X^\pi)$-valuation taking b_ρ to $[\![\rho]\!]_{C^\pi}$. By the naturality of predicate liftings, (5) implies $x^\pi \in [\![\phi_i\eta]\!]_{X^\pi,\tau^\pi}$, because $\tau_Y(b_\rho) = \tau^\pi(b_\rho) \cap Y_a$ for $\rho : a$. But since $[\![\phi_i\eta]\!]_{X^\pi,\tau^\pi} = [\![\phi_i\eta\sigma]\!]_{C^\pi}$, we have arrived at a contradiction to $x^\pi \models^{a_i}_{C^\pi} \pi \equiv \neg\phi_i\eta\sigma$. □

From Theorem 5.5, we obtain a multi-sorted version of the generic PSPACE decision procedure of Schröder and Pattinson (2009). This requires the computation of matchings of given pseudovaluations, and we require that the rules associated with features are *closed under contraction*, that is, it suffices to consider matchings $((\phi_1,\ldots,\phi_n)/\psi,\sigma)$ where $\psi\sigma$ does not contain duplicate literals, with the consequence that there are only finitely many matches to check in every recursion step. Formally, these notions are defined as follows.

Definition 5.6. We say a clause is *contracted* if all its literals are distinct. An instance of a rule is *contracted* if its conclusion is a contracted clause, and a matching is contracted if the corresponding rule instance is contracted. Finally, a feature $\mathsf{F} = (\Lambda,\mathscr{R})$ is *closed under contraction* if every application of a rule instance with duplicate literals in the conclusion can be replaced by a contracted instance of a (possibly different) rule. Formally, for every V-instance $R\sigma$ of a rule $R \equiv (\phi_1;\ldots;\phi_n)/\psi$ over V in $\mathscr{R}$, there exists a contracted V-instance $R'\sigma'$ of a rule $R' \equiv (\phi'_1;\ldots;\phi'_n)/\psi' \in \mathscr{R}$ such that $\psi\sigma$ and $\psi'\sigma'$ are propositionally equivalent and for all i, $\phi'_i\sigma'$ is propositionally entailed by $\phi_i\sigma$.

To close a feature $\mathsf{F} = (\Lambda,\mathscr{R})$ under contraction, we just add a rule $(\phi'_1;\ldots;\phi'_n)/\psi'$ for every rule $(\phi_1;\ldots;\phi_n)/\psi$ over V in $\mathscr{R}$ and every V-substitution σ, where ϕ'_i is some suitably chosen propositional equivalent of $\phi_i\sigma$ and ψ' is obtained from $\psi\sigma$ by removing duplicate literals. It is clear that the resulting set of rules inherits one-step soundness and strict one-step completeness from $\mathscr{R}$.

The following corollary follows immediately from Theorem 5.5.

Corollary 5.7. If every feature in G is strictly one-step complete and closed under contraction, then a formula $\phi : s$ is satisfiable over a $[\![\mathsf{G}]\!]$-structure if and only if $H \models \phi$ for some pseudovaluation H for ϕ such that for every contracted matching $((\phi_1; \ldots; \phi_n)/\psi, \sigma)$ of H, one of the formulas $\neg\phi_i\sigma$ is satisfiable.

Since rules are generally too large to pass around directly, we will assume that every rule is represented by a *code*. For the features discussed in Example 3.2, the codes can be taken as the parameters of the rules. Formal definitions are as follows.

Definition 5.8. Let $\mathsf{F} = (\Lambda, \mathscr{R})$ be a feature. We assume that the rules in $\mathscr{R}$ are represented by *codes*, that is, strings over a given finite alphabet. Two rules are *equivalent* if their conclusions are identical and their premises are propositionally equivalent.

Under the assumptions of the above corollary, we have the following algorithm on an alternating Turing machine (Chandra and Stockmeyer 1981), which generalises an algorithm from Vardi (1989). Essentially, the algorithm just applies the satisfiability criterion of Corollary 5.7 recursively, decreasing the modal depth of the formula in every recursion step. To achieve polynomial space usage, the algorithm proceeds by depth-first search. Technically, this is encapsulated in the use of alternation.

