Coalgebraic Modal Logic in the Category of Nominal Sets

Anthony Voutas

A thesis submitted in partial fulfillment of the degree of
Bachelor of Science (Honours) at
The Department of Computer Science
Australian National University

May 2013

Typeset in Palatino by TEX and LATEX 2ε.

Except where otherwise indicated, this thesis is my own original work.

Anthony Voutas
30 May 2013

To my mother Pip, my father Louis, and my girlfriend Ellie.
Your tremendous love keeps me going.

Acknowledgements

I would like to acknowledge the many people who have helped me through the completion of this thesis. The first is my supervisor Ranald Clouston, who has always been available to provide encouragement, optimism, constructive criticism, helpful analysis, suggestions and good advice in general.

I would also like to acknowledge Dirk Pattinson and Rajeev Goré whose work I have used as a basis for my own, and who have both provided helpful comments and clarifications along the way. Additionally, I would like to thank Alwen Tiu for sharing his knowledge about CCS and the π-calculus which had an influence on the direction of this thesis.

I am thankful to the Australian National University, and in particular the Logic and Computation Group and the School of Computer Science, for the opportunities that have led me to this point. In this regard, I would also like to thank the current honours convenor, John Slaney, for managing the honours course, for running the Logic Summer School, for generally being excited about logic, and for sharing that excitement with others. The concept of studying logic seemed somewhat banal to me before I saw him put the magic into it.

I would also like to thank Nandita Sharma, for running the honours tutorials and for her advice with regards to time management among other things. Also, I would like to thank Bindi Mamouney for her help in all things admin related.

Last but certainly not least, I would like to thank my friends and family for their support, encouragement, practical advice, and most of all for their patience.

Abstract

Coalgebraic modal logic (CML) in the category of sets and functions (**Set**) has a well established generic proof theory. The key contribution of this thesis is to partially extend that proof theory into the category of nominal sets and equivariant functions (**Nom**). In doing this, we can study the interaction between name binding (modelled in **Nom**) and named transition structures (modelled by coalgebra). We achieve this extension by nominalising the work of [Pattinson 2012]. We show generic results in soundness and completeness, and we bring the И quantifier of nominal logic into the framework. We conclude that large parts of the generic proof theory of CML in **Set** lifts to **Nom** with varying degrees of difference. We also conclude that the generic proof theory of CML in **Nom** can be used to reason about systems with named transitions and name binding.

Contents

Chapter 1

Introduction

1.1 Context and Research Question

Modal logics are a variety of formal symbolic logic which augment propositional logic with modal operators which can be seen to be roughly analogous to linguistic modalities. Since the formal semantics of modal logics are often transition systems of various kinds, modal logics have become relevant to computer science as a way of describing and studying properties of computation and computer systems (which can often be represented in terms of states and transitions between them). There are many modal logics that are relevant to computer science, so a rich literature has emerged surrounding modal logics and their applications [Blackburn et al. 2006] [Milner et al. 1993] [Emerson 1990].

There are many different modal logics. Here is a non-exhaustive list of modal logics by name and what they are used to reason about: Intuitionistic logic (constructive truth in mathematics), Alethic logic (metaphysical possibility and necessity), Epistemic modal logic (epistemic possibility and necessity), Deontic logic (permission and obligation), Doxastic logic (belief), Probabilistic modal logic (probabilities), Graded modal logic (multiple possibility and almost necessity), Hennessy Milner Logic (labelled transition systems), Coalition Logic (cooperation in game theory), and Conditional Logic (counterfactuals).

There are also large classes of modal logics such as Tense logics (for events in time), Temporal logics (for temporary properties - often safety and liveness properties of computer systems), Spatial logics (for geometry and local properties), Neighbourhood logics and Normal Modal Logics (both of which are very versatile, though the former subsumes the latter). It is also worth noting that for many of the individual logics in the paragraph above there are variations on the logic in the literature which are also independent modal logics, requiring independent semantics, proof rules and decision procedures. These variations can enter the discourse either because the original logic is considered to be philosophically inadequate, or because the variations have different applications.

Each time a new modal logic is discovered and studied, properties such as soundness, completeness and decidability must be reexamined. For those logics which are decidable, new decision procedures (often not the most efficient possible ones at first) are developed. A well developed common semantic framework for modal logics

should simplify the process by distilling these questions down to just those things that make a new logic different.

As it turns out, coalgebraic modal logic (CML) [Cîrstea et al. 2011] is a well established, successful semantic framework for addressing many generic questions in the field of modal logic. For instance, generic theorems in CML make checking soundness, completeness and decidability for a new modal logic much simpler, because one must only answer the simpler questions of whether the semantics fit the coalgebraic semantics and whether the deduction rules for the logic are one step sound, one step complete, cut absorbing and contraction closed. The coalgebraic framework captures many examples in modal logic, including (but certainly not limited to) Neighbourhood logics (and therefore all Normal modal logics), Hennessy Milner Logic, Graded modal logic, Coalition logic, Probabilistic modal logic, Conditional Logic and various Deontic Logics.

Furthermore, a generic, asymptotically optimal algorithm has been developed based on tableau proof systems with global caching [Goré et al. 2010]. This algorithm can be adapted to any modal logic which fits within the framework, of which there are many. The global caching approach is also a practical one, in the sense that the algorithm, being agnostic about its search method, should benefit greatly from good search heuristics in the future.

The generic semantics and proof theory of coalgebraic modal logic has the potential to greatly simplify the work of creating and using modal logics, as well as modifying existing modal logics. Despite this, at the current point in time, the proof theory results of coalgebraic modal logic are constrained by one key simplifying assumption, where the general theory of coalgebra (and even coalgebraic modal logic) is not. The generic proof theory of coalgebraic modal logic focuses on the case of coalgebra in the category of sets and functions.

Coalgebra, being a child of category theory, has obvious meaning outside the category of sets and functions (**Set**), and coalgebras in the category of vector spaces and linear functions, **Nom** (the category of nominal sets and equivariant functions), and various other categories such as presheaf topoi have all been examined as part of the broader study of coalgebra and transition systems. Coalgebraic modal logic has been used outside the category of sets and functions to create expressive logics for various coalgebras. However, to the best of our knowledge, the generic proof theory of CML has not been applied to coalgebras outside the category of sets and functions. This is not intended as a criticism. The focus on coalgebras in the category of sets almost certainly simplified the development of the generic proof theory, and as so many classic examples are covered by the choice, it was a reasonable one to make.

Now that we have a well developed proof theory for the case of coalgebras in the category of sets and functions, it seems like a good time to start the process of expanding the results of coalgebraic modal logic to other categories. Due to its computational relevance, and its similarities with the category of sets and functions, we choose to start with **Nom**, the category of nominal sets and equivariant functions.

The theory surrounding nominal sets [Gabbay and Pitts 1999] [Gabbay and Pitts 2002] have been developed as a means for reasoning about object level names and the

binding of those names by quantifiers and other binders such as the λ of λ-calculus. The notion of binding has deep relevance to computer science as the binding of names is instrumental in the construction of generic functions in computational calculi such as the λ-calculus. Variable scoping through name binding is vital even outside of functional representations of computation.

The choice of **Nom** also allows us to introduce Nominal Logic's [Pitts 2003] И quantifier into the framework in a natural way, which has applications to modelling the logic of local/bound variables. This type of reasoning is relevant to many systems, especially mobile systems.

The research questions of this thesis could be stated as follows. Can the generic proof theory of coalgebraic modal logic be lifted to the case for modal logics whose semantics are captured by coalgebras in **Nom**? Can the И quantifier fit harmoniously into the coalgebraic framework as an addition to the syntax, semantics and proof theory? More generally, does the proof theory of coalgebraic modal logic generalise to categorical settings outside **Set**?

1.2 Document Structure

The structure of the document is as follows. First, there is the introduction above. In chapter 2 a more thorough and technical background is given to a number of topics including category theory, coalgebra, nominal sets, coalgebraic modal logic, and proof theory in modal logic. In chapter 3 the semantic framework of coalgebraic modal logic in the category of nominal sets is presented. In chapter 4 the proof theoretic framework of coalgebraic modal logic in the category of nominal sets is presented, including generic soundness results and some preliminary results in generic completeness. In chapter 5 the И quantifier is introduced into the framework, and modifications are made to the framework to accommodate it. In chapter 6 we discuss aspects of decidability and decision procedures for coalgebraic modal logics in the category of nominal sets. In chapter 7 we discuss an application of the framework to reasoning about nominal calculi. In chapter 8 we conclude with a summary of the results and a comment on the key differences between our framework and the existing coalgebraic modal logic on sets. We then present some potential avenues for further study.

Chapter 2

Background

2.1 Category Theory and coalgebra

In this section we fix notation in category theory and discuss coalgebra. Our presentation of coalgebraic modal logic doesn't require advanced concepts in category theory, only the relatively simple concepts of functor, natural transformation, and of course, coalgebra. The classic reference text for category theory is [Mac Lane 1978]. Awodey [Awodey 2006] gives a gentler introduction to category theory which requires less background but still covers many of the important concepts, and certainly all of the concepts that this thesis uses explicitly.

Formalising the concept of transition systems would capture both an abstract semantics of modal logics and the practical application of modal logics to computer science. Of course, formalising the notion of a transition system has direct applications to computer science[1] without even mentioning modal logics. A somewhat surprising development has arisen from the field of category theory. In category theory the concept of duality is central in the sense that every concept has a dual concept. Specifically, the concept of algebra has the dual concept of coalgebra (the prefix co is used to denote that something is a dual). Coalgebra has been seen to capture formally the informal concept of a transition structure.

So, we must provide a brief introduction to category theory. Informally, a category is a universe of mathematical discourse. Examples include the category of sets and functions, the category of topological spaces and continuous functions, and the category of groups and group homomorphisms.

Definition 2.1.1. A *category* $\mathcal{C}$ is a pair of collections, one of objects $\mathrm{obj}(\mathcal{C})$ (e.g. sets) and one of arrows $\mathrm{arr}(\mathcal{C})$ (e.g. functions) along with an operation and a few rules. First, each arrow has a domain and a codomain which are objects. We say $f : X \to Y$ to denote that the arrow f has domain X and codomain Y. Second, two arrows can be composed if the codomain of the first matches the domain of the second. The result is another arrow from the domain of the first to the codomain of the second, and this composition is associative. We use $f \circ g$ for the composite by first following g and then f. Third, each object has a special identity arrow with domain and codomain

[1] and indeed to any science of dynamic systems

being that object (e.g. the identity function on a set), and those arrows act as identities for the composition operation. We use id_X to denote the identity arrow of the object X.

The category of sets and functions fulfils all of these requirements. If we change sets for nominal sets and functions for equivariant functions (which will be discussed in section 2.2), the requirements are still met, because the composite of two equivariant functions is an equivariant function, and identity functions are also equivariant.

Definition 2.1.2. Functors are just maps between categories which preserve function composition and identities. Formally, a functor $F : \mathcal{C} \to \mathcal{C}'$ maps objects $X \in \mathcal{C}$ to objects $FX \in \mathcal{C}'$ and arrows $f : X \to Y \in \mathcal{C}$ to arrows $Ff : FX \to FY \in \mathcal{C}'$, such that $F(f \circ g) = F(f) \circ F(g)$ where $\circ$ is arrow composition.

The example to think of is the powerset functor, which maps sets to their powersets and functions $f : X \to Y$ to their image maps: $(\mathcal{P}f)(A) = \{y \in Y \mid \exists x \in A \,.\, y = f(x)\}$.

Definition 2.1.3. There are also contravariant functors that map arrows $f : X \to Y \in \mathcal{C}$ to arrows $f : FY \to FX \in \mathcal{C}'$ such that $F(f \circ g) = F(g) \circ F(f)$.

The example to think of is the contravariant powerset functor, which maps sets to their powersets and functions $f : X \to Y$ to their inverse image maps $f^{-1}(B) = \{x \in X \mid \exists y \in B \,.\, y = f(x)\}$.

Definition 2.1.4. For two functors $F, G : \mathcal{C} \to \mathcal{C}'$, a *natural transformation* between them is a collection of arrows from the objects in the image of one functor to the objects in the image of the other, satisfying a simple naturality property. The arrows α_X in the collection are indexed by objects of $\mathcal{C}$. The naturality property is that for any arrow $f : X \to Y$ of $\mathcal{C}$, the following diagram of $\mathcal{C}'$ commutes:

$$\begin{array}{ccc} F(X) & \xrightarrow{\alpha_X} & G(X) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(Y) & \xrightarrow[\alpha_Y]{} & G(Y) \end{array}$$

By commutativity of a diagram we mean that the composite arrow yielded by following a path through the diagram from A to B is the same as the composite yielded by following any other path from A to B. The naturality of this collection of arrows ensures that commutative diagrams in the image of F can be transformed via these arrows to commutative diagrams in the image of G. If the collection $\{\alpha_X\}_{X \in obj(\mathcal{C})}$ is a natural tranformation from a functor F to a functor G, then we say $\alpha : F \dot{\to} G$.

One can also have a natural transformation from a *contravariant* functor F to another contravariant functor G. That is, a collection of arrows $\alpha_X : F(X) \to G(X)$ such

that the following diagram commutes.

$$\begin{array}{ccc} F(X) & \xrightarrow{\alpha_X} & G(X) \\ {\scriptstyle F(f)}\uparrow & & \uparrow{\scriptstyle G(f)} \\ F(Y) & \xrightarrow[\alpha_Y]{} & G(Y) \end{array}$$

As an example, take the contravariant functors $\mathrm{Homset}(-,2), \mathcal{P}(-) : \mathbf{Set} \to \mathbf{Set}$. For every set X there is an arrow $\alpha_X : \mathrm{Homset}(X,2) \to \mathcal{P}(X)$ which takes characteristic functions on X to subsets of X by taking the inverse image of $\top$ under the characteristic function. That is, $\alpha_X(g) = g^{-1}(\top)$. Noting that $\mathrm{Homset}(f,2)(g) = g \circ f$ and that $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$, the collection α_X can be seen to be natural. The collection of maps $(\alpha_X)^{-1} : \mathcal{P}(X) \to \mathrm{Homset}(X,2)$ also happens to be natural, and since this is the case we say that $\mathrm{Homset}(-,2)$ and $\mathcal{P}(-)$ are naturally isomorphic.

We now turn our attention to coalgebra. First we examine coalgebras for the category of sets and functions.

Studies into coalgebra in the last two decades have explored this connection between coalgebras and transition structures [Jacobs and Rutten 1997] [Rutten 2000]. In algebra there is an evaluation function from a set of expressions to a set of values, while in coalgebra there is a transition function from a set of states to a set of successor objects. The duality of these two concepts is clarified by the use of functors in the construction of expression sets and successor sets. Algebra makes use of induction to define expressions and to prove properties about them, while coalgebra uses coinduction to define successors and to prove properties about them. Furthermore, the coalgebraic concept of bisimulation generalises the previous notions of bisimulation for specific state transition systems.

To see the duality between the algebraic and coalgebraic approaches, consider the example of lists and streams.

In the case of lists, we have construction by induction from a base case of the empty list, and the cons operation. We can also prove properties concerning lists inductively by proving them for the empty list and then showing that if they hold for the list then they hold for another symbol cons the list. Algebraically, if we take the set X of lists of symbols from the set L, we have $\mathsf{cons} : L \times X \to X$ and $\mathsf{empty} : 1 \to X$. Thus, by combining these functions disjointly, we have the list algebra

$$\mathit{eval} : 1 + L \times X \to X$$

as an algebra for the functor $1 + L \times (-) : \mathbf{Set} \to \mathbf{Set}$. An algebra for a functor F is just a function $f : FX \to X$ for some carrier set X. Meanwhile a coalgebra for a functor F is just a function $g : X \to FX$ for some carrier set X.

In the case of streams (lists that are possibly infinite), we have *decomposition* by *corecursion*, in which we split the stream into the head (a symbol) and tail (another stream). The result of this splitting on an empty stream is null. We can prove prop-

erties about streams coinductively (explain this). If X is the set of streams of symbols from the set L, we have the stream coalgebra

$$split : X \to 1 + L \times X$$

as a coalgebra for the functor $1 + L \times (-) : \mathbf{Set} \to \mathbf{Set}$. So, the list algebra and the stream coalgebra share the same functor.

More generally for a functor $F : \mathcal{C} \to \mathcal{C}$ (when the categories match we can call it an endofunctor), an algebra (F-algebra) is an arrow $\epsilon : FX \to X$ and a coalgebra (F-coalgebra) is an arrow $\gamma : X \to FX$. Thus the concept of coalgebra has an obvious extension to other categories.

Coalgebras in categories such as **Nom** (see section 2.2) and $\mathbf{Set}^{\mathbb{I}}$ (sets parametrised by a finite set of names) have been used to model name binding in process calculi such as the π-calculus [Fiore and Staton 2006]. Process calculi are vital to the study of mobile computation, and the study of process calculi also has wider implications for computation in general.

The general theory of coalgebra also has relevance outside computer science. In theoretical physics for instance, a Hopf algebra [Sweedler 1969] is defined as being simultaneously an algebra and a coalgebra for vector spaces over a common field, combined with some coherence conditions and an antipode map. Hopf algebras which are neither commutative nor co-commutative are sometimes called quantum groups due to their relevance to quantum mechanics.

2.2 Nominal Sets

In this section we provide background on the concepts of nominal sets and nominal logic. A more complete background can be found in [Clouston 2009].

Object level variables and binders are used in many different logics, especially logics with quantifiers. Binders are also used in computational calculi in which they make the scope of variables explicit and aid in the construction of generic functions and processes. Certain rules concerning the binding of object level variables in these systems require side conditions which are not captured in the syntax or semantics of the systems themselves.

Nominal sets have been proposed [Pitts 2003] as an explicit model of object level variables, which we will call names from now on. Nominal sets are basically sets in which each element is dependent on a certain set of names.

We start with an infinite set of *names* $\mathbb{A} = \{n_1, n_2, ...\}$, and consider the group $\mathbb{P}_{\mathbb{A}}$ of all finite name permutations (bijections $\mathbb{A} \to \mathbb{A}$ that only change finitely many names). Now we take a set X and a group action on X of finite name permutations. That is, $\cdot : \mathbb{P}_{\mathbb{A}} \times X \to X$ written infix, such that $\pi \cdot (\pi' \cdot x) = (\pi \circ \pi') \cdot x$ and $id \cdot x = x$, where id is the unit of the group given by the identity bijection.

From this, we can construct the set of names which $x \in X$ is dependent on. We say that x is *supported* by $S \subseteq \mathbb{A}$ if $\forall n_1, n_2 \in (\mathbb{A} \setminus S) \,.\, (n_1\ n_2) \cdot x = x$, where $(n_1\ n_2)$ is the permutation swapping just n_1 and n_2. If x is supported by a finite set of names, then

we say that x is finitely supported. If x is finitely supported then there is a unique smallest set of names that supports x, called *the support* of x or $\text{supp}(x)$.

Definition 2.2.1 (Nominal Set)**.** A nominal set is a set with a name permutation action as above such that every element of the set is finitely supported. So every element of the nominal set has a finite support of names, and the nominal set itself is closed under all name permutations.

Definition 2.2.2. We define a permutation action on the set of subsets of X.

$$\pi \cdot S \triangleq \{\pi \cdot x \mid x \in S\}$$

If we select only those subsets that are finitely supported w.r.t. this permutation action, then we get a nominal set $\mathcal{P}_{fs}(X)$. In general subsets of X might not be finitely supported. Take for example $X = \mathbb{A}$ and $S = \{a_{2n} \mid n \in \mathbb{N}\}$ every second name in some ordering of the names. Then S is not finitely supported.

Definition 2.2.3. We define a permutation action on the set of functions between nominal sets X and Y.

$$(\pi \cdot f)(x) \triangleq \pi \cdot_Y (f(\pi^{-1} \cdot_X x))$$

If we select only those functions that are finitely supported w.r.t. this permutation action, then we get a nominal set Y^X. In fact, $\mathcal{P}_{fs}(X)$ is just the same as 2^X where 2 is just the two element set $\{\top, \bot\}$ with a trivial name permutation action.

Those functions with empty support are called equivariant functions. Alternatively an equivariant function $f : X \to Y$ between nominal sets is a function $X \to Y$ such that

$$\pi \cdot_Y f(x) = f(\pi \cdot_X x)$$

Nominal sets and equivariant functions form a category **Nom**. Nominal sets and finitely supported functions also form a category, but this thesis focuses on the case of **Nom** for simplicity.

An important notion in **Nom** is that of orbit and orbit finiteness [Ciancia and Tuosto 2009].

