

Modular Algorithms for Heterogeneous Modal Logics

Lutz Schröder^{1*} and Dirk Pattinson²

¹ DFKI-Lab Bremen and Department of Computer Science, Universität Bremen

² Department of Computing, Imperial College London

Abstract. State-based systems and the modal logics for reasoning about them often heterogeneously combine a number of features such as non-determinism and probabilities. Here, we show that the combination of features can also be reflected algorithmically and 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 an algorithmic description to 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 that this is 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 e.g. 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 [7]. 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 [15,9] and graded modal logic [6,5], induced by joint actions of agents as in coalition logic [18], or non-monotonically conditioned as in conditional logic [3]. An attractive aspect of many of these logics is that they admit shallow models and are decidable in low complexity classes, in the absence of fixpoint operators typically *PSPACE* (e.g. [25,18,23]), i.e. the same as the standard modal logic *K* [2] 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: E.g. in the alternating model of probabilistic transition systems [8], states may have either non-deterministic or probabilistic transitions, whereas simple Segala systems [24] have a two-layered structure where non-deterministic transitions lead to probability distributions over successor states. Bartels et al. [1] present 12 different types of probabilistic transition systems that arise as combinations of basic features.

Here, we introduce a simple calculus that formalises the combination of features and establish that combined logics inherit the pleasant properties of their building blocks,

* Support by the DFG project HasCASL (KR 1191/7-2) is gratefully acknowledged

in particular shallow models and decidability in *PSPACE*. 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* [17] whose semantics is parametric in a set functor T ; models then appear as T -coalgebras. This pushes the generic *PSPACE*-decision procedure of [23], 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 $\mathbf{Set}^n \rightarrow \mathbf{Set}$, where n is the arity of the feature (e.g. 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 $T : \mathbf{Set}^n \rightarrow \mathbf{Set}^n$ 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 e.g. the ones presented in [1], which equip multi-sorted logics with a single-sorted semantics.

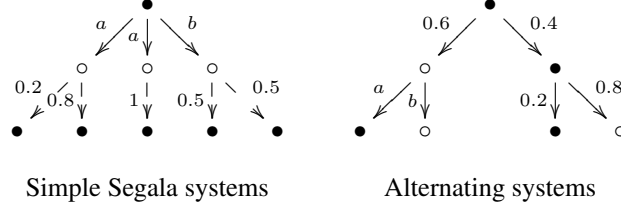
The central technical contribution of this work is the construction of a logically equivalent *flattening* of a given gluing, where flat gluings assign to each feature an individual sort in the semantics. Flat gluings are technically more tractable than general gluings. In particular, one can establish the shallow model property and a generic *PSPACE* algorithm for flat gluings. Together, these results imply *PSPACE* upper bounds for satisfiability w.r.t. general gluings, including the standard single-sorted semantics.

Related Work. Our work is closely related to the framework presented in [4,11], which focuses on completeness issues, with the main difference that our approach makes the multi-sorted nature of heterogeneous logics explicit by considering multi-sorted models. Our treatment of typed formulas resembles the use of *ingredients* in [11], but the multi-sorted semantics avoids the use of the next-operator of *loc.cit.* The main advantage of the new framework is that constructions such as cartesian product or disjoint union are no longer special cases and that the decision procedures of [22,23] generalise straightforwardly to the multi-sorted case. The multisorted approach to the complexity of composite modal logics complements transfer results obtained for the fusion of modal logics [26,10] in the sense that our framework is presently limited to logics axiomatised without nested modalities, but allows more flexible logic composition and covers also non-Kripke semantics.

2 Multisorted Modal Logics by Example

2.1 Logics for Probabilistic Systems

Segala systems [24] and alternating systems [8] 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. Contrastingly, alternating systems have two kinds of states engaging in purely probabilistic transitions and non-deterministic actions, respectively.



It has been shown in [12, Theorem 8] that *probabilistic modal logic* over a set A of actions characterises states of image-finite Segala systems up to bisimilarity. This logic has two sorts n and u of non-deterministic and probabilistic (‘uncertain’) formulas, respectively, and two families of modal operators

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

where L_p reads ‘with probability at least p ’. The sets $\mathcal{L}_n$ and $\mathcal{L}_u$ of non-deterministic and probabilistic formulas, respectively, are thus defined by the grammar

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

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

$$+ : u, n \rightarrow o \quad L_p : o \rightarrow u \quad \Box_a : o \rightarrow n$$

inducing the obvious three-sorted grammar. The binary modal operator $+$ implements the choice between probabilistic and non-deterministic transitions, being essentially a case statement: $\phi + \psi$ demands that ϕ holds if the present state is probabilistic whereas ψ holds if present state is non-deterministic.

