Multivalued Coalgebraic Modal Logic for Multiagent Systems and Multiplayer Games

MSc Thesis (*Afstudeerscriptie*)

written by

Nima Motamed
(born June 3rd, 1998 in Amsterdam, The Netherlands)

under the supervision of **prof. dr. Alexander Kurz** (Chapman University) and **prof. dr. Yde Venema**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
August 30th, 2021	dr. Maria Aloni (*chair*)
	dr. Alexandru Baltag
	dr. Nick Bezhanishvili
	prof. dr. Alexander Kurz (*co-supervisor*)
	prof. dr. Yde Venema (*co-supervisor*)

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Multivalued Coalgebraic Modal Logic for Multiagent Systems and Multiplayer Games

Nima Motamed

Institute for Logic, Language and Computation

Abstract

This thesis investigates coalgebraic generalizations of two multiagent modal logics from the literature, in which truth values are identified with sets of agents. In the first logic, which is due to Melvin Fitting, the truth value of a formula is identified as the set of agents for whom the formula is true, while in the second logic, which is due to Loes Olde Loohuis and Yde Venema, the truth value of a formula is identified as the set of players that have a winning strategy at a corresponding position in a multiplayer evaluation game. For the first logic, we identify a new base category of interest, from which the generalization comes forth naturally using the theory of coalgebraic modal logic, and give proofs of adequacy and expressivity. For the second logic, we define multiplayer evaluation games in which play proceeds nondeterministically, and use the well-known fact that predicate liftings induce transformations from coalgebras to neighbourhood frames. Finally, we prove that fragments of our generalizations are equiexpressive, and show how they can naturally describe situations with multiple agents.

Acknowledgements

First and foremost, I wish to sincerely thank Yde Venema and Alexander Kurz for their excellent supervision and constant support during every phase of my work. My interest in coalgebra originated in a course Yde taught, and can be fully attributed to the passion and care he displayed when presenting the theory of coalgebra as a grand, unifying framework for state-based systems. During the writing of the thesis, Yde took a strict and critical, yet simultaneously incredibly comforting approach. There is no doubt in my mind that I would not have been able to make it through the thesis without this approach, and I find it difficult to express how incredibly grateful I am to him.

Yde's approach struck a good balance with that of Alexander, who continuously encouraged me to broaden my horizons. During the writing of the thesis, he wore many hats besides that of a supervisor: he was a mentor, teaching me the ways of a mathematician; a lecturer, spending entire mornings explaining material to me; and at times even an art connoisseur, introducing me to the beautiful paintings of Saenredam. I cannot thank him enough for the patience he had with me.

I wish to also extend my gratitude to Alexandru Baltag and Nick Bezhanishvili for agreeing to be on the Thesis Committee and being willing to read the thesis, despite the defence having been rescheduled to late August, and to Maria Aloni for agreeing to chair the defence. On a similar note, I wish to thank the entire ILLC staff — nowhere else could I have explored my curiosities this well, with top-class courses on anything ranging from computation to cognition. Especially noteworthy is Ulle Endriss, who was both a kind mentor and an excellent lecturer.

Furthermore, I want to thank Jesse Postema, Jan Schutte, Wilco Kruijer, and Loes Gennissen for reading through so many drafts of the thesis, and for being able to give such detailed and extensive feedback, even though they are all in fields vastly different to mine, and even though most of them are plenty busy writing their own theses. I also owe a great debt to Putri van der Linden, who took the time to draw the wonderful illustration accompanying this thesis, based on her interpretation of the concept of 'slicing' in multiagent systems, which appears in the thesis.

Finally, I wish to express my utmost gratitude to my family and friends. They have all been patient and supportive — not only during the writing of the thesis, but in fact throughout my life. I sadly do not have adequate space to thank all of them, so I will only restrict myself to three. I thank Sem, for being nothing less than a brother. I thank my mother, for her endless care and perseverance. And most of all, I thank my late father. You were a true friend. I dedicate this thesis to your memory.

Contents

Chapter 1

Introduction

In this thesis, we consider two multivalued modal logics with a common conceptual basis: truth values are identified with sets of agents. We study generalizations and properties of these two logics, and find ways to relate them. We now begin describing the two logics.

The first logic we are considering comes from Fitting (2009). This logic is based on the observation that logics taking truth values from Boolean algebras other than the two-element Boolean algebra are conspicuously absent from the literature. As Fitting points out, allowing truth values from arbitrary finite (or more generally, complete atomic) Boolean algebras provides an intuitive method to express situations involving multiple agents. The idea is simple: taking sets of agents to be truth values (and hence considering the Boolean algebra arising from the powerset of the set of agents), we can identify the *Boolean-valued* truth value of a formula with the set of agents for whom the formula is true.

Let us see an example of how this identification of Boolean-valued truth values with sets of agents (or in other words, with *multiagent-valued* truth values) can aid in expressing situations with multiple agents. Consider two agents Alice ($\mathfrak{a}$) and Bob ($\mathfrak{b}$). If we want to express using conventional propositional logic that both Alice and Bob are wearing coats, we will usually introduce propositional variables $c_{\mathfrak{a}}$ (meaning that Alice is wearing a coat) and $c_{\mathfrak{b}}$ (meaning that Bob is wearing a coat), and then assert that the formula $c_{\mathfrak{a}} \wedge c_{\mathfrak{b}}$ has truth value 'true'. Taking sets of agents to be truth values, we can instead simply introduce a single propositional variable c, and assert that it has truth value $\{\mathfrak{a}, \mathfrak{b}\}$. Similarly, if we want to express that Alice is wearing a hat while Bob is not, we will conventionally do so by introducing propositional variables $h_{\mathfrak{a}}$ and $h_{\mathfrak{b}}$, and then assert that the formula $h_{\mathfrak{a}} \wedge \neg h_{\mathfrak{b}}$ has truth value 'true'. In the Boolean-valued setting, we can instead simplify again by taking only one propositional variable h, and then asserting that h has truth value $\{\mathfrak{a}\}$.

Fitting (2009) shows how we can also define a Boolean-valued *modal* logic. This again allows us to simplify the way situations with multiple agents are expressed. Supposing that we use an epistemic interpretation of the $\Box$-modality of modal logic, we will conventionally model the knowledge of Alice and Bob through a multiagent Kripke model with separate accessibility relations for Alice and Bob, with corresponding modalities $\Box_{\mathfrak{a}}$ and $\Box_{\mathfrak{b}}$. The statement that both Alice and Bob know that they themselves are wearing coats, can then be expressed by asserting that the formula $\Box_{\mathfrak{a}} c_{\mathfrak{a}} \wedge \Box_{\mathfrak{b}} c_{\mathfrak{b}}$ has truth value 'true'. In the Boolean-valued setting, we instead need only consider a single $\Box$-modality, with which we can assert that the formula $\Box c$ has truth value $\{\mathfrak{a}, \mathfrak{b}\}$.

The second logic we are considering comes from Olde Loohuis and Venema (2010). Analogously to the reasoning behind the logic of Fitting (2009), this logic is based on the observation that applications of games in logic typically only concern the interaction between *two* players, with little attention for *multiplayer* games in logic. The logic is built on a multiplayer generalization of evaluation games for classical two-valued basic modal logic interpreted over Kripke models. Formulas in the logic represent positions in certain multiplayer evaluation games, and the truth value of a formula is identified with the set of *players* that have a winning strategy at the position represented by the formula.

While both logics we consider are generalizations of basic modal logic interpreted over Kripke models, one could argue that their essence lies in the way they identify truth values of formulas as certain sets of agents — as the set of agents for whom the formula is true (in the logic of Fitting (2009)), or as the set of players that have a winning strategy at the formula (in the logic of Olde Loohuis and Venema (2010)). Two questions naturally arise. First, to what extent are properties of the two logics determined by how they identify truth values as sets of agents? Second, can logics with these two different but similar identifications of truth values be related, through e.g. an equiexpressivity result? To answer these questions, we will define more general modal logics with the same set of truth values. Properties of the logics that depend only on the identifications of truth values should persist in the generalizations, thus answering the first question. And similarly, any relations between the generalizations are more indicative of relations between logics with the aforementioned truth values — as opposed to the original logics which are just variants of basic modal logic over Kripke models — therefore answering the second question.

We will naturally define these more general modal logics using the theory of coalgebra and coalgebraic modal logic. Coalgebras are objects that can be viewed as generalizations of many transition systems (such as Kripke frames) considered throughout mathematics and theoretical computer science, and the theory of (universal) coalgebra (see Rutten (2000)) allows one

to *parametrically* discuss properties of said transition systems. Similarly, the theory of coalgebraic modal logic (see e.g. Kurz and Leal (2012)) generalizes many modal logics used to describe transition systems, and can also allow us to parametrically discuss properties of said logics.

There are coalgebraic multivalued logics in the literature. First and foremost, in an unpublished note, Kurz (2017) has worked on a preliminary coalgebraic generalization of the logic of Fitting (2009), which in fact was the initial motivation behind this thesis, though we expand greatly upon the methods therein. Bílková and Dostál (2013, 2016) and Bílková, Kurz, et al. (2013) consider general coalgebraic multivalued logics where the truth values come from arbitrary commutative integral residuated lattices or commutative quantales. Though this approach could work for our generalization, since the powerset of the set of agents forms a commutative quantale, it still falls short, as their approach does not consider any multivalued structure in their coalgebras, which is required to capture the logic of Fitting (2009). On the other hand, Babus and Kurz (2016) consider coalgebraic multivalued logics that both take truth values from a commutative quantale, and enrich their base category with commutative quantales. But, there still remains a problem: coalgebras over categories enriched with the powerset of the set of agents do not capture the kind of structures considered by Fitting (2009), meaning their approach would still not suffice for our stated purpose.

We will instead approach the coalgebraic generalization of the multiagent-valued logic of Fitting (2009) by defining a new base category over which we take our coalgebras, with sets as its objects, and agent-indexed families of functions as its morphisms. By then applying standard concepts of coalgebraic modal logic to this new base category, we naturally obtain transition structures and logics generalizing the work of Fitting (2009). In fact, we will show that we can proceed fully analogously to Fitting (2009), and *lift* two-valued coalgebraic modal logics to multiagent-valued modal logics in such a manner that the truth value of a formula is precisely the set of agents for whom the formula is true. Properties like adequacy and expressivity with respect to bisimilarity also lift from the two-valued logic to the multiagent-valued logic.

Analogously to how the logic of Olde Loohuis and Venema (2010) is built on multiplayer evaluation games based on a generalization of two-valued basic modal logic, a generalization of their logic should contain a way to associate multiplayer evaluation games to two-valued coalgebraic modal logics. It is of interest to note that there exists work relating coalgebra and games, such as that by Cîrstea and Sadrzadeh (2008) and Venema (2006) on two-valued evaluation games for coalgebraic fixed point logics in the style of Moss (1999), and by Baltag (2000) and König, Mika-Michalski, and Schröder (2020) on (bi)simulation games for coalgebras. Our approach does not bear much similarity to that of the above authors, however, with

the focus solely on evaluation games for multiplayer logics in the style of Olde Loohuis and Venema (2010).

As we will show, we can associate multiplayer evaluation games to two-valued coalgebraic modal logics by reconsidering the definition of the underlying game structure. We will argue that it is necessary to consider *nondeterministic* games, in which play proceeds by nondeterministically selecting at each position a player that has to make a move. Using nondeterministic games, we will then define the evaluation games by making use of the well-known fact from coalgebraic modal logic that modalities in the coalgebraic setting (i.e. predicate liftings) induce transformations to neighbourhood frames. Together with an assumption of monotonicity of the original coalgebraic modal logic, these transformations will allow us to define the evaluation games based purely on a game-theoretic definition of coalgebraic modal logic over neighbourhood frames.

Given that we are considering two coalgebraic modal logics with the same space of truth values, both of which arise as generalizations of two-valued coalgebraic modal logics, we will also consider how the two logics differ in expressive power. We will show that if we make some assumptions on the coalgebra type functor, and if we take differences between the kinds of coalgebras the two logics are interpreted over into account, then a certain fragment of the multiplayer logic is equiexpressive to multiagent-valued logic. And furthermore, we will show that adding the full expressive power of the multiplayer logic to the multiagent-valued logic enables it to express more situations involving multiple agents than it is originally capable of.

1.1 Thesis Outline and Contributions

Concretely, the structure of the thesis, as well as the original contributions in the thesis, are as follows.

- In Chapter 2, we present the basic definitions and propositions of coalgebra and coalgebraic modal logic, as known from the literature.
- In Chapter 3, we start out by presenting the basic (noncoalgebraic) multiagent-valued logic of Fitting (2009), along with original proofs of adequacy and expressivity, as well as original definitions of bounded morphisms for the structures he considers. Afterwards, we begin defining the coalgebraic multiagent-valued logic by introducing and studying a novel base category based on the definition of the aforementioned bounded morphisms. We then show how applying the standard theory of two-valued coalgebraic modal logic to this base category produces a multiagent-valued logic generalizing that of Fitting (2009). Finally, we conclude by giving proofs of adequacy and

expressivity.

- In Chapter 4, we similarly start out by presenting the basic, noncoalgebraic multiplayer logic of Olde Loohuis and Venema (2010), and argue that the game structure it is based on will not allow a proper coalgebraic generalization. Having argued that, we give an original definition of a nondeterministic multiplayer game structure, compare different notions of winning strategies for these nondeterministic games, and finally show how the nondeterministic games relate to the original game structure through certain embeddings.

  We then use nondeterministic games to generalize the logic of Olde Loohuis and Venema (2010), still at the level of Kripke-like structures. We show how the resulting logic is more expressive than the original logic, allowing for connectives that were undefinable in the original. Additionally, we also show that the logic is adequate with respect to bisimilarity.

  Finally, we then define an original coalgebraic generalization of the logic by using (mostly) well-known facts about modalities and neighbourhood frames in the coalgebraic setting, showing that we can lift monotone two-valued coalgebraic modal logics to coalgebraic multiplayer logics.

- In Chapter 5, we relate the (coalgebraic generalizations of the) two logics in two ways. We start by giving an original proof that fragments of the logics are equiexpressive under some assumptions on the coalgebra type functor. Afterwards, we prove that multiagent-valued logic has less (informal) expressive power than expected when it comes to expressing situations with multiple agents, and show how adding features of the multiplayer logic can mend this lack of expressive power.

- Finally, we conclude the thesis in Chapter 6 with directions for further work.

Chapter 2

Preliminaries

In this chapter, we will give (most of) the relevant definitions and theorems required to understand the rest of the thesis. These will cover coalgebras (Section 2.1), and coalgebraic modal logic (Section 2.2). We will presume familiarity with the basic notions of modal logic, and will therefore not treat these here, instead referring readers to Blackburn, de Rijke, and Venema (2002). We will also presume basic understanding of game-theoretic semantics for classical two-valued logic, as treated by e.g. Hintikka (1983). Finally, the theories of coalgebra and coalgebraic modal logic make heavy use of category theory, which we assume readers are familiar with, though we will only make use of the basic definitions and properties of categories, functors, natural transformations, adjunctions, (co)limits, and subobject classifiers. No knowledge of any advanced theorems is assumed.

2.1 Coalgebra

For a comprehensive introduction of the theory of coalgebra, we refer readers to Jacobs (2016) and Rutten (2000, 2019).

Definition 2.1.1. Given a category $\mathbf{C}$ and an endofunctor $\mathcal{T} : \mathbf{C} \to \mathbf{C}$, a *$\mathcal{T}$-coalgebra* is a pair $\mathbb{S} = \langle S, \sigma \rangle$, where S is an object of $\mathbf{C}$, and $\sigma : S \to \mathcal{T}S$ is a $\mathbf{C}$-morphism. The object S is referred to as the *carrier object* of $\mathbb{S}$, while σ is referred to as the *coalgebra map* of $\mathbb{S}$. The functor $\mathcal{T}$ is referred to as the (*coalgebra*) *type* of $\mathcal{T}$-coalgebras. A $\mathbf{C}$-morphism $f : S \to S'$ is a *$\mathcal{T}$-coalgebra morphism* $f : \mathbb{S} \to \mathbb{S}'$ between $\mathcal{T}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$, if the diagram

$$\begin{array}{ccc} S & \xrightarrow{f} & S' \\ {\scriptstyle\sigma}\downarrow & & \downarrow{\scriptstyle\sigma'} \\ \mathcal{T}S & \xrightarrow[\mathcal{T}f]{} & \mathcal{T}S' \end{array}$$

commutes. The category of $\mathcal{T}$-coalgebras and $\mathcal{T}$-coalgebra morphisms is denoted by $\mathbf{Coalg}_{\mathbf{C}}(\mathcal{T})$.

Dually, a *$\mathcal{T}$-algebra* is a pair $\mathbb{A} = \langle A, \alpha \rangle$, where A is an object of $\mathbf{C}$, and $\alpha : \mathcal{T}A \to A$ is a $\mathbf{C}$-morphism. A $\mathbf{C}$-morphism $f : A \to A'$ is a *$\mathcal{T}$-algebra morphism* $f : \mathbb{A} \to \mathbb{A}'$ between $\mathcal{T}$-algebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{A}' = \langle A', \alpha' \rangle$, if the diagram

$$\begin{array}{ccc} \mathcal{T}A & \xrightarrow{\mathcal{T}f} & \mathcal{T}A' \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\alpha'} \\ A & \xrightarrow[f]{} & A' \end{array}$$

commutes. The category of $\mathcal{T}$-algebras and $\mathcal{T}$-algebra morphisms is denoted by $\mathbf{Alg}_{\mathbf{C}}(\mathcal{T})$. ◁

For now, we will mainly be considering coalgebras over the category **Set** of sets and functions. These can be seen as general state-based transition systems, as we will shortly give examples of. With this in mind, we will refer to carrier sets of coalgebras over **Set** as *state spaces*, and to their coalgebra maps as *transition maps*. Elements of a state space are referred to as *states*. We will refer to endofunctors $\mathcal{T} : \mathbf{Set} \to \mathbf{Set}$ as *set functors*. Elements of $\mathcal{T}S$ for a set S are often referred to as *unfoldings* of S.

Example 2.1.2. (i) Consider the powerset functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$, sending sets to their powerset, and functions $f : X \to Y$ to the function $\mathcal{P}f$ sending subsets $U \subseteq X$ to the f-image $(\mathcal{P}f)(U) := f[U]$. We have that $\mathcal{P}$-coalgebras are precisely Kripke frames: $\mathcal{P}$-coalgebras $\langle S, \sigma \rangle$ are Kripke frames $\langle S, R \rangle$ in which $R[s] = \sigma(s)$ for states $s \in S$ (where $R[s] = \{t \in S \,;\, sRt\}$). Analogously, $\mathcal{P}$-coalgebra morphisms are precisely bounded morphisms between Kripke frames.

(ii) Considering the functor $\mathcal{A}ut_C$ (where C is a fixed set) defined for a set X as $\mathcal{A}ut_C X := 2 \times S^C$, we have that $\mathcal{A}ut_C$-coalgebras are precisely deterministic automata over an alphabet C. A state s is final iff the first component of $\sigma(s)$ (i.e. $(\mathsf{proj}_2 \circ \sigma)(s)$) is 1, and the c-transition of s for a letter $c \in C$ is the state $(\mathsf{proj}_{S^C} \circ \sigma)(s)(c)$.

(iii) Considering the functor $\mathcal{S}t_C$ defined for a set X as $\mathcal{S}t_C X := C \times S$, we have that $\mathcal{S}t_C$-coalgebras are so-called stream systems over C, in which every state s generates an infinite stream $\boldsymbol{c} \in C^\omega$ with c_0 defined as the first component of $\sigma(s)$ and c_{i+1} defined as the first component of $\sigma(s')$ where s' is the second component of $\sigma(s)$.

(iv) Considering the functor $\mathcal{D}ist$ sending sets X to the set of discrete probability distributions over X, we have that $\mathcal{D}ist$-coalgebras are precisely discrete-time Markov chains over X, with $\sigma(s)$ being the probability distribution over 'next' states from s.