Definition 2.2.4. if X is a nominal set, for $x \in X$ we define the *orbit* of x as

$$orb(x) = \{\pi \cdot x \mid \pi \in \mathbb{P}_{\mathbb{A}}\}$$

We have that $orb(x) \subseteq X$, since X is closed under permutation, and we have that if $orb(x) \cap orb(y) \neq \varnothing$ then $orb(x) = orb(y)$. This makes it possible to partition X into its orbits. If X has finitely many distinct orbits we call it *orbit finite*.

In category theory, there is a generic notion of the "finite" objects in a category, namely the notion of *finitely presentable* objects. Orbit finite nominal sets are the finitely presentable objects in **Nom** [Petrisan 2012, Proposition 2.3.7]. Thus orbit-finiteness is the appropriate notion of finiteness in **Nom** for many different purposes.

Now that we have some key notions about nominal sets, we can introduce some notions of nominal logic [Pitts 2003]. The first notion is that of freshness. If $x \in X$ a nominal set, and $a \in \mathbb{A}$ then $a\#x$ (pronounced a is fresh for x) means that $a \notin supp(x)$. For a subset $S \subseteq \mathbb{A}$, we say $S\#x$ to mean that $\forall a \in S \,.\, a\#x$. For finite subsets we usually write, for example: $a, b\#x$. If we want to say that a is fresh for multiple elements of different nominal sets $x_i \in X_i$, we can say $a\#x_1, x_2, x_3$.

We now present the $И$ quantifier. First we must recognise that logical formulae involving object level variables form a nominal set, where the support of a formula is the set of free names of that formula. Now we have:

$$Иn.\phi = \exists n'\#\phi \,.\, \phi[n'/n]$$

Where $\phi[n'/n]$ means ϕ with n' substituted for every instance of n. Since any two fresh names are logically indistinguishable for ϕ we actually have an equivalent definition

$$Иn.\phi = \forall n'\#\phi \,.\, \phi[n'/n]$$

These two definitions are equivalent by the some-any theorem of nominal logic. It also makes the quantifier self-dual, and means that it can be seen to distribute over other propositional connectives also. In other words:

$$\begin{aligned} Иn.\neg\phi &= \neg Иn.\phi \\ Иn.(\phi_1 \wedge \phi_2) &= (Иn.\phi_1) \wedge (Иn.\phi_2) \\ Иn.(\phi_1 \vee \phi_2) &= (Иn.\phi_1) \vee (Иn.\phi_2) \\ Иn.(\phi_1 \to \phi_2) &= (Иn.\phi_1) \to (Иn.\phi_2) \end{aligned}$$

2.3 Coalgebraic Modal Logic

In this section we cover the background of coalgebraic modal logic (CML) itself. We present the current framework in **Set**, as well as work on the framework outside of **Set**. We use [Pattinson 2012] as a key reference.

Since coalgebra successfully captures the notion of transition system, it makes sense to use it as a semantic framework for modal logic. Early work on the connection between coalgebras and modal logics [Kurz 1998] [Moss 1999] has been expanded upon with generic results concerning soundness, completeness and decidability of coalgebraic modal logics [Pattinson 2003] as well as a finite model property [Schröder 2007]. As mentioned above, there is also a generic decision procedure with good optimality bounds [Goré et al. 2010]. A recent summary of both the state of play and visions for the future of the field can be found in [Cîrstea et al. 2011].

The Coalgebraic Modal Logic framework we will use in this thesis is the predicate lifting model [Pattinson 2003]. There is one prominent competing model given by Moss [Moss 1999]. It is based on an infinitary syntax that is generated from the functor defining the logic. The reason for our choice of framework is that we are interested in lifting the results of the generic proof theory of coalgebraic modal logic on **Set** into

other categorical settings, and the work in proof theory for coalgebraic modal logic comes from the predicate lifting model.

It is worth noting that this coalgebraic modal logic framework is specific to rank 1 modal logics. That is, modal logics in which each modal operator can only look one transition ahead. Among other logics, temporal logics are ruled out by this choice, and though there has been work on coalgebraic modal logic beyond rank 1 [Pattinson and Schröder 2008b] [Círstea 2010], we make no comment on it here.

So, we first have a generic finitary syntax

$$\phi, \psi = p \mid \phi \wedge \psi \mid \neg\phi \mid \heartsuit(\phi)$$

Where $\heartsuit$ is a modal operator drawn from a set of modal operators Λ. Given above we have the unary example, but we are also allowed n-ary operators, and all results presented here generalise to the n-ary case easily. This fragment will generate the rest of the propositional connectives in the normal way.

As for the semantics, each different modal logic is represented by a successor functor. We will fix a functor T in the category of sets and functions. A model for this modal logic will be given by a coalgebra $\gamma : X \to T(X)$ and a valuation $v : V \to \mathcal{P}(X)$. Inside this model we have interpretations for the propositional connectives as follows

$$\begin{aligned} [\![p]\!]_{X,v} &= v(p) \\ [\![\phi_1 \wedge \phi_2]\!]_{X,v} &= [\![\phi_1]\!]_{X,v} \cap [\![\phi_2]\!]_{X,v} \\ [\![\neg\phi]\!]_{X,v} &= X \setminus [\![\phi]\!]_{X,v} \end{aligned}$$

For the modal operators we must introduce the notion of predicate lifting. A predicate lifting is a natural transformation from the contravariant powerset functor $\mathcal{P}$ to $\mathcal{P} \circ T$ (where functor composition amounts to composing the maps which constitute the functor). We interpret the modal operators as predicate liftings, and focus on the component at X. That is, $[\![\heartsuit]\!]_X : \mathcal{P}(X) \to \mathcal{P}(T(X))$. This should be conceptualised as mapping predicates on X (the state space) to predicates on TX (the successor space). We regain the interpretation of formulae as subsets of X by

$$\begin{aligned} [\![\heartsuit(\phi)]\!]_{TX,v} &= [\![\heartsuit]\!]_X([\![\phi]\!]_{X,v}) \\ [\![\heartsuit(\phi)]\!]_{X,v} &= \gamma^{-1}([\![\heartsuit(\phi)]\!]_{TX,v}) \end{aligned}$$

This generalises to n-ary predicate liftings by taking natural transformations of the form $\mathcal{P}(-)^n \dot{\to} \mathcal{P}(T(-))$.

Example 2.3.1. The normal modal logic K is modelled by coalgebras of the type

$$X \to \mathcal{P}(X)$$

Since every state has a set of successors.

Example 2.3.2. Hennessy Milner Logic (multimodal K) is modelled by coalgebras of

the type

$$X \to \mathcal{P}(X)^A$$

where A is the set of transition labels. Since every state $x \in X$ has a set of successors along a particular $a \in A$, we can conceptualise this as x having a successor which is a function from transition labels to sets of states.

An obvious question to ask at this point is whether a given syntax matches a given functor. Namely, we want to know whether Λ is the correct set of modalities for T. This is the question of expressivity. This basically amounts to whether coalgebraic bisimilarity and logical equivalence coincide. We have this, at least for the case of **Set**, whenever T is finitary and Λ is big enough that it can be used to logically separate distinct successors.

This expressivity question has been asked in the literature of coalgebraic modal logic beyond **Set**. An interesting development here has been the realisation that expressivity comes for free in the presence of a certain type of adjunction between the algebraic syntax and the coalgebraic semantics. We do not follow this route, and instead we aim to get expressivity in the same way as in the case of **Set**, though we do not investigate this thoroughly, as we are primarily interested in developing the generic proof theory in **Nom**.

The generic proof theory in **Set** consists of generic results that simplify the work of proving soundness and completeness results for new examples. A set of rules is assumed, and certain properties of that rule set in relation to T, namely one step soundness and one step completeness, are shown to give soundness and completeness results respectively. These one step properties are much easier to verify on the level of an individual logic.

Additionally, generic results in decidability simplify the job of proving a logic to be decidable and constructing a decision procedure for it. Given that the rule set absorbs cut and is contraction closed, it will be decidable. Better yet, a generic decision procedure with good optimality properties has been developed [Goré et al. 2010]. This means that the work of developing a proof procedure for a new rank 1 modal logic is reduced to fitting it into the framework and then proving these relatively simple properties. Implementation of a generic algorithm is a complicated endeavour, but the developement of a modular approach that allows the specifics of the rules and functors to be added ad hoc is a reasonable expectation in the short to medium term.

Beyond **Set**, there have been a few attempts at building a framework for coalgebraic modal logic. Perhaps the most notable of these is the Kan extension approach to constructing expressive logics for accessible functors in locally presentable categories [Klin 2007]. This work examines expressivity specifically, and does not attempt proof theory. This is a common trend in the literature. Coalgebraic Modal Logic has also been examined over measurable spaces [Schubert 2009] and analytic spaces [Schubert 2007] and but again the focus is on expressivity. Coalgebra over slice categories have been examined as a means of bringing the valuation into the coalgebra itself [Pattinson and Schröder 2008a].

On the other hand, there are many rank 1 modal logics that cannot be captured in

the **Set** framework for various reasons. One challenge is the use of the И quantifier in the logic, which requires a nominal setting for the semantics and a different kind of rule system than the one the **Set** framework offers. Examples of these logics typically come from the field of concurrency [Nicola and Loreti 2008] [Caires and Cardelli 2002] [Caires and Cardelli 2003]. Concurrency theory uses names in a fundamental way to represent resources and channels.

There are other challenges with these kinds of logics, for example, the use of fixed point operators [Venema 2004]. Each of these challenges needs to be addressed in isolation first for the sake of simplicity. In the fullness of time these isolated approaches will need to be combined in order to deal with the many different concepts in logics of concurrency. The way we choose to deal with the problem of the И quantifier is to change our base category to **Nom** and rebuild the generic proof theory of coalgebraic modal logic in this new setting.

In general, we want to be able to reproduce the generic proof theory in a number of different categories to model different aspects of logic within modal logics. For example, the propositional logic of CML is determined by the standard set algebra operations in **Set**. Many other categories have different internal logics. This observation may provide a possible avenue for adding modalities to subclassical logics in a generic way. In addition to this, other concepts in the field of modal logic might be captured by a change of category to another internally classical category (such as **Nom**).

When moving to another category, we will likely need a subobject classifier [Goldblatt 2006] and exponentiation to define predicate liftings (though there is a possibility that another categorical monoid might also give us something useful). The propositional logic is determined by the internal logic of the category, and specifically the meanings of intersection, union and complement. Topoi give us a subobject classifier and exponentiation, and in general the internal logic will be intuitionistic, though there are classical topoi, such as **Set** and **Nom**. Possible Topoi worth examining include presheaf topoi [Goldblatt 2006]. The Shanuel topos, which is equivalent to **Nom**, is based on a presheaf topoi, and many of the other categories which have been used to model names are also based on presheaf topoi.

2.4 Modal logic and tableau

In this section we provide some background on proof theory in modal logic. This grounds the later discussion about proof theory in coalgebraic modal logic, which is distinguished in the sense that it applies generically to all logics in the framework.

Modal logics can be specified by specifying a syntax, semantics and a system of reasoning, be it a Hilbert style axiom system or a system of rules for deduction. The obvious questions that arise out of this specification are whether the system of reasoning is sound and complete for the semantics and whether a decision procedure can be written to automatically check the satisfiability/validity of a statement in the syntax. Hilbert style axiom systems have a certain philosophical appeal, but deduction rules

are much easier to turn into a decision procedure. Proof theory in modal logic has essentially converged to sequent/tableau systems due to this generic framework being useful for many modal and temporal logics.

Tableau proof search takes a statement in the syntax of the logic in question, converts it to negation normal form, and then deconstructs it recursively using inverted deduction rules in an attempt to find dependencies on conflicting literals. This deconstruction is recursive on the structure of the statement, which is tree-like, and if a tableau has conflicting literals in all of its branches it is called closed and the formula is unsatisfiable. There can be some duplication between different parts of the tableau tree, and various methods have been proposed to avoid this duplication. Duplication is asymptotically significant in the worst case. For many modal logics it means the difference between worst case 2-EXPTIME and worst case EXPTIME [Goré and Nguyen 2007a].

Two important techniques in tableau proof search are caching for avoiding duplication, and blocking for avoiding infinite loops. Caching is a method in which the satisfiability status of formulae are remembered during the search and the algorithm checks for cache hits before evaluating the satisfiability of a formula. This way, the satisfiability of each formula is only computed once. Blocking is a method in which formulae are checked against their ancestors in the tableau tree in order to avoid loops. When a loop is detected and contains no conflicting literals the tableau marks the branch as open/satisfiable.

Resolving caching with blocking is often a complicated endeavour. In general the deconstruction of the formula relies in a nondeterministic choice of tableau rules (recall that these are the inverted deduction rules). This nondeterminism results in a forest of tableau, and if any of these tableau close we know that the formula is unsatisfiable. In the case of a forest of tableau, naively caching the satisfiability of formula between tableau can be unsound. This unsoundness arises when a formula is marked as satisfiable in an open tableau due only to the nondeterministic choice of rule applications which lead to a loop being marked as open when another choice would lead to it being marked as closed. Without caching, this nondeterministic openness would not be problematic, since we only need a single closed tableau and we will find it eventually. However, caching this formula as satisfiable is incorrect, and can lead to the decision procedure not finding a refutation of an unsatisfiable formula.

In the case of the description logic 'Attributive Concept Language with Complements' (ALC) [Schmidt-Schauß and Smolka 1991] (the modal logic equivalent being Hennessey Milner Logic, by which we mean the normal modal logic K modified by allowing multiple separate transition relations, each with its own transition label - we have modalities $\langle a \rangle$ and $[a]$ for all a in the set of transition labels), a solution to the problem of caching in the face of nondeterminism was given as global caching [Goré and Nguyen 2007a]. Despite its high storage space requirements, the global caching technique is conceptually simpler than the methods of Donini and Massacci [2000] and Ding and Haarslev [2006], and so is more readily optimised in a provably sound way with propagation and cutoffs [Goré and Nguyen 2007b]. This simplicity has also lead to an investigation of the generality of the method for use in other modal logics,

leading to the development of a generic global caching algorithm which is applicable to an entire class of coalgebraic modal logics [Goré et al. 2010].

2.5 Global Caching

In this section we address the original global caching algorithm [Goré and Nguyen 2007a].

The original global caching algorithm is a decision procedure for ALC. Global caching uses a tableau proof method, but in order to avoid complicated cross-pass subtleties, it seeks to build a single and-or graph, and probe it for satisfiability. Here the "and-nodes" are those that require all of their children to be satisfiable for the node to be satisfiable, that is, the $\langle a \rangle$-nodes. The "or-nodes" are nodes that require at least one of their children to be satisfiable. That is, $(\vee)$-nodes (one of the two children must be satisfiable) and $(\wedge)$-nodes (the single child must be satisfiable). Leaf nodes contain either a set of noncontradicting literals, or the single literal $\bot$.

Global caching is achieved in the construction of the graph structure. No duplicate nodes are created, instead directed edges can be added to point to existing nodes. The key is that in this graph, unsatisfiability of a node can only be inherited from a child already labelled as unsatisfiable, or from a child labelled as $(\bot)$. If such a node exists, its parents are examined for unsatisfiability, and so on.

The global caching method will mark the root node as unsatisfiable iff the formula is unsatisfiable with respect to the global assumptions. This is because a consistent marking on the and-or tree can be used to construct a multimodal Kripke frame in which the global assumptions hold at all worlds and the formula holds at one.

Starting at the root, a saturation path (in a consistent marking) corresponds to a path from the starting node to an $(\wedge)$-node. This constitutes a sequence of choices of or-children which represents a consistent model of the non modal concepts in the formula (given the global assumptions). Thus, we can take this model an create an initial world in our frame which validates it. Then we must create successor worlds to this world for all the AND children. The marking has already labelled the arcs to these children with the relations that the new worlds will have with our current world, so we force those relations between the current world and the new worlds. For these new worlds we will have to take a saturation path to define the model for that world. Children of and-nodes may have already been visited in this process and if so, we simply force the relations between the current world and that existing world, instead of creating a new world. An and-node with no children is satisfiable (because the marking is consistent) and no successor worlds must be added to the frame. This process terminates because at worst it can expand all subformulae of the formula and will never create duplicate worlds. Moreover it never forces a contradiction, because the consistent marking contains no contradictory nodes.

As mentioned above, once we have the and-or graph, we only need to find the node labelled $\bot$ and push unsatisfiability up the graph based on the rule that if any child of an and-node is unsatisfiable then the and-node is unsatisfiable, and if all chil-

dren of an or-node are unsatisfiable then the or-node is unsatisfiable.

In this way the root node is marked unsatisfiable iff the formula is unsatisfiable wrt the global assumptions. If the root node is unsatisfiable it is because there is no consistent marking, because all markings include the $\perp$ node. If the root node is left unmarked, there will be a consistent marking in the and-or graph, because starting from the root node, at least one child of every or-node reached is not marked unsatisfiable and all children of and-nodes reached are not marked unsatisfiable, and the only way to be unsatisfiable in the and-or graph is to have unsatisfiable children (any for an and-node and all for an or-node) and since the graph is finite the cause of the unsatisfiability is ultimately the $\perp$ node.

This basic version of the algorithm requires the storage of all subformulae of the formula, and as such requires exponential storage space. This can be avoided in certain cases by performing on-the-fly propagation of unsatisfiability and satisfiability (satisfiability can be propagated from a non $\perp$ leaf node by dual rules). This cuts down the search space because in the best case it is only necessary to expand one child of any contradictory AND node (the contradictory child).

Chapter 3

Semantic Framework

In this chapter we introduce the core notions of the semantic framework for coalgebraic modal logic in the category of nominal sets and equivariant functions. Much of this is adapted from the semantics framework for coalgebraic modal logic in the category of sets and functions [Pattinson 2012].

3.1 Equivariant Coalgebra and Finitely Supported Predicate Liftings

We start with the modified definition of coalgebras. A coalgebra is an equivariant function from a nominal set of states X (intuitively comprised of "state expressions", such that each expression corresponds to a state $x \in X$) to a nominal set of state successors TX, where T is a functor in the category of nominal sets and equivariant functions.

Definition 3.1.1 (T-coalgebra)**.** A coalgebra for a functor T is given by a nominal set of states X and an equivariant "successor" function:

$$\gamma : X \to TX$$

Which picks out a successor for each state (the former being an element of the nominal successor set given by the action of the functor T on the state space).

Now, we define the notion of a predicate lifting, which is meant to capture the semantics of a modal operator as a collection of maps (one for each possible state space) from predicates on the state space (given by nonmodal formulae) to predicates on the successor space (given by modalised formulae).

In traditional coalgebraic modal logic the modal operators in a logic belong to a set (the set of modalities of the logic). This is generally obvious by the way individual coalgebraic modal logics are presented, and the set does not have any extra required properties (though the elements do), so it is often left implicit. In Nominal CML, the set Λ of modal operators in a logic is to be a nominal set, because we wish to capture the concept of modalities dependent on a finite support of names. The potentially nonempty support of modal operators needs to be highlighted by the semantics. It

becomes useful to make explicit the previously implicit map in the traditional case:

$$\Lambda \xrightarrow{[\![-]\!]} \mathrm{Nat}(\mathcal{P}, \mathcal{P} \circ T)$$

In the nominal case, the contravariant powerset functor becomes the contravariant finitely supported powerset functor, which takes each nominal set X to the nominal set of finitely supported subsets of X. Furthermore, as we just acknowledged, since modal operators are finitely supported, their interpretation should be finitely supported, thus leading us to the notion of a finitely supported natural transformation, which will be made clear shortly. Now the traditional interpretation has become an equivariant map:

$$\Lambda \xrightarrow{[\![-]\!]} \mathrm{Nat}_{fs}(\mathcal{P}_{fs}, \mathcal{P}_{fs} \circ T)$$

The appropriate concept for the interpretation of these nominal modalities is one where the natural transformation in question is not made up from equivariant components, but from components of finite support. Each component of a predicate lifting is intended to interpret syntactically identical modalities in different state sets. Often we will choose the semantics such that the components have the same support as the whole, but we do allow for components to have a smaller support than the whole.