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 $\mathcal{L}_a$ and $\mathcal{L}_b$ featuring modal operators $\Box$ and $\heartsuit$, respectively, can be modelled by a logic with sorts a , b , f and four modal operators with associated source and target sorts

$$[\pi_1] : a \rightarrow f \quad [\pi_2] : b \rightarrow f \quad \Box : f \rightarrow a \quad \heartsuit : f \rightarrow b.$$

The $[\pi_i]$ are postulated to commute with all boolean connectives. The well-known *fusion* $\mathcal{L}_a \otimes \mathcal{L}_b$ (cf. e.g. [13]) disjointly combines the axioms and modalities of $\mathcal{L}_a$ and $\mathcal{L}_b$. One can translate back and forth between the fusion and formulas of sort f , taking e.g. the operator $\Box$ of the fusion to the composite operator $[\pi_1]\Box$. 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 that it is for many purposes preferable to work with the more restrictive well-typed combinations considered here.

2.3 Conditional Logic

The standard conditional logic CK [3] has a binary modal operator $\Rightarrow$, where $\phi \Rightarrow \psi$ is read as a non-monotonic conditional. The right hand argument of $\Rightarrow$ behaves essentially like the normal modal logic K , in particular obeys the usual K -axiom when the left argument is fixed. Indeed we can embed CK into a two-sorted *extended conditional logic* with sorts c, k and modal operators

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

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 . This shows how a given complex logic can be broken down into simpler building blocks.

3 Compositional Syntax of Multisorted 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 1. An n -ary *feature* is a pair $F = (\Lambda, \mathcal{R})$ consisting of a set Λ of modal operators L with profiles $L : i_1, \dots, i_k \rightarrow *$, where $1 \leq i_1, \dots, i_k \leq n$ are formal argument sorts and $*$ is a formal target sort, and a set $\mathcal{R}$ of *one-step rules* of the form $(\phi_1; \dots; \phi_n)/\psi$, where for $i = 1, \dots, n$, ϕ_i is a propositional formula over a set V_i of propositional variables, and ψ is a disjunctive clause over atoms of the form $L(a_1, \dots, a_k)$ with $L : i_1, \dots, i_k \rightarrow *$ in Λ and $a_j \in V_{i_j}$, $j = 1, \dots, k$.

Note that the rule format disallows nested modalities in the conclusion, so that rules describe the one-step behaviour of a system. As in the single sorted case [22], 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 2. We describe the features implicit in the Examples of Sect. 2. Figure 1 shows the associated proof rules, already in a special format needed in Sect. 5. The rules for non-determinism and uncertainty are taken from [23]; the others are obtained by the same principles. The sum expression in the uncertainty rule refers to the (propositionally expressible) arithmetic of characteristic functions [23].

Non-Determinism: Given a set A of actions, the unary feature N_A has modal operators $\Box_a : 1 \rightarrow *$ for $a \in A$. We write K instead of N_A if A is a singleton.

Uncertainty: The unary feature U has modal operators $L_p : 1 \rightarrow *$ for $p \in [0, 1] \cap \mathbb{Q}$.
Choice: The binary feature S has a single modal operator $+$: $1, 2 \rightarrow *$.
Fusion: The binary feature P has two modal operators $[\pi_i] : i \rightarrow *$, $i = 1, 2$.
Conditionality: The binary feature C has a binary modal operator $\dot{\Rightarrow}$: $1, 2 \rightarrow *$.

Nondeterminism :	$\frac{\bigwedge_{j=1}^n \alpha_j \rightarrow \beta}{\bigwedge_{j=1}^n \Box_a \alpha_j \rightarrow \Box_a \beta} \quad (n \geq 0, a \in A)$
Uncertainty :	$\frac{\sum_{j=1}^n r_j \alpha_j \geq k}{\bigvee_{j=1}^n \text{sgn}(r_j) L_{p_j} \alpha_j} \quad \left(\begin{array}{l} n \geq 1, r_j \in \mathbb{Z} - \{0\}, k \in \mathbb{Z} \\ \sum_{j=1}^n r_j p_j \begin{cases} < k & \text{if } \forall j. r_j < 0 \\ \leq k & \text{otherwise} \end{cases} \end{array} \right)$
Choice :	$\frac{(\bigwedge_{j=1}^m \alpha_j \rightarrow \bigvee_{k=1}^n \beta_k) : 1 \quad (\bigwedge_{j=1}^m \gamma_j \rightarrow \bigvee_{k=1}^n \delta_k) : 2}{\bigwedge_{j=1}^m (\alpha_j + \gamma_j) \rightarrow \bigvee_{k=1}^n (\beta_k + \delta_k)} \quad (m, n \geq 0)$
Fusion :	$\frac{(\bigwedge_{j=1}^m \alpha_j \rightarrow \bigvee_{k=1}^n \beta_k) : i}{\bigwedge_{j=1}^m [\pi_i] \alpha_j \rightarrow \bigvee_{k=1}^n [\pi_i] \beta_k} \quad (i = 1, 2; m, n \geq 0)$
Conditionality :	$\frac{(\bigwedge_{j=1}^m \alpha_j \rightarrow \bigvee_{k=1}^n \beta_k) : 2}{\bigwedge_{j=1}^m (\gamma \dot{\Rightarrow} \alpha_j) \rightarrow \bigvee_{k=1}^n (\gamma \dot{\Rightarrow} \beta_k)} \quad (m, n \geq 0)$

Fig. 1. Proof rules for the features of Example 2

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

Definition 3. Let Φ be a set of features, and let $\mathcal{S}$ be a set of sorts. *Feature expressions* t are terms over the set $\mathcal{S}$ of variables where the features appear as function symbols, i.e.