(v) Considering the functor $\mathscr{Tree}$ defined as $\mathscr{Tree}X := (X \times X) + 1$, we have that $\mathscr{Tree}$-coalgebras represent (potentially infinite) binary trees, with $\sigma(s) = 0$ representing that the state s is a leaf of the tree, while $\sigma(s) = \langle t, u \rangle$ represents that the states t and u are the children of s. ◁

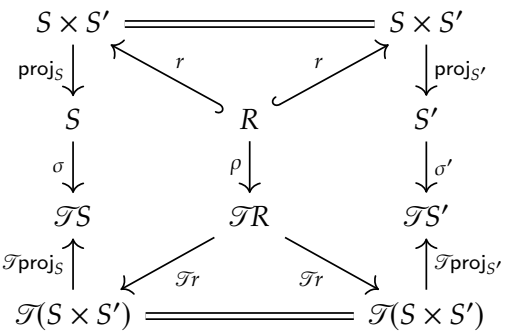
The theory of coalgebras allows us to treat all the above examples uniformly with respect to the coalgebra type. It gives general definitions and proofs applicable to all coalgebras. For example, in the study of a particular class of transition systems, one is often interested in determining whether states exhibit the same 'behaviour'. Using coalgebras, one could say that the behaviour of a state is precisely that which is preserved by a coalgebra morphism. This allows us to give the following general definition of *behavioural equivalence*.

Definition 2.1.3. Given a set functor $\mathscr{T}$, together with $\mathscr{T}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$, we say states $s \in S$ and $s' \in S'$ are *behaviourally equivalent* (and we write $\mathbb{S}, s \simeq \mathbb{S}', s'$) if there is some $\mathscr{T}$-coalgebra $\mathbb{X}$ with $\mathscr{T}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$ such that $f(s) = f'(s')$. ◁

In the study of particular classes of transition systems, behavioural equivalence is often captured by a suitable notion of *bisimilarity*. An elegant definition of coalgebraic bisimilarity using coalgebra morphisms is given by Aczel and Mendler (1989), which is the one we will use, though there are several other possible definitions (see Staton (2009) for an overview). Though originally defined for the category of sets, we will give a slightly more general definition that only requires the category to have binary products, but is still sufficient for our purposes.

Definition 2.1.4. Given a category $\mathbf{C}$ with binary products, an object R is an *internal binary relation* between objects X and Y, if it is (the domain of) a subobject $r : R \hookrightarrow X \times Y$.

Given an endofunctor $\mathscr{T} : \mathbf{C} \to \mathbf{C}$ and $\mathscr{T}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$, an internal binary relation R between S and S' is a (*coalgebraic*) *bisimulation* between $\mathbb{S}$ and $\mathbb{S}'$ (and we write $R : \mathbb{S} \leftrightarrows \mathbb{S}'$) if there exists a $\mathbf{C}$-morphism $\rho : R \to \mathscr{T}R$ (making $\mathbb{R} := \langle R, \rho \rangle$ a $\mathscr{T}$-coalgebra) such that $\mathsf{proj}_S \circ r$ and $\mathsf{proj}_{S'} \circ r$ become $\mathscr{T}$-coalgebra morphisms from $\mathbb{R}$ to $\mathbb{S}$ and $\mathbb{S}'$, respectively. That is, the diagram



needs to commute.

Assuming $\mathbf{C} = \mathbf{Set}$, then internal binary relations are just binary relations. If there exists $B \subseteq S \times S'$ such that $B : \mathbb{S} \leftrightarrows \mathbb{S}'$ with sBs' for states $s \in S$ and $s' \in S'$, the states s and s' are (*coalgebraically*) *bisimilar*, and we write $\mathbb{S}, s \leftrightarrows \mathbb{S}', s'$. ◁

Remark 2.1.5. We could have given an even more general definition in which binary products need not exist, by defining internal binary relations to be objects R with jointly monic morphisms $p_X : R \to X$ and $p_Y : R \to Y$. But since we will only be considering categories with binary products throughout this thesis, the definition given above will suffice. ◁

Bisimilarity is a specific instance of behavioural equivalence.

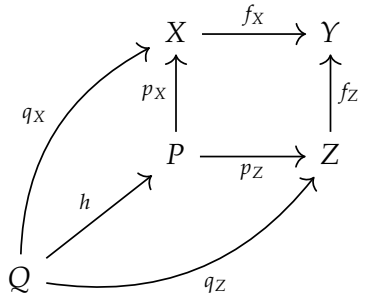
Proposition 2.1.6. *Let $\mathcal{T}$ be a set functor, and take $\mathcal{T}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$. It holds that*

$$\mathbb{S}, s \leftrightarrows \mathbb{S}', s' \text{ implies } \mathbb{S}, s \simeq \mathbb{S}', s'$$

for all $s \in S$ and $s' \in S'$.

Using an example by Aczel and Mendler (1989, after Proposition 6.2), it can be verified that behavioural equivalence and bisimilarity do not generally coincide. They coincide in specific situations: of particular interest, is the situation where the coalgebra type $\mathcal{T}$ *preserves weak pullbacks*.

Definition 2.1.7. Given a category $\mathbf{C}$, a *weak pullback* of $\mathbf{C}$-morphisms $f_X : X \to Y$ and $f_Z : Z \to Y$ is a triple $\langle P, p_X, p_Z \rangle$ where the $\mathbf{C}$-morphisms $p_X : P \to X$ and $p_Z : P \to Z$ satisfy $f_X \circ p_X = f_Z \circ p_Z$, such that for all $\mathbf{C}$-morphisms $q_X : Q \to X$ and $q_Z : Q \to Z$ satisfying $f_X \circ q_X = f_Z \circ q_Z$, there exists a (not necessarily unique) $\mathbf{C}$-morphism $h : Q \to P$ making the diagram



commute. An endofunctor $\mathcal{T} : \mathbf{C} \to \mathbf{C}$ is said to *preserve weak pullbacks* if given such a weak pullback $\langle P, p_X, p_Z \rangle$ of f_X and f_Z, it holds that the triple $\langle \mathcal{T}P, \mathcal{T}p_X, \mathcal{T}p_Z \rangle$ is a weak pullback of $\mathcal{T}f_X$ and $\mathcal{T}f_Z$. ◁

Proposition 2.1.8. *Let $\mathcal{T}$ be a set functor that preserves weak pullbacks, and take $\mathcal{T}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$. It holds that*

$$\mathbb{S}, s \simeq \mathbb{S}', s' \text{ implies } \mathbb{S}, s \leftrightarrows \mathbb{S}', s'$$

for all $s \in S$ and $s' \in S'$.

2.2 Coalgebraic Modal Logic

Though we will give some motivating explanations for the definitions of coalgebraic modal logic, this will be far from an extensive treatment of the matter. For a good general conceptual introduction to coalgebraic modal logic, we refer readers to Cîrstea, Kurz, et al. (2009), while more technically-minded readers should also consider Bonsangue and Kurz (2005), Kupke, Kurz, and Pattinson (2004), Kupke and Pattinson (2011), and Kurz and Leal (2012). The parts of the coalgebraic modal logic appearing in this thesis require knowledge of algebraic logic and Stone duality, which we assume the reader is familiar with — see Davey and Priestley (2002).

To define a coalgebraic modal logic, we start with a category **Space** of 'state spaces' over which we will be taking coalgebras, and a category **Algebra** of 'algebras' (not to be confused with a category of $\mathcal{T}$-algebras), encoding a propositional logic with each object in **Algebra** generally being viewed as consisting of formulas/predicates. We will informally assume both categories to be 'set-like', to aid in giving intuitive descriptions. [1]

We require there to be a dual adjunction between **Space** and **Algebra**, which we refer to as a *logical connection* following Pavlovic, Mislove, and Worrell (2006).

Definition 2.2.1. An adjunction $\mathcal{Th} \dashv \mathcal{Pred}$ consisting of functors $\mathcal{Th} : \mathbf{Algebra} \to \mathbf{Space}^{\mathrm{op}}$ and $\mathcal{Pred} : \mathbf{Space}^{\mathrm{op}} \to \mathbf{Algebra}$ is a *logical connection*. The functor $\mathcal{Th}$ is called the *theory functor*, and the functor $\mathcal{Pred}$ is called the *predicate functor*. ◁

As implied by the names, we interpret $\mathcal{Pred}$ as sending spaces X in **Space** to the algebra of the predicates over X, while $\mathcal{Th}$ sends algebras A in **Algebra** to the space of logically consistent theories of A (as also explained by Bezhanishvili et al. (2020)). By requiring $\mathcal{Th} \dashv \mathcal{Pred}$ to be an adjunction, we enforce that the notions of predicates and theories are well-behaved in the sense also described by e.g. Klin (2007). Intuitively, we can describe the

[1] Though not relevant for our presentation of the material, the category **Algebra** is usually assumed to be an *algebraic* category, in the sense that it comes equipped with a forgetful functor $\mathcal{Forg} : \mathbf{Algebra} \to \mathbf{Set}$ with a left adjoint $\mathcal{Free} \dashv \mathcal{Forg}$, referred to as the *free algebra functor* of **Algebra**.

semantics encoded by the logical connection equivalently as either transforming (using **Algebra**-morphisms $[\![-]\!] : A \to \mathscr{P}red\, X$) propositional logics (consisting of formulas) to state spaces of states at which the formulas of the logic are satisfied, or as transforming (using **Space**-morphisms $[\![-]\!]^\flat : X \to \mathscr{T}h A$) state spaces to the object of formulas 'satisfied' by its states.

Having defined the logical connection, we can then move on to actually define a coalgebraic modal logic. We first specify the syntax of the logic — that is, its *modal similarity type*. The modal similarity type is encoded in an endofunctor $\mathscr{S}im : \mathbf{Algebra} \to \mathbf{Algebra}$, intuitively associating to each algebra a single layer of modal operators.[2] The resulting category of $\mathscr{S}im$-algebras will then encode the modal logics syntactically.

Next, to specify the semantics given a coalgebra type $\mathscr{T} : \mathbf{Space} \to \mathbf{Space}$, we proceed by defining the *one-step semantics*, which is a natural transformation

$$\mathsf{one} : \mathscr{S}im \circ \mathscr{P}red \Rightarrow \mathscr{P}red \circ \mathscr{T},$$

intuitively defining the semantics of the modal operators in terms of unfoldings. The one-step semantics allows us to functorially associate to each $\mathscr{T}$-coalgebra a $\mathscr{S}im$-algebra consisting of its predicates. This process is referred to as *algebraification*.

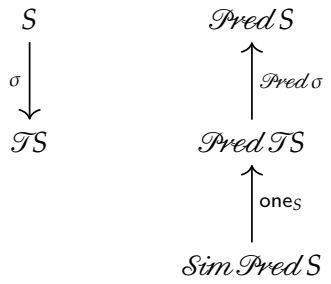
Definition 2.2.2. The *algebraification functor*

$$\mathscr{A}lg : \mathbf{Coalg}_{\mathbf{Space}}(\mathscr{T})^{\mathrm{op}} \to \mathbf{Alg}_{\mathbf{Algebra}}(\mathscr{S}im)$$

sends $\mathscr{T}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ to the $\mathscr{S}im$-algebra

$$\mathscr{A}lg\mathbb{S} := \langle \mathscr{P}red\, S, \mathscr{P}red\, \sigma \circ \mathsf{one}_S \rangle\,,$$

as in the following diagram:



The algebraification functor sends $\mathscr{T}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{S}'$ to $\mathscr{S}im$-algebra morphisms $\mathscr{A}lg f : \mathscr{A}lg\mathbb{S}' \to \mathscr{A}lg\mathbb{S}$ defined as $\mathscr{A}lg f := \mathscr{P}red\, f$. ◁

[2] Though not required for our purposes, the endofunctor $\mathscr{S}im$ can encode not only a modal similarity type, but also equations such as $\Box(p \wedge q) = \Box p \wedge \Box q$ for the usual $\Box$-operator of modal logic. We refer to Bonsangue and Kurz (2006) for a thorough treatment of this.

It follows from the naturality of **one** that $\mathscr{Alg}$ is indeed a well-defined functor.

Assuming that there exists an initial $\mathscr{Sim}$-algebra $\mathbb{Lang} := \langle \mathsf{Lang}, \lambda \rangle$, we refer to Lang as being the algebra of *formulas* of $\mathscr{Sim}$. The initiality of $\mathbb{Lang}$ then gives rise to the semantics of our coalgebraic modal logic.

Definition 2.2.3. Given a $\mathcal{T}$-coalgebra $\mathbb{S} = \langle S, \sigma \rangle$, the *semantic mapping* for $\mathbb{S}$ is the unique $\mathscr{Sim}$-algebra morphism $[\![-]\!]_{\mathbb{S}} : \mathbb{Lang} \to \mathscr{Alg}\mathbb{S}$, which is an **Algebra**-morphism $[\![-]\!]_{\mathbb{S}} : \mathsf{Lang} \to \mathscr{Pred}S$ from the algebra of formulas of $\mathscr{Sim}$ to the algebra of predicates over S. ◁

It will be of value to us to look at this from a concrete perspective. Consider **Space** = **Set** and **Algebra** = **BA** (i.e. the category of Boolean algebras and Boolean algebra homomorphisms). The logical connection in this case is the well-known dual adjunction between **Set** and **BA**. This dual adjunction arises by 'homming into' the dualizing object given by the two-element set 2 and the two-element Boolean algebra $\mathbf{2}$. That is, $\mathscr{Pred}X$ is the set of functions from X to 2 (which is isomorphic to $\mathscr{P}X$), with obvious Boolean algebra structure. And $\mathscr{Th}\mathbb{B}$ is the set of Boolean algebra homomorphisms from $\mathbb{B}$ to $\mathbf{2}$, or in other words, the set of ultrafilters of $\mathbb{B}$.

To define the functor $\mathscr{Sim}$ concretely given this logical connection, we start from a more conventional conception of a modal similarity type as a pair $\mathbb{Sim} = \langle \mathsf{Sym}, \mathsf{ar} \rangle$ consisting of a set Sym of modality symbols, and an arity function $\mathsf{ar} : \mathsf{Sym} \to \omega$. Taking a set Prop of propositional variables, we can then define a generator functor $\mathscr{Gen} : \mathbf{BA} \to \mathbf{Set}$ sending Boolean algebras $\mathbb{B}$ with domain B to the set

$$\mathscr{Gen}\mathbb{B} := \left\{ \heartsuit(\boldsymbol{b}) \,;\, \heartsuit \in \mathsf{Sym}, \boldsymbol{b} \in B^{\mathsf{ar}\heartsuit} \right\} + \mathsf{Prop},$$

and which is defined on Boolean algebra homomorphisms in the obvious way. Then using the free Boolean algebra functor $\mathscr{Free}_{\mathbf{BA}} : \mathbf{Set} \to \mathbf{BA}$, we can define $\mathscr{Sim} := \mathscr{Free}_{\mathbf{BA}} \circ \mathscr{Gen}$. It is then easily verified (using e.g. the initial sequence of $\mathscr{Sim}$) that there exists an initial $\mathscr{Sim}$-algebra $\mathbb{Lang}$, with carrier object Lang being a Boolean algebra with a domain that we can consider to consist of formulas[3] φ defined inductively as

$$\varphi ::= p \mid \top \mid \bot \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\neg\varphi) \mid (\heartsuit(\underbrace{\varphi, \ldots, \varphi}_{\mathsf{ar}\heartsuit \text{ times}})),$$

where $p \in \mathsf{Prop}$ and $\heartsuit \in \mathsf{Sym}$. The Boolean algebra structure and the coalgebra map λ are defined in the straightforward manner.

To specify the one-step semantics, we make use of collections of *predicate liftings*.

[3] Technically, the Boolean algebra is a Lindenbaum-Tarski algebra, with a domain that can more accurately be considered to consist of equivalence classes of formulas, with respect to the relation of logical equivalence under the laws of Boolean algebra. This holds by definition of the free Boolean algebras used in the definition of $\mathscr{Sim}$.

Definition 2.2.4. A (two-valued) *n-ary predicate $\mathcal{T}$-lifting* (for $n < \omega$) is a natural transformation

$$\mathbf{lift} : (\mathscr{Forg} \circ \mathscr{Pred})^n \Rightarrow \mathscr{Forg} \circ \mathscr{Pred} \circ \mathcal{T},$$

where $\mathscr{Forg}$ is the forgetful functor from **BA** to **Set**. In other words, n-ary predicate $\mathcal{T}$-liftings are natural transformations $\mathbf{lift} : (2^-)^n \Rightarrow 2^{\mathcal{T}(-)}$. $\lhd$

Since our logic includes propositional variables, we will consider the functor $\mathcal{T}_{\mathsf{Prop}}$, defined as $\mathcal{T}_{\mathsf{Prop}}X := \mathcal{T}X \times \mathcal{P}\mathsf{Prop}$, of which the coalgebras are referred to as *$\mathcal{T}$-models*, and which we will denote as triples $\langle S, \sigma, \mathsf{col}\rangle$, where $\sigma : S \to \mathcal{T}S$ and $\mathsf{col} : S \to \mathcal{P}\mathsf{Prop}$.

Given a Sym-indexed collection $\mathbf{Lift} = \left\langle \mathbf{lift}^{\heartsuit} \right\rangle_{\heartsuit \in \mathsf{Sym}}$ of $(\mathsf{ar}\heartsuit)$-ary predicate $\mathcal{T}$-liftings (not $\mathcal{T}_{\mathsf{Prop}}$), we can define the one-step semantics for a set S as the unique Boolean algebra homomorphism induced, using the universal property of the free Boolean algebra $\mathscr{Sim}\mathscr{Pred}S$, by the function

$$\mathsf{one}'_S : \mathscr{Gen}\,\mathscr{Pred}S \to \mathscr{Forg}\,\mathscr{Pred}\,\mathcal{T}_{\mathsf{Prop}}S$$

defined as

$$\begin{aligned} \mathsf{one}'_S(\heartsuit(\boldsymbol{q})) &:= \left\{\langle U, P\rangle \in \mathcal{T}_{\mathsf{Prop}}S \,;\, U \in \mathsf{lift}^{\heartsuit}_S(\boldsymbol{q})\right\} \text{ and} \\ \mathsf{one}'_S(p) &:= \left\{\langle U, P\rangle \in \mathcal{T}_{\mathsf{Prop}}S \,;\, p \in P\right\}, \end{aligned}$$

where $p \in \mathsf{Prop}$, $\heartsuit \in \mathsf{Sym}$ and $\boldsymbol{q} \in (2^S)^n$.

So specifying a concrete modal similarity type $\mathbb{Sim}$ and collection **Lift** of predicate liftings suffices to define the other components. Because of this, we refer to pairs $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift}\rangle$ as (*two-valued*) *coalgebraic modal logics over* $\mathcal{T}$. Writing out the semantics $[\![-]\!]^{\mathbb{Log}}_{\mathbb{S}} : \mathsf{Lang} \to \mathscr{Pred}S$ (where the domain of $\mathscr{Pred}S$ is $2^S \cong \mathcal{P}S$) that we get through algebraification for a $\mathcal{T}$-model $\mathbb{S}$, we find that we can give the following compositional characterization.

Proposition 2.2.5. *Given a $\mathcal{T}$-model $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$, it holds that*

$$\begin{aligned} [\![p]\!]^{\mathbb{Log}}_{\mathbb{S}} &= \widehat{\mathsf{col}}\,(p), \\ [\![\top]\!]^{\mathbb{Log}}_{\mathbb{S}} &= S, \\ [\![\bot]\!]^{\mathbb{Log}}_{\mathbb{S}} &= \varnothing, \\ [\![\varphi \wedge \psi]\!]^{\mathbb{Log}}_{\mathbb{S}} &= [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}} \cap [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}, \\ [\![\varphi \vee \psi]\!]^{\mathbb{Log}}_{\mathbb{S}} &= [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}} \cup [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}, \\ [\![\neg\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}} &= S \smallsetminus [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}, \text{ and} \\ [\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})]\!]^{\mathbb{Log}}_{\mathbb{S}} &= \sigma^{-1}[\mathsf{lift}^{\heartsuit}_S([\![\varphi_1]\!]^{\mathbb{Log}}_{\mathbb{S}}, \ldots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]^{\mathbb{Log}}_{\mathbb{S}})]. \end{aligned}$$

Note that we use the transpose $\widehat{\mathsf{col}} : \mathsf{Prop} \to \mathscr{P}S$ as defined in Section 2.3.

Using the semantics, we can define logical equivalence on states.