Definition 3.1.2 (predicate lifting for T)**.** A predicate lifting is a finitely supported natural transformation

$$\lambda : \mathcal{P}_{fs}(-) \xrightarrow[fs]{\cdot} \mathcal{P}_{fs}(T-)$$

That is, a collection of functions λ_X (each with support a subset of $\sup(\lambda)$), one for each nominal set X, such that the following diagram commutes for all equivariant $f : X \to Y$.

$$\begin{array}{ccc} \mathcal{P}_{fs}(X) & \xrightarrow{\lambda_X} & \mathcal{P}_{fs}(TX) \\ {\scriptstyle f^{-1}}\uparrow & & \uparrow{\scriptstyle (Tf)^{-1}} \\ \mathcal{P}_{fs}(Y) & \xrightarrow[\lambda_Y]{} & \mathcal{P}_{fs}(TY) \end{array}$$

The case of polyadic predicate liftings is an easy generalisation of this, by taking components $(\mathcal{P}_{fs}(X))^n \to \mathcal{P}_{fs}(TX)$. In fact, all the results in the thesis lift easily to the case of polyadic predicate liftings. As discussed, we will interpret modalities as predicate liftings.

Definition 3.1.3. The finitely supported predicate liftings $\mathrm{Nat}_{fs}(\mathcal{P}_{fs}, \mathcal{P}_{fs} \circ T)$ form a nominal set in which $(\pi \cdot \lambda)_X = \pi \cdot (\lambda_X)$.

Note that $\pi \cdot (\lambda_X)$ is given by definition 2.2.3.

Theorem 3.1.4. *$Nat_{fs}(\mathcal{P}_{fs}, \mathcal{P}_{fs} \circ T)$ is closed under the given permutation action.*

Proof. Consider the following diagram:

$$\begin{array}{ccccccc}
\mathcal{P}_{fs}(X) & \xrightarrow{\pi^{-1}\cdot} & \mathcal{P}_{fs}(X) & \xrightarrow{\lambda_X} & \mathcal{P}_{fs}(TX) & \xrightarrow{\pi\cdot} & \mathcal{P}_{fs}(TX) \\
{\scriptstyle f^{-1}}\uparrow & & {\scriptstyle f^{-1}}\uparrow & & \uparrow{\scriptstyle (Tf)^{-1}} & & \uparrow{\scriptstyle (Tf)^{-1}} \\
\mathcal{P}_{fs}(Y) & \xrightarrow[\pi^{-1}\cdot]{} & \mathcal{P}_{fs}(Y) & \xrightarrow[\lambda_Y]{} & \mathcal{P}_{fs}(TY) & \xrightarrow[\pi\cdot]{} & \mathcal{P}_{fs}(TY)
\end{array}$$

The left and right squares commute because f is equivariant. The middle one commutes because λ is a predicate lifting. Therefore, the outer rectangle commutes, which is exactly the requirement for $\pi \cdot \lambda$ to be a finitely supported predicate lifting in $\mathrm{Nat}_{fs}(\mathcal{P}_{fs}, \mathcal{P}_{fs} \circ T)$.

□

3.1.1 Example: Nominal Hennessy Milner Logic

In this section we present the example of Nominal Hennessy Milner Logic. Hennessy Milner Logic is a multimodal version of the normal modal logic K. Nominal Hennessy Milner Logic is a modified version in which the transition labels have a naming structure applied to them. For simplicity, we take the label set to simply be the set of names $\mathbb{A}$. We will use $n, m \in \mathbb{A}$ for names/transition labels.

The models for Nominal Hennessy Milner Logic are coalgebras for the covariant functor $\mathcal{P}_{fs}(-)^{\mathbb{A}}$ on NOM. That is, equivariant functions of the form:

$$\gamma : X \to \mathcal{P}_{fs}(X)^{\mathbb{A}}$$

The set of modalities Λ is given by

$$\Lambda = \{\langle n \rangle \mid n \in \mathbb{A}\} \cup \{[n] \mid n \in \mathbb{A}\}$$

Where the permutation action simply changes the name within the modality without changing the type. We now define $[\![\langle n \rangle]\!] : \mathcal{P}_{fs}(-) \underset{fs}{\dot{\to}} \mathcal{P}_{fs}(T-)$. In a coalgebra (X, γ), if A is a finitely supported subset of X, then:

$$[\![\langle n \rangle]\!]_X(A) = \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(n) \cap A \neq \varnothing\}$$

Note that this does not depend on the choice of γ, but later, when interpreting formulae as subsets of X, γ becomes important. We now show that $[\![\langle n \rangle]\!]$ is a predicate

lifting in the sense given above by showing that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{P}_{fs}(X) & \xrightarrow{[\![\ \langle n \rangle\ ]\!]_X} & \mathcal{P}_{fs}(\mathcal{P}_{fs}(X)^{\mathbb{A}}) \\ {\scriptstyle f^{-1}}\uparrow & & \uparrow{\scriptstyle (\mathcal{P}_{fs}(f)^{\mathbb{A}})^{-1}} \\ \mathcal{P}_{fs}(Y) & \xrightarrow[[\![\ \langle n \rangle\ ]\!]_Y]{} & \mathcal{P}_{fs}(\mathcal{P}_{fs}(Y)^{\mathbb{A}}) \end{array}$$

Proof.

$$\begin{aligned} (\mathcal{P}_{fs}(f)^{\mathbb{A}})^{-1} \circ [\![\ \langle n \rangle\ ]\!]_Y(B) &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid \mathcal{P}_{fs}(f)^{\mathbb{A}}(g) \in [\![\ \langle n \rangle\ ]\!]_Y(B)\} \\ &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid \mathcal{P}_{fs}(f) \circ g \in [\![\ \langle n \rangle\ ]\!]_Y(B)\} \end{aligned}$$

Now

$$\begin{aligned} \mathcal{P}_{fs}(f) \circ g \in [\![\ \langle n \rangle\ ]\!]_Y(B) &\iff \mathcal{P}_{fs}(f) \circ g(n) \cap B \neq \varnothing \\ &\iff g(n) \cap f^{-1}(B) \neq \varnothing \\ &\iff g \in [\![\ \langle n \rangle\ ]\!]_X(f^{-1}(B)) \end{aligned}$$

So

$$\begin{aligned} (\mathcal{P}_{fs}(f)^{\mathbb{A}})^{-1} \circ [\![\ \langle n \rangle\ ]\!]_Y(B) &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g \in [\![\ \langle n \rangle\ ]\!]_X(f^{-1}(B))\} \\ &= [\![\ \langle n \rangle\ ]\!]_X \circ f^{-1}(B) \end{aligned}$$

□

The interpretation of the dual modality $[\![\ [n]\ ]\!]$ defined:

$$[\![\ [n]\ ]\!]_X(A) = \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(n) \subseteq A\}$$

can be shown to be a predicate lifting via similar reasoning, or by Theorem 3.2.3 below.

3.2 Syntax and Semantics

Definition 3.2.1 (Signature)**.** The signature of a coalgebraic modal logic is

$$\mathcal{F}(\Lambda) \ni \varphi, \psi ::= p \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \heartsuit(\varphi)$$

Though we may demand that formulae are in negation normal form, so that

$$\mathcal{F}(\Lambda) \ni \varphi, \psi ::= p \mid \overline{p} \mid \top \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \heartsuit(\varphi) \mid \overline{\heartsuit}(\varphi)$$

Where $\overline{\heartsuit}$ is the dual of the modal operator $\heartsuit$. See theorem 3.2.3.

Note that $\mathcal{F}(\Lambda)$ is a nominal set, where the permutation distributes over all propositional connectives and applies to the modal operators as in:

$$\pi \cdot (\heartsuit(\phi)) = (\pi \cdot \heartsuit)(\pi \cdot \phi)$$

Definition 3.2.2 (Interpretation)**.** Given a T-coalgebra (X, γ) and an equivariant valuation $v : V \to \mathcal{P}_{fs}(X)$ the interpretation of formulae is given as follows. Interpretation is an equivariant map

$$[\![-]\!]_{X,v} : \mathcal{F}(\Lambda) \to \mathcal{P}_{fs}(X)$$

from formulae to finitely supported subsets of the state space X.

In what follows, we note that all of the interpretations are finitely supported (in particular the complement of a finitely supported subset is finitely supported), and we also note that the resulting interpretation function is equivariant in a sense that will be made clearer in the next section.

The interpretation of formulae when the outermost connective is nonmodal:

$$\begin{aligned}
[\![\top]\!]_{X,v} &= X \qquad [\![\bot]\!]_{X,v} = \varnothing \\
[\![p]\!]_{X,v} &= v(p) \qquad [\![\neg\phi]\!]_{X,v} = X \setminus [\![\phi]\!]_{X,v} \\
[\![\phi_1 \wedge \phi_2]\!]_{X,v} &= [\![\phi_1]\!]_{X,v} \cap [\![\phi_2]\!]_{X,v} \\
[\![\phi_1 \vee \phi_2]\!]_{X,v} &= [\![\phi_1]\!]_{X,v} \cup [\![\phi_2]\!]_{X,v} \\
[\![\phi_1 \to \phi_2]\!]_{X,v} &= [\![\phi_2]\!]_{X,v} \cup X \setminus [\![\phi_1]\!]_{X,v}
\end{aligned}$$

For modal formulae, we first interpret the modal operator

$$[\![\heartsuit]\!] : \mathcal{P}_{fs}(-) \stackrel{\cdot}{\to}_{fs} \mathcal{P}_{fs}(T-)$$

Note that $[\![\heartsuit]\!]$ is the interpretation of the modal operator as a predicate lifting, the details of which must be given for each individual logic. Now, we can define the interpretation of a modal formulae:

$$\begin{aligned}
[\![\heartsuit(\varphi)]\!]_{TX,v} &= [\![\heartsuit]\!]_X([\![\varphi]\!]_{X,v}) \\
[\![\psi]\!]_{X,v} &= \gamma^{-1}([\![\psi]\!]_{TX,v})
\end{aligned}$$

At a specific state $x \in X$:

$$X, x, v \vDash \phi \text{ iff } x \in [\![\phi]\!]_{X,v}$$

in particular

$$X, x, v \vDash \heartsuit(\varphi) \text{ iff } \gamma(x) \in [\![\heartsuit]\!]_X([\![\varphi]\!]_{X,v})$$

Validity of formulae for $\mathsf{Coalg}(T)$ under a valuation v is defined as follows

$$\begin{aligned}
X, v \vDash \phi &\text{ iff } [\![\phi]\!]_{X,v} = X \\
TX, v \vDash \heartsuit(\varphi) &\text{ iff } [\![\heartsuit]\!]_X([\![\varphi]\!]_{X,v}) = TX
\end{aligned}$$

Generic reasoning about dual modalities gives the following

Theorem 3.2.3. *The dual of a predicate lifting is a predicate lifting, so if*

$$\overline{\heartsuit} = \neg\heartsuit\neg$$

and $[\![\heartsuit]\!]$ *is a predicate lifting, then so too is* $[\![\overline{\heartsuit}]\!]$.

Proof. Let $\neg : \mathcal{P}_{fs}(A) \to \mathcal{P}_{fs}(A)$ refer to the function which takes the complement of a finitely supported subset of A. Note that this complement is still finitely supported. The function $\neg$ is defined for any nominal A. Now the following diagram

$$\begin{array}{ccccccc}
\mathcal{P}_{fs}(X) & \xrightarrow{\neg} & \mathcal{P}_{fs}(X) & \xrightarrow{[\![\heartsuit]\!]_X} & \mathcal{P}_{fs}(TX) & \xrightarrow{\neg} & \mathcal{P}_{fs}(TX) \\
{\scriptstyle f^{-1}}\uparrow & & {\scriptstyle f^{-1}}\uparrow & & \uparrow{\scriptstyle (Tf)^{-1}} & & \uparrow{\scriptstyle (Tf)^{-1}} \\
\mathcal{P}_{fs}(Y) & \xrightarrow[\neg]{} & \mathcal{P}_{fs}(Y) & \xrightarrow[[\![\heartsuit]\!]_Y]{} & \mathcal{P}_{fs}(TY) & \xrightarrow[\neg]{} & \mathcal{P}_{fs}(TY)
\end{array}$$

represents the dual predicate lifting. Since all the squares commute, the outside commutes and $[\![\overline{\heartsuit}]\!]$ is indeed a predicate lifting. □

3.3 Substitutions and valuations

A substitution is traditionally a function $\sigma : V \to \mathcal{F}(\Lambda)$, in which propositional variables are replaced by arbitrary formulae. Meanwhile, a valuation is traditionally a function $v : V \to \mathcal{P}(X)$ which interprets propositional variables as a subset of the state space.

Since we have presented the semantics in terms of nominal sets and equivariant functions, it makes sense to attempt to give an account of equivariant substitutions and equivariant valuations. The case of equivariant substitutions makes certain things clear about what to do about valuations, and in some sense substitutions are more primitive, and more problematic.

Given an equivariant substitution $\sigma : V \to \mathcal{F}(\Lambda)$, a permutation $(a\,b)$, and a propositional variable $p \in V$, we have:

$$(a\,b) \cdot \sigma(p) = \sigma((a\,b) \cdot p)$$

If variables are to be nameless, and we want substitutions to preserve that namelessness, then we can't have interesting substitutions, since it is by no means the case that $(a\,b) \cdot \sigma(p) = \sigma(p)$ for interesting σ. So, we have two options. One is to defer the support of the propositional variables using a notion of permutation moderated variables, or suspensions [Urban et al. 2004]. Alternatively, we could allow finitely supported substitutions and keep propositional variables nameless. We choose the latter.

This change does not require a change of category for the coalgebraic semantics (which are still given in terms of equivariant maps), it merely requires that substitutions be finitely supported functions. In practice substitutions are only ever applied to finitely many variables, and those variables are substituted for finite formulae. So, any practical instance of a substitution can be made finitely supported by being the identity on all variables that are not present, or by treating all those variables in some other uniform way.

It is important that the concepts of valuations and substitutions agree. Specifically, we want to recover the following important relationship:

$$[\![\phi\sigma]\!]_{X,v} = [\![\phi]\!]_{X,\hat{v}}$$

Where $\hat{v}(p) = [\![\sigma(p)]\!]_{X,v}$. We therefore need, for any valuation v and substitution σ, that $\hat{v}$ is an acceptable valuation. Given that $\hat{v}$ is a valuation of the appropriate type, this relationship can be established by induction on the structure of the formula ϕ, with the base case of propositional variables being almost immediately obvious.

This result is critical to the one-step theorems, because they rely on a relationship between R-derivability (given in terms of syntax and using substitutions) and validity (given in terms of semantics and using valuations). Adding substitutions to the one-step definitions is possible, but in that case it becomes more difficult to prove that individual logics fit the definition, and it introduces a syntactic element that seems less than ideal. Simultaneously, the alternative of adding valuations to the definition of derivability is unacceptable, because derivability must be a purely syntactic notion.

It is for these reasons that if substitutions have to be finitely supported, it makes sense to take finitely supported valuations, so that $\hat{v}$ is a valuation for any valuation v and substitution σ.

3.4 Equivariance of Interpretation

We now show that despite having finitely supported substitutions and valuations, the interpretation function given above is equivariant in the following sense:

Theorem 3.4.1 (equivariant interpretation)**.** *For all $\phi \in \mathcal{F}(\Lambda)$, $v : V \to \mathcal{P}_{fs}(X)$ a valuation, and π a name permutation:*

$$\pi \cdot [\![\phi]\!]_{X,v} = [\![\pi \cdot \phi]\!]_{X,\pi\cdot v}$$

Proof. We prove this by induction on the structure of ϕ.

Base case: ($\phi = p$)

$$[\![\pi \cdot p]\!]_{X,\pi\cdot v} = (\pi \cdot v)(\pi \cdot p)$$

By definition 2.2.3 we have $(\pi \cdot v)(\pi \cdot p) = \pi(v(\pi^{-1}(\pi \cdot p)))$

$$\begin{aligned}
[\![\pi \cdot p]\!]_{X,\pi\cdot v} &= \pi(v(\pi^{-1}(\pi \cdot p))) \\
&= \pi(v(p)) \\
&= \pi \cdot (v(p)) \\
&= \pi \cdot [\![p]\!]_{X,v}
\end{aligned}$$

Inductive case: ($\phi = \heartsuit(\psi)$ - other connectives are trivial)

Lemma: (Note we are inductively assuming that $\pi \cdot [\![\psi]\!]_{X,v} = [\![\pi \cdot \psi]\!]_{X,\pi\cdot v}$)

$$\begin{aligned}
\pi \cdot [\![\heartsuit(\psi)]\!]_{TX,v} &= \pi \cdot ([\![\heartsuit]\!]_X([\![\psi]\!]_{X,v}) \\
&= (\pi \cdot [\![\heartsuit]\!]_X)(\pi \cdot [\![\psi]\!]_{X,v}) \\
&= (\pi \cdot [\![\heartsuit]\!]_X)([\![\pi \cdot \psi]\!]_{X,\pi\cdot v})
\end{aligned}$$

Now, by definition 3.1.3, we have $(\pi \cdot [\![\heartsuit]\!])_X = \pi \cdot [\![\heartsuit]\!]_X$ so:

$$\begin{aligned}
\pi \cdot [\![\heartsuit(\psi)]\!]_{TX,v} &= (\pi \cdot [\![\heartsuit]\!])_X([\![\pi \cdot \psi]\!]_{X,\pi\cdot v}) \\
&= [\![\pi \cdot \heartsuit]\!]_X([\![\pi \cdot \psi]\!]_{X,\pi\cdot v}) \\
&= [\![(\pi \cdot \heartsuit)(\pi \cdot \psi)]\!]_{TX,\pi\cdot v} \\
&= [\![\pi \cdot (\heartsuit(\psi))]\!]_{TX,\pi\cdot v}
\end{aligned}$$

Note that since the interpretation of modalities $[\![-]\!] : \mathsf{Op} \to \mathrm{Nat}_{fs}(\mathcal{P}_{fs}, \mathcal{P}_{fs} \circ T)$ is equivariant we have that $\pi \cdot [\![\heartsuit]\!] = [\![\pi \cdot \heartsuit]\!]$.

Main result:

$$\begin{aligned}
\pi \cdot [\![\heartsuit(\psi)]\!]_{X,v} &= \pi \cdot \gamma^{-1}([\![\heartsuit(\psi)]\!]_{TX,v}) \\
&= \gamma^{-1}(\pi \cdot [\![\heartsuit(\psi)]\!]_{TX,v}) \quad \gamma \text{ equivariant} \\
&= \gamma^{-1}([\![\pi \cdot (\heartsuit(\psi))]\!]_{TX,\pi\cdot v}) \\
&= [\![\pi \cdot (\heartsuit(\psi))]\!]_{X,\pi\cdot v}
\end{aligned}$$

□

3.5 Coalgebra morphisms and truth invariance

In this section we prove a result about the interpretation of a single formula across different coalgebra. We use the notion of coalgebra morphism to capture the relationship between interpretations.

Definition 3.5.1 (Coalgebra morphism)**.** A coalgebra morphism $f : (X, \gamma) \to (Y, \delta)$ is

an equivariant function $f : X \to Y$ such that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle\gamma}\downarrow & & \downarrow{\scriptstyle\delta} \\ TX & \xrightarrow[Tf]{} & TY \end{array}$$

Lemma 3.5.2 (Coalgebra morphisms preserve modalities)**.** *Suppose $f : (X, \gamma) \to (Y, \delta)$ is a coalgebra morphism, λ is a predicate lifting, and $A \in \mathcal{P}_{fs}(Y)$. Then*

$$f^{-1} \circ \delta^{-1} \circ \lambda_Y(A) = \gamma^{-1} \circ \lambda_X \circ f^{-1}(A)$$

Proof.

$$\begin{array}{ccccc} \mathcal{P}_{fs}(X) & \xrightarrow{\lambda_X} & \mathcal{P}_{fs}(TX) & \xrightarrow{\gamma^{-1}} & \mathcal{P}_{fs}(X) \\ {\scriptstyle f^{-1}}\uparrow & & \uparrow{\scriptstyle (Tf)^{-1}} & & \uparrow{\scriptstyle f^{-1}} \\ \mathcal{P}_{fs}(Y) & \xrightarrow[\lambda_Y]{} & \mathcal{P}_{fs}(TY) & \xrightarrow[\delta^{-1}]{} & \mathcal{P}_{fs}(Y) \end{array}$$

The square on the left commutes because λ is a predicate lifting. The square on the right can be obtained by applying the contravariant powerset functor to diagrammatic definition of coalgebra morphisms, and thus it commutes as well. Together this implies the outside of the whole diagram commutes, which is precisely the lemma. □

Theorem 3.5.3 (Truth is invariant under coalgebra-morphisms)**.** *If $f : (X, \gamma) \to (Y, \delta)$ is a coalgebra morphism, $v : V \to \mathcal{P}_{fs}(Y)$ is a valuation, and $\varphi \in \mathcal{F}(\Lambda)$, then*

$$f^{-1}([\![\, \varphi \,]\!]_{Y,v}) = [\![\, \varphi \,]\!]_{X, f^{-1} \circ v}$$

Proof. Note first that $f^{-1} \circ v : V \to \mathcal{P}_{fs}(X)$ is indeed a valid valuation.