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

A *gluing* of Φ over $\mathcal{S}$ is a family $G = (t_a)_{a \in \mathcal{S}}$ of feature expressions, denoted by $(a_1 \rightarrow t_{a_1}, \dots, a_n \rightarrow t_{a_n})$ for $\mathcal{S} = \{a_1, \dots, a_n\}$; in this case we also write $a_i \rightarrow t_{a_i} \in G$.

A gluing $G = (t_a)_{a \in \mathcal{S}}$ induces a multi-sorted modal logic, as follows. The set $\text{Types}(G)$ of *G-types* consists of the proper subterms of the t_a , where the sorts $a \in \mathcal{S}$ are called *base types* and the expressions $t \in \text{Types}(G) \setminus \mathcal{S}$ are the *composite types*. (Types are related to the *ingredients* of [11].) We call a gluing *flat* if $\mathcal{S} = \text{Types}(G)$, i.e. there are no composite types, which is the case if every term t_a is of the form $F(a_1, \dots, a_n)$. Typed *G-formulas* $\phi : s$, $s \in \text{Types}(G)$, are inductively generated by closure under boolean operators $\perp, \neg, \wedge$ at each type (with further boolean operators defined in the standard way) and by the following typing rules for composite types (left) and base types (right)

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

where the left hand rule has side condition $F(s_1, \dots, s_n) \in \text{Types}(\mathbf{G})$ and the right hand rule has side condition $a \rightarrow F(s_1, \dots, s_n) \in \mathbf{G}$, and in both cases $L : i_1, \dots, i_n \rightarrow *$ in $\mathbf{F}$. We write $\mathcal{F}_s(\mathbf{G})$ for the set of $\mathbf{G}$ -formulas of type s and denote the family $(\mathcal{F}_s(\mathbf{G}))_{s \in \text{Types}(\mathbf{G})}$ by $\mathcal{F}(\mathbf{G})$.

Similarly, the proof system induced by $\mathbf{G}$ is described in terms of a $\text{Types}(\mathbf{G})$ -indexed family of derivability predicates $\vdash_s \subseteq \mathcal{F}_s(\mathbf{G})$ defined inductively by closure under propositional reasoning at each type and the deduction rules for composite types (left) and base types (right), distinguished only by the type discipline,

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

where $F(s_1, \dots, s_n) \in \text{Types}(\mathbf{G})$ in the left hand rule, $a \rightarrow F(s_1, \dots, s_n) \in \mathbf{G}$ in the right hand rule, and in both cases, $(\phi_1; \dots; \phi_n)/\psi$ is a rule of $\mathbf{F}$ and σ is a substitution mapping variables $a \in V_i$ to formulas $\sigma(a) : s_i$.

A given logic can be syntactically generated by different gluings, typically including both flat and non-flat ones, determining different classes of semantic structures (cf. Sect. 4). The core of this work 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 Sect. 2, are typically non-flat.

Example 4. From the features $\mathbf{S}$, $\mathbf{U}$, and $\mathbf{N}_A$ (Example 2), we can form gluings

$$\mathbf{G}_1 \equiv (a \rightarrow \mathbf{S}(\mathbf{U}(a), \mathbf{N}_A(a))) \text{ and } \mathbf{G}_2 \equiv (a \rightarrow \mathbf{S}(u, n), u \rightarrow \mathbf{U}(a), n \rightarrow \mathbf{N}_A(a)).$$

Here, $\mathbf{G}_1$ has types a , $\mathbf{N}_A(a)$, $\mathbf{U}(a)$, whereas $\mathbf{G}_2$ is flat with types a , n , u . Modulo identifications $\mathbf{N}_A(a) = n$ and $\mathbf{U}(a) = u$, both gluings give rise to the (typed) formulas describing alternating systems (Sect. 2.1). The remaining example logics from Sect. 2 are captured by the following gluings (where we omit the obvious flat versions):

Probabilistic modal logic of Segala Systems: $s \rightarrow \mathbf{N}_A(\mathbf{U}(s))$.

Fusion: The fusion of logics $\mathcal{L}_a$ and $\mathcal{L}_b$ as in Sect. 2.2, regarded as features, is $f \rightarrow \mathbf{P}(\mathcal{L}_a(f), \mathcal{L}_b(f))$.