Definition 2.2.6. Given $\mathscr{T}$-models $\mathbb{S}$ and $\mathbb{S}'$, we say states s in $\mathbb{S}$ and s' in $\mathbb{S}'$ are $\mathbb{L}\mathrm{og}$-*equivalent* (and we write $\mathbb{S}, s \equiv^{\mathbb{L}\mathrm{og}} \mathbb{S}', s'$) if

$$s \in [\![\varphi]\!]^{\mathbb{L}\mathrm{og}}_{\mathbb{S}} \text{ iff } s' \in [\![\varphi]\!]^{\mathbb{L}\mathrm{og}}_{\mathbb{S}'}$$

for all formulas $\varphi \in \mathsf{Lang}$. ◁

Considering the notions of behavioural equivalence, it is desirable for a coalgebraic modal logic to be *adequate* in the sense that behaviourally equivalent states are logically equivalent, and *expressive* in the sense that logically equivalent states are behaviourally equivalent. Adequacy is fortunately guaranteed.

Proposition 2.2.7. *Let $\mathbb{S}$ and $\mathbb{S}'$ be $\mathscr{T}$-models. Then it holds that*

$$\mathbb{S}, s \simeq \mathbb{S}', s' \text{ implies } \mathbb{S}, s \equiv^{\mathbb{L}\mathrm{og}} \mathbb{S}', s'$$

for all states s in $\mathbb{S}$ and s' in $\mathbb{S}'$.

For expressiveness, two assumptions are required. First, the coalgebra type $\mathscr{T}$ functor must be finitary. While this is defined as requiring that $\mathscr{T}$ preserves ω-filtered colimits, we can give a simpler characterization, as shown by Adámek and Trnková (1990): $\mathscr{T}$ is finitary iff for every set X and unfolding $U \in \mathscr{T}X$, there is some finite $Y \subseteq X$ and unfolding $U' \in \mathscr{T}Y$ such that $U = (\mathscr{T}\mathsf{incl})(U')$, where $\mathsf{incl} : Y \hookrightarrow X$ is the inclusion function from Y to X.

The second assumption is that **Lift** must be *separating*.

Definition 2.2.8. We say that **Lift** is *separating for $\mathscr{T}$* if for all sets X and $U, U' \in \mathscr{T}X$ with $U \neq U'$, there exists some modality symbol $\heartsuit \in \mathsf{Sym}$ with $V_1, \ldots, V_{\mathsf{ar}\heartsuit} \in \mathscr{P}X$ such that it holds that either $U \in \mathsf{lift}^{\heartsuit}_X(V_1, \ldots, V_{\mathsf{ar}\heartsuit})$ or $U' \in \mathsf{lift}^{\heartsuit}_X(V_1, \ldots, V_{\mathsf{ar}\heartsuit})$, but not both. ◁

Proposition 2.2.9. *Let $\mathscr{T}$ be a finitary set functor, and let $\mathbb{L}\mathrm{og} = \langle \mathbb{S}\mathrm{im}, \mathbf{Lift} \rangle$ be a coalgebraic modal logic over $\mathscr{T}$ such that* **Lift** *is separating for $\mathscr{T}$. Then for all $\mathscr{T}$-models $\mathbb{S}$ and $\mathbb{S}'$, it holds that*

$$\mathbb{S}, s \equiv^{\mathbb{L}\mathrm{og}} \mathbb{S}', s' \text{ implies } \mathbb{S}, s \simeq \mathbb{S}', s'$$

for all states s in $\mathbb{S}$ and s' in $\mathbb{S}'$.

2.3 Notation

We will use a lot of typographical conventions when writing mathematics. Though we will define special notation whenever it is introduced, we make note of the following conventions which will be used throughout the entire thesis.

- Sets and relations (and objects of 'general' categories) are denoted using uppercase italics (e.g. X and R), though 'special' distinguished sets and relations will be denoted by uppercase sans-serif letters (e.g. Fin and Adm).
- Functions (and morphisms of 'general' categories) are denoted using lowercase italics (e.g. f and g), though 'special' distinguished functions will be denoted by lowercase sans-serif letters (e.g. turn and strat).
- Sets of agents are denoted using uppercase Gothic letters (e.g. $\mathfrak{A}$ and $\mathfrak{B}$), while individual agents are denoted using lowercase Gothic letters (e.g. $\mathfrak{a}$, $\mathfrak{b}$ and $\mathfrak{c}$).
- Categories are denoted using upright, serif bold letters (e.g. $\mathbf{C}$ and $\mathbf{Set}$).
- Functors are denoted using calligraphic letters (e.g. $\mathscr{P}$, $\mathscr{Forg}$ and $\mathscr{Dist}$).
- Tuples of objects are enclosed using angular brackets (e.g. $\langle x, y \rangle$).
- Sequences (or more generally, 'homogeneous' tuples of objects of the same type) of objects are denoted using bold (italic or sans-serif) letters (e.g. $\boldsymbol{n}$ and $\mathbf{Lift}$). This includes natural transformations.
- Structures (which are generally 'heterogeneous' tuples of objects of potentially different types) are denoted using blackboard bold letters (e.g. $\mathbb{M}$, $\mathbb{S}$ and $\mathbb{Log}$).
- Transition maps of coalgebras are denoted using Greek letters e.g. (σ, ξ and ε). Greek letters φ, ψ and χ are reserved for formulas of logics.

We will also repeatedly use the following pieces of notation. Given a function $f : X \to Y^Z$ (where Y^Z is the set of functions from Z to Y), its *transpose* is the function $\widehat{f} : Z \to Y^X$ defined by 'swapping' the order of the first two arguments, i.e. $\widehat{f}(z) := x \in X \mapsto f(x)(z)$. The canonical projections of a product $\prod_{i \in I} X_i$ are denoted proj_i, and the canonical injections of a coproduct $\sum_{i \in I} X_i$ are denoted inj_i.

Finally, we will generally treat natural numbers n as ordinal numbers in the set-theoretic fashion, with $n = \{0, 1, \ldots, n-1\}$, and will therefore denote the set of all natural numbers by ω.

Chapter 3

Multiagent-Valued Logic

In this chapter, we will define and generalize multiagent-valued logic as treated by Fitting (2009).[1] In Section 3.1, we give the basic definitions and properties of the structures and logic defined in *How True*. Afterwards, we generalize these structures and logics to the coalgebraic setting in Section 3.2.

3.1 Boolean-Valued Basic Modal Logic

Throughout this thesis, we will usually fix a countable set Prop of propositional variables, along with a potentially infinite set $\mathfrak{A}$ of agents. Whenever we assume $\mathfrak{A}$ to be finite, we will make said assumption explicit.

As stated in Chapter 1, we will consider logics of which the powerset $\mathscr{P}\mathfrak{A}$ serves as the space of truth values. This powerset forms a Boolean algebra $\mathbb{P}_{\mathfrak{A}}$ with the usual operations of intersection, union, and complementation. As it is known (Tarski 1935) that all complete atomic Boolean algebras are isomorphic to such a powerset Boolean algebra (which are themselves complete and atomic), our logics can be viewed as taking truth values from arbitrary complete atomic Boolean algebras. With this in mind, we will refer to these logics as being *Boolean-valued*, while we still explicitly work with the interpretation that the truth values correspond to sets of agents from $\mathfrak{A}$.

3.1.1 Syntax, Models and Slices

The basic language Lang we consider is precisely that of basic modal logic interpreted over Kripke models.

Definition 3.1.1. The *Boolean-valued basic modal language* Lang is inductively defined as

$$\mathsf{Lang} \ni \varphi ::= p \mid (\varphi \vee \varphi) \mid (\neg\varphi) \mid (\Diamond\varphi),$$

[1] Hereafter, this article is referred to as *How True*.

where $p \in \mathsf{Prop}$. $\triangleleft$

We diverge slightly from the original definitions in *How True* by taking only disjunction, negation, and the modality $\Diamond$ to be the connectives in our language. This is done to facilitate comparisons with our work in Chapter 4. The other connectives from *How True* are defined as abbreviations in the usual manner, i.e.

$$\begin{aligned} \varphi \wedge \psi &:= \neg(\neg\varphi \vee \neg\psi), \\ \varphi \to \psi &:= \neg\varphi \vee \psi, \\ \Box\varphi &:= \neg\Diamond\neg\varphi. \end{aligned}$$

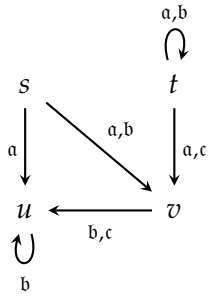
Things become interesting when we consider the semantics. Instead of interpreting Lang over 'ordinary' Kripke models, *How True* instead interprets it over Kripke models in which the accessibility relation itself is also in a sense Boolean-valued. These are instances of so-called *Boolean-valued (binary) relations* between sets X and Y, which are defined to be functions $R : X \times Y \to \mathscr{P}\mathfrak{A}$. We will however diverge from the original terminology, and instead refer to these as *agent-indexed relations*. The reason for this will first be expanded upon when we introduce the Slicing Slogan later in this section, before being made more rigorous when we discuss the coalgebraic generalization of the logic in Section 3.2.

Definition 3.1.2. An *agent-indexed Kripke frame* $\mathbb{F}$ is a pair $\mathbb{F} = \langle S, R\rangle$, where S is a set of *states*, and $R : S \times S \to \mathscr{P}\mathfrak{A}$ is an *agent-indexed relation* referred to as an (agent-indexed) *accessibility relation*. An *agent-indexed Kripke model* $\mathbb{M}$ is a pair $\langle \mathbb{F}, \mathsf{col}\rangle$, where $\mathbb{F} = \langle S, R\rangle$ is an agent-indexed Kripke frame, and $\mathsf{col} : S \to (\mathscr{P}\mathfrak{A})^{\mathsf{Prop}}$ is an *agent-indexed colouring*. We often unfold the inner pair in the definition of agent-indexed Kripke models, writing them as triples $\mathbb{M} = \langle S, R, \mathsf{col}\rangle$. $\triangleleft$

Again we diverge from *How True* in that we work with *colourings* $\mathsf{col} : S \to (\mathscr{P}\mathfrak{A})^{\mathsf{Prop}}$ instead of (agent-indexed) *valuations* $\mathsf{val} : S \times \mathsf{Prop} \to \mathscr{P}\mathfrak{A}$. This clearly is only a matter of presentation and irrelevant to any results, since $(X^Y)^Z \cong (X^Z)^Y$ for any sets X, Y and Z. We use colourings for ease of comparison with the coalgebraic generalization in Section 3.2.

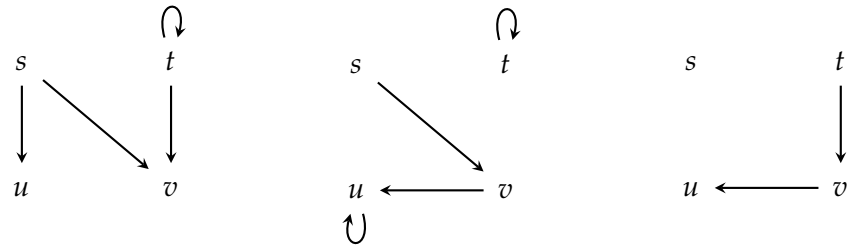
Before we consider the matter of how the semantics of agent-indexed basic modal logic should be defined using agent-indexed Kripke models, it is essential for our presentation of the material to introduce the following notion. We are able to take apart an agent-indexed Kripke model into component Kripke models by only considering a single agent's part of the model. These component Kripke models are referred to as *slices*.

Definition 3.1.3. Given an agent-indexed Kripke frame $\mathbb{F} = \langle S, R\rangle$ and an agent $\mathfrak{a} \in \mathfrak{A}$, the $\mathfrak{a}$-*slice of* $\mathbb{F}$ is the ordinary Kripke frame $\mathbb{F}_\mathfrak{a} = \langle S, R_\mathfrak{a}\rangle$, where



$\mathsf{col}(u)(p) := \{\mathfrak{a}, \mathfrak{c}\}$, $\mathsf{col}(v)(q) := \{\mathfrak{a}, \mathfrak{b}\}$, $\mathsf{col}(t)(p) := \{\mathfrak{c}\}$

Figure 3.1.1: Example of an agent-indexed Kripke model $\mathbb{M}$, defined in Example 3.1.4.



$\mathsf{col}_{\mathfrak{a}}(u) := \{p\}$, $\mathsf{col}_{\mathfrak{a}}(v) := \{q\}$ (a) $\mathbb{M}_{\mathfrak{a}}$.

$\mathsf{col}_{\mathfrak{b}}(v) := \{q\}$ (b) $\mathbb{M}_{\mathfrak{b}}$.

$\mathsf{col}_{\mathfrak{c}}(u) := \{p\}$, $\mathsf{col}_{\mathfrak{c}}(t) := \{p\}$ (c) $\mathbb{M}_{\mathfrak{c}}$.

Figure 3.1.2: Slices of $\mathbb{M}$ from Figure 3.1.1.

$R_{\mathfrak{a}} \subseteq S \times S$ is defined by putting $sR_{\mathfrak{a}}t$ iff $\mathfrak{a} \in R(s,t)$. Given an agent-indexed Kripke model $\mathbb{M} = \langle \mathbb{F}, \mathsf{col} \rangle$, the $\mathfrak{a}$-*slice of* $\mathbb{M}$ is the ordinary Kripke model $\mathbb{M}_{\mathfrak{a}} = \langle \mathbb{F}_{\mathfrak{a}}, \mathsf{col}_{\mathfrak{a}} \rangle$, where $\mathsf{col}_{\mathfrak{a}} : S \to \mathscr{P}\mathsf{Prop}$ is defined by putting $\mathsf{col}_{\mathfrak{a}}(s) := \{p \in \mathsf{Prop} \,;\, \mathfrak{a} \in \mathsf{col}(s)(p)\}$. ◁

Let us look at an example of how slicing works.

Example 3.1.4. Consider the agent-indexed Kripke model $\mathbb{M}$ in Figure 3.1.1 — states are vertices, the agent-indexed accessibility relation is denoted by the (labelled) edges, and the agent-indexed colouring is written out beneath the frame. In case an edge or colouring is missing, it means that said edge/colouring holds for no agents. The slices of $\mathbb{M}$ are displayed in Figure 3.1.2. ◁

The way slices fully determine the structures we are considering motivates our usage of the term 'agent-indexed' in referring to the introduced structures. Slices are also of vital importance for our logic, being used in a theorem (which we present as Theorem 3.1.6) in *How True* which states

that the semantics of Boolean-valued basic modal logic is fully determined by the semantics of two-valued basic modal logic, applied to each slice of an agent-indexed Kripke model. Conceptually, it is this theorem that validates the title of *How True*. In that article, slices are presented *after* introducing the logic and its semantics, and the aforementioned theorem is seen as a consequence of how the semantics is defined. However, we believe it is fruitful to take a different point of view at Fitting's work, with it actually being built with the notion of slicing as a primitive notion. We will therefore present things in a different order, with the following Slicing Slogan being a fundamental and helpful slogan motivating all of our design choices, including those of the semantics.

Slicing Slogan. *Multiagent structures consist of single-agent slices, and single-agent slices define multiagent structures.*

Note that we make no claims about the generality of this slogan beyond its application in studying and generalizing the structures and logics considered in *How True*.

As a first instance of the Slicing Slogan, let us see how the notion of slicing naturally gives rise to the semantics of Boolean-valued basic modal logic. First consider the semantics of two-valued basic modal logic. Over an ordinary Kripke model $\mathbb{M} = \langle\langle S, R\rangle, \mathsf{col}\rangle$, these can be given as a *semantic mapping* $[\![-]\!]_{\mathbb{M}} : \mathsf{Lang} \to \mathscr{P}S$, mapping a formula $\varphi \in \mathsf{Lang}$ to the set $[\![\varphi]\!]_{\mathbb{M}} \subseteq S$ of states in $\mathbb{M}$ at which φ holds. Since $\mathscr{P}S \cong 2^S$, we can also view the semantics as a function $[\![-]\!]_{\mathbb{M}} : \mathsf{Lang} \to 2^S$, mapping formulas to elements of the function space 2^S. This offers an elegant and precise interpretation of $f \in 2^S$ as two-valued predicates in $\mathbb{M}$, mapping states $s \in S$ to truth values in 2. These two-valued predicates denote whether the predicate holds at s (i.e. $f(s) = 1$), or does not (i.e. $f(s) = 0$).

From an algebraic point of view, we can consider 2^S to be the domain of a modal algebra[2] $\mathbf{2}_{\mathbb{M}}$ consisting of functions mapping states to elements of the two-element Boolean algebra $\mathbf{2}$, usually referred to as a *complex algebra*. The Boolean operations of $\mathbf{2}_{\mathbb{M}}$ arise naturally from those of $\mathbf{2}$, with e.g. the join of predicates $f, g \in 2^S$ defined for $s \in S$ as $\left(f \vee^{\mathbf{2}_{\mathbb{M}}} g\right)(s) := f(s) \vee^{\mathbf{2}} g(s)$. The operator $\Diamond^{\mathbf{2}_{\mathbb{M}}}$ is defined for $f \in 2^S$ and $s \in S$ as $\left(\Diamond^{\mathbf{2}_{\mathbb{M}}} f\right)(s) := 1$ iff there is some $t \in S$ such that sRt and $f(t) = 1$. Now, the semantics of two-valued basic modal logic are then simply given by the operations on $\mathbf{2}_{\mathbb{M}}$, with $[\![\varphi \vee \psi]\!]_{\mathbb{M}} := [\![\varphi]\!]_{\mathbb{M}} \vee^{\mathbf{2}_{\mathbb{M}}} [\![\psi]\!]_{\mathbb{M}}$, $[\![\neg\varphi]\!]_{\mathbb{M}} := \neg^{\mathbf{2}_{\mathbb{M}}} [\![\varphi]\!]_{\mathbb{M}}$, and $[\![\Diamond\varphi]\!]_{\mathbb{M}} := \Diamond^{\mathbf{2}_{\mathbb{M}}} [\![\varphi]\!]_{\mathbb{M}}$.

Since the truth values in Boolean-valued logics do not come from $\mathbf{2}$, but instead from algebra $\mathbb{P}_{\mathfrak{A}}$, it is natural to now consider the semantic mapping to be a function $[\![-]\!]^{\mathfrak{A}}_{\mathbb{M}} : \mathsf{Lang} \to (\mathscr{P}\mathfrak{A})^S$, mapping formulas to 'Boolean-valued predicates' $f \in (\mathscr{P}\mathfrak{A})^S$, determining for each state $s \in S$ the set of

[2] That is, a Boolean algebra $\mathbb{B}$ with an additional unary operator $\Diamond^{\mathbb{B}}$ on its domain that is *normal* and *additive*, meaning that it preserves finite (possibly empty) joins.

agents $f(s)$ for whom the underlying predicate holds. Analogously to what we did with the two-valued semantics, we apply $\mathscr{P}\mathfrak{A} \cong 2^{\mathfrak{A}}$, and instead describe Boolean-valued predicates $f \in (2^{\mathfrak{A}})^S$ as determining for states $s \in S$ and agents $\mathfrak{a} \in \mathfrak{A}$ whether the underlying predicate holds in state s for $\mathfrak{a}$. Using the isomorphisms $(2^{\mathfrak{A}})^S \cong (2^S)^{\mathfrak{A}}$, we can see how Boolean-valued predicates $f \in (2^{\mathfrak{A}})^S$ fit the Slicing Slogan, as they consist of two-valued predicate 'slices' $\widehat{f}(\mathfrak{a})$ for each $\mathfrak{a} \in \mathfrak{A}$.[3]

Putting this into an algebraic context again, we consider $(2^S)^{\mathfrak{A}}$ to be the domain of a modal algebra $\mathbf{2}_{\mathfrak{A},\mathbb{M}}$. Using the modal algebras $\mathbf{2}_{\mathbb{M}_\mathfrak{a}}$ given by the $\mathfrak{a}$-slices of $\mathbb{M}$ for $\mathfrak{a} \in \mathfrak{A}$, we can define all operations on Boolean-valued predicates in $\mathbf{2}_{\mathfrak{A},\mathbb{M}}$ by applying the operations of these modal algebra slices $\mathbf{2}_{\mathbb{M}_\mathfrak{a}}$ to the component two-valued predicate slices. This is uniformly done for all the operations, with for $f, g \in (2^S)^{\mathfrak{A}}$ the join being defined as $\left(f \vee^{\mathbf{2}_{\mathfrak{A},\mathbb{M}}} g\right)(\mathfrak{a}) := f(\mathfrak{a}) \vee^{\mathbf{2}_{\mathbb{M}_\mathfrak{a}}} g(\mathfrak{a})$, and the operator $\Diamond^{\mathbf{2}_{\mathfrak{A},\mathbb{M}}}$ being defined as $\left(\Diamond^{\mathbf{2}_{\mathfrak{A},\mathbb{M}}} f\right)(\mathfrak{a}) := \Diamond^{\mathbf{2}_{\mathbb{M}_\mathfrak{a}}} f(\mathfrak{a})$.