We proceed by induction on the structure of φ, using lemma 3.5.2 for the modal case.

Base case: ($\varphi = p$)

$$f^{-1}([\![\, p \,]\!]_{Y,v}) = f^{-1}(v(p)) = [\![\, p \,]\!]_{X, f^{-1} \circ v}$$

Inductive case 1: ($\varphi = \varphi_1 \wedge \varphi_2$)

$$\begin{aligned} f^{-1}([\![\, \varphi_1 \,]\!]_{Y,v} \cap [\![\, \varphi_2 \,]\!]_{Y,v}) &= f^{-1}([\![\, \varphi_1 \,]\!]_{Y,v}) \cap f^{-1}([\![\, \varphi_2 \,]\!]_{Y,v}) \\ &= [\![\, \varphi_1 \,]\!]_{X, f^{-1} \circ v} \cap [\![\, \varphi_2 \,]\!]_{X, f^{-1} \circ v} \end{aligned}$$

Inductive case 2: ($\varphi = \neg\psi$)

$$f^{-1}(Y \setminus [\![\psi]\!]_{Y,v}) = X \setminus f^{-1}([\![\psi]\!]_{Y,v}) = X \setminus [\![\psi]\!]_{X,f^{-1}\circ v}$$

Inductive case 3: ($\varphi = \heartsuit(\psi)$)

$$\begin{aligned} f^{-1}([\![\heartsuit(\psi)]\!]_{Y,v}) &= f^{-1} \circ \delta^{-1} \circ [\![\heartsuit]\!]_Y([\![\psi]\!]_{Y,v}) \\ &= \gamma^{-1} \circ [\![\heartsuit]\!]_X(f^{-1}([\![\psi]\!]_{Y,v})) \quad \text{by lemma 3.5.2} \\ &= \gamma^{-1} \circ [\![\heartsuit]\!]_X([\![\psi]\!]_{X,f^{-1}\circ v}) \\ &= [\![\heartsuit(\psi)]\!]_{X,f^{-1}\circ v} \end{aligned}$$

□

3.6 Proof Rules

In this section we present the framework for specifying the proof rules of coalgebraic modal logics in the category of nominal sets. Notably, the framework recognises that all coalgebraic modal logics in **Nom** are propositional logic with the addition of modalities. The propositional reasoning is distinguished from the modal reasoning in the sense that it is local, while the modal reasoning is transitional.

Definition 3.6.1. A rank 1 modal formula is a formula $\phi \in \mathcal{F}(\Lambda)$ such that $\phi = \heartsuit\psi$ for some modality $\heartsuit$ and some propositional formula ψ.

Definition 3.6.2 (One-step rule)**.** A one step rule over a nominal set of modalities Λ is a pair (ψ, ψ'), written as ψ/ψ', where $\psi \in \mathsf{Prop}(V)$ and $\psi' \in \mathsf{Cl}(\Lambda(V))$. $\mathsf{Prop}(V)$ is the set of disjunctive clauses over propositional variables in V and $\mathsf{Cl}(\Lambda(V))$ is the set of disjunctive clauses over rank 1 modal formulae.

Definition 3.6.3 (R-derivable formulae)**.** Suppose R is a nominal set of one-step rules over the modal signature Λ. The set of R-derivable formulae is the smallest set which

- contains $\varphi\sigma$ whenever φ is a propositional tautology and $\sigma : V \to \mathcal{F}(\Lambda)$ is a substitution;
- is closed under modus ponens, i.e. it contains ψ whenever it contains $\varphi \to \psi$ and φ; and
- contains $\psi'\sigma$ whenever it contains $\psi\sigma$, $\psi/\psi' \in \mathsf{R}$, and $\sigma : V \to \mathcal{F}(\Lambda)$ is a substitution.

We write $\mathsf{R} \vdash \varphi$ in the case that φ is an R-derivable formula. Note that this set is defined inductively, as membership of any formula can be reduced to it being a propositional tautology in the base case, or to the presence of other formula in the inductive case, with minimality ensuring there are no closed loops of inference in the inductive case. Since R is nominal, and the above construction of the set of R-derivable formulae

is equivariant, it follows that this set of R-derivable formulae is a nominal subset of $\mathcal{F}(\Lambda)$ (i.e. closed under permutation).

Example 3.6.4. We recall the example of Nominal Hennessy Milner Logic. The set R contains the rules

$$\frac{\bigwedge_{i=1}^{n} p_i \to q}{\bigwedge_{i=1}^{n} [a]p_i \to [a]q}$$

for all $a \in \mathbb{A}$. This R is a nominal set with the obvious permutation action.

Chapter 4

Soundness and Completeness

In this chapter we address the ways in which the coalgebraic framework simplifies the tasks of proving the soundness and completeness of modal logics. Specifically, we present the one step properties and the results that flow from them, including an orbit-finite model property.

4.1 Soundness

The problem of showing the soundness of a coalgebraic modal logic is simplified in this section by reducing it to proving a property called *one step soundness*. This property relates the modal rules of R to the one step semantics of the functor for the logic.

4.1.1 One Step Soundness

Definition 4.1.1 (One Step Soundness)**.** A nominal rule set R is *one-step sound* if, for every rule $\phi/\psi \in R$ and every nominal set X and valuation $v : V \to \mathcal{P}_{fs}(X)$, we have $TX, v \vDash \psi$ whenever $X, v \vDash \phi$.

Theorem 4.1.2. *Suppose R is one-step sound. Then* $\mathsf{Coalg}(T) \vDash \phi$ *whenever* $R \vdash \phi$*, for all* $\phi \in \mathcal{F}(\Lambda)$.

Proof. Pick $(X, \gamma) \in \mathsf{Coalg}(T)$ and a valuation $v : V \to \mathcal{P}_{fs}(X)$. We need to show that $X, v \vDash \phi$. We use induction on the definition of R-derivability.

The base case is where ϕ is a propositional tautology with arbitrary formulae substituted in for the propositions by $\sigma : V \to \mathcal{F}(\Lambda)$. By definition 3.2.2, if ψ is a propositional tautology then $[\![\psi]\!]_{X,v} = X$ for all valuations v. To show that $[\![\psi\sigma]\!]_{X,v} = X$ we pick $\hat{v}(p) = [\![\sigma(p)]\!]_{X,v}$, and then $[\![\psi\sigma]\!]_{X,v} = [\![\psi]\!]_{X,\hat{v}} = X$. Note here that $\hat{v}$ is an acceptable valuation, as it is finitely supported and maps each propositional variable to a finitely supported subset of X. Therefore $X, v \vDash \phi$.

In the modus ponens case we have that ϕ is the consequent of an implication such that the implication and the antecedent are interpreted as X (by induction). By definition 3.2.2 (Specifically the interpretation of implication) we have that the consequent must also be interpreted as X.

In the rule application case we have that $\phi = \psi'\sigma$, $\psi/\psi' \in \mathsf{R}$, and $\mathsf{R} \vdash \psi\sigma$. By induction we can assume that $\psi\sigma$ is valid - in other words $X, v \vDash \psi\sigma$. We then know that $X, \hat{v} \vDash \psi$ and by one step soundness we know that $TX, \hat{v} \vDash \psi'$, so $TX, v \vDash \psi'\sigma$ and thus $X, v \vDash \psi'\sigma$ (the preimage of TX under any map from X is X).

This concludes the proof. □

4.1.2 Example: Nominal Hennessy Milner Logic

Recall the rule set R for Nominal Hennessy Milner Logic is

$$\frac{\bigwedge_{i=1}^{n} p_i \to q}{\bigwedge_{i=1}^{n} [a]p_i \to [a]q}$$

for all $n \in \mathbb{N}$ and all $a \in \mathbb{A}$. Recall also that this R is a nominal set with the obvious name permutation action.

Proposition 4.1.3. R *as given above is one step sound with respect to the semantics of Nominal Hennessy Milner Logic given previously*

Proof. We must show that if $X, v \vDash \bigwedge_{i=1}^{n} p_i \to q$ then $TX, v \vDash \bigwedge_{i=1}^{n} [a]p_i \to [a]q$.
So, we know that

$$\begin{aligned} X &= [\![\bigwedge_{i=1}^{n} p_i \to q]\!]_{X,v} \\ &= [\![\bigvee_{i=1}^{n} \neg p_i \vee q]\!]_{X,v} \\ &= \bigcup_{i=1}^{n} [\![\neg p_i]\!]_{X,v} \cup [\![q]\!]_{X,v} \\ &= \bigcup_{i=1}^{n} (X \setminus [\![p_i]\!]_{X,v}) \cup [\![q]\!]_{X,v} \end{aligned}$$

In other words for any finitely supported subset $A \subseteq X$, we have

$$A \cap \bigcup_{i=1}^{n} (X \setminus [\![p_i]\!]_{X,v}) \neq \varnothing \text{ OR } A \subseteq [\![q]\!]_{X,v}$$

And we must show that

$$[\![\bigwedge_{i=1}^{n} [a]p_i \to [a]q]\!]_{TX,v} \left(= \bigcup_{i=1}^{n} (TX \setminus [\![[a]p_i]\!]_{TX,v}) \cup [\![[a]q]\!]_{TX,v} \right) = TX$$

We know from the semantics that

$$\begin{aligned} [\![[a]p_i]\!]_{TX,v} &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \subseteq [\![p_i]\!]_{X,v}\} \\ [\![[a]q]\!]_{TX,v} &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \subseteq [\![q]\!]_{X,v}\} \end{aligned}$$

So

$$\begin{aligned}\bigcup_{i=1}^{n}(TX \setminus [\![\ [a]p_i\ ]\!]_{TX,v}) &= \bigcup_{i=1}^{n}\{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \not\subseteq [\![\ p_i\ ]\!]_{X,v}\}\\ &= \bigcup_{i=1}^{n}\{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \cap (X \setminus [\![\ p_i\ ]\!]_{X,v}) \neq \varnothing\}\\ &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid \exists i\,.\, g(a) \cap (X \setminus [\![\ p_i\ ]\!]_{X,v}) \neq \varnothing\}\\ &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \cap \bigcup_{i=1}^{n}(X \setminus [\![\ p_i\ ]\!]_{X,v}) \neq \varnothing\}\end{aligned}$$

Then

$$\begin{aligned}&[\![\ \bigwedge_{i=1}^{n}[a]p_i \to [a]q\ ]\!]_{TX,v}\\ &= \bigcup_{i=1}^{n}(TX \setminus [\![\ [a]p_i\ ]\!]_{TX,v}) \cup [\![\ [a]q\ ]\!]_{TX,v}\\ &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \cap \bigcup_{i=1}^{n}(X \setminus [\![\ p_i\ ]\!]_{X,v}) \neq \varnothing\} \cup [\![\ [a]q\ ]\!]_{TX,v}\\ &= \{g \in \mathcal{P}_{fs}(X)^{\mathbb{A}} \mid g(a) \cap \bigcup_{i=1}^{n}(X \setminus [\![\ p_i\ ]\!]_{X,v}) \neq \varnothing \text{ OR } g(a) \subseteq [\![\ q\ ]\!]_{X,v}\}\\ &= TX\end{aligned}$$

□

4.2 Completeness

In this section we discuss notions surrounding completeness. In particular, we discuss a property called *one step completeness* and a property called *the orbit-finite model property*. We also show that the rule set for Hennessy Milner Logic is One Step Complete with respect to the nominal sets semantics.

4.2.1 One Step Completeness

This section we define one step completeness. One step completeness is a useful notion to the generic proof theory of coalgebraic modal logic in the category of sets and functions. Essentially, in that setting, it implies completeness and is far less laborious to establish for a given logic. In fact it goes further than that to establish a finite model property for the logic.

Definition 4.2.1 (Propositional Consequence)**.** If $\Psi \subseteq \mathcal{F}(\Lambda)$ is a set of formulae and $\phi \in \mathcal{F}(\Lambda)$, we say that ϕ is a *propositional consequence* of Ψ, written $\Psi \vdash_{\mathsf{PL}} \phi$ if there

are $\phi_1, \ldots \phi_n \in \Psi$ such that $\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi$ is a substitution instance of a propositional tautology (note that substitutions can replace a variable with any formula, including modal ones).

Definition 4.2.2 (One Step Completeness)**.** A rule set R is *one-step complete*, if for every nominal set X, valuation $\sigma : V \to \mathcal{P}_{fs}(X)$ and clause $\chi \in \mathsf{Cl}(\Lambda(V))$ (see definition 3.6.2) we have

$$\{\psi\tau \mid X, \sigma \vDash \phi\tau,\ \tau : V \to \mathsf{Prop}(V),\ \phi/\psi \in \mathsf{R}\} \vdash_{\mathsf{PL}} \chi$$

whenever $TX, \sigma \vDash \chi$.

In the case of coalgebraic modal logic in the category of nominal sets, there are some issues that still require resolution. We only provide a partial solution and some suggestions in the direction of a generic result.

4.2.2 Orbit-Finite Model Property

In this section the goal is to show that one step completeness implies completeness via an orbit-finite model property. This will make life easier, as checking one step completeness for a rule set is much simpler than doing a canonical model construction. We proceed now by attempting a generic canonical model construction. What results is the shape of a possible proof, however we have left some conjectures open along the way.

We are given a functor T, a formula $\phi \in \mathcal{F}(\Lambda)$, and a rule set R (where R is one step complete for T). We want to show that if $\mathrm{Coalg}(T) \vDash \phi$ then $\mathsf{R} \vdash \phi$. That is, if a formula is semantically valid, it is provable with R. Recall that we know from the soundness result that the converse holds for one step sound rule systems.

In order to show this completeness result we want to, for all $\phi \in \mathcal{F}(\Lambda)$, construct an intermediate statement A_ϕ such that

$$\mathrm{Coalg}(T) \vDash \phi \quad \Longrightarrow \quad A_\phi \quad \Longrightarrow \quad \mathsf{R} \vdash \phi$$

Essentially we want A_ϕ to be the statement $C, \sigma \vDash \phi$, for some canonical C and σ which depend on ϕ. The first implication is trivial since $\mathrm{Coalg}(T) \vDash \phi$ quantifies over all models, so, regardless of which C, σ we choose the implication holds.

Thus for a given $\phi \in \mathcal{F}(\Lambda)$ we wish to construct a model such that if $C, \sigma \vDash \phi$ then $\mathsf{R} \vdash \phi$. We obviously need to use R to define the model. We first need to introduce a few more concepts.

Definition 4.2.3.

$$\sim\phi = \begin{cases} \neg\phi & \text{if } \phi \neq \neg\psi \text{ for any } \psi \\ \psi & \text{if } \phi = \neg\psi \text{ for some } \psi \end{cases}$$

Definition 4.2.4 (Closed)**.** A set of formulae $\Sigma \subseteq \mathcal{F}(\Lambda)$ is *closed* if

- If $\psi \in \Sigma$ then $\mathsf{Sf}(\psi) \subseteq \Sigma$ (where $\mathsf{Sf}(\phi)$ is the set of all subformulae of ϕ);

- If $\psi \in \Sigma$ then $\sim\psi \in \Sigma$; and
- If $\psi \in \Sigma$ then $\pi \cdot \psi \in \Sigma$ for all $\pi \in \mathbb{P}_{\mathbb{A}}$

Definition 4.2.5 (Consistent)**.** A set of formulae $\Sigma \subseteq \mathcal{F}(\Lambda)$ is *inconsistent* if there are $\phi_1, ..., \phi_n \in \Sigma$ such that $R \vdash \phi_1 \wedge ... \wedge \phi_n \to \bot$. Sets of formulae are *consistent* if they are not inconsistent. Note that nontrivial closed sets are inconsistent.

Definition 4.2.6 (Consistently Σ-Maximal)**.** A subset M of a closed, orbit-finite set $\Sigma \subseteq \mathcal{F}(\Lambda)$ is *consistently Σ-maximal* if it is maximal (with respect to the subset inclusion) among the consistent subsets of Σ. In other words, $M \subseteq \Sigma$ is consistent but $M \cup \{\phi\}$ is inconsistent whenever $\phi \in \Sigma \setminus M$.

Lemma 4.2.7. *Note that if M is consistently Σ-maximal and $\psi \in \Sigma$, then*

$$\psi \in M \ \text{ or } \ \sim\psi \in M$$

Proof. Assume (for a contradiction) that there is a $\psi \in \Sigma$ such that neither ψ nor $\sim\psi$ are in M.

Since $\psi \in \Sigma \setminus M$, we know that $M \cup \{\psi\}$ is inconsistent. That is, there exists $\phi_1, ..., \phi_n \in M$ such that $\mathsf{R} \vdash \phi_1 \wedge ... \wedge \phi_n \wedge \psi \to \bot$. Similarly, there exists $\varphi_1, ..., \varphi_n \in M$ such that $\mathsf{R} \vdash \varphi_1 \wedge ... \wedge \varphi_n \wedge \ \sim\psi \to \bot$ (note that Σ is closed, so $\sim\psi \in \Sigma$). Since the set of R-derivable formulae is closed under modus ponens and contains all substitution instances of propositional tautologies, $\mathsf{R} \vdash \phi_1 \wedge ... \wedge \phi_n \wedge \varphi_1 \wedge ... \wedge \varphi_n \to \bot$, which means that M is inconsistent, which is a contradiction. □

Definition 4.2.8 (S_Σ)**.** Given a closed, orbit-finite $\Sigma \subseteq \mathcal{F}(\Lambda)$ (which is nominal because it is closed) we define a set S_Σ as:

$$S_\Sigma = \{M \in \mathcal{P}_{fs}(\Sigma) \mid M \text{ consistently } \Sigma\text{-maximal}\}$$

Lemma 4.2.9. *Suppose $\Sigma \subseteq \mathcal{F}(\Lambda)$ and $S \in \mathcal{P}_{fs}(\Sigma)$ is consistent. Then there exists a consistently Σ-maximal set M with $S \subseteq M$. In other words, there is a consistent maximisation of any consistent $S \in \mathcal{P}_{fs}(\Sigma)$.*

Proof. With sets this is a standard result, but with nominal sets, we must show that any finitely supported subset of Σ can be maximised in a way that doesn't result in an infinite support.

This can be achieved by treating the all the names outside the support in a uniform way, so that if one adds a formula to S, it automatically adds every permutation of that formula involving names outside the support of S. □

We now return to the matter of defining a model for which if $C, \sigma \vDash \phi$ then $\mathsf{R} \vdash \phi$. We take Σ to be the closure of the set of subformulae of ϕ (including ϕ itself).

We intend to use S_Σ as the canonical state space, so we must show that it is a nominal set.

Lemma 4.2.10. *S_Σ is a nominal set.*

Proof. Firstly, in order to show that it is a nominal set, we must recall the permutation action for sets

$$\pi \cdot M = \{\pi \cdot \phi \mid \phi \in M\}$$

Then we must show that $\pi \cdot M$ is finitely supported, consistent and maximal among the finitely supported consistent subsets of Σ. This is trivial, since the set of R-derivable formulae is closed under permutation. □

We aim to construct a canonical model with a coalgebra $\gamma : S_\Sigma \to T(S_\Sigma)$ and a valuation $\sigma : V \to \mathcal{P}_{fs}(S_\Sigma)$. Note that this will not result in a finite model property, but instead an orbit-finite model property, which is the appropriate notion of finiteness in **Nom** (see section 2.2).

We leave the definitions of the coalgebra γ and the valuation σ to the appropriate places in the proofs that follow.

We want to prove a canonical model theorem, but in order to do this we must establish a truth lemma of the form:

For $\varphi \in \Sigma$ and $M \in S_\Sigma$ we have

$$S_\Sigma, M, \sigma \vDash \varphi \iff \varphi \in M$$

We need to prove the lemma by induction on the structure of φ, but in order to do this, we must establish the existence of a coalgebra which satisfies the lemma. This existence lemma is the difficult part. We do not give a full proof, though we do give an account of how it might be proved, assuming an (admittedly strong) lemma.