Algorithm 5.9. To decide satisfiability of a G-formula $\phi : s$, where $s = \mathsf{F}(s_1, \ldots, s_n) \in \mathsf{Types}(\mathsf{G})$ if s is composite and $a \to \mathsf{F}(s_1, \ldots, s_n) \in \mathsf{G}$ if $s = a$ is a base type:

(1) **Initialise:** Construct the set $MA(\phi)$.
(2) **Existential:** Guess a pseudovaluation H for ϕ with $H \models \phi$.
(3) **Universal:** Non-deterministically choose a contracted matching $((\phi_1; \ldots; \phi_n)/\psi, \sigma)$ of H.
(4) **Existential:** Guess $i \in \{1, \ldots, n\}$ and a clause χ from the conjunctive normal form (CNF) of ϕ_i and recursively check that $\neg\chi\sigma : s_i$ is satisfiable.

The algorithm succeeds if all possible choices at steps marked *universal* lead to successful termination, and for all steps marked *existential*, there exists a choice leading to successful termination. Note that the algorithm terminates successfully in Step (3) if there are no matching rules. In particular, the algorithm terminates either in Step (2) or in Step (3) if ϕ has rank 0. Step (3) breaks down into choosing a contracted clause ρ over $MA(\phi)$ such that $H \models \neg\rho$, and then choosing a code of a rule that matches ρ. Note that it suffices to guess one rule from each equivalence class of rules (Definition 5.8).

The crucial requirement for the effectivity of Algorithm 5.9 is that Steps (3) and (4) can be performed in polynomial time – the central condition is that we only need to guess rule codes of polynomially bounded size in Step (3). Formally, we have the following set of conditions.

Definition 5.10. A feature $\mathsf{F} = (\Lambda, \mathscr{R})$ is PSPACE-*tractable* if all contracted matchings have a code of polynomially bounded size, and it can be decided in NP:

— whether a given code is the code of some rule in $\mathscr{R}$;
— whether a rule matches a given pseudovaluation; and
— whether a clause belongs to the CNF of a given premise of a given rule.

Under these conditions, Algorithm 5.9 runs in polynomial time, which proves the following theorem since APTIME = PSPACE (Chandra and Stockmeyer 1981) – the details are as in Schröder and Pattinson (2009).

Theorem 5.11 (Space complexity). If every feature in G is strictly one-step complete, closed under contraction and PSPACE-tractable, then the satisfiability problem for $\mathscr{F}(\mathsf{G})$-formulas over G-coalgebras is in PSPACE.

All the rule sets presented in Figure 1 are strictly one-step complete, closed under contraction and PSPACE-tractable (this is either obvious or shown in Schröder and Pattinson (2009)). Thus, Theorem 5.11 guarantees that satisfiability for logics arising through arbitrary gluings of the features from Example 3.2 are in PSPACE.

Remark 5.12. The recursive structure of the algorithm allows for a modular implementation that interconnects separate matching routines for each feature. In particular, the same algorithmic structure may alternatively be applied to effective heuristic matching routines, leading to more efficient overall reasoning.

6. Conclusions

We have introduced a calculus of *gluings*, which describe ways of combining logical features such as uncertainty, non-determinism and choice. We have shown that the satisfaction problem of a gluing is in PSPACE if this is true for the features involved. This has been achieved by equipping the logics under consideration with a multi-sorted coalgebraic semantics. Crucially, we have shown that the satisfiability problem for a gluing is equivalent to that for a corresponding flattened gluing; flat gluings are technically more tractable than general gluings, and essentially allow a straightforward generalisation of arbitrary generic results from single-sorted coalgebraic logic by just adding multi-sorted notation. As an example, we have carried this out in detail for the generic PSPACE algorithm for satisfiability (Schröder and Pattinson 2009). However, the same principle applies without further ado to other results such as generic EXPTIME algorithms for TBox reasoning in coalgebraic modal and hybrid logics (Schröder *et al.* 2009; Goré *et al.* 2010a; Goré *et al.* 2010b) or Hennessy–Milner-type expressivity results (Pattinson 2004; Schröder 2007); in fact, Schröder *et al.* (2009) makes extensive use of this mechanism in the examples. Our results thus pave the way for modularised tool support for a large class of heterogeneous logics. The nature of the type of logic combination studied here is essentially that of a typed fusion, but the study of further combination mechanisms such as products or $\mathscr{E}$-connections (Kutz *et al.* 2004) in the coalgebraic framework is the subject of future work. A further point of interest is the imposition of frame conditions on heterogeneous coalgebras, thus obtaining axiomatic interaction between the component logics – some initial results along these lines may be found in Pattinson and Schröder (2008). The modularity mechanisms described here are the basis of proof support for heterogeneous modal logics in the generic modal proof tool CoLoSS (Coalgebraic Logic Satisfiability Solver) – see Calin *et al.* (2009).