The modal algebra $\mathbf{2}_{\mathfrak{A},\mathbb{M}}$ as we have just defined it can now again be used to give the semantics of Boolean-valued basic modal logic. Working this out a bit, we see that the following definition precisely gives the semantics arising from $\mathbf{2}_{\mathfrak{A},\mathbb{M}}$, only considering Boolean-valued predicates to be elements of $(\mathscr{P}\mathfrak{A})^S$, instead of the equivalent $(2^S)^{\mathfrak{A}}$.

Definition 3.1.5. Given an agent-indexed Kripke model $\mathbb{M} = \langle \langle S, R \rangle, \mathsf{col} \rangle$, the *semantics of Boolean-valued basic modal logic* is given by the function $[\![-]\!]^{\mathfrak{A}}_{\mathbb{M}} : \mathsf{Lang} \to (\mathscr{P}\mathfrak{A})^S$ defined inductively by putting

$$\begin{aligned}
[\![p]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &:= \mathsf{col}(s)(p), && \text{(for } p \in \mathsf{Prop}\text{)} \\
[\![\varphi \vee \psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &:= [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \cup [\![\psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s), \\
[\![\neg\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &:= \mathfrak{A} \setminus [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s), \text{ and} \\
[\![\Diamond\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &:= \bigcup_{t \in S} \left(R(s,t) \cap [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(t)\right)
\end{aligned}$$

for $s \in S$. ◁

The fact that these semantics actually correspond to the slice-based semantics arising from $\mathbf{2}_{\mathfrak{A},\mathbb{M}}$ is the main content of what we refer to as the Basic Slicing Theorem (Fitting 2009, Theorem 4.2). To clarify that this is in fact what is stated by the theorem, we first give a slightly different, but equivalent and arguably more elegant presentation of the theorem, before Fitting's original formulation.

[3] As defined in Section 2.3, given a function $f : X \to Y^Z$, we write $\widehat{f} : Z \to Y^X$ for the function obtained by swapping the order of the first two arguments (or more formally, through $(Y^Z)^X \cong (Y^X)^Z$).

Theorem 3.1.6 (Basic Slicing Theorem)**.** *Let* $\mathbb{M}$ *be an agent-indexed Kripke model. For each formula* $\varphi \in \mathsf{Lang}$ *and agent* $\mathfrak{a} \in \mathfrak{A}$*, it holds that*

$$\widehat{[\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = [\![\varphi]\!]_{\mathbb{M}_{\mathfrak{a}}}.$$

Equivalently, we have that

$$[\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) := \left\{\mathfrak{a} \in \mathfrak{A}\,;\, s \in [\![\varphi]\!]_{\mathbb{M}_{\mathfrak{a}}}\right\}$$

for states s *in* $\mathbb{M}$*.*

Both formulations of the theorem can be proven by a simple induction on formulas $\varphi \in \mathsf{Lang}$.

Writing out the semantics for the logical connectives defined as abbreviations, we see that

$$\begin{aligned}
[\![\varphi \wedge \psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &= [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \cap [\![\psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s), \\
[\![\varphi \to \psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &= [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \Rightarrow [\![\psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \\
&= \left(\mathfrak{A} \setminus [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s)\right) \cup [\![\psi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s), \text{ and} \\
[\![\Box\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) &= \bigcap_{t \in S} \left(R(s,t) \Rightarrow [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s)\right) \\
&= \bigcap_{t \in S} \left((\mathfrak{A} \setminus R(s,t)) \cup [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s)\right).
\end{aligned}$$

The Basic Slicing Theorem (Theorem 3.1.6) confirms the Boolean-valued semantics for these defined connectives is truly just built out of the two-valued semantics for those same connectives, justifying their usage.

Before moving on, let us see an example of the semantics in action.

Example 3.1.4 (continued)**.** Considering the agent-indexed Kripke model in Figure 3.1.1, we compute the semantics of e.g. the formula $\Diamond q$ to find

$$\begin{aligned}
[\![\Diamond q]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) = \quad & \left(R(s,s) \cap [\![q]\!]^{\mathfrak{A}}_{\mathbb{M}}(s)\right) \\
& \cup \left(R(s,t) \cap [\![q]\!]^{\mathfrak{A}}_{\mathbb{M}}(t)\right) \\
& \cup \left(R(s,u) \cap [\![q]\!]^{\mathfrak{A}}_{\mathbb{M}}(u)\right) \\
& \cup \left(R(s,v) \cap [\![q]\!]^{\mathfrak{A}}_{\mathbb{M}}(v)\right) \\
= \quad & (\varnothing \cap \varnothing) \\
& \cup (\varnothing \cap \varnothing) \\
& \cup (\{\mathfrak{a}\} \cap \varnothing) \\
& \cup (\{\mathfrak{a}, \mathfrak{b}\} \cap \{\mathfrak{a}, \mathfrak{b}\}) \\
= \quad & \{\mathfrak{a}, \mathfrak{b}\}.
\end{aligned}$$

Similarly, we find that

$$[\![\Diamond q]\!]^{\mathfrak{A}}_{\mathbb{M}}(t) = \{\mathfrak{a}\}$$
$$[\![\Diamond q]\!]^{\mathfrak{A}}_{\mathbb{M}}(u) = \varnothing, \text{ and}$$
$$[\![\Diamond q]\!]^{\mathfrak{A}}_{\mathbb{M}}(v) = \varnothing.$$

We can see that the Basic Slicing Theorem (Theorem 3.1.6) indeed holds here, with $\Diamond q$ being true at s in $\mathbb{M}_{\mathfrak{a}}$ and $\mathbb{M}_{\mathfrak{b}}$, and at t in $\mathbb{M}_{\mathfrak{a}}$. ◁

3.1.2 Propositional Constants

While the Basic Slicing Theorem (Theorem 3.1.6) does provide an elegant characterization of the semantics of Boolean-valued basic modal logic, it also shows inherent limitations in the expressive power of the logic. Though two-valued logic does allow one to define formulas φ for some fixed truth value t such that $[\![\varphi]\!]_{\mathbb{M}}(s) = t$ for any Kripke model $\mathbb{M}$ (e.g. $p \vee \neg p$ for $\top$, $p \wedge \neg p$ for $\bot$), this same property no longer holds in the Boolean-valued setting. In fact, the only truth values for which such formulas exist are $\mathfrak{A}$ and $\varnothing$, with the corresponding formulas again being $p \vee \neg p$ and $p \wedge \neg p$, respectively. To see that such formulas do not exist for truth values $\mathfrak{B} \in \mathscr{P}\mathfrak{A} \setminus \{\mathfrak{A}, \varnothing\}$, one need only consider agent-indexed Kripke models $\mathbb{M}$ in which every agent has the same slice: the Basic Slicing Theorem guarantees that $[\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \in \{\mathfrak{A}, \varnothing\}$.

To combat this lack of expressivity, *How True* introduces propositional constants for each set of agents into the logic. This is a natural choice, as these sets are now the truth values in our logic.

Definition 3.1.7. The *extended Boolean-valued basic modal language* $\mathsf{Lang}_{\mathfrak{A}}$ is defined as $\mathsf{Lang}_{\mathfrak{A}} := \mathsf{Lang} \cup \{\ulcorner \mathfrak{B} \urcorner ; \mathfrak{B} \subseteq \mathfrak{A}\}$, where $\ulcorner \mathfrak{B} \urcorner$ is a purely syntactic constant. Given an agent-indexed Kripke model $\mathbb{M}$ with set of states S, the *semantics of extended Boolean-valued basic modal logic* is given by the function $[\![-]\!]^{\mathfrak{A},\mathrm{Ext}}_{\mathbb{M}} : \mathsf{Lang}_{\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$, defined by putting $[\![\varphi]\!]^{\mathfrak{A},\mathrm{Ext}}_{\mathbb{M}} := [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}$ for $\varphi \in \mathsf{Lang}$, and $[\![\ulcorner \mathfrak{B} \urcorner]\!]^{\mathfrak{A},\mathrm{Ext}}_{\mathbb{M}}(s) := \mathfrak{B}$ for $\mathfrak{B} \subseteq \mathfrak{A}$ and $s \in S$. ◁

From now on, we will always be working with extended Boolean-valued basic modal logic, and will therefore drop the 'extended' part both in our text, as well as in the semantics (writing $[\![-]\!]^{\mathfrak{A}}_{\mathbb{M}}$ instead of $[\![-]\!]^{\mathfrak{A},\mathrm{Ext}}_{\mathbb{M}}$).

These new propositional constants can be used in conjunction with other logical connectives to create rich expressions. For example, using the abbreviation for implication, we find that

$$[\![\ulcorner \mathfrak{B} \urcorner \to \varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) = (\mathfrak{A} \setminus \mathfrak{B}) \cup [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s),$$

which effectively allows us to use an assertion of $\ulcorner \mathfrak{B} \urcorner \to \varphi$ to put a lower bound on the truth value of φ, since

$$[\![\ulcorner \mathfrak{B} \urcorner \to \varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) = \mathfrak{A} \text{ iff } \mathfrak{B} \subseteq [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s).$$

Conversely, we find that $[\![\varphi \to \ulcorner \mathfrak{B} \urcorner]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) = \mathfrak{A}$ iff $[\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \subseteq \mathfrak{B}$. Such formulas containing implications between propositional constants and other formulas are referred to as *bounding formulas* in *How True*, and are used there to produce tableaux. Bounding formulas will also be of vital importance to our work in Chapter 5.

Though not addressed in *How True*, the Basic Slicing Theorem (Theorem 3.1.6) no longer directly applies after introducing the new propositional constants, since these do not exist in the language of basic modal logic. But considering how a propositional constant $\ulcorner \mathfrak{B} \urcorner$ effectively 'behaves' as $\top$ for agents $\mathfrak{a} \in \mathfrak{B}$, and as $\bot$ for agents $\mathfrak{a} \notin \mathfrak{B}$, we are able to give translations of the formulas in our extended language to the original language in such a way that a version of the theorem still applies.

For an agent $\mathfrak{a} \in \mathfrak{A}$, inductively define the translation $\mathsf{tr}_{\mathfrak{a}} : \mathsf{Lang}_{\mathfrak{A}} \to \mathsf{Lang}$ by putting

$$\begin{aligned}
\mathsf{tr}_{\mathfrak{a}}(p) &:= p, && (p \in \mathsf{Prop}) \\
\mathsf{tr}_{\mathfrak{a}}(\ulcorner \mathfrak{B} \urcorner) &:= \begin{cases} \top & \text{if } \mathfrak{a} \in \mathfrak{B} \\ \bot & \text{if } \mathfrak{a} \notin \mathfrak{B} \end{cases}, && (\mathfrak{B} \subseteq \mathfrak{A}) \\
\mathsf{tr}_{\mathfrak{a}}(\neg\varphi) &:= \neg \mathsf{tr}_{\mathfrak{a}}(\varphi), \\
\mathsf{tr}_{\mathfrak{a}}(\varphi \vee \psi) &:= \mathsf{tr}_{\mathfrak{a}}(\varphi) \vee \mathsf{tr}_{\mathfrak{a}}(\psi), \text{ and} \\
\mathsf{tr}_{\mathfrak{a}}(\Diamond\varphi) &:= \Diamond \mathsf{tr}_{\mathfrak{a}}(\varphi).
\end{aligned}$$

Though not given in *How True*, the following extended version of the slicing theorem holds when using this translation.

Theorem 3.1.8 (Extended Slicing Theorem)**.** *Let* $\mathbb{M}$ *be an agent-indexed Kripke model. For each formula* $\varphi \in \mathsf{Lang}_{\mathfrak{A}}$ *and agent* $\mathfrak{a} \in \mathfrak{A}$*, it holds that*

$$\widehat{[\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]_{\mathbb{M}_{\mathfrak{a}}}.$$

This theorem is yet again proven using a simple induction on formulas $\varphi \in \mathsf{Lang}_{\mathfrak{A}}$.

The Extended Slicing Theorem shows that the extended logic still has some glaring limitations expressivity-wise. Informally speaking, it is still the case that there are no formulas φ such that determining whether its evaluation contains some agent $\mathfrak{a}$ requires one to also consider the slice of another agent $\mathfrak{b}$. This holds because in determining what the truth value of a formula is, we need only independently evaluate the formula in each slice, before aggregating the results of these evaluations into a single truth value. This makes the logic unsuited for expressing general multiagent two-valued logics. We will revisit this limitation in Section 5.2, as we will be equipped to provide a possible solution by then.

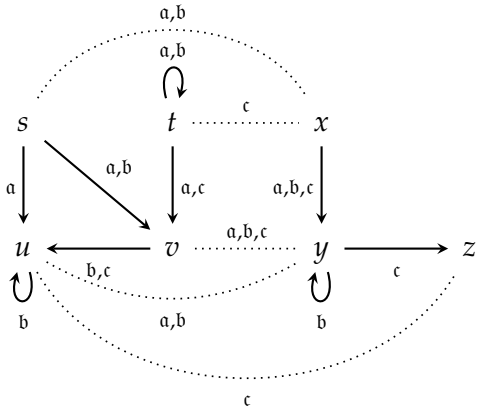


Figure 3.1.3: An agent-indexed bisimulation between two agent-indexed Kripke models, defined in Example 3.1.10.

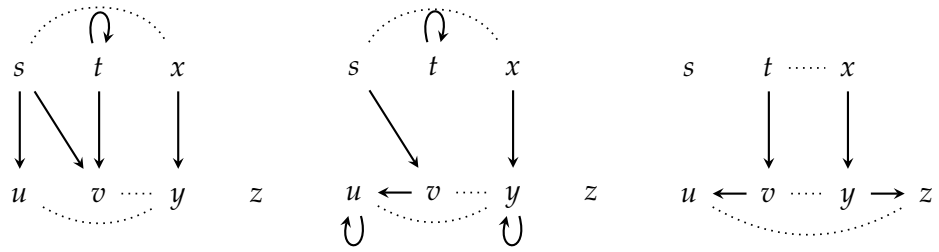
3.1.3 Bisimulations and Bounded Morphisms

As is standard within the field of modal logic, it is important to have a proper notion of bisimulation in order to study behavioural equality between two structures. Agent-indexed Kripke models are no exception to this requirement. In *How True*, a notion of bisimulation between agent-indexed Kripke models is given that was originally defined by Fitting (2003), which we refer to as *agent-indexed bisimulations*. While *How True* solely discusses mechanical ways of determining whether an agent-indexed bisimulation between two models exists, we will focus on their semantic properties.

The definition of agent-indexed bisimulations fully arises through the Slicing Slogan. Where an ordinary bisimulation between Kripke models is a binary relation between the two models' state spaces satisfying some conditions, through application of the Slicing Slogan we will require the multiagent analogue to be an agent-indexed relation consisting of bisimulation slices.

Definition 3.1.9. Given agent-indexed Kripke models $\mathbb{M}$ and $\mathbb{M}'$ with respective state spaces S and S', an agent-indexed relation $B : S \times S' \to \mathscr{P}\mathfrak{A}$ is called an *agent-indexed bisimulation* $B : \mathbb{M} \leftrightarrows \mathbb{M}'$ if $B_{\mathfrak{a}} : \mathbb{M}_{\mathfrak{a}} \leftrightarrows \mathbb{M}'_{\mathfrak{a}}$ for all agents $\mathfrak{a} \in \mathfrak{A}$. If $\mathfrak{B} \subseteq B(s, s')$ for states $s \in S$ and $s' \in S'$, we also write $B : \mathbb{M}, s \leftrightarrows_{\mathfrak{B}} \mathbb{M}', s'$. If there is some B such that $B : \mathbb{M}, s \leftrightarrows_{\mathfrak{B}} \mathbb{M}', s'$, we write $\mathbb{M}, s \leftrightarrows_{\mathfrak{B}} \mathbb{M}', s'$. If $\mathfrak{B}$ is a singleton $\{\mathfrak{a}\}$, we leave out the brackets, and we write $\leftrightarrows_{\mathfrak{a}}$ instead of $\leftrightarrows_{\{\mathfrak{a}\}}$. ◁

Example 3.1.10. Consider the agent-indexed Kripke models in Figure 3.1.3: the left one is from Figure 3.1.1, while the right one is new. We ignore colourings for clarity of presentation: we could just assume the colourings assign no propositional variables to any of the states. The dotted lines de-



(a) The $\mathfrak{a}$-slices. (b) The $\mathfrak{b}$-slices. (c) The $\mathfrak{c}$-slices.

Figure 3.1.4: Slices of the agent-indexed Kripke models and bisimulation from Figure 3.1.3.

note an agent-indexed bisimulation between the two models, as can be verified by looking at the slices in Figure 3.1.4. ◁

Though *How True* does not expand upon this, as we will see shortly as well as later in Section 3.2, agent-indexed bisimulations as defined thusly are indeed the natural candidate for bisimulations between agent-indexed Kripke models. For now we focus on two arguments not given in *How True*: with respect to our definition of bisimulations, Boolean-valued basic modal logic is both *adequate* (meaning that bisimilar states are logically equivalent) and *expressive* (meaning that logically equivalent states are bisimilar), as also treated in Section 2.2.

We first need to define what it means for two states to be logically equivalent. Following the Slicing Slogan, we will consider a sliced version of logical equivalence, in which two states are logically equivalent for a specific agent.

Definition 3.1.11. Given agent-indexed Kripke models $\mathbb{M}$ and $\mathbb{M}'$ and states s and s' from $\mathbb{M}$ and $\mathbb{M}'$ respectively, and an agent $\mathfrak{a} \in \mathfrak{A}$, we say s and s' are *logically* $\mathfrak{a}$*-equivalent* with respect to Boolean-valued basic modal logic (and we write $\mathbb{M}, s \equiv^{\mathrm{BML}}_{\mathfrak{a}} \mathbb{M}', s'$) if

$$\mathfrak{a} \in [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s) \text{ iff } \mathfrak{a} \in [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}'}(s')$$

for all formulas $\varphi \in \mathsf{Lang}_{\mathfrak{A}}$. If $\mathbb{M}, s \equiv^{\mathrm{BML}}_{\mathfrak{a}} \mathbb{M}', s'$ for all agents $\mathfrak{a} \in \mathfrak{A}$, then we just say s and s' are *logically equivalent* with respect to Boolean-valued basic modal logic, and we write $\mathbb{M}, s \equiv^{\mathrm{BML}} \mathbb{M}', s'$. ◁

Using this sliced definition of logical equivalence, we can express the adequacy of our logic in the following, also sliced form.

Theorem 3.1.12. *Let* $\mathbb{M}$ *and* $\mathbb{M}'$ *be agent-indexed Kripke models with states* s *and* s' *from* $\mathbb{M}$ *and* $\mathbb{M}'$ *respectively. It holds that*

$$\mathbb{M}, s \leftrightarrows_{\mathfrak{a}} \mathbb{M}', s' \text{ implies } \mathbb{M}, s \equiv^{\mathrm{BML}}_{\mathfrak{a}} \mathbb{M}', s'$$

for all agents $\mathfrak{a} \in \mathfrak{A}$.