Conjecture 4.2.11. *Given $M \in \mathcal{P}_{fs}(S_\Sigma)$, if*

$$TS_\Sigma = \bigcup_{\heartsuit\phi \in (\Sigma \setminus M)} [\![\heartsuit p_\phi]\!]_{TS_\Sigma,\sigma} \cup \bigcup_{\heartsuit\phi \in M} [\![\neg\heartsuit p_\phi]\!]_{TS_\Sigma,\sigma}$$

then there exist finite $A \subset \Sigma \setminus M$ and $B \subset M$ such that

$$TS_\Sigma = \bigcup_{\heartsuit\phi \in A} [\![\heartsuit p_\phi]\!]_{TS_\Sigma,\sigma} \cup \bigcup_{\heartsuit\phi \in B} [\![\neg\heartsuit p_\phi]\!]_{TS_\Sigma,\sigma}$$

Now, in order to prove the existence lemma, we need to first show the following:

Lemma 4.2.12. *Take a closed, finite $\Sigma \subseteq \mathcal{F}(\Lambda)$ and*

$$v : V \to \mathcal{P}_{fs}(S_\Sigma) \quad \textit{and} \quad \rho : V \to \Sigma$$

such that $v(p) = \{M \in S_\Sigma \mid \rho(p) \in M\}$. Then we have

$$\mathsf{R} \vdash \phi\rho \iff S_\Sigma, v \vDash \phi$$

for all $\phi \in \mathsf{Prop}(V)$.

Proof. We first show that the following claim implies the lemma: $\phi\rho$ is R-consistent iff ϕ is satisfiable over v. Explicitly:

$$\mathsf{R} \nvdash \sim\phi\rho \iff [\![\phi]\!]_{S_\Sigma,v} \neq \varnothing \tag{4.1}$$

First, if $[\![\phi]\!]_{S_\Sigma,v} \neq \varnothing$ then $[\![\sim\phi]\!]_{S_\Sigma,v} \neq S_\Sigma$, and so $S_\Sigma, v \nvDash \sim\phi$. By contraposition, if $S_\Sigma, v \vDash \sim\phi$, then $[\![\phi]\!]_{S_\Sigma,v} = \varnothing$, so by condition 4.1 we have $\mathsf{R} \vdash \sim\phi\rho$. Since ϕ was arbitrary, given condition 4.1, if $S_\Sigma, v \vDash \phi$ then $\mathsf{R} \vdash \phi\rho$.

Now, if $\mathsf{R} \vdash \phi\rho$, then $\mathsf{R} \vdash \sim\sim\phi\rho$, so by condition 4.1: $[\![\sim\phi]\!]_{S_\Sigma,v} = \varnothing$. Therefore: $S_\Sigma, v \vDash \phi$.

We now show condition 4.1 where ϕ is a conjunctive clause, namely:

$$\phi = p_{i_1} \wedge \ldots \wedge p_{i_n} \wedge \overline{p_{j_1}} \wedge \ldots \wedge \overline{p_{j_m}}$$

We construct a set

$$\Psi = \{\rho(p_{i_k}) \mid 1 \leq k \leq n\} \cup \{ \sim\rho(p_{j_k}) \mid 1 \leq k \leq m\} \subseteq \Sigma$$

Then $\phi\rho$ is R-consistent iff Ψ is R-consistent iff there exists an A such that $\Psi \subseteq A \in S_\Sigma \subseteq S_\Sigma$. If there is such an A then $\phi\rho \in A$, and since $\phi\rho \in A$ iff $A \in [\![\phi]\!]_{S_\Sigma,v}$, we have that ϕ is satisfiable under v. $\phi\rho \in A$ iff $A \in [\![\phi]\!]_{S_\Sigma,v}$ can be established by induction on the structure of ϕ:

$\phi = p$

$$\phi\rho \in A \iff \rho(p) \in A \iff A \in v(p) \iff A \in [\![\phi]\!]_{S_\Sigma,v}$$

$\phi = \phi_1 \wedge \phi_2$

$$\begin{aligned} &\phi\rho \in A \iff \phi_1\rho \wedge \phi_2\rho \in A \iff \phi_1\rho \in A \text{ and } \phi_2\rho \in A \\ &\iff A \in [\![\phi_1]\!]_{S_\Sigma,v} \text{ and } A \in [\![\phi_2]\!]_{S_\Sigma,v} \iff A \in [\![\phi]\!]_{S_\Sigma,v} \end{aligned}$$

$\phi = \neg\psi$

$$\begin{aligned} \phi\rho \in A \iff \neg\psi\rho \in A \iff \psi\rho \notin A \iff A \notin [\![\psi]\!]_{S'_\Sigma,v} \\ \iff A \in S_\Sigma \setminus [\![\psi]\!]_{S_\Sigma,v} \iff A \in [\![\phi]\!]_{S_\Sigma,v} \end{aligned}$$

Conversely, if ϕ is satisfiable, then there is a state M with $\phi\rho \in M$, and this state can be taken to be $M \in S_\Sigma$, since ρ maps into Σ. Thus $\Psi \subseteq M \in S_\Sigma$ and so $\phi\rho$ is R-consistent.

Now, for $\phi = \phi_1 \vee \phi_2$ by induction we have

$$\begin{aligned} \mathsf{R} \vdash \sim\phi\rho \iff \mathsf{R} \vdash \sim\phi_1\rho \wedge \ \sim\phi_2\rho \iff \mathsf{R} \vdash \sim\phi_1\rho \text{ and } \mathsf{R} \vdash \sim\phi_2\rho \\ \iff [\![\phi_1]\!]_{S_\Sigma,v} = \varnothing \text{ and } [\![\phi_2]\!]_{S_\Sigma,v} = \varnothing \iff [\![\phi]\!]_{S_\Sigma,v} = \varnothing \end{aligned}$$

And since $\phi \in \mathsf{Prop}(V)$ can be put in disjunctive normal form, this concludes the proof. □

Lemma 4.2.13. *Take a closed, finite $\Sigma \subseteq \mathcal{F}(\Lambda)$, then, for all $M \in S_\Sigma$ there exists $t \in T(S_\Sigma)$ such that for all $\heartsuit(\phi) \in \Sigma$*

$$t \in [\![\heartsuit]\!]_{S_\Sigma}(\Sigma_\phi) \iff \heartsuit(\phi) \in M$$

Proof given conjecture 4.2.11. We will prove the above by contradiction. Suppose that there exists $M \in S_\Sigma$, such that for all $t \in T(S_\Sigma)$, there exists a $\heartsuit(\phi) \in \Sigma$ such that either

$$t \in [\![\heartsuit]\!]_{S_\Sigma}(\Sigma_\phi) \qquad \text{and} \qquad \heartsuit(\phi) \notin M$$

Or

$$t \notin [\![\heartsuit]\!]_{S'_\Sigma}(\Sigma_\phi) \qquad \text{and} \qquad \heartsuit(\phi) \in M$$

In the first case,

$$t \in \bigcup_{\heartsuit(\phi) \in \Sigma \setminus M} [\![\heartsuit(p_\phi)]\!]_{TS_\Sigma, \sigma}$$

under the substitution $\sigma(p_\phi) = \Sigma_\phi$. In the second case,

$$t \in \bigcup_{\heartsuit(\phi) \in M} [\![\neg\heartsuit(p_\phi)]\!]_{TS_\Sigma, \sigma}$$

under the same substitution $\sigma(p_\phi) = \Sigma_\phi$. In any case, we have that

$$TS_\Sigma = \bigcup_{\heartsuit(\phi) \in \Sigma \setminus M} [\![\heartsuit(p_\phi)]\!]_{TS_\Sigma, \sigma} \cup \bigcup_{\heartsuit(\phi) \in M} [\![\neg\heartsuit(p_\phi)]\!]_{TS_\Sigma, \sigma}$$

under σ.

Using conjecture 4.2.11, we can get finite $A \subset \Sigma \setminus M$ and $B \subset M$ such that

$$TS_\Sigma = \bigcup_{\heartsuit\phi \in A} [\![\heartsuit p_\phi]\!]_{TS_\Sigma, \sigma} \cup \bigcup_{\heartsuit\phi \in B} [\![\neg\heartsuit p_\phi]\!]_{TS_\Sigma, \sigma}$$

So, given

$$\chi = \bigvee_{\heartsuit(\phi) \in A} \heartsuit(p_\phi) \cup \bigvee_{\heartsuit(\phi) \in B} \neg\heartsuit(p_\phi)$$

we have that $T(S_\Sigma), \sigma \vDash \chi$. Since R is one-step complete, we have that:

$$\{\psi\tau \mid \phi/\psi \in \mathsf{R},\ \tau : V \to \mathsf{Prop}(V),\ S_\Sigma, \sigma \vDash \phi\tau\} \vdash_{\mathsf{PL}} \chi$$

With $\rho : V \to \mathcal{F}(\Lambda)$ a substitution, we can take the substitution instance of a propositional tautology given by the $\vdash_{\mathsf{PL}}$ statement above and substitute again by ρ to get another substitution instance of a propositional tautology. Thus:

$$\{\psi\tau\rho \mid \phi/\psi \in \mathsf{R},\ \tau : V \to \mathsf{Prop}(V),\ S_\Sigma, \sigma \vDash \phi\tau\} \vdash_{\mathsf{PL}} \chi\rho \tag{4.2}$$

Now, if $\mathsf{R} \vdash \phi\tau\rho$, and $\phi/\psi \in \mathsf{R}$ then $\mathsf{R} \vdash \psi\tau\rho$, by definition. If all the elements of the set in 4.2 are R-derivable, then so to is $\chi\rho$, since the set of R-derivable formulae

contain all substitution instances of propositional tautologies (in particular the substitution instance of a propositional tautology given by (4.2)), and the set of R-derivable formulae is closed under modus ponens. Thus in order to show $\mathsf{R} \vdash \chi\rho$, we can show $\mathsf{R} \vdash \phi\tau\rho$ whenever $S_\Sigma, \sigma \vDash \phi\tau$. We do this for $\rho : p_\phi \mapsto \phi$ (which comes under Lemma 4.2.12), and thus show that

$$\begin{aligned}\chi\rho &= \bigvee_{\heartsuit(\phi)\in A} \heartsuit(\phi) \vee \bigvee_{\heartsuit(\phi)\in B} \neg\heartsuit(\phi) \\ &\overset{\mathsf{PL}}{=} \bigwedge_{\heartsuit(\phi)\in B} \heartsuit(\phi) \to \bigvee_{\heartsuit(\phi)\in A} \heartsuit(\phi)\end{aligned}$$

is R-derivable. Since $B \subset M$ and since M is maximal among the consistent subsets of Σ, M must contain an element of A. However, $A \subset \Sigma \setminus M$, which means M is not maximal, which is a contradiction. □

Then we have this:

Corollary 4.2.14 (Existence Lemma)**.** *Suppose Σ is closed and finite. Then there exists a T-coalgebra $\gamma : S_\Sigma \to T(S_\Sigma)$ such that*

$$\gamma(M) \in [\![\heartsuit]\!]_{S_\Sigma}(\Sigma_\phi) \iff \heartsuit(\phi) \in M$$

for all $\heartsuit(\phi) \in \Sigma$.

Proof. Choose $\gamma(M) = t$ satisfying Lemma 4.2.13. This is equivariant because the property defining t is equivariant:

$$\begin{aligned}&\pi \cdot t \in [\![\heartsuit]\!]_{S_\Sigma}(\Sigma_\phi) \\ &\iff t \in \pi^{-1} \cdot [\![\heartsuit]\!]_{S_\Sigma}(\Sigma_\phi) \\ &\iff t \in [\![\pi^{-1} \cdot \heartsuit]\!]_{S_\Sigma}(\Sigma_{\pi^{-1}\cdot\phi}) \\ &\iff (\pi^{-1} \cdot \heartsuit)(\pi^{-1} \cdot \phi) \in M \\ &\iff \pi^{-1} \cdot \heartsuit(\phi) \in M \\ &\iff \heartsuit(\phi) \in \pi \cdot M\end{aligned}$$

□

Lemma 4.2.15 (Truth Lemma)**.** *For $\varphi \in \Sigma$ and $M \in S_\Sigma$ we have*

$$S_\Sigma, M, \sigma \vDash \varphi \iff \varphi \in M$$

Proof. We proceed by induction on the structure of φ. For the sake of brevity we use $M, \sigma \vDash \psi$ to mean $S_\Sigma, M, \sigma \vDash \varphi$. Recall that M is a state, and so it either validates formulae or it negates them.

In the base case, $\varphi = p$ so $M, \sigma \vDash p \iff M \in \sigma(p)$, and therefore, we define $\sigma(q) = \{M \in S_\Sigma \mid q \in M\}$. So $M, \sigma \vDash p \iff p \in M$.

In the case that $\varphi = \psi_1 \wedge \psi_2$, we know the following

$$\begin{aligned} M, \sigma \vDash \psi_1 \wedge \psi_2 &\iff M, \sigma \vDash \psi_1 \text{ and } M, \sigma \vDash \psi_2 \\ &\iff \psi_1 \in M \text{ and } \psi_2 \in M \\ &\iff \psi_1 \wedge \psi_2 \in M \end{aligned}$$

In the case that $\varphi = \neg\psi$, we know the following

$$\begin{aligned} M, \sigma \vDash \neg\psi &\iff M, \sigma \nvDash \psi \\ &\iff \psi \notin M \\ &\iff \neg\psi \in M \quad \text{see lemma 4.2.7} \end{aligned}$$

In the case that $\varphi = \heartsuit\psi$, we know the following

$$\begin{aligned} M, \sigma \vDash \heartsuit\psi &\iff \gamma(M) \in [\![\heartsuit\psi]\!]_{TS_\Sigma, \sigma} \\ &\iff \gamma(M) \in [\![\heartsuit]\!]_{S_\Sigma}([\![\psi]\!]_{S_\Sigma, \sigma}) \end{aligned}$$

Our inductive assumption is that $M, \sigma \vDash \psi \iff \psi \in M$. So $[\![\psi]\!]_{S_\Sigma, \sigma} = \Sigma_\psi := \{M \in S_\Sigma \mid \psi \in M\}$.

Thus, in order to complete the proof, we must show that

$$\gamma(M) \in [\![\heartsuit]\!]_{S_\Sigma}(\Sigma_\psi) \iff \heartsuit\psi \in M$$

This is exactly Lemma 4.2.14. □

Theorem 4.2.16 (Canonical Model Theorem)**.** *Given that Σ is the smallest closed subset of $\mathcal{F}(\Lambda)$ containing ϕ, whenever $S_\Sigma, \sigma \vDash \phi$ then $\mathsf{R} \vdash \phi$*

Proof. We know by lemma 4.2.15 that $\phi \in M$ for all $M \in S_\Sigma$. Since all M are R-consistent none of them contain $\sim\phi$. Since $\sim\phi \in \Sigma$ (since Σ is closed), we know that $\{\sim\phi\}$ is R-inconsistent. Therefore, $\mathsf{R} \vdash (\sim\phi) \to \bot$, which means $\mathsf{R} \vdash \neg(\sim\phi)$, which means $\mathsf{R} \vdash \phi$. □

4.2.3 Example: Nominal Hennessy Milner Logic

Recall the rule set R for Nominal Hennessy Milner Logic:

$$\frac{\bigwedge_{i=1}^{n} p_i \to q}{\bigwedge_{i=1}^{n} [a]p_i \to [a]q}$$

Proposition 4.2.17. R *is one step complete for the given semantics*

Proof. We need to show that for all $\chi \in \mathsf{Cl}(\Lambda(V))$ if $TX, v \vDash \chi$ then $\Psi \vdash_{\mathsf{PL}} \chi$, where

$$\Psi = \left\{ \bigwedge_{i \in I} [a]p_i \to [a]q \;\middle|\; X, v \vDash \bigwedge_{i \in I} p_i \to q,\ I \text{ finite},\ q, p_i \in V\ a \in \mathbb{A} \right\}$$

Without loss of generality we can assume (where $A, B \subseteq \mathbb{A}$ are finite)

$$\chi = \bigwedge_{a\in A, i\in I} [a]p_{a_i} \to \bigvee_{b\in B, j\in J} [b]q_{b_j}$$

Then by assumption ($TX, v \vDash \chi$)

$$[\![\bigwedge_{a\in A, i\in I} [a]p_{a_i}]\!]_{TX,v} \subseteq [\![\bigvee_{b\in B, j\in J} [b]q_{b_j}]\!]_{TX,v}$$

Now to simplify the notation, let $A_{a_i} = v(p_{a_i})$ and $B_{b_j} = v(q_{b_j})$, so

$$\begin{aligned} [\![\bigwedge_{a\in A, i\in I} [a]p_{a_i}]\!]_{TX,v} &= \bigcap_{a\in A, i\in I} [\![[a]p_{a_i}]\!]_{TX,v} \\ &= \bigcap_{a\in A, i\in I} [\![[a]]\!]_X(A_{a_i}) \end{aligned}$$

and

$$[\![\bigvee_{b\in B, j\in J} [b]q_{b_j}]\!]_{TX,v} = \bigcup_{b\in B, j\in J} [\![[b]]\!]_X(B_{b_j})$$

First we establish a simple equivalence:

$$C \in \bigcap_{i\in I} \mathcal{P}_{fs}(A_i) \iff \forall i \,.\, C \overset{fs}{\subseteq} A_i \iff C \overset{fs}{\subseteq} \bigcap_{i\in I} A_i \iff C \in \mathcal{P}_{fs}(\bigcap_{i\in I} A_i)$$

Then, we return to examine $[\![\bigwedge_{a\in A, i\in I} [a]p_{a_i}]\!]_{TX,v}$

$$\begin{aligned} h \in \bigcap_{a\in A, i\in I} [\![[a]]\!]_X(A_{a_i}) &\iff \forall a, i \,.\, h \in [\![[a]]\!]_X(A_{a_i}) \\ &\iff \forall a, i \,.\, h(a) \in \mathcal{P}_{fs}(A_{a_i}) \\ &\iff \forall a \,.\, h(a) \in \bigcap_{i\in I} \mathcal{P}_{fs}(A_{a_i}) \\ &\iff \forall a \,.\, h(a) \in \mathcal{P}_{fs}(\bigcap_{i\in I} A_{a_i}) \\ &\iff \forall a \,.\, h \in [\![[a]]\!]_X(\bigcap_{i\in I} A_{a_i}) \\ &\iff h \in \bigcap_{a\in A} [\![[a]]\!]_X(\bigcap_{i\in I} A_{a_i}) \end{aligned}$$

So $\bigcap_{a\in A, i\in I} [\![[a]]\!]_X(A_{a_i}) = \bigcap_{a\in A} [\![[a]]\!]_X(\bigcap_{i\in I} A_{a_i})$.

Now let $f : \mathbb{A} \to \mathcal{P}_{fs}(X)$ such that for all $a \in \mathbb{A}$ we have $f(a) = \bigcap_{i\in I} A_{a_i}$. Now we

know:

$$\forall a \in A \,.\, f \in [\![\, [a] \,]\!]_X(\bigcap_{i\in I} A_{a_i})$$
$$\therefore f \in \bigcap_{a\in A} [\![\, [a] \,]\!]_X(\bigcap_{i\in I} A_{a_i})$$
$$\therefore f \in \bigcup_{b\in B, j\in J} [\![\, [b] \,]\!]_X(B_{b_j})$$

Therefore, for some b and j we have

$$f(b) \subseteq B_{b_j}$$

In other words, if $b \in A$ then

$$\bigcap_{i\in I} A_{b_i} \subseteq B_{b_j}$$

Thus we know

$$X, v \vDash \bigwedge_{i\in I} p_{b_i} \to q_{b_j}$$

Therefore, for some $b \in A \cap B$ and $j \in J$:

$$\left(\bigwedge_{i\in I} [b]p_{b_i} \to [b]q_{b_j}\right) \in \Psi$$

and the following is a substitution instance of a propositional tautology:

$$\left(\bigwedge_{i\in I} [b]p_{b_i} \to [b]q_{b_j}\right) \to \chi$$

So $\Psi \vdash_{\mathsf{PL}} \chi$.

Now we want to be able to say that $b \in A$. Basically, we can extend χ by adding terms to the conjunctive part or the disjunctive part without altering validity. So we can assume that $A = B$ in χ itself without loss of generality.

This concludes the proof in the case that $J \neq \varnothing$ (note that R contains the necessary rule for the $I = \varnothing$ case).

If $J = \varnothing$ then $TX, v \nvDash \chi$, since in this case

$$\chi = \bigwedge_{a\in A, i\in I} [a]p_{a_i} \to \bot = \neg \bigwedge_{a\in A, i\in I} [a]p_{a_i}$$

and one can easily construct a T-coalgebra in which the successor of a particular state in X is $\mapsto \varnothing$, and then one can see that $\bigwedge_{a\in A, i\in I}[a]p_{a_i}$ would be true at that state, invalidating $\neg \bigwedge_{a\in A, i\in I}[a]p_{a_i}$. □

Corollary 4.2.18. *Thus far, Nominal Hennessy Milner Logic is equivalent to traditional Hennessy Milner Logic (assuming conjecture 4.2.11).*

Proof. They have the same rule sets and those rule sets are sound and complete for both versions of the semantics. □

In the freshness section, we will introduce the fresh quantifier into Nominal Hennessy Milner Logic, which will make it quite different from traditional Hennessy Milner Logic.