Extended conditional logic: $c \rightarrow \mathbf{C}(c, \mathbf{K}(c))$. Note in particular that in the induced proof system, we can derive the standard rule

$$(RCK) \frac{\bigwedge_{i=1}^n \alpha_i \rightarrow \beta}{\bigwedge_{i=1}^n (\gamma \Rightarrow \alpha_i) \rightarrow (\gamma \Rightarrow \beta)} \quad (n \geq 0)$$

of the conditional logic CK [3], where $\gamma \Rightarrow \alpha$ abbreviates $\gamma \xRightarrow{\bullet} \Box \alpha$.

4 Multi-Sorted vs. Single-Sorted Coalgebraic Semantics

We now generalise the coalgebraic interpretation of modal logic, introduced in [17], 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 recall some basic notions of multi-sorted coalgebra (cf. e.g. [16]), generalising the single-sorted setting [19]:

Definition 5. We write $\mathbf{Set}$ for the category of sets and functions. Let $\mathbf{Set}^{\mathcal{S}}$ denote the category of $\mathcal{S}$ -sorted sets and $\mathcal{S}$ -sorted functions, with objects being families $X = (X_a)_{a \in \mathcal{S}}$ (or just (X_a)) of sets X_a , and morphisms $f : (X_a) \rightarrow (Y_a)$ being families $f = (f_a)_{a \in \mathcal{S}}$ of maps $f_a : X_a \rightarrow Y_a$. We write $\mathbf{Set}^n$ for $\mathbf{Set}^{\{1, \dots, n\}}$. A functor $T : \mathbf{Set}^{\mathcal{S}} \rightarrow \mathbf{Set}^{\mathcal{S}}$ may be regarded a family $T = (T_a)_{a \in \mathcal{S}}$ of functors $T_a : \mathbf{Set}^{\mathcal{S}} \rightarrow \mathbf{Set}$. A T -coalgebra $A = (X, \xi)$ is a pair (X, ξ) where X is an $\mathcal{S}$ -sorted set and $\xi = (\xi_a) : X \rightarrow TX$ is an $\mathcal{S}$ -sorted function (i.e. $\xi_a : X_a \rightarrow T_a X$) called the *transition function*. A *morphism* between T -coalgebras (X, ξ) and (Y, ζ) is an $\mathcal{S}$ -sorted function $f : X \rightarrow Y$ such that $(Tf)\xi = \zeta f$ in $\mathbf{Set}^{\mathcal{S}}$.

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

The interpretation of modal operators is based on predicate liftings [17,20]; in the multi-sorted setting, this takes the following shape.

Definition 6. A *predicate lifting* λ of profile $\lambda : i_1, \dots, i_k \xrightarrow{\bullet} *$ for a functor $T : \mathbf{Set}^n \rightarrow \mathbf{Set}$, where $i_1, \dots, i_k \leq n$, is a natural transformation

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

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

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

Definition 7. Let $\mathbf{F} = (A, \mathcal{R})$ be an n -ary feature. A *structure* for $\mathbf{F}$ consists of an endofunctor $\llbracket \mathbf{F} \rrbracket : \mathbf{Set}^n \rightarrow \mathbf{Set}$ and an assignment of a predicate lifting $\llbracket L \rrbracket : i_1, \dots, i_k \xrightarrow{\bullet} *$ for T to every modal operator $L : i_1, \dots, i_k \rightarrow *$ in A , subject to the condition that every rule $R = \phi_1; \dots; \phi_n / \psi$ over V in $\mathcal{R}$ 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 $\llbracket \phi_i \rrbracket \tau = X_i$ for all i , then $\llbracket \psi \rrbracket \tau = TX$, where $\llbracket \phi_i \rrbracket \tau \subseteq X_i$ and $\llbracket \psi \rrbracket \tau \subseteq TX$ are defined by the usual clauses for boolean operators and $\llbracket L(a_1, \dots, a_k) \rrbracket \tau = \llbracket L \rrbracket(\tau(a_1), \dots, \tau(a_k))$.

When features are equipped with structures, every feature expression t over the set $\mathcal{S}$ of sorts defines a functor $\llbracket t \rrbracket : \mathbf{Set}^{\mathcal{S}} \rightarrow \mathbf{Set}$ by

$$\llbracket a \rrbracket = P_a : \mathbf{Set}^{\mathcal{S}} \rightarrow \mathbf{Set} \quad (a \in \mathcal{S}) \quad \text{and} \quad \llbracket \mathbf{F}(t_1, \dots, t_n) \rrbracket = \llbracket \mathbf{F} \rrbracket \circ \langle \llbracket t_1 \rrbracket, \dots, \llbracket t_n \rrbracket \rangle,$$

where P_a is projection to the a -th component and $\langle \cdot \rangle$ represents tupling. Thus, a gluing $\mathbf{G} = (t_a)_{a \in \mathcal{S}}$ induces a functor $\llbracket \mathbf{G} \rrbracket : \mathbf{Set}^{\mathcal{S}} \rightarrow \mathbf{Set}^{\mathcal{S}}$.