Acknowledgements

The authors wish to thank the anonymous referees for helpful comments, and Erwin R. Catesbeiana for suggestions on multi-sorted consistency.

References

Adámek, J., Herrlich, H. and Strecker, G. E. (1990) *Abstract and Concrete Categories*, Wiley Interscience.

Alur, R., Henzinger, T. A. and Kupferman, O. (2002) Alternating-time temporal logic. *Journal of the ACM* **49** 672–713.

Barr, M. (1993) Terminal coalgebras in well-founded set theory. *Theoretical Computer Science* **114** 299–315.

Bartels, F., Sokolova, A. and de Vink, E. (2003) A hierarchy of probabilistic system types. In: Gumm, H.-P. (ed.) Coalgebraic Methods in Computer Science, CMCS 03. *Electronic Notes in Theoretical Computer Science* **82**.

Berghofer, S. and Wenzel, M. (1999) Inductive datatypes in HOL – lessons learned in formal-logic engineering. In: Bertot, Y., Dowek, G., Hirschowitz, A., Paulin, C. and Théry, L. (eds.) Theorem Proving in Higher Order Logics, TPHOLs 1999. *Springer-Verlag Lecture Notes in Computer Science* **1690** 19–36.

Blackburn, P., de Rijke, M. and Venema, Y. (2001) *Modal Logic*, Cambridge University Press.

Calin, G., Myers, R., Pattinson, D. and Schröder, L. (2009) CoLoSS: The Coalgebraic Logic Satisfiability Solver (system description). In: Areces, C. and Demri, S. (eds.) Methods for Modalities, M4M-5, 2007. *Electronic Notes in Theoretical Computer Science* **231** 41–54.

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

Chellas, B. (1980) *Modal Logic*, Cambridge University Press.

Cîrstea, C. and Pattinson, D. (2007) Modular construction of modal logics. *Theoretical Computer Science* **388** (1-3) 83–108.

D'Agostino, G. and Visser, A. (2002) Finality regained: A coalgebraic study of Scott-sets and multisets. *Archive for Mathematical Logic* **41** 267–298.

de Vink, E. and Rutten, J. (1999) Bisimulation for probabilistic transition systems: A coalgebraic approach. *Theoretical Computer Science* **221** 271–293.

Fine, K. (1972) In so many possible worlds. *Notre Dame Journal of Formal Logic* **13** 516–520.

Goré, R., Kupke, C. and Pattinson, D. (2010a) Optimal tableau algorithms for coalgebraic logics. In: Esparza, J. and Majumdar, R. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, TACAS 10. *Springer-Verlag Lecture Notes in Computer Science* **6015** 114–128.

Goré, R., Kupke, C., Pattinson, D. and Schröder, L. (2010b) Global caching for coalgebraic description logics. In: Giesl, J. and Haehnle, R. (eds.) International Joint Conference on Automated Reasoning, IJCAR 2010. *Springer-Verlag Lecture Notes in Computer Science* **6173** 46–60.

Halpern, J. Y. (2003) *Reasoning About Uncertainty*, MIT Press.

Hansson, H. and Jonsson, B. (1990) A calculus for communicating systems with time and probabilities. In: *Real-Time Systems, RTSS 90*, IEEE 278–287.

Hansson, H. A. (1994) *Time and Probability in Formal Design of Distributed Systems*, Elsevier.

Heifetz, A. and Mongin, P. (2001) Probabilistic logic for type spaces. *Games and Economic Behavior* **35** 31–53.

Hemaspaandra, E. (1994) Complexity transfer for modal logic. In: Abramsky, S. (ed.) *Logic in Computer Science, LICS 94*, IEEE 164–173.

Jacobs, B. (2001) Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications* **35** 31–59.

Jonsson, B., Yi, W. and Larsen, K. G. (2001) Probabilistic extensions of process algebras. In: Bergstra, J., Ponse, A. and Smolka, S. (eds.) *Handbook of Process Algebra*, Elsevier.