Proof. By assumption, there must be some agent-indexed bisimulation B such that $B_{\mathfrak{a}} : \mathbb{M}_{\mathfrak{a}}, s \leftrightarrows \mathbb{M}'_{\mathfrak{a}}, s'$. Take any $\varphi \in \mathsf{Lang}_{\mathfrak{A}}$. Suppose without loss of generality that $\mathfrak{a} \in [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}}(s)$. Then by the Extended Slicing Theorem (Theorem 3.1.8) we find that $s \in [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]_{\mathbb{M}_{\mathfrak{a}}}$. Since $\mathbb{M}_{\mathfrak{a}}, s \leftrightarrows \mathbb{M}'_{\mathfrak{a}}, s'$, it follows from the adequacy of two-valued basic modal logic that $s' \in [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]_{\mathbb{M}'_{\mathfrak{a}}}$ as well. Finally, applying the Extended Slicing Theorem again, we get that $\mathfrak{a} \in [\![\varphi]\!]^{\mathfrak{A}}_{\mathbb{M}'}(s')$. □

Most properties we wish to prove about Boolean-valued basic modal logic can be proven quite simply by using the same property in the two-valued case, along with (potentially) the Extended Slicing Theorem, like we did here. Therefore we generally omit proofs of such statements. One instance of such a property is expressivity over image-finite agent-indexed Kripke models.

Theorem 3.1.13. *Let* $\mathbb{M}$ *and* $\mathbb{M}'$ *be agent-indexed Kripke models such that the slices* $\mathbb{M}_{\mathfrak{a}}$ *and* $\mathbb{M}'_{\mathfrak{a}}$ *for all agents* $\mathfrak{a} \in \mathfrak{A}$ *are image-finite. Then for all states* s *in* $\mathbb{M}$ *and* s' *in* $\mathbb{M}'$*, it holds that*

$$\mathbb{M}, s \equiv^{\mathrm{BML}}_{\mathfrak{a}} \mathbb{M}', s' \text{ implies } \mathbb{M}, s \leftrightarrows_{\mathfrak{a}} \mathbb{M}', s'.$$

for all agents $\mathfrak{a} \in \mathfrak{A}$*.*

While bisimulations are a unifying concept all throughout modal logic, they were historically preceded by the notion of a *bounded morphism*. These are the natural analogue of a structure-preserving map between two Kripke models, and are in fact precisely *functional* bisimulations. There are no analogues to bounded morphisms in *How True*, however. In order to define a notion of *agent-indexed bounded morphisms* in accordance with the Slicing Slogan, we will require a structure with ordinary bounded morphisms as its slices. In order to do this, the notion of an *agent-indexed function* will be essential. These are defined similarly to agent-indexed relations, but will be of such great importance throughout our exploration of Boolean-valued modal logic that we highlight them here.

Definition 3.1.14. Given sets X and Y, an *agent-indexed function between* X *and* Y is a function $f : X \to Y^{\mathfrak{A}}$. For such f, we write $f : X \rightsquigarrow Y$ to make explicit that it is an agent-indexed function. Given an agent $\mathfrak{a} \in \mathfrak{A}$, the $\mathfrak{a}$-*slice* of f is the function $f_{\mathfrak{a}} : X \to Y$ defined as $f_{\mathfrak{a}}(x) := f(x)(a)$ for $x \in X$. We say f is *injective* (resp. *surjective*, *bijective*) if $f_{\mathfrak{a}}$ is injective (resp. surjective, bijective) for each agent $\mathfrak{a} \in \mathfrak{A}$. The *agent-indexed graph of* f is the agent-indexed relation $\mathscr{Graph}_{\mathrm{AGENT}} f : X \times Y \to \mathscr{P}\mathfrak{A}$ defined as $(\mathscr{Graph}_{\mathrm{AGENT}} f)(x, y) := \{\mathfrak{a} \in \mathfrak{A} ; f_{\mathfrak{a}}(x) = y\}$. ◁

Remark 3.1.15. Though we define agent-indexed functions as functions $f : X \to Y^{\mathfrak{A}}$, it is arguably conceptually clearer to consider them to be families

$\langle f_{\mathfrak{a}} \rangle_{\mathfrak{a} \in \mathfrak{A}}$ of functions $f_{\mathfrak{a}} : X \to Y$ (i.e. functions $f : \mathfrak{A} \to Y^X$), as this would make the slices explicit. It will often however be technically clearer to work with the first definition $f : X \to Y^{\mathfrak{A}}$, since the domains of the agent-indexed function and the underlying function match. As all three of these definitions are equivalent for our purposes, we will freely switch between these two presentations as needed to make notation clearer, though we will implicitly assume that the first definition is the 'actual' one. Having said that, the alternative definitions might be better suited when generalizing the current work even further by creating agent-indexed versions of categories other than **Set**. ◁

Definition 3.1.16. Given agent-indexed Kripke frames $\mathbb{M}$ and $\mathbb{M}'$ with state spaces S and S', respectively, an agent-indexed function $f : S \rightsquigarrow S'$ is an *agent-indexed bounded morphism between* $\mathbb{M}$ *and* $\mathbb{M}'$ (written as $f : \mathbb{M} \to \mathbb{M}'$) if the slice $f_{\mathfrak{a}}$ is a bounded morphism $f_{\mathfrak{a}} : \mathbb{M}_{\mathfrak{a}} \to \mathbb{M}'_{\mathfrak{a}}$ for all agents $\mathfrak{a} \in \mathfrak{A}$. ◁

As with two-valued bounded morphisms, we have that agent-indexed bounded morphisms are precisely functional bisimulations.

Proposition 3.1.17. *Let $f : \mathbb{M} \to \mathbb{M}'$ be an agent-indexed bounded morphism between agent-indexed Kripke models with respective state spaces S and S', and let $B : S \times S' \to \mathcal{P}\mathfrak{A}$ be an agent-indexed bisimulation such that $B_{\mathfrak{a}}$ is functional for each agent $\mathfrak{a} \in \mathfrak{A}$. Then it holds that*

(*i*) $\mathscr{G}\mathit{raph}_{\text{AGENT}} f : \mathbb{M}, s \leftrightarrows_{\mathfrak{a}} \mathbb{M}'$, $f_{\mathfrak{a}}(s)$ *for all $s \in S$ and agents $\mathfrak{a} \in \mathfrak{A}$, and*

(*ii*) $g : \mathbb{M} \to \mathbb{M}'$, *where $g : S \rightsquigarrow S'$ is defined by putting $g_{\mathfrak{a}}(s) := s'$ for $s \in S$ with s' being the unique state in S' such that $sB_{\mathfrak{a}}s'$.*

Corollary 3.1.18. *For agent-indexed bounded morphisms $f : \mathbb{M} \to \mathbb{M}'$ and states s in $\mathbb{M}$, it holds that $\mathbb{M}, s \equiv^{\text{BML}}_{\mathfrak{a}} \mathbb{M}'$, $f_{\mathfrak{a}}(s)$ for all agents $\mathfrak{a} \in \mathfrak{A}$.*

3.2 Boolean-Valued Coalgebraic Modal Logic

Having given an overview of the theory of Boolean-valued basic modal logic built on top of agent-indexed Kripke models, we now set our sights on a more general goal: defining generalized Boolean-valued modal logics over arbitrary state-based systems. We will do this using the framework of *coalgebraic modal logic*. These logics, like the original logic from *How True*, will be defined as natural Boolean-valued 'liftings' of an underlying two-valued logic.

3.2.1 Agent-Indexed Coalgebras

As our first step towards a coalgebraic generalization, we will try to fit agent-indexed Kripke models as a structure into the framework of coalgebra. For simplicity, let us first consider frames. We define the category

AgentKF of agent-indexed Kripke frames and agent-indexed bounded morphisms.[4] To properly model these as coalgebras, we require a category $\mathbf{C}$ and an endofunctor $\mathscr{T} : \mathbf{C} \to \mathbf{C}$ such that the category $\mathbf{Coalg}_{\mathbf{C}}(\mathscr{T})$ of $\mathscr{T}$-coalgebras is equivalent to **AgentKF**.

An initial guess would be to use $\mathbf{C} = \mathbf{Set}$ and $\mathscr{T} = \mathscr{P}^{\mathfrak{A}}$, and this guess would not be entirely wrong. If we purely consider the objects in the categories, we can see that $\mathscr{P}^{\mathfrak{A}}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ are precisely agent-indexed Kripke frames, with the agent-indexed accessibility relation $R : S \times S \to \mathscr{P}\mathfrak{A}$ being recoverable by defining $R(s,t) = \{\mathfrak{a} \in \mathfrak{A} \,;\, t \in \sigma(s)(\mathfrak{a})\}$. And conversely, we can consider agent-indexed Kripke frames to be $\mathscr{P}^{\mathfrak{A}}$-coalgebras by defining the transition map $\sigma : S \to (\mathscr{P}S)^{\mathfrak{A}}$ as $\sigma(s)(\mathfrak{a}) := \{t \in S \,;\, sR_{\mathfrak{a}}t\}$.

So far this works out. But what do $\mathscr{P}^{\mathfrak{A}}$-coalgebra morphisms look like? As our base category is **Set**, a $\mathscr{P}^{\mathfrak{A}}$-coalgebra morphism $f : \mathbb{S} \to \mathbb{S}'$ is a function $f : S \to S'$ such that the diagram

$$\begin{array}{ccc} S & \xrightarrow{\quad f \quad} & S' \\ {\scriptstyle\sigma}\downarrow & & \downarrow{\scriptstyle\sigma'} \\ (\mathscr{P}S)^{\mathfrak{A}} & \xrightarrow[(\mathscr{P}f)^{\mathfrak{A}}]{} & (\mathscr{P}S')^{\mathfrak{A}} \end{array}$$

commutes. Unfortunately, not all agent-indexed bounded morphisms correspond to $\mathscr{P}^{\mathfrak{A}}$-coalgebra morphisms. In fact, $\mathscr{P}^{\mathfrak{A}}$-coalgebra morphisms can all be viewed as agent-indexed functions $f : S \rightsquigarrow S'$ for which $f_{\mathfrak{a}} = f_{\mathfrak{b}}$ for all agents $\mathfrak{a}, \mathfrak{b} \in \mathfrak{A}$. So clearly, $\mathbf{Coalg}_{\mathbf{Set}}(\mathscr{P}^{\mathfrak{A}})$ will not be equivalent to **AgentKF**.

Since the problem stems from the fact that $\mathscr{P}^{\mathfrak{A}}$-coalgebra morphisms are functions in **Set**, an obvious next step is to change our base category to one in which morphisms correspond to agent-indexed functions. As we will see, this move is the right one.

Definition 3.2.1. We denote by **ASet** the category of *sets and agent-indexed functions*. Given two agent-indexed functions $f : X \rightsquigarrow Y$ and $g : Y \rightsquigarrow Z$, their *composition* $g \circ f : X \rightsquigarrow Z$ is defined in the obvious way by composing along slices, i.e. $(g \circ f)_{\mathfrak{a}} := g_{\mathfrak{a}} \circ f_{\mathfrak{a}}$. If f and g are functions $f : X \to Y^{\mathfrak{A}}$ and $g : Y \to Z^{\mathfrak{A}}$, we define it as $(g \circ f)(x)(\mathfrak{a}) := g(f(x)(\mathfrak{a}))(\mathfrak{a})$. The *agent-indexed identity* $\mathrm{id}_X^{\mathbf{ASet}} : X \rightsquigarrow X$ on a set X is defined as $(\mathrm{id}_X^{\mathbf{ASet}})_{\mathfrak{a}} := \mathrm{id}_X^{\mathbf{Set}}$. ◁

It is not difficult to verify that these specifications produce a well-defined category. This is quite a simple category to work with, as virtually all we can say about **Set** extends easily to a statement about **ASet**. Most of this follows from the following basic but essential observation.

[4] Though we have not explicitly defined them, it should be clear how an agent-indexed bounded morphism between frames is defined: it is an agent-indexed function between state spaces such that every one of its slices is a bounded morphism between the slices of the frames.

Fact 3.2.2. *Agent-indexed functions f and g are equal iff the $\mathfrak{a}$-slice functions $f_{\mathfrak{a}}$ and $g_{\mathfrak{a}}$ are equal for all agents $\mathfrak{a} \in \mathfrak{A}$.*

As a consequence of this observation, diagrams in **ASet** commute iff the diagram in **Set** obtained by taking $\mathfrak{a}$-slices of all agent-indexed functions in the original diagram commutes for all $\mathfrak{a} \in \mathfrak{A}$. Furthermore, mono-, epi- and isomorphisms in **ASet** are precisely injective, surjective and bijective agent-indexed functions (as defined in Definition 3.1.14), respectively.

Before showing how coalgebras over **ASet** can capture agent-indexed Kripke frames and models, we first show how some general important properties of **Set** extend to **ASet**.

Proposition 3.2.3. *The category* **ASet**

(*i*) *has all small products,*

(*ii*) *has all small coproducts,*

(*iii*) *is Cartesian closed, and*

(*iv*) *has subobject classifiers.*

Proof. (i) Take a set $\{X_i\}_{i \in I}$ of sets indexed by a set I. Since **Set** is complete, the **Set**-product $\prod_{i \in I} X_i$ exists. We equip this **Set**-product with agent-indexed projections $\mathsf{proj}_i^{\mathbf{ASet}} : \prod_{i \in I} X_i \rightsquigarrow X_i$ for each $i \in I$ by applying the **Set**-projections along each slice — i.e. $(\mathsf{proj}_i^{\mathbf{ASet}})_{\mathfrak{a}} := \mathsf{proj}_i^{\mathbf{ASet}}$. To see that $\prod_{i \in I} X_i$ along with these agent-indexed projections actually forms the **ASet**-product of the sets X_i, consider some set P with agent-indexed functions $f^{(i)} : P \rightsquigarrow X_i$ for each $i \in I$. For each such $f^{(i)}$, we have that $f^{(i)}_{\mathfrak{a}}$ is a function $f^{(i)}_{\mathfrak{a}} : P \to X_i$ for each $\mathfrak{a} \in \mathfrak{A}$. So applying the universal property of the **Set**-product, we see that for every $\mathfrak{a} \in \mathfrak{A}$ there is a unique function $g_{\mathfrak{a}} : P \to \prod_{i \in I} X_i$ such that

$$f^{(i)}_{\mathfrak{a}} = \mathsf{proj}_i^{\mathbf{Set}} \circ g_{\mathfrak{a}}. \tag{3.1}$$

Considering these functions $g_{\mathfrak{a}}$ as slices, we obtain an agent-indexed function $g : P \rightsquigarrow \prod_{i \in I} X_i$. And since Equation (3.1) holds for each $\mathfrak{a} \in \mathfrak{A}$ and $i \in I$, we immediately get that $f^{(i)} = \mathsf{proj}_i^{\mathbf{ASet}} \circ g$ for each $i \in I$ as well. The uniqueness of g is finally guaranteed by the uniqueness of each of the functions $g_{\mathfrak{a}}$. So $\prod_{i \in I} X_i$ along with the agent-indexed projections is the **ASet**-product $\prod_{i \in I} X_i$.

(ii) This can be shown using approximately the same line of reasoning used to show **ASet** has small products, only with all arrows reversed. The **ASet**-coproduct is the **Set**-coproduct $\sum_{i \in I} X_i$ with agent-indexed injections $\mathsf{inj}_i^{\mathbf{ASet}} : X_i \rightsquigarrow \sum_{i \in I} X_i$ whose slices are the injections $\mathsf{inj}_i^{\mathbf{Set}}$.

(iii) Since we have shown that **ASet** has all small products, it suffices to show that exponential objects exist.[5] Given sets X and Y, we define their exponential as $\mathsf{Exp}_{\mathbf{ASet}}(X,Y) := \{f\,;\, f : X \rightsquigarrow Y\}$. Its evaluation map $\mathsf{eval}^{\mathbf{ASet}}_{X,Y} : \mathsf{Exp}_{\mathbf{ASet}}(X,Y) \times X \rightsquigarrow Y$ is given as $(\mathsf{eval}^{\mathbf{ASet}}_{X,Y})_{\mathfrak{a}}(f,x) := f_{\mathfrak{a}}(x)$. To see that $\mathsf{Exp}_{\mathbf{ASet}}(X,Y)$ is indeed the exponential of X and Y, take any set Z and agent-indexed function $g : Z \times X \rightsquigarrow Y$. Define the exponential transpose $h : Z \rightsquigarrow \mathsf{Exp}_{\mathbf{ASet}}(X,Y)$ of g by putting $(h_{\mathfrak{a}}(z))_{\mathfrak{b}}(x) := g_{\mathfrak{b}}(z,x)$. We then have that

$$\begin{aligned}\left(\mathsf{eval}^{\mathbf{ASet}}_{X,Y}\right)_{\mathfrak{a}} \circ \left(h_{\mathfrak{a}} \times \left(\mathsf{id}^{\mathbf{ASet}}_{X}\right)_{\mathfrak{a}}\right) &= \left(\mathsf{eval}^{\mathbf{ASet}}_{X,Y}\right)_{\mathfrak{a}} \circ \left(h_{\mathfrak{a}} \times \mathsf{id}^{\mathbf{Set}}_{X}\right) \\ &= \langle z,x\rangle \in Z \times X \mapsto \left(h_{\mathfrak{a}}(z)\right)_{\mathfrak{a}}(x) \\ &= \langle z,x\rangle \in Z \times X \mapsto g_{\mathfrak{a}}(z,x) \\ &= g_{\mathfrak{a}}\end{aligned}$$

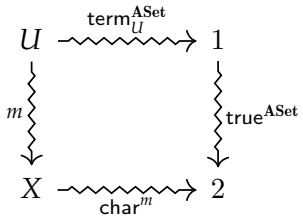
for every agent $\mathfrak{a} \in \mathfrak{A}$. So $\mathsf{eval}^{\mathbf{ASet}}_{X,Y} \circ \left(h \times \mathsf{id}^{\mathbf{ASet}}_{X}\right) = g$. To see that h is the only agent-indexed function satisfying this equality, note that for any alternative $h' : Z \rightsquigarrow \mathsf{Exp}_{\mathbf{ASet}}(X,Y)$ with $\mathsf{eval}^{\mathbf{ASet}}_{X,Y} \circ \left(h' \times \mathsf{id}^{\mathbf{ASet}}_{X}\right) = g$, it holds for every $\mathfrak{a} \in \mathfrak{A}$ and $\langle z,x\rangle \in Z \times X$ that $(h'_{\mathfrak{a}}(z))_{\mathfrak{a}}(x) = g_{\mathfrak{a}}(z,x) = (h_{\mathfrak{a}}(z))_{\mathfrak{a}}(x)$. Thus $\mathsf{Exp}_{\mathbf{ASet}}(X,Y)$ with $\mathsf{eval}^{\mathbf{ASet}}_{X,Y}$ is the exponential of X and Y.

(iv) First note that the one-element set 1 is the terminal object in **ASet**, with the unique $\mathsf{term}^{\mathbf{ASet}}_{X} : X \rightsquigarrow 1$ for sets X consisting of the functions $\mathsf{term}^{\mathbf{Set}}_{X}$ for all of its slices. Based on our discussion of logic in the agent-indexed setting so far, it is natural to guess that the subobject classifier of **ASet** consists of the set $\mathscr{P}\mathfrak{A}$. But this is actually not the case: like in **Set**, we have that the subobject classifier in **ASet** is 2, with corresponding agent-indexed truth function $\mathsf{true}^{\mathbf{ASet}} : 1 \rightsquigarrow 2$ defined as $\mathsf{true}^{\mathbf{ASet}}_{\mathfrak{a}} := \mathsf{true}^{\mathbf{Set}}$.