The coalgebraic semantics of $\mathcal{F}(\mathbf{G})$ is now given w.r.t. $\llbracket \mathbf{G} \rrbracket$ -coalgebras $C = (X, \xi)$. For a type $s \in \text{Types}(\mathbf{G})$, an s -state of C is an element $x \in \llbracket s \rrbracket X$. The semantics of a

formula $\phi : s$ is a set $\llbracket \phi \rrbracket_C \subseteq \llbracket s \rrbracket X$ of s -states. We have the usual clauses for propositional connectives, and the semantics of modal operators is given by the following clauses for composite types (top, assuming $F(s_1, \dots, s_n) \in \text{Types}(\mathbf{G})$) and base types (bottom, for $a \rightarrow F(s_1, \dots, s_n) \in \mathbf{G}$):

$$\begin{aligned}\llbracket L(\phi_1, \dots, \phi_n) : F(s_1, \dots, s_n) \rrbracket_C &= \llbracket L \rrbracket(\llbracket \phi_1 \rrbracket_C, \dots, \llbracket \phi_n \rrbracket_C) \\ \llbracket L(\phi_1, \dots, \phi_n) : a \rrbracket_C &= \xi_a^{-1} \circ \llbracket L \rrbracket(\llbracket \phi_1 \rrbracket_C, \dots, \llbracket \phi_n \rrbracket_C)\end{aligned}$$

where, in both cases, $L : i_1, \dots, i_n \rightarrow *$ in $\mathbf{F}$. We write $x \models_C^s \phi$ if $\phi : s$ and $x \in \llbracket \phi \rrbracket_C$.

Note that the requirement that rules are one-step sound immediately yields soundness of the logic w.r.t. the semantics described above; this is as in [4].

Example 8. The standard semantics for the features of Example 2 is induced by the following structures.

Non-Determinism: A structure for $\mathbf{N}_A$ is given by $\llbracket \mathbf{N}_A \rrbracket = \mathcal{P}(A \times _)$ and

$$\llbracket \Box_a \rrbracket_X(C) = \{B \in \mathcal{P}(A \times X) \mid \{x : (a, x) \in B\} \subseteq C\}.$$

Note that (single-sorted) coalgebras for $\mathcal{P}(A \times _)$ are labelled transition systems, and the lifting associated with $\Box_a$ gives rise to Hennessy-Milner logic [17].

Uncertainty: Put $\llbracket \mathbf{U} \rrbracket = D_\omega$, where D_ω is the *finite distribution functor* D_ω that maps a set X to the set of probability distributions on X with finite support. The modal operators L_p are interpreted by

$$\llbracket L_p \rrbracket_X(A) = \{P \in D_\omega X \mid PA \geq p\}.$$

(Single-sorted) D_ω -coalgebras are finitely branching probabilistic transition systems. For $\mathbf{G} = (s \rightarrow \mathbf{N}_A(\mathbf{U}(s)))$ (Example 4), we have $\llbracket \mathbf{G} \rrbracket = \mathcal{P} \circ D_\omega$, so that $\llbracket \mathbf{G} \rrbracket$ -coalgebras are precisely Segala systems, while coalgebras for the corresponding flat signature have an explicit separation between non-deterministic and probabilistic states.

Choice: Let $\llbracket \mathbf{S} \rrbracket$ be the disjoint sum functor $\llbracket \mathbf{S} \rrbracket(X, Y) = X + Y$, and interpret the modality $+$ by

$$\llbracket + \rrbracket_{X,Y}(A, B) = A + B \subseteq X + Y.$$

Fusion: Let $\llbracket \mathbf{P} \rrbracket$ be the binary product functor $\llbracket \mathbf{P} \rrbracket(X, Y) = X \times Y$, and put

$$\llbracket \pi_1 \rrbracket_{X,Y} A = \{(x, y) \mid x \in A\} \text{ and } \llbracket \pi_2 \rrbracket_{X,Y} B = \{(x, y) \mid y \in B\}.$$

Conditionality: Define the functor $\llbracket \mathbf{C} \rrbracket$ by $\llbracket \mathbf{C} \rrbracket(X, Y) = \mathcal{Q}X \rightarrow Y$, with $\mathcal{Q}$ denoting contravariant powerset and $\rightarrow$ denoting function space, and put

$$\llbracket \overset{\bullet}{\Rightarrow} \rrbracket_{X,Y}(A, B) = \{f : \mathcal{Q}X \rightarrow Y \mid f(A) \in B\}.$$

For $\mathbf{G} = (c \rightarrow \mathbf{C}(c, K(c)))$ (Example 4), we have $\llbracket \mathbf{G} \rrbracket X = \mathcal{Q}X \rightarrow \mathcal{P}X$, and $\llbracket \mathbf{G} \rrbracket$ -coalgebras are conditional frames [3].