Chapter 5

Freshness

In this chapter we introduce the И quantifier into the framework. Recall that the И quantifier essentially models the binding of variables, which we call names in order to distinguish them from propositional variables. We add И to the generic syntax, provide the semantics of the quantifier and modify the notion of R-derivability. We will then lift the soundness and completeness results given in the previous chapter.

We call the lifted framework И-CML.

5.1 Semantics

In this section we present the new syntax, intuitive meaning and formal semantics of the И quantifier. We will also examine properties such as equivariance of interpretation and preservation of interpretations under coalgebra maps by adding the И case to the relevant induction proofs.

First, the new generic syntax is given by:

$$\mathcal{F}(\Lambda) \ni \varphi, \psi ::= p \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \heartsuit(\varphi) \mid И n \,.\, \varphi$$

The semantics of the И quantifier is given by

$$[\![\, И n.\phi \,]\!]_{X,v} = \left\{ x \in X \,\middle|\, \exists n' \#(x, \phi, v) \,.\, x \in [\![\, \phi[n'/n] \,]\!]_{X,v} \right\}$$

Where $\phi[n'/n]$ means ϕ with n' substituted for all instances of n.

We now consider the induction steps for the proofs of the equivariance property and the preservation of interpretations under coalgebra maps.

5.1.1 Equivariance of Interpretation

Recall that Theorem 3.4.1 was proven by induction on the structure of formulae. We now prove the И case of this induction, thus lifting the theorem to include formulae from the new syntax.

Proof. We want to show that $\pi \cdot [\![\, И n.\phi \,]\!]_{X,v} = [\![\, \pi \cdot И n.\phi \,]\!]_{X,\pi\cdot v}$. Firstly:

$$\pi \cdot [\![\, И n.\phi \,]\!]_{X,v} = \pi \cdot \{ x \in X \mid \exists n' \#(x, \phi, v) \,.\, x \in [\![\, \phi[n'/n] \,]\!]_{X,v} \}$$

For $x \in X$ if there is a fresh name $n'\#(x, \phi, v)$ such that $x \in [\![\phi[n'/n]]\!]$ then by the some/any theorem we can choose a fresh name with the same property, except that it is also fresh for π. The converse, in which we just forget about the fact that n' is fresh for π, is trivial. Thus:

$$\begin{aligned}\pi \cdot [\![И n.\phi]\!]_{X,v} &= \pi \cdot \{x \in X \mid \exists n'\#(x, \phi, v, \pi) \,.\, x \in [\![\phi[n'/n]]\!]_{X,v}\} \\ &= \{x \in X \mid \exists n'\#(x, \phi, v, \pi) \,.\, x \in \pi \cdot [\![\phi[n'/n]]\!]_{X,v}\}\end{aligned}$$

Now, by the induction hypothesis

$$\begin{aligned}\pi \cdot [\![И n.\phi]\!]_{X,v} &= \{x \in X \mid \exists n'\#(x, \phi, v, \pi) \,.\, x \in [\![\pi \cdot (\phi[n'/n])]\!]_{X,\pi\cdot v}\} \\ &= \{x \in X \mid \exists n'\#(x, \phi, v, \pi) \,.\, x \in [\![(\pi \cdot \phi)[n'/\pi(n)]]\!]_{X,\pi\cdot v}\}\end{aligned}$$

Again, by the some/any theorem we can remove the condition that $n'\#\pi$, and we have:

$$\begin{aligned}\pi \cdot [\![И n.\phi]\!]_{X,v} &= [\![И \pi(n).\pi \cdot \phi]\!]_{X,\pi\cdot v} \\ &= [\![\pi \cdot И n.\phi]\!]_{X,\pi\cdot v}\end{aligned}$$

□

5.1.2 Preservation of Interpretations Under Coalgebra Maps

Recall that Theorem 3.5.3 was proven by induction on the structure of formulae. We now prove the И case of this induction, thus lifting the theorem to include formulae from the new syntax.

Proof. We want to show that, given f is a coalgebra morphism from (C, γ) to (D, δ), we have $f^{-1}([\![И n.\psi]\!]_{D,v}) = [\![И n.\psi]\!]_{C,f^{-1}\circ v}$.

$$\begin{aligned}f^{-1}([\![И n.\psi]\!]_{D,v}) &= f^{-1}\left(\{y \in D \mid \exists n'\#(y, \psi, v) \,.\, y \in [\![\psi[n'/n]]\!]_{D,v}\}\right) \\ &= \{x \in C \mid \exists n'\#(f(x), \psi, v) \,.\, f(x) \in [\![\psi[n'/n]]\!]_{D,v}\}\end{aligned}$$

Now, since f is equivariant $supp(f(x)) \subseteq supp(x)$. Therefore, we must use the some/any theorem to choose n' fresh for the rest of $supp(x)$ also.

$$f^{-1}([\![И n.\psi]\!]_{D,v}) = \{x \in C \mid \exists n'\#(x, \psi, v) \,.\, x \in f^{-1}([\![\psi[n'/n]]\!]_{D,v})\}$$

Now, by the induction hypothesis:

$$\begin{aligned}f^{-1}([\![И n.\psi]\!]_{D,v}) &= \{x \in C \mid \exists n'\#(x, \psi, v) \,.\, x \in [\![\psi[n'/n]]\!]_{C,f^{-1}\circ v}\} \\ &= [\![И n.\psi]\!]_{C,f^{-1}\circ v}\end{aligned}$$

□

5.2 R-derivable formulae

In the following we will introduce the rules of the И quantifier. Instead of adding them to R itself, we make a change to the notion of R-derivable formulae, and reserve R for the one step modal rules.

We are proposing a treatment of nominal logic in line with the treatment of propositional logic of the existing framework. In one sense, we want to encompass all substitution instances of nominal tautologies. However, since И is a binder, we want only capture avoiding substitution instances. This requirement is implicit in the identification of α-equivalent formulae, but requires an explicit change to be made to the original definition of R-derivability. Furthermore, we need to define what a nominal tautology is.

First, consider the И-normalisation of a formula in the new syntax.

$$\begin{aligned} \neg И n.\phi &= И n.\neg\phi \\ И n.p &= p \\ И n.(\phi_1 \wedge \phi_2) &= (И n.\phi_1) \wedge (И n.\phi_2) \\ И n.(\heartsuit\phi) &= \heartsuit(И n.\phi) \quad \text{given } n\#\heartsuit \end{aligned}$$

The first of these is true from the some/any theorem of nominal logic. The second comes from the fact that propositions are equivariant. The third is also a result of the some/any theorem. The fourth comes from observing the fact that it is irrelevant if a fresh name is chosen before or after a transition if that transition does not depend on that fresh name.

These equivalences allow us to define the normal form of a formula. We apply the first equation in tandem with the familiar propositional formulae in order to put the formula into negation normal form. Taking the other three equations as left to right reduction steps and applying them to all subformulae until none of the rules can be applied again, we arrive at the И-normal form, in which any instance of И occurs directly before a modality, though possibly with multiplicity as in:

$$И n_1.И n_2....И n_i.\heartsuit\phi$$

Where $n_1, n_2, ..., n_i \in supp(\heartsuit)$. This is also assuming that the fourth equation is applied in a broader sense to allow $И n$ to jump past other $И m$ on its way to $\heartsuit$. Essentially, while we may have a finite sequence of И quantifiers in front of a modality, the order of that sequence is irrelevant. This is true even if some of the names used in that sequence match. In particular if $n_k = n_j$ for $k < j$ then the $И n_k$ can be removed as the later one binds all instances of n_j, and so the outer binder is essentially fresh for $\heartsuit\phi$.

Therefore, we can say that the formula given is $И A.\heartsuit\phi$ where $A = \{n_1, n_2, ..., n_i\}$ (in essence even though not in our traditional syntax). We also know that in the И-normal form we have $A \subseteq supp(\heartsuit)$ for every such occurrence of a modal operator.

In the case of Hennessy Milner Logic over $\mathbb{A}$, for example, each modality will either be proceeded directly by a single И quantifier, or none at all. When we add И

to the nominal sets version of Hennessy Milner Logic we call it И-HML, pronounced Fresh Hennessy Milner Logic.

We will identify formulae that have the same И-normal form. Another way of saying this is that we assume that all formulae are in И-normal form. We also identify all formulae in И-normal form which differ only by a reordering of И quantifiers, as discussed. One complication of this identification is that substitutions must be followed by renormalisation.

Returning to the definition of R-derivable formulae, recall that the set of R-derivable formulae is defined as the smallest subset of $\mathcal{F}(\Lambda)$ such that certain properties hold. We keep the existing definition, but add further properties. Firstly, notice that the meaning of substitution instances of propositional tautologies has now changed, because propositions can now be substituted with formulae involving И, and we have introduced an equivalence between formulae with the same И-normal form.

Definition 5.2.1. The set of R-derivable formulae is the smallest subset of $\mathcal{F}(\Lambda)$ such that the original properties hold, and additionally it

- contains $\phi = \psi\sigma$ whenever it contains the И normal form of ψ and σ is a capture avoiding substitution for ψ;
- contains $И n.\phi$ whenever it contains ϕ and we have $n \in \mathbb{A}$; and
- contains $\phi[n'/n]$ whenever it contains $И n.\phi$ and we have $n' \# \phi$ and $n \in \mathbb{A}$.

The third rule above is a part of the traditional nominal logic. However, it causes difficulty for us, since in a particular model and world we have to choose n' fresh for ϕ, the current world, and the valuation. We cannot guarantee the freshness of n' with respect to those things in a syntactic way, although introducing an explicit syntactic freshness operator might solve this problem.

Given the way in which it is currently phrased, the soundness of the third rule is difficult to verify. It might not even be sound. However, for theorems of the form $И n.\phi$, it seems apparent that ϕ must also be a derivable theorem. If this was the case then we could recover $\phi[n'/n]$ as a derivable theorem just because the set of R-derivable formulae are closed under permutation. It isn't entirely obvious to the author how to show that ϕ must be a derivable theorem whenever $И n.\phi$ is, though one might examine the origin of $И n.\phi$ by cases (substitution, modus ponens, the second rule of the above).

5.3 Soundness

Recall that Theorem 4.1.2 was proved by an induction on the definition of the set of R-derivable formulae. Now that we have added to the notion of R-derivable formulae we must treat the extra cases.

First, we show that the set of semantic theorems is closed under permutation.

Lemma 5.3.1. *If* $Coalg(T) \vDash \phi$ *then* $Coalg(T) \vDash \pi \cdot \phi$

Proof. Assume for a contradiction that $\mathrm{Coalg}(T) \vDash \phi$ and $\mathrm{Coalg}(T) \nvDash \pi \cdot \phi$.

So there is a model X, v and a state $x \in X$ such that $x \notin [\![\pi \cdot \phi]\!]_{X,v}$ and therefore, by equivariance, $x \notin \pi \cdot ([\![\phi]\!]_{X,\pi^{-1}\cdot v})$, which in turn means that $\pi^{-1} \cdot x \notin [\![\phi]\!]_{X,\pi^{-1}\cdot v}$.

So ϕ is not a theorem, which is a contradiction.

□

Note that in the following proof we omit the case of the third rule of definition 5.2.1.

Theorem 5.3.2 (Soundness of the fresh rules)**.** *If* $\mathsf{R} \vdash \phi$ *then* $Coalg(T) \vDash \phi$

Proof. We know $\mathsf{R} \vdash \phi$ and show $\mathrm{Coalg}(T) \vDash \phi$.

We proceed by induction on the definition of R-derivability. All cases except the nominal ones above have already been shown.

First, whenever $\phi = \psi\sigma$ is introduced as a result of the И-normal form of ψ being R-derivable, we know by induction that the И-normal form of ψ is semantically valid. As a result, ψ is semantically valid, as the И-reductions are semantically valid. Any substitution instance of a valid formula is valid. Thus $\mathrm{Coalg}(T) \vDash \phi$.

Second, whenever $И n.\psi$ is introduced as a result of $\mathsf{R} \vdash \psi$ we know $\mathrm{Coalg}(T) \vDash \psi$ by the inductive hypothesis.

Now, we need to show that if $\mathrm{Coalg}(T) \vDash \psi$ then $\mathrm{Coalg}(T) \vDash И n.\psi$ for $n \in \mathbb{A}$.

For a contradiction, assume that $\mathrm{Coalg}(T) \nvDash И n.\psi$. That is, there exists a model $Y, w : V \to \mathcal{P}_{fs}(Y)$ and state y such that $y \notin [\![И n.\psi]\!]_{Y,w}$.

$$
\begin{aligned}
& y \notin [\![И n.\psi]\!]_{Y,w} \\
\iff\ & y \notin \{y' \in Y \mid \exists n' \#(y', \psi) \,.\, y' \in [\![\psi[n'/n]]\!]_{Y,w}\} \\
\iff\ & \neg \exists n' \#(y, \psi) \,.\, y \in [\![\psi[n'/n]]\!]_{Y,w} \\
\iff\ & \exists n' \#(y, \psi) \,.\, y \notin [\![\psi[n'/n]]\!]_{Y,w} \\
\iff\ & \exists n' \#(y, \psi) \,.\, y \notin [\![(n'\ n) \cdot \psi]\!]_{Y,w}
\end{aligned}
$$

However, we know that $\mathrm{Coalg}(T) \vDash \psi$, which means that $\mathrm{Coalg}(T) \vDash (n'\ n) \cdot \psi$ for fresh n' (by lemma 5.3.1), and so $[\![(n'\ n) \cdot \psi]\!]_{Y,w} = Y$. This is a contradiction.

□

5.4 Completeness

In this section we address the issue of the completeness of the new definition of derivability with respect to the И quantifier. Here we assume that the modal case for completeness is given, and we address the И case.

Recall that the completeness proof had two parts. First there was the truth lemma, and then the proof from consistent maximality that $\mathsf{R} \vdash (\sim \phi) \to \bot$. Adding И to the syntax and proof rules, the latter follows with no additional problem. The truth lemma involves an induction over the structure of ϕ, and now that induction is incomplete, so we must treat the case of formulae of the form $И n.\phi$.

There are a number of other results that rely in part on reasoning about the support of formulae, either implicitly or explicitly. These also have to be reexamined in light of the И quantifier being a binder which alters the support of a formula. We do not reexamine those results here. We do not expect them to pose significant trouble. Meanwhile, the truth lemma is central, and problematic.

For the truth lemma we must show for $M \in \mathcal{P}_{fs}(S_\Sigma)$

$$M, \sigma \vDash И n.\phi \iff И n.\phi \in M$$

where $M, \sigma \vDash \psi$ stands for $S_\Sigma, M, \sigma \vDash \psi$.

The basic structure of the proof is as follows:

$$\begin{aligned} M, \sigma \vDash И n.\phi &\iff M, \sigma \vDash \phi[n'/n] \text{ for fresh } n' \\ &\iff \phi[n'/n] \in M \text{ for fresh } n' \\ &\iff И n.\phi \in M \end{aligned}$$

The first step essentially follows from the basic definitions of the semantics

$$\begin{aligned} M, \sigma \vDash И n.\phi &\iff \exists n' \# (M, \phi, \sigma) \,.\, M \in [\![\, \phi[n'/n] \,]\!]_{S_\Sigma, \sigma} \text{ for fresh } n' \\ &\iff M, \sigma \vDash \phi[n'/n] \text{ for fresh } n' \end{aligned}$$

The second step is the inductive step, so that presents no challenges.

The third and final step, however, must invoke the consistent maximality of M with respect to R-derivability.

The validity of the second and third R-derivability rule of definition 5.2.1 is relevant to this question, but we need something stronger. The R-derivability rules refer only to R-theorems and not to formulae which happen to be true in particular models. We can conclude from assuming the rules (including the third rule which is questionable) that

$$R \vdash \phi[n'/n] \text{ for fresh } n' \iff R \vdash И n.\phi$$

This does not allow us to conclude, however, that $\phi[n'/n]$ for a particular fresh n' is semantically true if and only if $И n.\phi$. For that we need n' to vary as we vary the model and which world we are looking at.

The consistent maximality of M goes some way towards a solution. For instance, it allows us the following, assuming $\phi \in \Sigma$

$$\mathsf{R} \vdash \phi \implies \phi \in M$$

and

$$\begin{aligned} \phi \in \Sigma \setminus M &\implies M \cup \{\phi\} \text{ inconsistent} \\ &\implies \exists \psi_1, ..., \psi_n \in M \,.\, \mathsf{R} \vdash \psi_1 \wedge ... \wedge \psi_n \wedge \phi \to \bot \end{aligned}$$

additionally, for $\phi \in \Sigma$

$$\phi \in M \iff \sim\phi \notin M$$

It isn't entirely obvious whether or not these facts are sufficient to establish the required equivalence. This problem is on top of the problem with the third rule of definition 5.2.1, and may require a fundamentally stronger rule framework which allows reasoning with assumptions to capture the behaviour of И in the context of individual models and at individual worlds.

All in all, we have something in the direction of a solid framework. The main problems are with completeness, which is to be expected, given that completeness is almost always the more complicated direction.

Chapter 6

Decision Procedures

In this chapter we will discuss decidability issues and decision procedures for coalgebraic modal logic in the category of nominal sets. We will present a general discussion of decidability and then a discussion of the potential application of global caching specifically to И-HML.

6.1 Decidability

In this section we discuss decidability issues. We show a result that is relevant to decidability, and present a concept which is important to decidability in the framework of coalgebraic modal logic in **Set** [Pattinson 2012].

We consider decidability in the framework without И, though we do not go far beyond conjecture.

The orbit-finite model property isn't enough to give us decidability of satisfiability via model checking. This is because the functor T may expand orbit-finite nominal sets to orbit-infinite nominal sets, and then we could not enumerate all possible coalgebras to check if they satisfy a formula. Instead we take the usual approach in coalgebraic modal logic and use sequent calculi.

6.1.1 Strict Completeness

Definition 6.1.1 (Strict Completeness)**.** A rule set R is called strictly one step complete when the following property holds:

For all nominal sets X, valuations $v : V \to \mathcal{P}_{fs}(X)$ and $\chi \in \mathsf{Cl}(\Lambda(V))$ (see definition 3.6.2), if $TX, v \vDash \chi$ then there is a rule $\phi/\psi \in \mathsf{R}$ and a variable substitution $\tau : V \to V$ such that

$$X, v \vDash \phi\tau \quad \& \quad \psi\tau \vdash_{\mathsf{PL}} \chi$$

Notice that the key difference from the definition of one step completeness is that Ψ now a singleton.

Lemma 6.1.2. *Let $\chi = \chi_1 \vee \chi_2$ with $\chi_1 \in \mathsf{Cl}(V)$ and $\chi_2 \in \mathsf{Cl}(\Lambda(\mathcal{F}(\Lambda)))$. Then*

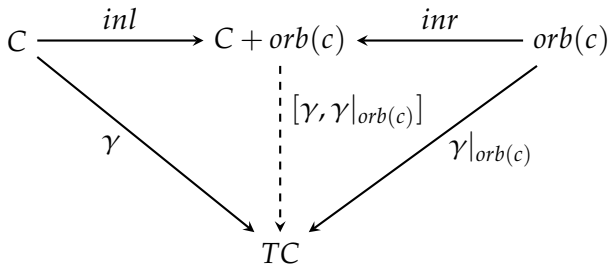
$$Coalg(T) \vDash \chi \iff Coalg(T) \vDash \chi_1 \; OR \; Coalg(T) \vDash \chi_2$$

Proof. ($\Longleftarrow$) is trivial.

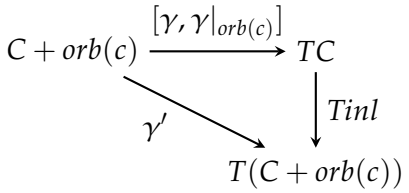
For ($\Longrightarrow$) we prove the contraposition. That is, we are assuming $\mathrm{Coalg}(T) \nvDash \chi_1$ and $\mathrm{Coalg}(T) \nvDash \chi_2$ and proving $\mathrm{Coalg}(T) \nvDash \chi$. First we note that if χ_1 is a propositional tautology, $\mathrm{Coalg}(T) \vDash \chi_1$ and we are done, so in the remainder we can assume that χ_1 is not a propositional tautology. Now we take a T-coalgebra falsifying χ_2 and fiddle with it to make a coalgebra which falsifies χ.