To see that this is indeed the subobject classifier, take any set X and injective agent-indexed function $m : U \rightsquigarrow X$. Then every slice $m_{\mathfrak{a}} : U \to X$ is an injective function, and so by definition of the subobject classifier in **Set**, for every $\mathfrak{a} \in \mathfrak{A}$ there exists a unique function $\mathsf{char}^{m_{\mathfrak{a}}} : X \to 2$ such that $m_{\mathfrak{a}}$ and $\mathsf{term}^{\mathbf{Set}}_{U}$ form the pullback of $\mathsf{true}^{\mathbf{Set}}$ and $\mathsf{char}^{m_{\mathfrak{a}}}$. Define $\mathsf{char}^{m} : X \rightsquigarrow 2$ by taking the functions $\mathsf{char}^{m_{\mathfrak{a}}}$ to be its slices. Then by definition of $\mathsf{true}^{\mathbf{ASet}}$ and $\mathsf{term}^{\mathbf{ASet}}_{U}$, it follows immediately that

[5] In order to reduce confusion between **Set** and **ASet**, we will sometime make explicit that we take an exponential object Y^X in category $\mathbf{C}$ by writing $\mathsf{Exp}_{\mathbf{C}}(X,Y)$ instead.

the diagram



commutes. And since $m_{\mathfrak{a}}$ and $\mathsf{term}_U^{\mathbf{Set}}$ form the pullback of $\mathsf{true}^{\mathbf{Set}}$ and $\mathsf{char}^{m_{\mathfrak{a}}}$, it follows quite simply that m and $\mathsf{term}_U^{\mathbf{ASet}}$ indeed form the pullback of char^m and $\mathsf{true}^{\mathbf{ASet}}$. □

Remark 3.2.4. While we gave an explicit and (arguably) pragmatic definition of **ASet** that will suit our purposes, the following may be of interest to readers with more background in category theory. Note that **ASet** actually can also be seen to arise as the Kleisli category for a monad. Consider the endofunctor $\mathscr{E} : \mathbf{Set} \to \mathbf{Set}$ defined on sets X as $\mathscr{E}X := X^{\mathfrak{A}}$, and on functions $f : X \to Y$ as $\big(\mathscr{E}f\big)(g) := f \circ g$ for $g \in X^{\mathfrak{A}}$. We equip $\mathscr{E}$ with the unit $\mathbf{unit} : \mathscr{Id}_{\mathbf{Set}} \Rightarrow \mathscr{E}$ defined as $\mathsf{unit}_X(x)(\mathfrak{a}) := x$, and multiplication $\mathbf{mult} : \mathscr{E} \circ \mathscr{E} \Rightarrow \mathscr{E}$ defined as $\mathsf{mult}_X(f)(\mathfrak{a}) := f(\mathfrak{a})(\mathfrak{a})$. It is easily verified that $\mathbb{E} = \langle \mathscr{E}, \mathbf{unit}, \mathbf{mult} \rangle$ indeed forms a **Set**-monad, with

$$\mathsf{mult}_X \circ \mathscr{E}\mathsf{mult}_X = f \in \mathscr{E}\mathscr{E}\mathscr{E}X \mapsto \big(\mathfrak{a} \in \mathfrak{A} \mapsto f(\mathfrak{a})(\mathfrak{a})(\mathfrak{a})\big) = \mathsf{mult}_X \circ \mathsf{mult}_{\mathscr{E}X}$$

and

$$\mathsf{mult}_X \circ \mathscr{E}\mathsf{unit}_X = \mathsf{id}_{\mathscr{E}X} = \mathsf{mult}_X \circ \mathsf{unit}_{\mathscr{E}X}$$

for all sets X. The morphisms in the Kleisli category $\mathbf{Kleisli}(\mathbb{B})$ between sets X and Y are then functions $f : X \to \mathscr{E}Y = Y^{\mathfrak{A}}$, i.e. precisely agent-indexed functions $f : X \rightsquigarrow Y$. Since composition of morphisms $f : X \to Y$ and $g : Y \to Z$ in $\mathbf{Kleisli}(\mathbb{E})$ is given as $g \circ_{\mathbf{Kleisli}(\mathbb{E})} f = \mathsf{mult}_Z \circ_{\mathbf{Set}} \mathscr{E}g \circ_{\mathbf{Set}} f$, we get that $(g \circ_{\mathbf{Kleisli}(\mathbb{E})} f)(x) = \mathfrak{a} \in \mathfrak{A} \mapsto g(f(x)(\mathfrak{a}))(\mathfrak{a})$ for $x \in X$, which is precisely the composition of g and f in **ASet**. It follows immediately that $\mathbf{Kleisli}(\mathbb{E}) \cong \mathbf{ASet}$.

Note that similarly to how we built a monad out of the functor $\mathscr{E}$, we can naturally build a comonad $\mathbb{D} = \langle \mathscr{D}, \mathbf{counit}, \mathbf{comult} \rangle$ out of the functor $\mathscr{D}X := X \times \mathfrak{A}$. Since $\mathscr{E}$ is right adjoint to $\mathscr{D}$ (in any Cartesian closed category), it follows quite simply that the Kleisli category $\mathbf{Kleisli}(\mathbb{E})$ is isomorphic to to the co-Kleisli category $\mathbf{coKleisli}(\mathbb{D})$, providing us with yet another way to view **ASet**. ◁

For now, we will again consider the question of how to treat agent-indexed Kripke frames and models as coalgebras. Similar to how ordinary Kripke frames are coalgebras for the powerset functor $\mathscr{P}$, agent-indexed Kripke frames are coalgebras for a suitable version of the powerset functor on **ASet**. We define the agent-indexed powerset functor $\mathscr{P}_{\mathfrak{A}} : \mathbf{ASet} \to \mathbf{ASet}$

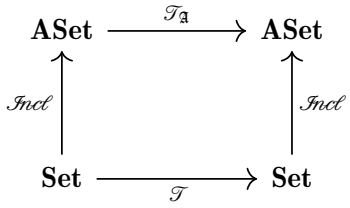
on sets X as $\mathscr{P}_{\mathfrak{A}}X := \mathscr{P}X$. On agent-indexed functions $f : X \rightsquigarrow Y$, we define $\mathscr{P}_{\mathfrak{A}}f : \mathscr{P}_{\mathfrak{A}}X \rightsquigarrow \mathscr{P}_{\mathfrak{A}}Y$ for $\mathfrak{a} \in \mathfrak{A}$ as $\left(\mathscr{P}_{\mathfrak{A}}f\right)_{\mathfrak{a}} := \mathscr{P}f_{\mathfrak{a}}$. It now follows immediately from the definition of agent-indexed functions, and from the fact that diagrams in **ASet** commute iff all of their slices commute in **Set**, that $\mathscr{P}_{\mathfrak{A}}$-coalgebra morphisms are precisely agent-indexed bounded frame morphisms. In other words, we have that $\mathbf{Coalg}_{\mathbf{ASet}}(\mathscr{P}_{\mathfrak{A}}) \cong \mathbf{AgentKF}$.

Note that the way we defined $\mathscr{P}_{\mathfrak{A}}$ relied in no way on the specifics of $\mathscr{P}$. We can similarly define the functor $\mathscr{P}_{\mathfrak{A},\mathsf{Prop}}$ based on the $\mathscr{P}$-model functor $\mathscr{P}_{\mathsf{Prop}}$ as defined in Section 2.2. More generally, we associate with any set functor a corresponding agent-indexed functor, which we refer to as its *agentization*.

Definition 3.2.5. Given a functor $\mathcal{T} : \mathbf{Set} \to \mathbf{Set}$, the *agentization of* $\mathcal{T}$ is the functor $\mathcal{T}_{\mathfrak{A}} : \mathbf{ASet} \to \mathbf{ASet}$ defined on sets X as $\mathcal{T}_{\mathfrak{A}}X := X$, and on agent-indexed functions $f : X \rightsquigarrow Y$ by putting $\left(\mathcal{T}_{\mathfrak{A}}f\right)_{\mathfrak{a}} := \mathcal{T}f_{\mathfrak{a}}$ for each agent $\mathfrak{a} \in \mathfrak{A}$.

Given a $\mathcal{T}_{\mathfrak{A}}$-coalgebra $\mathbb{S} = \langle S, \sigma \rangle$ and agent $\mathfrak{a} \in \mathfrak{A}$, the $\mathfrak{a}$-*slice of* $\mathbb{S}$ is the $\mathcal{T}$-coalgebra $\mathbb{S}_{\mathfrak{a}} := \langle S, \sigma_{\mathfrak{a}} \rangle$. ◁

This definition fits the Slicing Slogan: slices of $\mathcal{T}_{\mathfrak{A}}$-coalgebras are $\mathcal{T}$-coalgebras, and slices of $\mathcal{T}_{\mathfrak{A}}$-coalgebra morphisms are again $\mathcal{T}$-coalgebra morphisms between slices. The agentization $\mathcal{T}_{\mathfrak{A}}$ can also be truly considered an extension of $\mathcal{T}$, in the sense that the diagram



commutes, with $\mathscr{I}\!nc\ell$ being the inclusion functor keeping sets unchanged, and sending functions $f : X \to Y$ to the agent-indexed function defined as $(\mathscr{I}\!nc\ell\, f)_{\mathfrak{a}} := f$ for all $\mathfrak{a} \in \mathfrak{A}$. Even more strongly, we can show that the agentization $\mathcal{T}_{\mathfrak{A}}$ precisely captures the idea that 'multiagent'-structured $\mathcal{T}$-coalgebras should not only consist of $\mathcal{T}$-coalgebras for each agent, but also that conversely, $\mathcal{T}$-coalgebras for each agent determine 'multiagent'-structured $\mathcal{T}$-coalgebras. We formalize this statement by treating the set $\mathfrak{A}$ as the discrete category **Agents** in which agents $\mathfrak{a} \in \mathfrak{A}$ are the objects, and there are no morphisms besides identity morphisms. We then find the following isomorphism, fully in line with the Slicing Slogan.

Proposition 3.2.6. *For any set functor $\mathcal{T}$, denote by* **S** *the full subcategory of the category* $(\mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T}))^{\mathbf{Agents}}$ *of functors from* **Agents** *to* $\mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T})$ *with natural transformations, consisting of all functors $\mathcal{S}$:* **Agents** $\to \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T})$ *for which there is a set S such that $\mathcal{S}\mathfrak{a} = \langle S, \sigma_{\mathfrak{a}} \rangle$ for all $\mathfrak{a} \in \mathfrak{A}$. That is, we only consider $\mathcal{S}$*

that send agents to $\mathcal{T}$-coalgebras with the same carrier set. It then holds that

$$\mathbf{Coalg}_{\mathbf{ASet}}(\mathcal{T}_{\mathfrak{A}}) \cong \mathbf{S},$$

is an isomorphism.

Remark 3.2.7. We note that $\mathcal{T}_{\mathfrak{A}}$-coalgebras are structurally the same as $\mathcal{T}$-coalgebras with input, as considered by Hansen and Klin (2011). The difference lies in our usage of the base category **ASet** as opposed to **Set**. This difference in base category allows us to fully work with the Slicing Slogan through Proposition 3.2.6, as well as through proper 'sliced' notions of coalgebra morphisms and bisimulations, as we will see in Section 3.2.3. ◁

As an aside before moving on towards the matter of defining coalgebraic modal logics, we wish to note that generally speaking, endofunctors $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$ do not arise as agentizations of set functors $\mathcal{T} : \mathbf{Set} \to \mathbf{Set}$. Any $\mathcal{F}$ for which there is an agent-indexed function $f : X \rightsquigarrow Y$ with $\mathfrak{a}, \mathfrak{b} \in \mathfrak{A}$ such that $f_{\mathfrak{a}} = f_{\mathfrak{b}}$ but $\left(\mathcal{F}f\right)_{\mathfrak{a}} \neq \left(\mathcal{F}f\right)_{\mathfrak{b}}$ serves as a counterexample. We will largely not be considering such functors, as they do not play a role in showing how two-valued coalgebraic modal logics extend to Boolean-valued ones. Whenever we give results that *do* work for general endofunctors on **ASet**, we will make this explicit.

3.2.2 Logical Connection and Predicate Liftings

To define a Boolean-valued coalgebraic modal logic for coalgebras over the category **ASet**, we will start with a base logical connection, as in Definition 2.2.1. This is one between **ASet** and **BA**, giving us the base Boolean-valued propositional logic. As is the case for two-valued logic, the logical connection here arises by 'homming into' the dualizing object given by the set 2 and Boolean algebra $\mathfrak{2}$.[6]

Definition 3.2.8. The *Boolean-valued predicate functor* $\mathscr{P}red_{\mathfrak{A}} : \mathbf{ASet}^{\mathrm{op}} \to \mathbf{BA}$ sends sets X to the Boolean algebra $\mathfrak{2}_{\mathfrak{A},X}$ with domain $\mathsf{Exp}_{\mathbf{ASet}}(X, 2)$, i.e. $(2^{\mathfrak{A}})^X$, and operations derived naturally from the operations of the two-element Boolean algebra $\mathfrak{2}$ and the Slicing Slogan (cf. our explanation preceding Definition 3.1.5). It sends agent-indexed functions $f : X \rightsquigarrow Y$ to the Boolean algebra homomorphism $\mathscr{P}red_{\mathfrak{A}}f : \mathfrak{2}_{\mathfrak{A},Y} \to \mathfrak{2}_{\mathfrak{A},X}$ defined by putting $((\mathscr{P}red_{\mathfrak{A}}f)(g))_{\mathfrak{a}} := g_{\mathfrak{a}} \circ f_{\mathfrak{a}}$ for $g : Y \rightsquigarrow 2$ and $\mathfrak{a} \in \mathfrak{A}$.

The *Boolean-valued theory functor* $\mathscr{T}h_{\mathfrak{A}} : \mathbf{BA} \to \mathbf{ASet}^{\mathrm{op}}$ sends Boolean algebras $\mathbb{B}$ to the set $\mathsf{Exp}_{\mathbf{BA}}(\mathbb{B}, \mathfrak{2})$ (or equivalently, to the set of ultrafilters of $\mathbb{B}$). It sends Boolean algebra homomorphisms $f : \mathbb{B} \to \mathbb{C}$ to the agent-indexed function $\mathscr{T}h_{\mathfrak{A}}f : \mathsf{Exp}_{\mathbf{BA}}(\mathbb{C}, \mathfrak{2}) \rightsquigarrow \mathsf{Exp}_{\mathbf{BA}}(\mathbb{B}, \mathfrak{2})$ defined as $\left(\mathscr{T}h_{\mathfrak{A}}f\right)_{\mathfrak{a}}(g) := g \circ f$ for $g \in \mathsf{Exp}_{\mathbf{BA}}(\mathbb{C}, \mathfrak{2})$ and $\mathfrak{a} \in \mathfrak{A}$. ◁

[6] For those readers unfamiliar with the terminology, the phrase 'homming into' an object X refers to considering exponential objects $X^{(-)}$.

It is simple to verify that these data truly define functors.

Proposition 3.2.9. *The Boolean-valued predicate functor $\mathscr{Pred}_{\mathfrak{A}}$ is right adjoint to the Boolean-valued theory functor $\mathscr{Th}_{\mathfrak{A}}$.*

Proof sketch. The details of this proof are largely the same as those of the proof that $\mathscr{Th} \dashv \mathscr{Pred}$ in the two-valued setting. We only define the natural isomorphism $\mathbf{adj} : \mathscr{Hom}_{\mathbf{ASet}}(-, \mathscr{Th}_{\mathfrak{A}}-) \Rightarrow \mathscr{Hom}_{\mathbf{BA}}(-, \mathscr{Pred}_{\mathfrak{A}}-)$ together with its inverse, and do not fully write out the proof that it is natural and an isomorphism. Given a set X and Boolean algebra $\mathbb{B}$, we define $\mathsf{adj}_{X,\mathbb{B}} : \mathscr{Hom}_{\mathbf{ASet}}(X, \mathscr{Th}_{\mathfrak{A}}\mathbb{B}) \to \mathscr{Hom}_{\mathbf{BA}}(\mathbb{B}, \mathscr{Pred}_{\mathfrak{A}}X)$ as

$$\mathsf{adj}_{X,\mathbb{B}}(f) := b \in B \mapsto \left(x \in X \mapsto \left\{\mathfrak{a} \in \mathfrak{A} \,;\, b \in f_{\mathfrak{a}}(x)\right\}\right),$$

where B is the domain of $\mathbb{B}$, f is an agent-indexed function $f : X \rightsquigarrow \mathscr{Th}_{\mathfrak{A}}\mathbb{B}$, and we apply the isomorphism $2^{\mathfrak{A}} \cong \mathcal{P}\mathfrak{A}$ to simplify notation. The inverse function $\mathsf{adj}^{-1}_{X,\mathbb{B}} : \mathscr{Hom}_{\mathbf{BA}}(\mathbb{B}, \mathscr{Pred}_{\mathfrak{A}}X) \to \mathscr{Hom}_{\mathbf{ASet}}(X, \mathscr{Th}_{\mathfrak{A}}\mathbb{B})$ as

$$\left(\mathsf{adj}^{-1}_{X,\mathbb{B}}(f)\right)_{\mathfrak{a}} := x \in X \mapsto \left\{b \in B \,;\, \mathfrak{a} \in f(b)(x)\right\},$$

where $f : \mathbb{B} \to \mathscr{Pred}_{\mathfrak{A}}X$ is a Boolean algebra homomorphism, and we apply $2^B \cong \mathcal{P}B$ to treat the elements of $\mathscr{Th}_{\mathfrak{A}}\mathbb{B}$ as sets/ultrafilters, again to simplify notation. □

Given the logical connection, we could proceed fully analogously to the situation in Section 2.2, defining the resulting general Boolean-valued coalgebraic modal logics through the use of a modal similarity type functor $\mathscr{Sim} : \mathbf{BA} \to \mathbf{BA}$, which is represented by generators through a generator functor $\mathscr{Gen} : \mathbf{BA} \to \mathbf{Set}$, together with one-step semantics $\mathbf{one} : \mathscr{Sim} \circ \mathscr{Pred}_{\mathfrak{A}} \Rightarrow \mathscr{Pred}_{\mathfrak{A}} \circ \mathcal{F}^{\mathrm{op}}_{\mathsf{Prop}}$ for some endofunctor $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$. Since our interest is especially in showing how two-valued coalgebraic modal logics extend to Boolean-valued ones, we will mainly be interested in concretely defining such extended Boolean-valued coalgebraic modal logics.

Consider a modal similarity type $\langle \mathsf{Sym}, \mathsf{ar}\rangle$ consisting of a set Sym of modality symbols and an arity function $\mathsf{ar} : \mathsf{Sym} \to \omega$. We define our generator and signature functors based on this type as follows.

Definition 3.2.10. Given a modal similarity type $\mathbb{Sim} = \langle \mathsf{Sym}, \mathsf{ar}\rangle$, the $\mathbb{Sim}$-*generator functor* $\mathscr{Gen} : \mathbf{BA} \to \mathbf{Set}$ sends Boolean algebras $\mathbb{B}$ with underlying sets B to

$$\mathscr{Gen}\mathbb{B} := \left\{\heartsuit(\boldsymbol{b}) \,;\, \heartsuit \in \mathsf{Sym}, \boldsymbol{b} \in B^{\mathsf{ar}\heartsuit}\right\} + \left\{\ulcorner \mathfrak{B} \urcorner \,;\, \mathfrak{B} \subseteq \mathfrak{A}\right\} + \mathsf{Prop},$$

where $\heartsuit(\boldsymbol{b})$ is purely a syntactic construction. The $\mathbb{Sim}$-*functor* $\mathscr{Sim} : \mathbf{BA} \to \mathbf{BA}$ is then defined as $\mathscr{Sim} = \mathscr{Free}_{\mathbf{BA}} \circ \mathscr{Gen}$, where $\mathscr{Free}_{\mathbf{BA}} : \mathbf{Set} \to \mathbf{BA}$ is the free Boolean algebra functor. ◁

The one-step semantics of the logic can now be given by predicate liftings for each modality symbol. As might be expected, these predicate liftings are Boolean-valued, as opposed to two-valued. Two-valued predicate liftings can however be simply extended to *Booleanized* Boolean-valued ones.

Definition 3.2.11. An *n-ary Boolean-valued predicate $\mathcal{F}$-lifting* for a functor $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$ is a natural transformation $\mathbf{lift} : \left(\mathcal{F}\mathit{org}_{\mathbf{BA},\mathbf{Set}} \circ \mathcal{P}\mathit{red}_{\mathfrak{A}}\right)^n \Rightarrow \mathcal{F}\mathit{org}_{\mathbf{BA},\mathbf{Set}} \circ \mathcal{P}\mathit{red}_{\mathfrak{A}} \circ \mathcal{F}$, where $\mathcal{F}\mathit{org}_{\mathbf{BA},\mathbf{Set}}$ is the obvious forgetful functor from **BA** to **Set**.