Modal logic talks only about the observable behaviour of states; this is formally expressed as invariance of the logic under morphisms:

Proposition 9. Let $f : C \rightarrow D$ be a morphism of $\llbracket G \rrbracket$ -coalgebras. Then for each G -formula $\phi : s$ and each s -state x in C , $x \models_C^s \phi$ iff $(\llbracket s \rrbracket f)(x) \models_D^s \phi$.

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

Definition 10. A G -formula $\phi : s$ is *satisfiable* in a G -model if there exist a $\llbracket G \rrbracket$ -coalgebra C and an s -state x in C such that $x \models_C^s \phi$.

A central contribution of this work is to show that for every gluing, we can construct a flat gluing with an equivalent satisfiability problem. For flat gluings, one can generalise existing model constructions and complexity results for coalgebraic modal logic [22,23,21], and the relevant criteria reduce to the component logics; for the shallow-model-based *PSPACE* algorithm of [23], this is discussed in more detail in Sect. 5. We thus obtain compositional algorithmic methods also for the standard single-sorted semantics present in the literature.

We start by constructing a flattening G^b of an arbitrary gluing G and then transform $\llbracket G \rrbracket$ -coalgebras to $\llbracket G^b \rrbracket$ -coalgebras preserving satisfaction of formulas.

Definition 11. Let G be a gluing over the set $\mathcal{S}$ of sorts. The *flattening* $G^b = (u_s)_{s \in \text{Types}(G)}$ of G is a flat gluing over the set $\mathcal{S}^b = \text{Types}(G)$ of sorts, defined by $u_s = t_a$ for $s = a \in \mathcal{S}$ (with immediate subexpressions of t_a regarded as sorts in $\mathcal{S}^b$) and $u_s = s$ otherwise (with s regarded as a sort in $\mathcal{S}^b$).

Example 12. Given the gluings $G_1 \equiv (a \rightarrow S(U(a), N_A(a))$ and $G_2 \equiv (a \rightarrow S(u, n), u \rightarrow U(a), n \rightarrow N_A(a))$ from Example 4, G_2 is the flattening of G_1 , up to renaming the *sorts* $U(a)$ and $N_A(a)$ of the flattening into u and n , respectively.

It is easy to see that *the flattening G^b syntactically induces the same logic as G* , i.e. the types, formulas, and proof systems coincide. Our main result is now stated as follows.

Theorem 13. A G -formula is *satisfiable* in a G -model iff it is *satisfiable* (as a G^b -formula) in a G^b -model.

Proof. (Sketch) ‘*Only if*’: Expand a $\llbracket G \rrbracket$ -coalgebra C to a $\llbracket G^b \rrbracket$ -coalgebra C^b by inserting identity functions for the components of the structure map corresponding to composite types. Induction on the definition of the semantics then shows that the semantics w.r.t. C and C^b agree.

‘*If*’: Turn a $\llbracket G^b \rrbracket$ -coalgebra $D = (X_b, \xi_b)_{b \in \mathcal{S}^b}$ into the $\llbracket G \rrbracket$ -coalgebra $D^\# = (X_a, \gamma_a)_{a \in \mathcal{S}}$, where

$$\gamma_a = \llbracket F \rrbracket(\zeta_{s_1}, \dots, \zeta_{s_n}) \circ \xi_a$$

for $a \rightarrow F(s_1, \dots, s_n)$ in G , and the maps $\zeta_s : X_s \rightarrow \llbracket s \rrbracket(X_a)_{a \in \mathcal{S}}$ for $s \in \text{Types}(G)$ are defined recursively by

$$\zeta_a = id_{X_a} \quad (a \in \mathcal{S}) \quad \text{and} \quad \zeta_{F(s_1, \dots, s_n)} = \llbracket F \rrbracket(\zeta_{s_1}, \dots, \zeta_{s_n}) \circ \xi_{F(s_1, \dots, s_n)}.$$

One can then construct a coalgebra morphism $D \rightarrow (D^\#)^b$, and Proposition 9 yields the claim. $\square$

In our running example, the situation is as follows:

Example 14. Consider the gluings G_1 and G_2 over $\mathcal{S} = \{a, u, n\}$ from Example 4 and recall from Example 12 that $G_1^b = G_2$. Let $C = (X, \xi : X \rightarrow D_\omega X + \mathcal{P}(A \times X))$ be a $\llbracket G_1 \rrbracket$ -coalgebra. Then $C^b = ((X_s), (\xi_s))$ where $X_a = X$, $X_u = D_\omega X$, $X_n = \mathcal{P}(A \times X)$, $\xi_a = s$, $\xi_u = id_{X_u}$, and $\xi_n = id_{X_n}$.