Kurucz, A. (2006) Combining modal logics. In: van Benthem, J., Blackburn, P. and Wolter, F. (eds.) *Handbook of Modal Logic*, Elsevier.

Kutz, O., Lutz, C., Wolter, F. and Zakharyaschev, M. (2004) $\mathscr{E}$-connections of abstract description systems. *Artificial Intelligence* **156** 1–73.

Larsen, K. and Skou, A. (1991) Bisimulation through probabilistic testing. *Information and Computation* **94** 1–28.

Lewis, D. (1975) Intensional logics without iterative axioms. *Journal of Philosophical Logic* **3** 457–466.

Moss, L. (1999) Coalgebraic logic. *Annals of Pure and Applied Logic* **96** 277–317.

Mossakowski, T., Schröder, L., Roggenbach, M. and Reichel, H. (2006) Algebraic-coalgebraic specification in CoCasl. *Journal of Logic and Algebraic Programming* **67** 146–197.

Myers, R., Pattinson, D. and Schröder, L. (2009) Coalgebraic hybrid logic. In: de Alfaro, L. (ed.) Foundations of Software Science and Computation Structures, FoSSaCS 2009. *Springer-Verlag Lecture Notes in Computer Science* **5504** 137–151.

Olivetti, N., Pozzato, G. L. and Schwind, C. (2007) A sequent calculus and a theorem prover for standard conditional logics. *ACM Transactions on Computational Logic* **8**.

Pattinson, D. (2004) Expressive logics for coalgebras via terminal sequence induction. *Notre Dame Journal of Formal Logic* **45** 19–33.

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. *Springer-Verlag Lecture Notes in Computer Science* **4962** 66–80.

Pauly, M. (2002) A modal logic for coalitional power in games. *Journal of Logic and Computation* **12** 149–166.

Rutten, J. (2000) Universal coalgebra: A theory of systems. *Theoretical Computer Science* **249** 3–80.

Schröder, L. (2007) A finite model construction for coalgebraic modal logic. *Journal of Logic and Algebraic Programming* **73** 97–110.

Schröder, L. (2008) Expressivity of coalgebraic modal logic: The limits and beyond. *Theoretical Computer Science* **390** 230–247.

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 2007. *Springer-Verlag Lecture Notes in Computer Science* **4596** 459–471.

Schröder, L. and Pattinson, D. (2008a) 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*, AAAI Press 307–218.

Schröder, L. and Pattinson, D. (2008b) 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. *Springer-Verlag Lecture Notes in Artificial Intelligence* **5243** 324–331.

Schröder, L. and Pattinson, D. (2009) PSpace bounds for rank-1 modal logics. *ACM Transactions on Computational Logic* **10** (2:13) 1–33.

Schröder, L. and Pattinson, D. (2010) Rank-1 modal logics are coalgebraic. *Journal of Logic and Computation* **20** (5) 1113–1147.

Schröder, L., Pattinson, D. and Kupke, C. (2009) Nominals for everyone. In: Boutilier, C. (ed.) *International Joint Conferences on Artificial Intelligence, IJCAI 09*, AAAI Press 917–922.

Segala, R. (1995) *Modelling and Verification of Randomized Distributed Real-Time Systems*, Ph.D. thesis, Massachusetts Institute of Technology.

Segala, R. (2006) Probability and nondeterminism in operational models of concurrency. In: Baier, C. and Hermanns, H. (eds.) Concurrency Theory, CONCUR 2006. *Springer-Verlag Lecture Notes in Computer Science* **4137** 64–78.

Segala, R. and Turrini, A. (2005) Comparative analysis of bisimulation relations on alternating and non-alternating probabilistic models. In: *Quantitative Evaluation of Systems, QEST 2005*, IEEE 44–53.

Sokolova, A., de Vink, E. and Woracek, H. (2009) Coalgebraic weak bisimulation for action-type systems. *Scientific Annals of Computer Science* **19** 93–144.

Tobies, S. (2001) PSpace reasoning for graded modal logics. *Journal of Logic and Computation* **11** 85–106.

Vardi, M. (1989) On the complexity of epistemic reasoning. In: *Logic in Computer Science, LICS 89*, IEEE 243–251.

Wolter, F. (1998) Fusions of modal logics revisited. In: Kracht, M., de Rijke, M. and Wansing, H. (eds.) Advances in modal logic, AiML 98. *CSLI Lecture Notes* **1** 361–379.