Given an n-ary (two-valued) predicate $\mathcal{T}$-lifting **lift** (i.e. a natural transformation $\mathbf{lift} : \left(\mathcal{F}\mathit{org}_{\mathbf{BA},\mathbf{Set}} \circ \mathcal{P}\mathit{red}\right)^n \Rightarrow \mathcal{F}\mathit{org}_{\mathbf{BA},\mathbf{Set}} \circ \mathcal{P}\mathit{red} \circ \mathcal{T}$ for a set functor $\mathcal{T}$), the *Booleanization of* **lift** is the n-ary Boolean-valued predicate $\mathcal{T}_{\mathfrak{A}}$-lifting $\mathbf{lift}_{\mathfrak{A}}$ defined over a set X as

$$\left((\mathsf{lift}_{\mathfrak{A}})_X\left(q_1, \ldots, q_n\right)\right)_{\mathfrak{a}} := \mathsf{lift}_X(q_{1,\mathfrak{a}}, \ldots, q_{n,\mathfrak{a}})$$

for $\left\langle q_1, \ldots, q_n \right\rangle \in \left((2^{\mathfrak{A}})^X\right)^n$ and $\mathfrak{a} \in \mathfrak{A}$. ◁

So n-ary Boolean-valued predicate $\mathcal{F}$-liftings **lift** are mappings sending n Boolean-valued predicates $q_i : S \to \mathcal{P}\mathfrak{A}$ on a set S to a Boolean-valued predicate $\mathsf{lift}_S(q_1, \ldots, q_n) : \mathcal{T}S \to \mathcal{P}\mathfrak{A}$ over the set $\mathcal{F}S$ of $\mathcal{F}$-unfoldings of S. And Booleanizations of two-valued predicate $\mathcal{T}$-liftings are defined neatly in accordance with the Slicing Slogan: the $\mathfrak{a}$-slice of the Boolean-valued predicate $(\mathsf{lift}_{\mathfrak{A}})_X\left(q_1, \ldots, q_n\right)$ is the predicate $\mathsf{lift}_X(q_{1,\mathfrak{a}}, \ldots, q_{n,\mathfrak{a}})$ obtained from the $\mathfrak{a}$-slices $q_{1,\mathfrak{a}}, \ldots, q_{n,\mathfrak{a}}$.

Example 3.2.12. Consider the two-valued predicate lifting $\mathcal{P}$-lifting $\mathbf{lift}^{\Diamond}$ for the $\Diamond$-modality of basic modal logic, defined (using $2^S \cong \mathcal{P}S$) as $\mathsf{lift}^{\Diamond}_S(Q) := \{U \in \mathcal{P}S \,;\, U \cap Q \neq \varnothing\}$ for $Q \in \mathcal{P}S$. Then its Booleanization is the Boolean-valued predicate $\mathcal{P}_{\mathfrak{A}}$-lifting $\mathbf{lift}^{\Diamond}_{\mathfrak{A}}$ mapping Boolean-valued predicates $q \in (\mathcal{P}\mathfrak{A})^S$ over S to the Boolean-valued predicate

$$\left(\mathsf{lift}^{\Diamond}_{\mathfrak{A}}\right)_S(q) = U \in \mathcal{P}S \mapsto \left\{\mathfrak{a} \in \mathfrak{A} \,;\, U \cap \widehat{q}(\mathfrak{a}) \neq \varnothing\right\}.$$

Though superficially similar to the definition of the two-valued $\Diamond$-modality, it might not be fully clear now that this lifting will indeed capture the semantics of the $\Diamond$-modality in Boolean-valued basic modal logic, and will need to wait until Theorem 3.2.19. ◁

Putting together modal similarity types $\langle \mathsf{Sym}, \mathsf{ar} \rangle$ and a family $\mathbf{Lift} = \left\langle \mathbf{lift}^{\heartsuit} \right\rangle_{\heartsuit \in \mathsf{Sym}}$ of $(\mathsf{ar}\heartsuit)$-ary Boolean-valued predicate $\mathcal{F}$-liftings for each $\heartsuit \in \mathsf{Sym}$, we have all we need to fully specify the Boolean-valued coalgebraic modal logic through its one-step semantics. We refer to such pairings $\langle \langle \mathsf{Sym}, \mathsf{ar} \rangle, \mathbf{Lift} \rangle$ as *Boolean-valued coalgebraic modal logics*, or if we wish to make the coalgebra type specific, Boolean-valued coalgebraic modal logics over $\mathcal{F}$. Note that we wish to interpret these logics over $\mathcal{F}$-models, that is, $\mathcal{F}_{\mathsf{Prop}}$-coalgebras.

Definition 3.2.13. Given a Boolean-valued coalgebraic modal logic $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ over $\mathscr{F}$, the *one-step semantics of* $\mathbb{Log}$ is the natural transformation $\mathbf{one}^{\mathbb{Log}} : \mathscr{Sim} \circ \mathscr{Pred}_{\mathfrak{A}} \Rightarrow \mathscr{Pred}_{\mathfrak{A}} \circ \mathscr{F}^{\mathrm{op}}_{\mathsf{Prop}}$ defined for a set S as the unique Boolean algebra homomorphism induced by the function

$$\mathsf{one}'_S : \mathscr{Gen}\mathscr{Pred}_{\mathfrak{A}} S \to \mathscr{Forg}_{\mathbf{BA},\mathbf{Set}} \mathscr{Pred}_{\mathfrak{A}} \mathscr{F}_{\mathsf{Prop}} S$$

defined as

$$\begin{aligned}
\mathsf{one}'_S(\heartsuit(\boldsymbol{q})) &:= \langle U, P \rangle \in \mathscr{F}S \times \mathscr{P}\mathsf{Prop} \mapsto \mathsf{lift}^{\heartsuit}_S(\boldsymbol{q})(U), \\
&\qquad (\heartsuit \in \mathsf{Sym},\ \boldsymbol{q} \in (\mathscr{Pred}_{\mathfrak{A}} S)^{\mathsf{ar}\heartsuit}) \\
\mathsf{one}'_S(\ulcorner \mathfrak{B} \urcorner) &:= \langle U, P \rangle \in \mathscr{F}S \times \mathscr{P}\mathsf{Prop} \mapsto \mathsf{char}^{\mathfrak{B}}, \\
&\qquad (\mathsf{char}^{\mathfrak{B}} \text{ is characteristic of } \mathfrak{B} \subseteq \mathfrak{A}) \\
\mathsf{one}'_S(p) &:= \langle U, P \rangle \in \mathscr{F}S \times \mathscr{P}\mathsf{Prop} \mapsto \left(\mathfrak{a} \in \mathfrak{A} \mapsto \mathsf{char}^{P}(p)\right), \\
&\qquad (\mathsf{char}^{P} \text{ is characteristic of } P \subseteq \mathsf{Prop})
\end{aligned}$$

using the universal property of free Boolean algebras. ◁

As was the case for two-valued coalgebraic modal logic, it follows quite simply from the naturality of $\mathbf{lift}^{\heartsuit}$ for each $\heartsuit \in \mathsf{Sym}$ that **one** is in fact a natural transformation. And so we can again define an algebraification functor, like in Definition 2.2.2.

Definition 3.2.14. Given a Boolean-valued coalgebraic modal logic $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ over an endofunctor $\mathscr{F}$, the $\mathbb{Log}$*-algebraification functor* $\mathscr{Alg}_{\mathbb{Log}} : \mathbf{Coalg}_{\mathbf{ASet}}(\mathscr{F}_{\mathsf{Prop}})^{\mathrm{op}} \to \mathbf{Alg}_{\mathbf{BA}}(\mathscr{Sim})$ sends $\mathscr{F}$-models (i.e. $\mathscr{F}_{\mathsf{Prop}}$-coalgebras) $\mathbb{S} = \langle S, \sigma \rangle$ to the $\mathscr{Sim}$-algebra

$$\mathscr{Alg}_{\mathbb{Log}} \mathbb{S} := \left\langle \mathscr{Pred}_{\mathfrak{A}} S, \mathscr{Pred}_{\mathfrak{A}} \sigma \circ \mathsf{one}^{\mathbb{Log}}_S \right\rangle,$$

It sends $\mathscr{F}_{\mathsf{Prop}}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{S}'$ to the $\mathscr{Sim}$-algebra morphism $\mathscr{Alg}_{\mathbb{Log}} f := \mathscr{Pred}_{\mathfrak{A}} f$. ◁

As was also the case in the two-valued setting, the category $\mathbf{Alg}_{\mathbf{BA}}(\mathscr{Sim})$ has an initial algebra $\mathbb{L}\mathrm{ang}_{\mathbb{Sim},\mathfrak{A}} = \left\langle \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}, \lambda \right\rangle$ with its carrier being the obvious Boolean algebra built on the set of formulas φ[7] over the modal similarity type $\mathbb{Sim}$ along with propositional constants $\ulcorner \mathfrak{B} \urcorner$, and the obvious Boolean algebra homomorphism sending elements of $\mathscr{Sim}\mathsf{Lang}_{\mathbb{Sim}}$ to the formulas they represent as its transition map.

Definition 3.2.15. Given a modal similarity type $\mathbb{Sim} = \langle \mathsf{Sym}, \mathsf{ar} \rangle$, the *Boolean-valued language of* $\mathbb{Sim}$ is the set $\mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$ underlying the carrier Boolean

[7] Though technically, $\mathsf{Lang}_{\mathbb{Sim}}$ consists of equivalence classes $[\varphi]$ of formulas with respect to the axioms of Boolean algebras, we will always work with it as if it were just the set of formulas φ.

algebra of the initial $\mathscr{Sim}$-algebra $\mathbb{Lang}_{\mathbb{Sim},\mathfrak{A}}$. The language can be inductively defined (up to using standard abbreviations) as

$$\mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}} \ni \varphi ::= p \mid \ulcorner \mathfrak{B} \urcorner \mid (\varphi \vee \varphi) \mid (\neg\varphi) \mid (\heartsuit(\underbrace{\varphi, \ldots, \varphi}_{\mathsf{ar}\heartsuit \text{ times}})),$$

where $p \in \mathsf{Prop}$, $\mathfrak{B} \subseteq \mathfrak{A}$, and $\heartsuit \in \mathsf{Sym}$. ◁

The unique $\mathscr{Sim}$-algebra morphisms from $\mathbb{Lang}_{\mathbb{Sim},\mathfrak{A}}$ then finally give the semantics of the logic, using algebraifications. So we can give the definition of the logic in full now.

Definition 3.2.16. Given a Boolean-valued coalgebraic modal logic $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ over $\mathcal{F}$, together with an $\mathcal{F}$-model $\mathbb{S} = \langle S, \sigma \rangle$, the *semantics of* $\mathbb{Log}$ is the unique $\mathscr{Sim}$-algebra morphism $[\![-]\!]^{\mathbb{Log}}_{\mathbb{S}} : \mathbb{Lang}_{\mathbb{Sim},\mathfrak{A}} \to \mathscr{Alg}_{\mathbb{Log}}\mathbb{S}$. Forgetting the Boolean algebra structure and applying $2^{\mathfrak{A}} \cong \mathscr{P}\mathfrak{A}$, we will often view the semantics as a function $[\![-]\!]^{\mathbb{Log}}_{\mathbb{S}} : \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$. ◁

Working out the semantics, we find the following inductive characterization.

Proposition 3.2.17. *Let $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ be a Boolean-valued coalgebraic modal logic over $\mathcal{F}$, and $\mathbb{S} = \langle S, \sigma, \mathsf{col} \rangle$ be an $\mathcal{F}$-model.*[8] *The semantics of $\mathbb{Log}$ satisfies*

$$\begin{aligned}
[\![p]\!]^{\mathbb{Log}}_{\mathbb{S}}(s) &= \widehat{\mathsf{col}(s)}\,(p), && (p \in \mathsf{Prop}) \\
[\![\ulcorner \mathfrak{B} \urcorner]\!]^{\mathbb{Log}}_{\mathbb{S}}(s) &= \mathfrak{B}, \\
[\![\varphi \vee \psi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s) &= [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s) \cup [\![\psi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s), \\
[\![\neg\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s) &= \mathfrak{A} \smallsetminus [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s), \text{ and} \\
[\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})]\!]^{\mathbb{Log}}_{\mathbb{S}} &= (\mathscr{P}\!red_{\mathfrak{A}}\sigma)\left(\mathsf{lift}^{\heartsuit}_S\left([\![\varphi_1]\!]^{\mathbb{Log}}_{\mathbb{S}}, \ldots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]^{\mathbb{Log}}_{\mathbb{S}}\right)\right)
\end{aligned}$$

for all $s \in S$.

Proof. By definition of the algebraification functor (Definition 3.2.14) and the fact that $[\![-]\!]^{\mathbb{Log}}_{\mathbb{S}}$ is a $\mathscr{Sim}$-algebra morphism, the square in the diagram

$$\begin{array}{ccc}
\mathscr{Sim}\,\mathsf{Lang}_{\mathbb{Sim}} & \xrightarrow{\mathscr{Sim}[\![-]\!]^{\mathbb{Log}}_{\mathbb{S}}} & \mathscr{Sim}\,\mathscr{P}\!red_{\mathfrak{A}}S \\
\downarrow{\scriptstyle \lambda} & & \downarrow{\scriptstyle \mathsf{one}^{\mathbb{Log}}_S} \\
 & & \mathscr{P}\!red_{\mathfrak{A}}\mathcal{F}_{\mathsf{Prop}}S \qquad \mathcal{F}_{\mathsf{Prop}}S \\
 & & \downarrow{\scriptstyle \mathscr{P}\!red_{\mathfrak{A}}\sigma'} \qquad \uparrow{\scriptstyle \sigma'} \\
\mathsf{Lang}_{\mathbb{Sim}} & \xrightarrow[[\![-]\!]^{\mathbb{Log}}_{\mathbb{S}}]{} & \mathscr{P}\!red_{\mathfrak{A}}S \qquad S
\end{array}$$

[8] As also done in Section 2.2, we will often denote $\mathcal{F}$-models by triples $\langle S, \sigma, \mathsf{col} \rangle$, where $\sigma : S \rightsquigarrow \mathcal{F}S$ and $\mathsf{col} : S \rightsquigarrow \mathscr{P}\mathsf{Prop}$.

commutes in **BA**. Considering a propositional variable p, we thus get by definition of the one-step semantics (Definition 3.2.13) and the Boolean-valued predicate functor (Definition 3.2.8) that

$$\begin{aligned}
[\![p]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}} &= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma')\left(\mathsf{one}_S^{\mathbb{L}\mathbb{o}\mathbb{g}}(p)\right)\\
&= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma')\left(f\right)\\
&\quad (\text{where } f = \langle U, P\rangle \in \mathcal{F}S \times \mathcal{P}\mathsf{Prop} \mapsto \left(\mathfrak{a} \in \mathfrak{A} \mapsto \mathsf{char}^P(p)\right))\\
&= s \in S \mapsto \left(\mathfrak{a} \in \mathfrak{A} \mapsto (\widehat{f})_{\mathfrak{a}}(\sigma'_{\mathfrak{a}}(s))\right)\\
&= s \in S \mapsto \left(\mathfrak{a} \in \mathfrak{A} \mapsto \mathsf{char}^{\mathsf{col}_{\mathfrak{a}}(s)}(p)\right)\\
&= s \in S \mapsto \left\{\mathfrak{a} \in \mathfrak{A}\,;\, p \in \mathsf{col}_{\mathfrak{a}}(s)\right\}\\
&= s \in S \mapsto \widehat{\mathsf{col}(s)}\,(p).
\end{aligned}$$

Similarly, we find for $\mathfrak{B} \subseteq \mathfrak{A}$ that

$$\begin{aligned}
[\![\ulcorner\mathfrak{B}\urcorner]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}} &= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma')\left(\mathsf{one}_S^{\mathbb{L}\mathbb{o}\mathbb{g}}(\ulcorner\mathfrak{B}\urcorner)\right)\\
&= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma')\left(f\right) \quad (\text{where } f = \langle U, P\rangle \in \mathcal{F}S \times \mathcal{P}\mathsf{Prop} \mapsto \mathsf{char}^{\mathfrak{B}})\\
&= s \in S \mapsto \left(\mathfrak{a} \in \mathfrak{A} \mapsto \mathsf{char}^{\mathfrak{B}}(\mathfrak{a})\right)\\
&= s \in S \mapsto \mathfrak{B}.
\end{aligned}$$

The statements for the semantics of $\varphi \vee \psi$ and $\neg\varphi$ follow directly from the definition of the Boolean operations on $\mathscr{P}\!\mathit{red}_{\mathfrak{A}}S$, together with the fact that $[\![-]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}} : \mathsf{Lang}_{\mathbb{S}\mathrm{im},\mathfrak{A}} \to \mathscr{P}\!\mathit{red}_{\mathfrak{A}}S$ is a Boolean algebra homomorphism. The statement for $\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})$ is also simple to prove. We use the abbreviation $[\![\boldsymbol{\varphi}]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}} = \left\langle [\![\varphi_1]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}}, \ldots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}}\right\rangle$. We then find that

$$\begin{aligned}
[\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}} &= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma')\left(\mathsf{one}_S^{\mathbb{L}\mathbb{o}\mathbb{g}}(\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit}))\right)\\
&= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma')\left(f\right)\\
&\quad (\text{where } f = \langle U, P\rangle \in \mathcal{F}S \times \mathcal{P}\mathsf{Prop} \mapsto \mathsf{lift}_S^{\heartsuit}([\![\boldsymbol{\varphi}]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}})(U))\\
&= s \in S \mapsto \left(\mathfrak{a} \in \mathfrak{A} \mapsto \mathsf{lift}_S^{\heartsuit}([\![\boldsymbol{\varphi}]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}})(\sigma_{\mathfrak{a}}(s))(\mathfrak{a})\right)\\
&= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma)\left(\mathsf{lift}_S^{\heartsuit}\left([\![\boldsymbol{\varphi}]\!]^{\mathbb{L}\mathbb{o}\mathbb{g}}_{\mathbb{S}}\right)\right). \qquad \square
\end{aligned}$$

Let us now at long last reconsider the matter of extending two-valued coalgebraic modal logics. Using Booleanizations of two-valued predicate liftings, we can associate with any two-valued coalgebraic modal logic over a set functor $\mathcal{T}$, a corresponding Boolean-valued coalgebraic modal logic over the agentization $\mathcal{T}_{\mathfrak{A}}$, which we again refer to as its *Booleanization*.

Definition 3.2.18. For a two-valued coalgebraic modal logic $\mathbb{Log} = \langle\langle \mathsf{Sym}, \mathsf{ar}\rangle, \mathbf{Lift}\rangle$ over a set functor $\mathcal{T}$[9], the *Booleanization of* $\mathbb{Log}$ is the Boolean-valued coalgebraic modal logic $\mathbb{Log}_{\mathfrak{A}} = \langle\langle \mathsf{Sym}, \mathsf{ar}\rangle, \mathbf{Lift}_{\mathfrak{A}}\rangle$ over $\mathcal{T}_{\mathfrak{A}}$, where $\mathbf{Lift}_{\mathfrak{A}} = \left\langle \mathbf{lift}^{\heartsuit}_{\mathfrak{A}} \right\rangle_{\heartsuit \in \mathsf{Sym}}$ consists of the Booleanizations of the liftings in $\mathbf{Lift}$. ◁

We can now show that Booleanizations of two-valued predicate liftings (and two-valued coalgebraic modal logics) are indeed proper generalizations of the original liftings (and coalgebraic modal logics) in accordance with the Slicing Slogan. We do this through a general Coalgebraic Slicing Theorem (cf. Theorems 3.1.6 and 3.1.8). Recall the definition of the translations $\mathsf{tr}_{\mathfrak{a}}$ for $\mathfrak{a} \in \mathfrak{A}$ as used in Theorem 3.1.8, and note that we can also define translations $\mathsf{tr}_{\mathfrak{a}} : \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}} \to \mathsf{Lang}_{\mathbb{Sim}}$ taking formulas in the Boolean-valued language to formulas in the two-valued one, by mapping constants $\ulcorner \mathfrak{B} \urcorner$ to either $\top$ or $\bot$, dependent on whether $\mathfrak{a} \in \mathfrak{B}$ or not. Using the translation, we can state the following theorem.