Conversely, given a $\llbracket G_2 \rrbracket$ -coalgebra $D = ((X_s), (\xi_s))$, we construct a $\llbracket G_1 \rrbracket$ -coalgebra $D^\# = (X, \xi)$ by putting $X = X_a$ and $\xi = (\xi_u + \xi_n) \circ \xi_a$. The triple (id_X, ξ_u, ξ_s) is a homomorphism $D \rightarrow (D^\#)^b$.

5 Applications to Model Construction and Complexity

We have seen in Sect. 3 that the same multi-sorted logic can arise from different gluings of given features, where the difference manifests itself only on a semantic level. The different interpretations of the logic are related by Theorem 13 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 [23] generalise to flat gluings; this enables us to derive upper *PSPACE* bounds for arbitrary gluings, in particular for heterogeneous logics equipped with their standard single-sorted semantics as in Sect. 2.

The shallow model construction requires the involved structures to be strictly one-step complete in the following sense, where the notation $\llbracket - \rrbracket_\tau$ is as in Definition 7. (Strict) one-step completeness implies weak completeness of the rule system [17,22].

Definition 15. An n -ary feature F is *strictly one-step complete* for a structure $T = \llbracket F \rrbracket : \mathbf{Set}^n \rightarrow \mathbf{Set}$ if, whenever $\llbracket \chi \rrbracket_\tau = T(X_1, \dots, X_n)$ for a sorted set $(V_1, \dots, V_n)$ 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 $L(a_{i_1}, \dots, a_{i_k})$, where $L : i_1, \dots, i_k \rightarrow *$ in F and $a_{i_j} \in V_{i_j}$, then χ is propositionally entailed by a clause $\psi\sigma$, where $(\phi_i)/\psi$ is a rule of F and σ is a $(V_1, \dots, V_n)$ -substitution (i.e. $\sigma(a) \in V_i$ for $a \in V_i$) such that $\llbracket \phi_i \sigma \rrbracket_\tau = X_i$ for all i .

(The formulation above corrects the formulation given in [23] in admitting only a single rule application in a strict derivation.) One shows analogously to the single-sorted case [22] 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. In [23], *rule resolution*, a systematic procedure for obtaining strictly one-step complete rule sets, has been described, which straightforwardly generalises to the multi-sorted setting. Throughout this section, we fix a gluing G of a set Φ of features over a set $\mathcal{S}$ of sorts; moreover we assume that every feature is equipped with a structure.

Definition 16. The set $MA(\phi)$ of *modal atoms* of an G -formula ϕ is defined recursively by $MA(\phi \wedge \psi) = MA(\phi) \cup MA(\psi)$, $MA(\neg\phi) = MA(\phi)$, and $MA(L(\rho_1, \dots, \rho_n)) = \{L(\rho_1, \dots, \rho_n)\}$. 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 that $s = F(s_1, \dots, s_n) \in \text{Types}(G)$ if s is composite and $a \rightarrow F(s_1, \dots, s_n) \in G$ if $s = a$ is a base type, we say that a rule $R = (\phi_1; \dots; \phi_n)/\psi$ 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 called a *matching* of H .

Our shallow model theorem now takes the following form.

Theorem 17. *If every feature in $\mathcal{G}$ is strictly one-step complete, then a formula $\phi : s$ is satisfiable in a $\mathcal{G}$ -model iff $H \models \phi$ for some pseudovaluation H for ϕ such that for every matching $((\phi_1; \dots; \phi_n)/\psi, \sigma)$ of H , one of the formulas $\neg\phi_i\sigma$ is satisfiable.*

The proof first reduces to flat gluings by Theorem 13 and then recursively constructs a shallow model whose root state is a pseudovaluation and whose branches are models of negated substitution instances of rule premises as in the statement.

From Theorem 17, we obtain a multi-sorted version of the generic *PSPACE* decision procedure of [23]. This requires to compute matchings of given pseudovaluations, and we require that the rules associated with features are *reduction closed*, i.e. it suffices to consider matchings $((\phi_1, \dots, \phi_n)/\psi, \sigma)$ where $\psi\sigma$ does not contain duplicate literals, with the consequence that there are only finitely many matches to check in every recursion step. Since rules are generally too large to pass around directly, we assume that every rule is represented by a *code*, i.e. a string over some alphabet. For the features discussed in Example 2, the codes can be taken as the parameters of the rules.

The crucial requirement for the effectivity of the algorithm is that one has a polynomial bound on codes of matching rules and that a number of minor infrastructure operations can be performed in polynomial time (cf. [23] for details), in which case we call a rule set *PSPACE-tractable*. All rule sets presented in Fig. 1 are strictly one-step complete, reduction closed, and *PSPACE-tractable* (this is either clear or shown in [23]). We obtain:

Theorem 18 (Space Complexity). *If every feature in $\mathcal{G}$ is strictly one-step complete, reduction closed, and *PSPACE-tractable*, then the satisfiability problem for $\mathcal{F}(\mathcal{G})$ -formulas over $\llbracket \mathcal{G} \rrbracket$ -coalgebras is in *PSPACE*.*

In particular, satisfiability for logics arising through arbitrary gluings of the features from Example 2 are in *PSPACE*.