Since $\mathrm{Coalg}(T) \nvDash \chi_2$ there exists a coalgebra (C, γ), a valuation $v : V \to \mathcal{P}_{fs}(C)$ and a state $c \in C$ such that $c \notin [\![\chi_2]\!]_{C,v}$. We construct a new coalgebra. First we clone c to create a state with the same successor as c. Then we change the valuation on that cloned state so that χ_1 does not hold at the new state. Since we require a nominal set, we will have to clone not just c, but the entire orbit of c in C, which we call $orb(c) = \{\pi \cdot c \mid \pi \in \mathbb{P}_{\mathbb{A}}\}$. The successors will be carried over, and the valuation on this new $orb(c) - \{c\}$ is irrelevant but we will just make it the same as that of the new c. That is $v'(p) = v(p) + orb(c)$ if p is a negative literal in χ_1 and $v'(p) = v(p)$ otherwise.

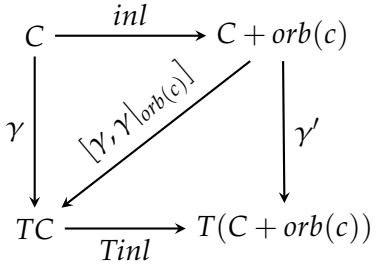
First, let's look at the carrier set $C + orb(c)$, and define a function to TC



Now define the coalgebra function $\gamma' : C + orb(c) \to T(C + orb(c))$ by



Now we can see that *inl* is a coalgebra morphism by joining the triangles from the above diagrams



Therefore, by Theorem 3.5.3 we have

$$inl^{-1}([\![\phi]\!]_{C+orb(c),v'}) = [\![\phi]\!]_{C,inl^{-1}\circ v'}$$

and from the definition of v' we can see that $inl^{-1} \circ v' = v$ so we have that

$$inl^{-1}([\![\phi]\!]_{C+orb(c),v'}) = [\![\phi]\!]_{C,v}$$

which we use in the following proof.

We start with the fact that $c \notin [\![\chi_2]\!]_{C,v}$. Since χ_2 is a clause over modal formulae, c has the desired property exactly when for all positive $\heartsuit\phi$ in χ_2 we have $c \notin [\![\heartsuit\phi]\!]_{C,v}$ AND for all negative $\neg\heartsuit\phi$ in χ_2 we have $c \in [\![\heartsuit\phi]\!]_{C,v}$. Now

$$\begin{aligned}
& c \in [\![\heartsuit\phi]\!]_{C,v} \\
\iff & \gamma(c) \in [\![\heartsuit\phi]\!]_{TC,v} \\
\iff & \gamma(c) \in [\![\heartsuit]\!]_C([\![\phi]\!]_{C,v}) \\
\iff & \gamma(c) \in [\![\heartsuit]\!]_C \circ inl^{-1}([\![\phi]\!]_{C+orb(c),v'}) \\
\iff & \gamma(c) \in (Tinl)^{-1} \circ [\![\heartsuit]\!]_{C+orb(c)}([\![\phi]\!]_{C+orb(c),v'}) \\
\iff & Tinl \circ \gamma(c) \in [\![\heartsuit]\!]_{C+orb(c)}([\![\phi]\!]_{C+orb(c),v'}) \\
\iff & Tinl \circ \gamma(c) \in [\![\heartsuit\phi]\!]_{T(C+orb(c)),v'} \\
\iff & Tinl \circ \gamma|_{orb(c)}(c) \in [\![\heartsuit\phi]\!]_{T(C+orb(c)),v'} \\
\iff & \gamma'(inr(c)) \in [\![\heartsuit\phi]\!]_{T(C+orb(c)),v'} \\
\iff & inr(c) \in [\![\heartsuit\phi]\!]_{C+orb(c),v'}
\end{aligned}$$

Apart from the result we took from the fact that *inl* is a coalgebra morphism, all of the steps here can be seen in the diagrams above, in the definition of predicate liftings, or in the definition of the relationship between interpretations on the successor space and interpretations on the state space.

Now, because this is true for all $\heartsuit\phi$, we can conclude that $inr(c) \notin [\![\chi_2]\!]_{C+orb(c),v'}$. This concludes the difficult part of the proof, but it remains to show that $inr(c) \notin [\![\chi_1]\!]_{C+orb(c),v'}$. This would follow if $inr(c) \notin v'(p)$ for all positive literals p in χ_1 and $inr(c) \in v'(q)$ for all negative literals q in χ_1. From the definition of v' this is obvious. □

When we are have a disjunction, this result, along with soundness and completeness, will allow us to split the problem of deciding it into two problems. Firstly, we check the propositional part of the disjunction for tautologies, and secondly we check the modal part using the rest of the decision procedure.

The framework in **Set** relies on strict completeness and another property called contraction closure. We conjecture that the same will be true of the framework in **Nom**. We do not foresee И creating undecidability, as it self dual, and does not create undecidability when added to propositional logic.

6.2 Global Caching for Fresh Hennessy Milner Logic

In this section we will informally discuss a potential modification of the original global caching algorithm for the Attributive Concept Language with Complements (ALC). Since ALC is the description logic version of HML, and since the nominal version of HML is theorem identical to traditional HML, global caching is obviously applicable to it. What we wish to consider is whether we can modify that original global caching algorithm to admit the И quantifier, and provide a decision procedure for И-HML.

Recall that we chose the transitions to be from $\mathbb{A}$ for simplicity. In general we can choose the transitions to be from any nominal set, and there are not great changes to the logic. We consider first the case of И-HML with $\mathbb{A}$ transitions. We require that formulae be put into И-normal form. Following this, we can see that each modality in the formula is either free or bound/fresh. The $\langle a \rangle$ modalities correspond to $\exists$-transitions in the AND-OR graph.

As we build the global caching AND-OR graph, we keep a set of fresh names on standby to stand in for the fresh transitions, and instantiate the bound transitions with one of those fresh names when the $\exists$ rule is applicable to the transition.

Given Δ is the set of global assumptions, we have the following rule in addition to the existing rules in the global caching algorithm. Ensuring that n is totally fresh:

$$\frac{И n.\langle n \rangle \phi,\ И n.[n]\psi_1,\ ...,\ И n.[n]\psi_i,\ \Gamma}{\phi, \psi_1, ..., \psi_i, \Delta}(И n.\langle n \rangle)$$

Note that before applying this rule we must convert all $И m.[m]\varphi$ to the α-equivalent $И n.[n]\varphi[n/m]$ to ensure that all of the fresh universals are carried over to the fresh successor instantiated by the rule.

It is important to note that the above rule does not mean that α-equivalent formulae should be treated differently. In reality it means quite the contrary. It means that for all fresh universal properties $И n.[n]\psi$, the unmodalised ψ must be transmitted along any and every fresh existential. We are in effect creating a fresh successor, and all of the fresh universals must apply to it. It is simpler to rename each of the relevant formula so that the bound names match, because when the modalities are removed in the new state, those names are no longer bound.

This is an additional $\exists$ rule which can be used for fresh transitions. This is essentially identical to the existing $\exists$ rule, except that we carry with us all of the fresh universals for each fresh existential, but we are allowed to instantiate different fresh existentials with different fresh names.

In the more general case of И-HML with transitions drawn from a nominal set, we still have a И-normal form. In this case there can be partially fresh transitions.

The parts of the transition being instantiated which aren't fresh must match the unfresh parts of the universals being carried with it. To ensure this we convert all $И m.[x']\varphi$ to the α-equivalent $И n.[(n\,m) \cdot x'](n\,m) \cdot \varphi$. If $(n\,m) \cdot x' = x$ then we know

that the unfresh parts match, and we can use the following rule:

$$\frac{И n.\langle x\rangle \phi,\ И n.([x]\psi_i),\ ...,\ И n.([x]\psi_1),\ \Gamma}{\phi, \psi_1, ..., \psi_i, \Delta}(И n.\langle x\rangle)$$

We conjecture that the above methods, or a sufficient extrapolation of them, result in a sound and complete decision procedure for И-HML. This is left as future work.

Chapter 7

Nominal Calculi

Development of the framework of coalgebraic modal logic outside **Set** should be directed in part by the characteristics of existing modal logics. The addition of И to the standard framework, as achieved by the change of category performed in this thesis, has been influenced by these concerns. Specifically, there already exist modal logics which use the И quantifier, especially in the field of concurrency and mobility. Some examples include spatial logics for reasoning about concurrency [Caires and Cardelli 2003], π-calculus variants for reasoning about mobility [Sangiogi and Walker 2001], and MOMO, a modal logic for reasoning about mobility [Nicola and Loreti 2008]. The logics of nominal calculi may serve as examples of coalgebraic modal logics in **Nom**.

In this chapter, we will highlight one particular logic and discuss the relationship between our framework and the logic. This will be followed in the future work section by a more general discussion of these types of logics and how coalgebraic modal logic might be extended to capture their other nonmodal aspects.

7.1 MOMO: A modal logic for reasoning about mobility

The concept of multilabelled transition systems (MLTSs) have been introduced [Nicola and Loreti 2008] as a semantic framework for examining nominal calculi. The framework is applied to the asynchronous π-calculus and to a fragment of the experimental programming language KLAIM called μ-KLAIM. MOMO is proposed as a logic for these multilabelled transition systems, and specific variants are given for these two examples. Only the syntax and semantics of MOMO are given. Proof rules are not discussed.

Models for MOMO are given by MLTSs, and the interpretation of a logical formula is the set of states (generally process expressions) which validate that formula. Thus, MOMO has a different dialect (given by different sets of names/resources/labels/label-predicates) for each different mobile computational calculi. That said, we focus on the generic aspects of the logic, and show in the following discussion that Hennessy Milner Logic with the fresh quantifier (ИHML) effectively captures a large fragment of MOMO (up to some minor differences). ИHML is examined as an example throughout this thesis, and the specific details of the logic can be found (there).

First, we will present the concept of multilabelled transition systems as they were

originally presented [Nicola and Loreti 2008]. MLTS will be central to the semantics of MOMO.

Definition 7.1.1. An MLTS is a tuple:

$$\mathcal{M} = \langle\, \mathcal{S},\, \mathcal{R},\, \mathcal{L},\, \mathbf{N},\, \rightarrow,\, \dashrightarrow,\, \hookrightarrow \rangle$$

Where $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{L}$ are sets of states, resources and labels respectively, $\mathbf{N}$ is a "naming structure" for those sets, and

$$\begin{aligned} \rightarrow &: \mathcal{S} \times \mathcal{L} \times \mathcal{S} \\ \dashrightarrow &: \mathcal{S} \times (\{\oplus, \ominus\} \times \mathcal{R}) \times \mathcal{S} \\ \hookrightarrow &: \mathcal{S} \times \mathbb{A} \times \mathcal{S} \end{aligned}$$

are relations corresponding to computation transitions, resource creation/consumption, and name revelation respectively, and where $\mathbb{A}$ is a countably infinite set of names. If $(s, \lambda, s') \in \rightarrow$ we write $s \xrightarrow{\lambda} s'$, and a similar notation applies to the other relations.

The concept of *naming structure* used in the original definition differs in a few ways from what would be achieved if $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{L}$ were simply declared to be nominal sets. Firstly, arbitrary name substitutions are used as opposed to name permutations (bijective name substitutions). Secondly, the support of an element is not constructed from the action of the substitutions on that element, but rather it is supposed to exist as a primitive. It is only constrained in the sense that we have $\eta(\sigma \cdot x) = \sigma(\eta(x))$ where η is the support function, σ is a substitution and $\sigma(X) = \{\sigma(x) \mid x \in X\}$. Though this constraint is certainly satisfied by the nominal sets framework, it is not immediately clear whether or not it is enough to recover the full nominal sets notion of support.

The only other difference is that for nominal sets we have $\pi_1 \cdot (\pi_2 \cdot x) = (\pi_1 \circ \pi_2) \cdot x$, but this is omitted in the definition of naming structure for MLTS. This omission may be accidental, since it certainly applies to all the relevant examples. Arguably, all of the relevant examples can be captured by just requiring the sets $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{L}$ to be nominal, and all of these minor differences can be done away with. This would be convenient, as the theory of nominal sets is well developed in the literature. This means that the proofs and concepts surrounding the topic are accessible, and we know from these investigations that the formal concept matches many practical examples. Meanwhile the alternate concept of naming structure does not have a well developed theory to support it, and is weaker than the nominal sets concept, meaning that it may not have all of the relevant properties which make nominal sets theory so useful.

Figure 7.1 gives the full generic syntax of MOMO.

A model for MOMO will be a multilabelled transition system, a set of label predicates pn with an interpretation function $\mathsf{B} : pn \rightarrow \mathcal{P}(\mathcal{L})$, and a logical environment $\varepsilon : V \rightarrow \mathcal{P}(\mathcal{S})$ (a valuation of the logical variables).

$\phi, \psi ::=$	
$\mathbf{T}$	true
$\mid \phi \wedge \psi$	conjunction
$\mid \neg\phi$	negation
$\mid \kappa$	logical variable
$\mid \nu\kappa \,.\, \phi$	maximal fixed point
$\mid \rho \rightarrow \phi$	resource consumption
$\mid \rho \leftarrow \phi$	resource production
$\mid \langle\alpha\rangle\phi$	transition existence
$\mid \exists n \,.\, \phi$	name quantification
$\mid \{n_1 = n_2\}$	name matching
$\mid n ® \phi$	name revelation
$\mid \curlyvee n \,.\, \phi$	fresh name quantification

Figure 7.1: The generic syntax of MOMO and the intuitive meanings of the syntax

Definition 7.1.2. The MLTS interpretation of MOMO formulae is given as a subset of $\mathcal{S}$ by:

Kernel Fragment:

$$\begin{aligned}
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\mathbf{T}]\!]\varepsilon &= S \\
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\neg\phi]\!]\varepsilon &= S - \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi]\!]\varepsilon \\
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi_1 \wedge \phi_2]\!]\varepsilon &= \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi_1]\!]\varepsilon \cap \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi_1]\!]\varepsilon \\
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\kappa]\!]\varepsilon &= \varepsilon(\kappa) \\
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\nu\kappa.\phi]\!]\varepsilon &= \nu\mathcal{F}
\end{aligned}$$

Where:

$$\mathcal{F}(S) = \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi]\!](\varepsilon[\kappa \mapsto S])$$

State Formulae:

$$\begin{aligned}
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\rho \rightarrow \phi]\!]\varepsilon &= \left\{ s_1 \;\middle|\; s_1 \overset{\ominus\rho}{\dashrightarrow} s_2 \text{ and } s_2 \in \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi]\!]\varepsilon \right\} \\
\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\rho \leftarrow \phi]\!]\varepsilon &= \left\{ s_1 \;\middle|\; s_1 \overset{\oplus\rho}{\dashrightarrow} s_2 \text{ and } s_2 \in \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi]\!]\varepsilon \right\}
\end{aligned}$$

Temporal Formulae:

$$\mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\langle\alpha\rangle\phi]\!]\varepsilon = \left\{ s_1 \;\middle|\; \exists s_2 \in \mathbb{M}^{\langle \mathcal{M}, \mathsf{B} \rangle}[\![\phi]\!]\varepsilon \,.\, s_1 \xrightarrow{\lambda} s_2,\ \lambda \in \mathsf{B}(\alpha) \right\}$$

Nominal Formulae:

$$\mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ \exists n.\phi\ ]\!]\varepsilon = \bigcup_{n'\in\mathcal{N}} \mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ \phi[n'/n]\ ]\!]\varepsilon$$

$$\mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ n\textcircled{R}\phi\ ]\!]\varepsilon = \left\{ s_1 \;\middle|\; \exists s_2 \in \mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ \phi\ ]\!]\varepsilon \,.\, s_1 \hookrightarrow^n s_2 \right\}$$

$$\mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ \text{И}n.\phi\ ]\!]\varepsilon = \left\{ s \;\middle|\; \exists n'.n' \notin fn(\phi) \cup \eta(s) \text{ and } s \in \mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ \phi[n'/n]\ ]\!]\varepsilon \right\}$$

Where $fn(\phi)$ is the free names of the formula ϕ and $\eta(s)$ is the set of names in the support of s given by the naming structure $\mathbf{N}$. Note that $\mathbb{M}^{\langle \mathcal{M},\mathsf{B}\rangle}[\![\ \{n_1 = n_2\}\ ]\!]\varepsilon$ is $\mathcal{S}$ if n_1 and n_2 are the same name, and $\varnothing$ otherwise.

The original MLTS paper does not specify what the free names of a formula are, and this is problematic because even though the support of the labels is given by the naming structure imposed on $\mathcal{L}$, a notion of what the support of the label predicates should be is never given. We can define it in a reasonable way. Replacing the naming structure with the nominal sets framework, and addressing the support of the various labelled transition operators in the way that we have treated nominal modalities in this thesis, we can be explicit about the support of a formula.

7.2 Coalgebraic semantics for MOMO

We now examine how the extended framework of coalgebraic modal logic treats the example of MOMO. As may be evident from the existing semantics, many of the concepts involve the existence of certain types of transitions. We aim to capture resource creation and consumption, computational transition, and name revelation with Hennessy-Milner modalities. Our semantics for the И quantifier is different in the sense that ours uses the nominal sets framework and MOMO uses a custom notion of naming structure. We propose that a change to our semantics does not present a significant change in terms of the examples, and that our semantics benefit from the well developed theories of nominal sets and coalgebraic modal logic.

The only issue we leave out of our coalgebraic semantics for MOMO is the fixed point operators. Coalgebraic approaches to fixed point operators have been attempted, but these are not the focus of this thesis. A brief discussion of the fixed point operators can be found in the future work section. From now on we distinguish between MOMO with fixed points, and MOMO without fixed points. The latter we will call Momo.

In order to present the coalgebraic semantics of Momo, we need a coalgebra in **Nom** and finitely supported predicate liftings for the modalities of Momo. We address the modalities in order of complexity, first name revelation, then resource creation and consumption, and finally the label predicates for computational transitions. We will alter the coalgebra as we proceed, so that in the end we arrive at a coalgebra which is a model for Momo. In the following, we drop the valuation on the coalgebra side and the logical environment on the MLTS side. They are equivalent concepts and they are superfluous to the presentation below.

Another aspect of the MLTS model that distinguishes it from the coalgebraic semantics we will present is that it accepts models of transition structures that are not constrained by the requirement of equivariance. These nonequivariant models are intuitively not relevant to examples of computational calculi as such calculi define transitions in terms of the current state without invoking specific names. In other words, there is generally a state expression for each state, and the method of determining the transitions which can be made from this state expression does not invoke specific names, but rather uses the structure of the state expression in question (and the names therein). For these reasons we do not see a problem with this change of semantics.

We first address the name revelation case. Recall that in the MOMO semantics we have a set of states $\mathcal{S}$. We correspondingly have a set of states S. We will start with a coalgebra $\gamma : S \to \mathcal{P}_{fs}(S)^{\mathbb{A}}$, recognising that the MOMO semantics for name revelation is essentially that of the transition existential $\langle a \rangle$ of Hennessy Milner Logic, and since the transition labels here are just the names, we can apply the semantics of $\langle a \rangle$ from ИHML without difficulty.

$$\begin{aligned}
[\![\ n\circledR\ ]\!]_S : \mathcal{P}_{fs}(S) &\xrightarrow{fs} \mathcal{P}_{fs}(\mathcal{P}_{fs}(S)^{\mathbb{A}}) \\
[\![\ n\circledR\ ]\!]_S(X) &= \left\{ d \in \mathcal{P}_{fs}(S)^{\mathbb{A}} \;\middle|\; d(n) \cap X \neq \varnothing \right\} \\
[\![\ n\circledR\phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}}} &= [\![\ n\circledR\ ]\!]_S([\![\ \phi\ ]\!]_S) \\
[\![\ n\circledR\phi\ ]\!]_S &= \gamma^{-1}([\![\ n\circledR\phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}}})
\end{aligned}$$

The support of this predicate lifting is n. To verify that this defines a predicate lifting, we can just refer to the corresponding observation for ИHML.

We now address the case of resource creation and consumption. This can also be cast as an addition to our existing coalgebra by recognising the similarity between the resource modalities and the transition existential. Our coalgebra is now $\gamma : S \to \mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}$, where the left $\mathcal{R}$ contains resource creators $\{\oplus\rho_1, \oplus\rho_2, ...\}$, and the right $\mathcal{R}$ contains resource consumers $\{\ominus\rho_1, \ominus\rho_2, ...\}$.

The reader may recall that we chose $\mathbb{A}$ as the set of labels for simplicity when we defined И-HML, but in fact any nominal set can be used as the set of labels.