Remark 19. The recursive structure of the algorithm allows for a modular implementation which interconnects separate matching routines for each feature. In particular, the same algorithmic structure may alternatively be applied to effective heuristic matching routines, leading to approximative but more efficient solutions.

6 Conclusions

We have introduced a calculus of *gluings*, which describe ways of combining logical features like uncertainty, non-determinism, and choice. We have shown that the satisfaction problem of a gluing is in *PSPACE* if this is true for involved features. This has been achieved by equipping the logics under consideration with a multi-sorted coalgebraic semantics. Crucially, we have shown that the satisfiability problem of a gluing is equivalent to that of a corresponding flattened gluing, which is technically more tractable. Our results pave the way for modularized tool support for a large class of heterogeneous logics. The study of $\mathcal{E}$ -connections [14] in the coalgebraic framework is the subject of future work.

References

1. F. Bartels, A. Sokolova, and E. de Vink. A hierarchy of probabilistic system types. In *Coalgebraic Methods in Computer Science*, vol. 82 of *ENTCS*. Elsevier, 2003.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge, 2001.
3. B. Chellas. *Modal Logic*. Cambridge, 1980.
4. C. Cîrstea and D. Pattinson. Modular construction of modal logics. *Theoret. Comput. Sci.* To appear. Earlier version in *Concurrency Theory, CONCUR 04*, vol. 3170 of *LNCS*, pp. 258–275, Springer, 2004.
5. G. D’Agostino and A. Visser. Finality regained: A coalgebraic study of Scott-sets and multisets. *Arch. Math. Logic*, 41:267–298, 2002.
6. K. Fine. In so many possible worlds. *Notre Dame J. Formal Logic*, 13:516–520, 1972.
7. J. Y. Halpern. *Reasoning About Uncertainty*. MIT Press, 2003.
8. H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Real-Time Systems, RTSS 90*, pp. 278–287. IEEE, 1990.
9. A. Heifetz and P. Mongin. Probabilistic logic for type spaces. *Games and Economic Behavior*, 35:31–53, 2001.
10. E. Hemaspaandra. Complexity transfer for modal logic. In *Logic in Computer Science, LICS 94*, pp. 164–173. IEEE, 1994.
11. B. Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theor. Inform. Appl.*, 35:31–59, 2001.
12. B. Jonsson, W. Yi, and K. G. Larsen. Probabilistic extensions of process algebras. In J. Bergstra, A. Ponse, and S. Smolka, eds., *Handbook of Process Algebra*. Elsevier, 2001.
13. A. Kurucz. Combining modal logics. In J. van Benthem, P. Blackburn, and F. Wolter, eds., *Handbook of Modal Logic*. Elsevier, 2006.
14. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyashev. $\mathcal{E}$ -connections of abstract description systems. *Artificial Intelligence*, 156:1–73, 2004.
15. K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inform. Comput.*, 94:1–28, 1991.
16. T. Mossakowski, L. Schröder, M. Roggenbach, and H. Reichel. Algebraic-coalgebraic specification in CoCASL. *J. Logic Algebraic Programming*, 67:146–197, 2006.
17. D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame J. Formal Logic*, 45:19–33, 2004.
18. M. Pauly. A modal logic for coalitional power in games. *J. Logic Comput.*, 12:149–166, 2002.
19. J. Rutten. Universal coalgebra: A theory of systems. *Theoret. Comput. Sci.*, 249:3–80, 2000.
20. L. Schröder. Expressivity of coalgebraic modal logic: the limits and beyond. *Theoret. Comput. Sci.* To appear. Earlier version in *Foundations of Software Science And Computation Structures*, vol. 3441 of *LNCS*, pp. 440–454, Springer, 2005.
21. L. Schröder. A semantic PSPACE criterion for the next 700 rank 0-1 modal logics. Available at <http://www.informatik.uni-bremen.de/~lschrode/papers/rank01pspace.pdf>.
22. L. Schröder. A finite model construction for coalgebraic modal logic. In *Foundations Of Software Science And Computation Structures*, vol. 3921 of *LNCS*, pp. 157–171. Springer, 2006.
23. L. Schröder and D. Pattinson. PSPACE reasoning for rank-1 modal logics. In *Logic in Computer Science, LICS 06*, pp. 231–240. IEEE, 2006.
24. R. Segala. *Modelling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.
25. S. Tobies. PSPACE reasoning for graded modal logics. *J. Logic Comput.*, 11:85–106, 2001.
26. F. Wolter. Fusions of modal logics revisited. In *Advances in modal logic*, vol. 1 of *CSLI Lect. Notes*, pp. 361–379. CSLI, 1998.