Now the semantics of resource creation and consumption are captured again by ИHML.

$$\begin{aligned}
[\![\ \rho\leftarrow\ ]\!]_S : \mathcal{P}_{fs}(S) &\xrightarrow{fs} \mathcal{P}_{fs}(\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}) \\
[\![\ \rho\leftarrow\ ]\!]_S(X) &= \left\{ d \in \mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}} \;\middle|\; d(\oplus\rho) \cap X \neq \varnothing \right\} \\
[\![\ \rho\leftarrow\phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}} &= [\![\ \rho\leftarrow\ ]\!]_S([\![\ \phi\ ]\!]_S) \\
[\![\ \rho\leftarrow\phi\ ]\!]_S &= \gamma^{-1}([\![\ \rho\leftarrow\phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}})
\end{aligned}$$

$$\begin{aligned}
[\![\ \rho \to]\!]_S : \mathcal{P}(S)_{fs} &\xrightarrow{fs} \mathcal{P}_{fs}(\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}) \\
[\![\ \rho \to]\!]_S(X) &= \left\{ d \in \mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}} \;\middle|\; d(\ominus\rho) \cap X \neq \varnothing \right\} \\
[\![\ \rho \to \phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}} &= [\![\ \rho \to]\!]_S([\![\ \phi\ ]\!]_S) \\
[\![\ \rho \to \phi\ ]\!]_S &= \gamma^{-1}([\![\ \rho \to \phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}}})
\end{aligned}$$

The support of these predicate liftings is the support of ρ, and again, since it matches the semantics of $\langle a \rangle$ in ИHML, it is indeed a predicate lifting.

We will add another set of transition labels for the temporal modality. This time we have a slightly more complicated job, because the transitions in the MOMO semantics are labelled with elements of $\mathcal{L}$, but the modalities are given by the label predicates pn, and we just have $\mathsf{B} : pn \to \mathcal{P}(\mathcal{L})$. So, we make the label predicates the additional transition labels, make pn a nominal set, and treat this case as the other two, and then show that the semantics works out to match. Note that we will have $\mathsf{B} : pn \to \mathcal{P}_{fs}(\mathcal{L})$ on our view.

The coalgebraic semantics can be given as follows:

$$\begin{aligned}
[\![\ \langle\alpha\rangle\ ]\!]_S : \mathcal{P}(S)_{fs} &\xrightarrow{fs} \mathcal{P}_{fs}(\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn}) \\
[\![\ \langle\alpha\rangle\ ]\!]_S(X) &= \left\{ d \in \mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn} \;\middle|\; d(\alpha) \cap X \neq \varnothing \right\} \\
[\![\ \langle\alpha\rangle\phi\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn}} &= [\![\ \langle\alpha\rangle\ ]\!]_S([\![\ \phi\ ]\!]_S) \\
[\![\ \langle\alpha\rangle\phi\ ]\!]_S &= \gamma^{-1}([\![\ \langle\alpha\rangle\ ]\!]_{\mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn}})
\end{aligned}$$

In order to show that the two versions of the semantics match, we first need to take an arbitrary labelled transition relation $\to \subseteq S \times \mathcal{L} \times S$ and define the coalgebra to match it, and vice versa. Then we need to check that they agree on which states satisfy $\langle\alpha\rangle\phi$.

So, given an arbitrary labelled transition relation $\to \subseteq S \times \mathcal{L} \times S$, we first define the function of type $S \times \mathcal{L} \to \mathcal{P}(S)$ that takes a state/label pair (s, λ) and returns all states s' such that $s \xrightarrow{\lambda} s'$. From this function we can curry to get a function of type $S \to \mathcal{P}(S)^{\mathcal{L}}$. We need to use this function to define the coalgebra map, so call it $g : S \to \mathcal{P}(S)^{\mathcal{L}}$.

Now let the coalgebra map be $f : S \to \mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn}$ defined on the pn as follows.

$$f(s)(\alpha) = \bigcup_{\lambda \in \mathsf{B}(\alpha)} g(s)(\lambda)$$

There are questions about support and equivariance here that have to be answered.

After answering them we have

$$\begin{aligned}
[\![\langle\alpha\rangle\phi]\!]_S &= f^{-1}(\left\{d \in \mathcal{P}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn} \,\middle|\, d(\alpha) \cap [\![\phi]\!]_S \neq \varnothing\right\}) \\
&= \left\{s \in S \,\middle|\, f(s) \in \left\{d \in \mathcal{P}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn} \,\middle|\, d(\alpha) \cap [\![\phi]\!]_S \neq \varnothing\right\}\right\} \\
&= \{s \in S \mid f(s)(\alpha) \cap [\![\phi]\!]_S \neq \varnothing\} \\
&= \left\{s \in S \,\middle|\, \bigcup_{\lambda \in \mathsf{B}(\alpha)} g(s)(\lambda) \cap [\![\phi]\!]_S \neq \varnothing\right\} \\
&= \left\{s \in S \,\middle|\, \exists s' \in \bigcup_{\lambda \in \mathsf{B}(\alpha)} g(s)(\lambda) \cap [\![\phi]\!]_S\right\} \\
&= \left\{s \in S \,\middle|\, \exists s' \in [\![\phi]\!]_S \,.\, \exists \lambda \in \mathsf{B}(\alpha) \,.\, s \xrightarrow{\lambda} s'\right\}
\end{aligned}$$

Inductively we can assume that $[\![\phi]\!]_S = \mathbb{M}^{\langle \mathcal{M}, \mathsf{B}\rangle}[\![\phi]\!]$, so we get that the semantics of $\langle\alpha\rangle\phi$ match when we go from the original semantics to the new semantics.

Given an arbitrary coalgebra map $f : S \to \mathcal{P}_{fs}(S)^{\mathbb{A}+\mathcal{R}+\mathcal{R}+pn}$, we will generate a labelled transition relation.

Take:

$$\begin{aligned}
\mathcal{L} &= \mathcal{P}(pn) \\
\to &= \left\{(s, \lambda, s') \,\middle|\, s' \in \bigcap_{\alpha \in \lambda} f(s)(\alpha)\right\} \\
\mathsf{B}(\alpha) &= \{\lambda \in \mathcal{L} \mid \alpha \in \lambda\}
\end{aligned}$$

Then (writing $[\![\cdot]\!]$ for $\mathbb{M}^{\langle \mathcal{M}, \mathsf{B}\rangle}[\![\cdot]\!]$)

$$\begin{aligned}
[\![\langle\alpha\rangle\phi]\!] &= \left\{s_1 \,\middle|\, \exists s_2 \in [\![\phi]\!] \,.\, \exists \lambda \in \mathsf{B}(\alpha) \,.\, s_1 \xrightarrow{\lambda} s_2\right\} \\
&= \{s_1 \mid \exists s_2 \in [\![\phi]\!] \,.\, \exists \lambda \,.\, \alpha \in \lambda \,.\, (s_1, \lambda, s_2) \in \to\} \\
&= \left\{s_1 \,\middle|\, \exists s_2 \in [\![\phi]\!] \,.\, \exists \lambda \,.\, \alpha \in \lambda \,.\, s_2 \in \bigcap_{\alpha' \in \lambda} f(s_1)(\alpha')\right\} \\
&= \left\{s_1 \,\middle|\, \exists s_2 \in [\![\phi]\!] \,.\, \exists \lambda \,.\, \alpha \in \lambda \,.\, \forall \alpha' \in \lambda \,.\, s_2 \in f(s_1)(\alpha')\right\} \\
&= \{s_1 \mid \exists s_2 \in [\![\phi]\!] \cap f(s)(\alpha)\} \\
&= \{s \in S \mid f(s)(\alpha) \cap [\![\phi]\!] \neq \varnothing\}
\end{aligned}$$

Using the same inductive trick as before, we can conclude that swapping from the new semantics to the old preserves the interpretation of $\langle a\rangle\phi$. It is interesting to note that the above makes $\mathcal{L}$ redundant in the semantics, which is why it is not present.

Thus, we have successfully captured the semantics of the transition modalities in MOMO. Name equality and the $\mathsf{И}$ quantifier come for free now that we have fit Momo into what amounts to $\mathsf{И}$HML with transition labels taken from a nominal sets instead

of from the particular nominal set $\mathbb{A}$. We have essentially simplified a large part of the logic in this way.

The only things left out of the new semantics are the fixed points, and name quantification. Name quantification is not addressed here, but we do present some information about fixed points in the future work section.

Chapter 8

Conclusion and Future Work

This chapter will summarise the key results of the previous chapters, discuss the relevance and importance of the work and present avenues for future research.

8.1 Summary of results

A number of novel results are present in this thesis.

We established a semantic framework for coalgebraic modal logic in **Nom**. We proved that the interpretation is equivariant and preserved by coalgebra morphisms.

We partially established a proof theory framework for coalgebraic modal logic in **Nom**. Proof theory for coalgebraic modal logic has, to the best of our knowledge, never been established outside of **Set**. We established generic soundness results, and went a good way towards establishing generic completeness results.

We incorporated the И quantifier into the semantic framework, and made efforts towards incorporating it into the proof theory, highlighting the challenges of this move.

We presented a small result relevant to generic decidability.

We showed that an existing logic for reasoning about nominal calculi can be simplified by the semantic framework for coalgebraic modal logic in **Nom**.

8.2 Relevance of the thesis

This thesis is relevant for two main reasons. Firstly, name binding in systems is commonplace, and the standard approaches to name binding in coalgebraic modal logic do not address generic proof theory. Secondly, the generic proof theory which is so successful in coalgebraic modal logic in **Set** has never been adapted to other categories before.

We do not see this as a complete answer to either of these demands. This thesis represents a preliminary foray into these areas. We do, however, see the results of the thesis as an important first step towards a deeper understanding of how the generic proof theory of coalgebraic modal logic can be adapted to reason about logics containing the И quantifier or other nominal constructs.

8.3 Future Work

Full generic completeness result

Whether by proving Conjecture 4.2.11 or arriving at completeness from a different direction, proving a generic completeness result in this context would firmly establish the first generic coalgebraic proof theory beyond **Set**.

Proof theory for fresh CML

Incorporating И into the generic proof theory framework of CML in **Nom** would be a useful step towards providing a generic proof theory for nominal calculi, and other name binding systems.

Global Caching for fresh HML

We have shown that И-HML with transitions drawn from a nominal set is a useful concept for capturing at a high level all manner of transition behaviour, from name revelation to resource creation and consumption, to computation. The ability to use global caching to reason about these kinds of systems could help in their analysis and design.

Generic Decidability results

Decidability results for CML in **Nom** and И-CML would further expand the usefulness of the generic proof theory.

Coalgebraic Bisimulation

Showing that bisimulation and logical equivalence coincide for CML in **Nom** would be useful to ensuring the logics are expressive for the functors. This will probably require a restriction to finitary functors and the introduction of a notion of separating structure [Pattinson 2012]. The other approach would be to exploit the adjunction between syntax and semantics to get expressivity for free. This approach might damage the current approach to proof theory by leaving us with something less than concrete.

Future Work: Fixed Points in MOMO

Fixed points in a language without negation can be interpreted in the following way [Venema 2004]:

$$[\![\mu x.p]\!]_{S,V} = \bigcap \{ S' \subseteq S \mid [\![p]\!]_{S,V[x \mapsto S']} \subseteq S' \}$$

$$[\![\nu x.p]\!]_{S,V} = \bigcup \{ S' \subseteq S \mid S' \subseteq [\![p]\!]_{S,V[x \mapsto S']} \}$$

Where:

$$V[x \mapsto S'](x) = S'$$
$$V[x \mapsto S'](y) = V(y) \qquad (\forall y \neq x)$$

However, MOMO allows negation, and fixed points can still be defined if each logical variable $(x \in X)$ is nested in an even number of negations.

Further study of fixed points in coalgebraic modal logic is required to address these concerns.

Nominal Calculi

Capturing more process calculi as coalgebas on nominal sets will provide more insight into how to address the generic concerns of the proof theory, and might also lead to simplifications of the logics of those process calculi. For example, the spatial logic for concurrency [Caires and Cardelli 2003] is a nominal calculi that uses the $\mathsf{И}$ quantifier. It is possible that a fragment of it is expressible in our framework.

Compositionality

Studying the composition of coalgebraic modal logics in nominal sets may lead to interesting composite systems. In particular, set coalgebras can be embedded into **Nom** as equivariant functions on nominal sets with a trivial permutation action. Given that a theory of compositionality is developed, set coalgebras can be composed with nominal coalgebras by first embedding the set coalgebras and then performing the composition in **Nom**.

Other categories

This work was a first step for generic CML proof theory outside **Set**. Other categories may be relevant to different types of systems. One obvious suggestion is to expand the **Nom** framework to the category of Nominal sets and finitely supported functions, or the category of FM-sets [Clouston 2009] and finitely supported functions. Presheaf topoi such as $\mathbf{Set}^{\mathbb{I}}$ which, like **Nom**, have been relevant in modelling names could also be investigated.

In general, the results of coalgebraic modal logic will have to be rehashed for every new category. This would be a laborious process. It would be appealing to have a more general theory. It may be possible to generalise the results of generic proof theory in CML to all topoi [Goldblatt 2006], or all topoi of a certain kind, such as presheaf topoi, or Grothendieck topoi.

In general, the internal logic of topoi is intuitionistic, so basing a modal logic in the most general topoi would result in an account of intuitionistic propositional logic augmented by modalities. Intuitionistic logic may make the proof theory more challenging to establish, so it may be worth considering primarily classical examples. In

any case, the internal logic of the category is something to keep in mind when expanding CML beyond **Set**.

Bibliography

AWODEY, S. 2006. *Category theory*, Volume 49. Oxford University Press. (p. 5)

BLACKBURN, P., VAN BENTHEM, J. F., AND WOLTER, F. 2006. *Handbook of modal logic*, Volume 3. Elsevier Science. (p. 1)

CAIRES, L. AND CARDELLI, L. 2002. A spatial logic for concurrency (part ii). In *CONCUR 2002Concurrency Theory*, pp. 209–225. Springer. (p. 13)

CAIRES, L. AND CARDELLI, L. 2003. A spatial logic for concurrency (part i). *Information and Computation 186*, 2, 194–235. (pp. 13, 57, 67)

CIANCIA, V. AND TUOSTO, E. 2009. A novel class of automata for languages on infinite alphabets. Technical Report CS-09-003, University of Leicester. (p. 9)

CÍRSTEA, C. 2010. Generic infinite traces and path-based coalgebraic temporal logics. *Electronic Notes in Theoretical Computer Science 264*, 2, 83–103. (p. 11)

CÎRSTEA, C., KURZ, A., PATTINSON, D., SCHRÖDER, L., AND VENEMA, Y. 2011. Modal logics are coalgebraic. *The Computer Journal 54*, 1, 31–41. (pp. 2, 10)

CLOUSTON, R. 2009. *Equational Logic for names and binders*. PhD thesis, University of Cambridge. (pp. 8, 67)

DING, Y. AND HAARSLEV, V. 2006. Tableau caching for description logics with inverse and transitive roles. In *Proc. DL-2006: International Workshop on Description Logics* (2006), pp. 143–149. (p. 14)

DONINI, F. M. AND MASSACCI, F. 2000. Exptime tableaux for ALC. *Artificial Intelligence 124*, 1, 87–138. (p. 14)

EMERSON, E. A. 1990. Temporal and modal logic. *Handbook of theoretical computer science 2*, 995–1072. (p. 1)

FIORE, M. AND STATON, S. 2006. A congruence rule format for name-passing process calculi from mathematical structural operational semantics. In *Logic in Computer Science, 2006 21st Annual IEEE Symposium on* (2006), pp. 49–58. IEEE. (p. 8)

GABBAY, M. AND PITTS, A. 1999. A new approach to abstract syntax involving binders. In *Logic in Computer Science, 1999. Proceedings. 14th Symposium on* (1999), pp. 214–224. IEEE. (p. 2)

GABBAY, M. J. AND PITTS, A. M. 2002. A new approach to abstract syntax with variable binding. *Formal aspects of computing 13*, 3-5, 341–363. (p. 2)

GOLDBLATT, R. 2006. *Topoi: the categorial analysis of logic*, Volume 98. Courier Dover Publications. (pp. 13, 67)

GORÉ, R., KUPKE, C., AND PATTINSON, D. 2010. Optimal tableau algorithms for coalgebraic logics. In J. ESPARZA AND R. MAJUMDAR Eds., *Tools and Algorithms for the Construction and Analysis of Systems*, Volume 6015 of *Lecture Notes in Computer Science*, pp. 114–128. Springer Berlin / Heidelberg. (pp. 2, 10, 12, 15)

GORÉ, R. AND NGUYEN, L. A. 2007a. Exptime tableaux for alc using sound global caching. In *Proceedings of the International Workshop on Description Logics (DL2007)* (Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June 2007). (pp. 14, 15)

GORÉ, R. AND NGUYEN, L. A. 2007b. Optimised exptime tableaux for alc using sound global caching, propagation and cutoffs. In *Proceedings of the International Workshop on Description Logics (DL2007)* (Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June 2007). (p. 14)

JACOBS, B. AND RUTTEN, J. 1997. A tutorial on (co)algebras and (co)induction. *European Association for Theoretical Computer Science 62*, 222–259. (p. 7)

KLIN, B. 2007. Coalgebraic modal logic beyond sets. *Electron. Notes Theor. Comput. Sci. 173*, 177–201. (p. 12)

KURZ, A. 1998. Specifying coalgebras with modal logic. *Electronic Notes in Theoretical Computer Science 11*, 56–70. (p. 10)

MAC LANE, S. 1978. *Categories for the working mathematician* (second ed.). Graduate Texts in Mathematics. Springer, New York. (p. 5)

MILNER, R., PARROW, J., AND WALKER, D. 1993. Modal logics for mobile processes. *Theoretical Computer Science 114*, 1, 149–171. (p. 1)

MOSS, L. S. 1999. Coalgebraic logic. *Annals of Pure and Applied Logic 96*, 1, 277–317. (p. 10)

NICOLA, R. D. AND LORETI, M. 2008. Multiple-labelled transition systems for nominal calculi and their logics. *Math. Struct. in Comp. Sci. 18*, 107–143. (pp. 13, 57, 58)

PATTINSON, D. 2003. Coalgebraic modal logic: soundness, completeness and decidability of local consequence. *Theoretical Computer Science 309*, 1-3, 177 – 193. (p. 10)

PATTINSON, D. 2012. Coalgebraic Modal Logic: Modalities Beyond Kripke Semantics. Unpublished course notes. (pp. ix, 10, 17, 51, 66)

PATTINSON, D. AND SCHRÖDER, L. 2008a. Admissibility of cut in coalgebraic logics. *Electronic Notes in Theoretical Computer Science 203*, 5, 221–241. (p. 12)

PATTINSON, D. AND SCHRÖDER, L. 2008b. Beyond rank 1: Algebraic semantics and finite models for coalgebraic logics. In *Foundations of Software Science and Computational Structures*, pp. 66–80. Springer. (p. 11)

PETRISAN, D. L. 2012. *Investigations into Algebra and Topology over Nominal Sets*. PhD thesis, University of Leicester. (p. 9)

PITTS, A. M. 2003. Nominal logic, a first order theory of names and binding. *Information and computation 186*, 2, 165–193. (pp. 3, 8, 10)

RUTTEN, J. 2000. Universal coalgebra: a theory of systems. *Theoretical Computer Science 249*, 3–80. (p. 7)

SANGIOGI, D. AND WALKER, D. 2001. *The π-calculus: A Theory of Mobile Processes* (first ed.). Cambridge Press, Cambridge, UK. (p. 57)

SCHMIDT-SCHAUSS, M. AND SMOLKA, G. 1991. Attributive concept descriptions with complements. *Artificial intelligence 48*, 1, 1–26. (p. 14)

SCHRÖDER, L. 2007. A finite model construction for coalgebraic modal logic. *Journal of Logic and Algebraic Programming 73*, 1, 97–110. (p. 10)

SCHUBERT, C. 2007. Coalgebraic logic over analytic spaces. Technical report, Technical Report 170, Chair for Software Technology, Technische Universität Dortmund. (p. 12)

SCHUBERT, C. 2009. Coalgebraic logic over measurable spaces: behavioral and logical equivalence. *Electronic Notes in Theoretical Computer Science 257*, 71–85. (p. 12)

SWEEDLER, M. E. 1969. *Hopf algebras*, Volume 202. WA Benjamin New York. (p. 8)

URBAN, C., PITTS, A. M., AND GABBAY, M. J. 2004. Nominal unification. *Theoretical Computer Science 323*, 1, 473–497. (p. 22)

VENEMA, Y. 2004. Automata and fixed point logics: a coalgebraic perspective. *Electronic Notes in Theoretical Computer Science 204*, 2004. (pp. 13, 66)