Theorem 3.2.19 (Coalgebraic Slicing Theorem)**.** *Let* $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift}\rangle$ *be a two-valued coalgebraic modal logic over* $\mathcal{T}$*, and* $\mathbb{S}$ *a* $\mathcal{T}_{\mathfrak{A}}$*-model. It holds that*

$$\widehat{[\![\varphi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}}}(\mathfrak{a}) = [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]^{\mathbb{Log}}_{\mathbb{S}_{\mathfrak{a}}}$$

for all $\varphi \in \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$ *and* $\mathfrak{a} \in \mathfrak{A}$*.*

Proof. The theorem follows almost immediately from Proposition 3.2.17, together with the definition of the Booleanization of a two-valued predicate lifting (Definition 3.2.11). □

Example 3.2.20. Consider the two-valued coalgebraic modal logic $\mathbb{Log}^{\mathrm{BML}}$ over $\mathcal{P}$ corresponding to two-valued basic modal logic with the $\Diamond$-modality. Then it follows almost immediately from Theorems 3.1.8 and 3.2.19 that the Booleanization $\mathbb{Log}^{\mathrm{BML}}_{\mathfrak{A}}$ corresponds precisely to the Boolean-valued basic modal logic from *How True* and Section 3.1, both syntactically and semantically. ◁

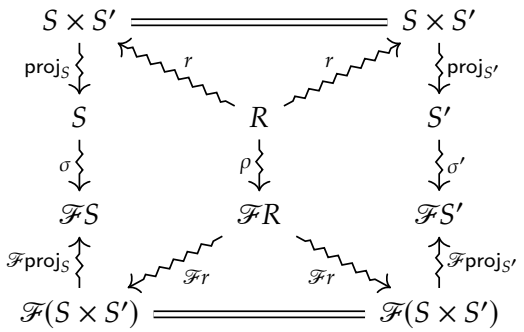
3.2.3 Bisimulations and Behavioural Equivalence

Having worked out how to generalize arbitrary two-valued coalgebraic modal logics (over functors $\mathcal{T} : \mathbf{Set} \to \mathbf{Set}$) to Boolean-valued ones (over functors $\mathcal{T}_{\mathfrak{A}} : \mathbf{ASet} \to \mathbf{ASet}$) in accordance with the Slicing Slogan, it is of interest to verify whether our definitions are able to properly capture general agent-indexed coalgebraic bisimulations and their relations to the logic. In

[9] We assume for the sake of simplicity that the generator functor for $\langle \mathsf{Sym}, \mathsf{ar}\rangle$ does not include propositional variables, as this would introduce unnecessary complexity in describing the formulas of the Booleanization. We hence also assume that $\mathcal{T}$ is not a model functor (i.e. $\mathcal{T} \neq \mathcal{T}'_{\mathsf{Prop}}$ for all set functors $\mathcal{T}'$).

doing so, it is natural to also consider similar questions about the notion of behavioural equivalence.

Let us start generally with coalgebraic bisimulations on $\mathscr{F}$-coalgebras for some $\mathscr{F} : \mathbf{ASet} \to \mathbf{ASet}$. Extending the definition of coalgebraic bisimulations we gave in the Definition 2.1.4, bisimulations between $\mathscr{F}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$ are built on internal binary relations $\langle R, p_S, p_{S'} \rangle$ in **ASet** between S and S'. These are sets R such that $p_S : R \rightsquigarrow S$ and $p_{S'} : R \rightsquigarrow S'$ are jointly monic — that is, for all $f, g : X \rightsquigarrow R$ it holds that if $p_S \circ f = p_S \circ g$ and $p_{S'} \circ f = p_{S'} \circ g$, then $f = g$. Since **ASet** is complete by Proposition 3.2.3, it has binary products, and thus we can more simply describe such internal binary relations in **ASet** as subobjects R of the **ASet**-product $S \times S'$, which are monic agent-indexed functions $r : R \rightsquigarrow S \times S'$, with $p_S = \mathsf{proj}_S \circ r$ and $p_{S'} = \mathsf{proj}_{S'} \circ r$. To make a bisimulation $R : \mathbb{S} \leftrightarrows \mathbb{S}'$ out of such subobjects, we equip it with $\mathscr{F}$-coalgebra structure $\rho : R \rightsquigarrow \mathscr{T}_{\mathfrak{A}} R$ such that p_S and $p_{S'}$ are $\mathscr{T}_{\mathfrak{A}}$-coalgebra morphisms. So putting this all together, a coalgebraic bisimulation $R : \mathbb{S} \leftrightarrows \mathbb{S}'$ between $\mathbb{S}$ and $\mathbb{S}'$ is a subobject $r : R \rightsquigarrow S \times S'$ such that the diagram



(3.2)

commutes.

Note the fact that **ASet** has subobject classifiers, as shown in Proposition 3.2.3. Because of this, it holds that subobjects $i : U \rightsquigarrow X$ of a set X are in a one-to-one correspondence with agent-indexed functions $f : X \rightsquigarrow 2$. So each slice $i_{\mathfrak{a}} : U \to X$ can be identified with a function $f_{\mathfrak{a}} : X \to 2$ determining a subset of X. We can then identify subobjects $i : U \rightsquigarrow X$ as *agent-indexed subsets of* X consisting of subsets $U_{\mathfrak{a}}$ of X for each agent $\mathfrak{a} \in \mathfrak{A}$. And thus using $(\mathscr{P}(S \times S'))^{\mathfrak{A}} \cong (\mathscr{P}\mathfrak{A})^{S \times S'}$, subobjects $r : R \rightsquigarrow S \times S'$ correspond precisely to agent-indexed relations $R : S \times S' \to \mathscr{P}\mathfrak{A}$. Since diagrams in **ASet** commute if the **Set**-diagram obtained by taking $\mathfrak{a}$-slices of all agent-indexed functions commutes for all $\mathfrak{a} \in \mathfrak{A}$, we can verify by slicing Diagram (3.2) that we can give the following definition of bisimilarity in **ASet**.

Definition 3.2.21. Given $\mathscr{F} : \mathbf{ASet} \to \mathbf{ASet}$ and $\mathscr{F}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$, we say that states $s \in S$ and $s' \in S'$ are $\mathfrak{B}$-*bisimilar* (and

we write $\mathbb{S}, s \leftrightarrows_{\mathfrak{B}} \mathbb{S}', s'$) for $\mathfrak{B} \subseteq \mathfrak{A}$ if there is an agent-indexed relation $B : S \times S' \to \mathcal{P}\mathfrak{A}$ that, viewed as an agent-indexed subset of $S \times S'$, is a bisimulation $B : \mathbb{S} \leftrightarrows \mathbb{S}'$ with $\mathfrak{B} \subseteq B(s, s')$. If $\mathfrak{B}$ is a singleton $\{\mathfrak{a}\}$, we will leave out the brackets, writing $\leftrightarrows_{\mathfrak{a}}$ instead of $\leftrightarrows_{\{\mathfrak{a}\}}$. ◁

We have that coalgebraic bisimulations on **ASet** for agentized functors admit a 'sliced' (i.e. agent-indexed) characterization similar to that of bisimulations for agent-indexed Kripke models (Definition 3.1.9), which are defined as agent-indexed relations $R : S \times S' \to \mathcal{P}\mathfrak{A}$ between state spaces S and S' of $\mathbb{M}$ and $\mathbb{M}'$, such that $R_{\mathfrak{a}} : \mathbb{M}_{\mathfrak{a}} \leftrightarrows \mathbb{M}'_{\mathfrak{a}}$ for all agents $\mathfrak{a} \in \mathfrak{A}$.

Proposition 3.2.22. *Let $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$ be $\mathcal{T}_{\mathfrak{A}}$-coalgebras for some set functor $\mathcal{T} : \mathbf{Set} \to \mathbf{Set}$. Then $\mathbb{S}, s \leftrightarrows_{\mathfrak{B}} \mathbb{S}', s'$ iff $\mathbb{S}_{\mathfrak{a}}, s \leftrightarrows \mathbb{S}'_{\mathfrak{a}}, s'$ for all agents $\mathfrak{a} \in \mathfrak{B}$.*

Proof. The direction from left to right follows immediately from the definition of bisimulations and the fact that diagrams in **ASet** commute iff all their slices commute in **Set**. The direction from right to left follows by noting that given bisimulations $B_{\mathfrak{a}} : \mathbb{S}_{\mathfrak{a}}, s \leftrightarrows \mathbb{S}'_{\mathfrak{a}}, s'$ for all $\mathfrak{a} \in \mathfrak{B}$, we can construct an agent-indexed bisimulation $B : \mathbb{S}, s \leftrightarrows_{\mathfrak{B}} \mathbb{S}', s'$ by using the $B_{\mathfrak{a}}$ as slices, while setting $\mathfrak{b}$-slices $B_{\mathfrak{b}}$ for $\mathfrak{b} \in \mathfrak{A} \setminus \mathfrak{B}$ to be empty. □

So coalgebraic bisimulations for agentized functors truly generalize the agent-indexed bisimulations as we saw them in Definition 3.1.9.

Considering the coalgebraic notion of behavioural equivalence given in Definition 2.1.3, we can also give a 'sliced' characterization: two states are behaviourally equivalent iff they are behaviourally equivalent in all slices. To do this, we first need to settle on what it generally means for two states to be behaviourally equivalent for some agents.

Definition 3.2.23. Given an endofunctor $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$ together with $\mathcal{F}$-coalgebras $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$, we say states $s \in S$ and $s' \in S'$ are *behaviourally $\mathfrak{B}$-equivalent* (and we write $\mathbb{S}, s \simeq_{\mathfrak{B}} \mathbb{S}', s'$) for $\mathfrak{B} \subseteq \mathfrak{A}$ if there is some $\mathcal{F}$-coalgebra $\mathbb{X}$ with $\mathcal{F}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$ such that $f_{\mathfrak{a}}(s) = f'_{\mathfrak{a}}(s')$ for all $\mathfrak{a} \in \mathfrak{B}$. If $\mathfrak{B}$ is a singleton $\{\mathfrak{a}\}$, we will leave out the brackets, writing $\simeq_{\mathfrak{a}}$ instead of $\simeq_{\{\mathfrak{a}\}}$. ◁

Clearly $\mathbb{S}, s \simeq \mathbb{S}', s'$ iff $\mathbb{S}, s \simeq_{\mathfrak{A}} \mathbb{S}', s'$, showing that this sliced definition properly generalizes the standard definition of behavioural equivalence on $\mathcal{F}$-coalgebras.

Using the definition, we can now give the 'sliced' characterization of behavioural equivalence.

Proposition 3.2.24. *Let $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$ be $\mathcal{T}_{\mathfrak{A}}$-coalgebras for some $\mathcal{T} : \mathbf{Set} \to \mathbf{Set}$. Then $\mathbb{S}, s \simeq_{\mathfrak{B}} \mathbb{S}', s'$ iff $\mathbb{S}_{\mathfrak{a}}, s \simeq \mathbb{S}'_{\mathfrak{a}}, s'$ for all agents $\mathfrak{a} \in \mathfrak{B}$.*

Proof. We prove the specific case where $\mathfrak{B} = \mathfrak{A}$ for the sake of simplicity. The case where $\mathfrak{B} \neq \mathfrak{A}$ is proven virtually identically, with some additional uninformative bookkeeping required to deal with agents $\mathfrak{a} \notin \mathfrak{B}$, and it is not difficult to derive the proof of the general statement from this specific case.

The direction from left to right is trivial, and follows directly from the fact that diagrams in **ASet** commute iff all slices of the diagram commute in **Set**: if $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$ are $\mathcal{T}_{\mathfrak{A}}$-coalgebra morphisms with $f(s) = f'(s')$, then $f_{\mathfrak{a}} : \mathbb{S}_{\mathfrak{a}} \to \mathbb{X}_{\mathfrak{a}}$ and $f'_{\mathfrak{a}} : \mathbb{S}'_{\mathfrak{a}} \to \mathbb{X}_{\mathfrak{a}}$ are $\mathcal{T}$-coalgebra morphisms with $f_{\mathfrak{a}}(s) = f'_{\mathfrak{a}}(s')$ for all $\mathfrak{a} \in \mathfrak{A}$.

The direction from right to left is still simple, though a bit more involved. Suppose there exist $\mathcal{T}$-coalgebras $\mathbb{Z}_{\mathfrak{a}} = \langle Z_{\mathfrak{a}}, \zeta_{\mathfrak{a}} \rangle$ and $\mathcal{T}$-coalgebra morphisms $f_{\mathfrak{a}} : \mathbb{S}_{\mathfrak{a}} \to \mathbb{Z}_{\mathfrak{a}}$ and $f'_{\mathfrak{a}} : \mathbb{S}'_{\mathfrak{a}} \to \mathbb{Z}_{\mathfrak{a}}$ satisfying $f_{\mathfrak{a}}(s) = f'_{\mathfrak{a}}(s')$ for all $\mathfrak{a} \in \mathfrak{A}$. Consider the **Set**-coproduct[10] $\sum_{\mathfrak{a}\in\mathfrak{A}} \mathbb{Z}_{\mathfrak{a}}$, which is the $\mathcal{T}$-coalgebra $\langle \sum_{\mathfrak{a}\in\mathfrak{A}} Z_{\mathfrak{a}}, \zeta \rangle$ with ζ being the unique morphism making the diagrams

$$\begin{array}{ccc} Z_{\mathfrak{b}} & \xrightarrow{\mathsf{inj}_{\mathfrak{b}}} & \sum_{\mathfrak{a}\in\mathfrak{A}} Z_{\mathfrak{a}} \\ {\scriptstyle \zeta_{\mathfrak{a}}}\downarrow & & \downarrow{\scriptstyle \zeta} \\ \mathcal{T}Z_{\mathfrak{b}} & \xrightarrow[\mathcal{T}\mathsf{inj}_{\mathfrak{b}}]{} & \mathcal{T}\sum_{\mathfrak{a}\in\mathfrak{A}} Z_{\mathfrak{a}} \end{array} \tag{3.3}$$

for all $\mathfrak{b} \in \mathfrak{A}$ commute, guaranteed to exist by the universal property of the coproduct. We construct a $\mathcal{T}_{\mathfrak{A}}$-coalgebra $\mathbb{X} = \langle \sum_{\mathfrak{a}\in\mathfrak{A}} Z_{\mathfrak{a}}, \xi \rangle$ by putting $\xi_{\mathfrak{a}} := \zeta$ for all $\mathfrak{a} \in \mathfrak{A}$. As $f_{\mathfrak{a}} : \mathbb{S}_{\mathfrak{a}} \to \mathbb{Z}_{\mathfrak{a}}$ is a $\mathcal{T}$-coalgebra morphism for all $\mathfrak{a} \in \mathfrak{A}$, we have that the diagram

$$\begin{array}{ccc} S & \xrightarrow{f_{\mathfrak{a}}} & Z_{\mathfrak{b}} \\ {\scriptstyle \sigma_{\mathfrak{a}}}\downarrow & & \downarrow{\scriptstyle \zeta_{\mathfrak{a}}} \\ \mathcal{T}S & \xrightarrow[\mathcal{T}f_{\mathfrak{a}}]{} & \mathcal{T}Z_{\mathfrak{b}} \end{array} \tag{3.4}$$

always commutes. Putting Diagrams (3.3) and (3.4) together, we find that the function $g_{\mathfrak{b}} := \mathsf{inj}_{\mathfrak{b}} \circ f_{\mathfrak{b}}$ is a $\mathcal{T}$-coalgebra morphism $g_{\mathfrak{b}} : \mathbb{S}_{\mathfrak{b}} \to \sum_{\mathfrak{a}\in\mathfrak{A}} \mathbb{Z}_{\mathfrak{a}}$ for all $\mathfrak{b} \in \mathfrak{A}$. Defining $g : \mathbb{S} \rightsquigarrow \sum_{\mathfrak{a}\in\mathfrak{A}} Z_{\mathfrak{a}}$ by putting together the slices $g_{\mathfrak{a}}$, we get by definition of **ASet** that $g : \mathbb{S} \to \mathbb{X}$ is a $\mathcal{T}_{\mathfrak{A}}$-coalgebra morphism. Using similar reasoning in which we replace all objects on the $\mathbb{S}$-side by those on the $\mathbb{S}'$-side (e.g. replacing $f_{\mathfrak{a}}$ by $f'_{\mathfrak{a}}$), we obtain a $\mathcal{T}_{\mathfrak{A}}$-coalgebra morphism $g' : \mathbb{S}' \to \mathbb{X}$ as well. Since $g_{\mathfrak{a}} = \mathsf{inj}_{\mathfrak{a}} \circ f_{\mathfrak{a}}$ and $g'_{\mathfrak{a}} = \mathsf{inj}_{\mathfrak{a}} \circ f'_{\mathfrak{a}}$ for all $\mathfrak{a} \in \mathfrak{A}$, it follows from the fact that $f_{\mathfrak{a}}(s) = f'_{\mathfrak{a}}(s')$ that $g_{\mathfrak{a}}(s) = g'_{\mathfrak{a}}(s')$ as well, and thus that $g(s) = g(s')$. So $\mathbb{S}, s \simeq \mathbb{S}', s'$. □

[10] When proving the general statement where $\mathfrak{B} \neq \mathfrak{A}$, one should consider the coproduct $(\sum_{\mathfrak{a}\in\mathfrak{B}} \mathbb{Z}_{\mathfrak{a}}) + (\sum_{\mathfrak{a}\in\mathfrak{A}\setminus\mathfrak{B}} \varnothing)$. The rest of the proof then proceeds as before, but using the initiality of the empty set in **Set** to equip it with $\mathcal{T}$-coalgebra structure.

It is natural to ask whether the notions of bisimilarity and behavioural equivalence on coalgebras in **ASet** coincide. As is the case in **Set**, we have that bisimilar states are behaviourally equivalent, but that the converse does not necessarily hold

Proposition 3.2.25. (*i*) *Given an endofunctor* $\mathscr{F} : \mathbf{ASet} \to \mathbf{ASet}$*, it holds that* $\mathbb{S}, s \leftrightarrows_{\mathfrak{B}} \mathbb{S}', s'$ *implies* $\mathbb{S}, s \simeq_{\mathfrak{B}} \mathbb{S}', s'$ *for all* $\mathscr{F}$*-coalgebras* $\mathbb{S}$ *and* $\mathbb{S}'$*, and* $\mathfrak{B} \in \mathfrak{B}$*, and*

(*ii*) *there are coalgebras* $\mathbb{S}$ *and* $\mathbb{S}'$ *over* **ASet** *such that* $\mathbb{S}, s \simeq_{\mathfrak{B}} \mathbb{S}', s'$ *but* $\mathbb{S}, s \not\leftrightarrows_{\mathfrak{B}} \mathbb{S}', s'$*.*

Proof. To show (i), we use a line of reasoning virtually identical to that that in the proof of the statement that bisimilarity implies behavioural equivalence in **Set**. To show (ii), we extend the example from Aczel and Mendler (1989, after Proposition 6.2), which we said in Section 2.2 shows that behavioural equivalence does not imply bisimilarity in **Set**, to **ASet** in the obvious manner by taking the agentization of the functor, and turning functions in the counterexample to agent-indexed functions with said functions for every slice. □

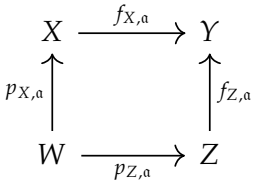
Recall from Proposition 2.1.8 that behavioural equivalence and bisimilarity in **Set** coincide if the coalgebra type functor preserves weak pullbacks. Using Propositions 3.2.22 and 3.2.24, it then follows immediately that if a set functor $\mathscr{T}$ preserves weak pullbacks, then $\mathbb{S}, s \simeq_{\mathfrak{B}} \mathbb{S}', s'$ implies $\mathbb{S}, s \leftrightarrows_{\mathfrak{B}} \mathbb{S}', s'$, for $\mathscr{T}_{\mathfrak{A}}$-coalgebras $\mathbb{S}$ and $\mathbb{S}'$. We can also prove more generally that if an endofunctor $\mathscr{F} : \mathbf{ASet} \to \mathbf{ASet}$ preserves weak pullbacks, then behavioural $\mathfrak{B}$-equivalence implies $\mathfrak{B}$-bisimilarity for $\mathscr{F}$-coalgebras. The proof for this statement is virtually identical to the proof of the statement in **Set**.

It is natural to ask under what circumstances agentizations $\mathscr{T}_{\mathfrak{A}}$ preserve weak pullbacks. This is potentially of interest for future work, as the property of preserving weak pullbacks is important throughout many areas of coalgebraic modal logic. As we will see, agentizations preserve weak pullbacks precisely when the original functor preserves weak pullbacks. To show this, we first require the following lemma.

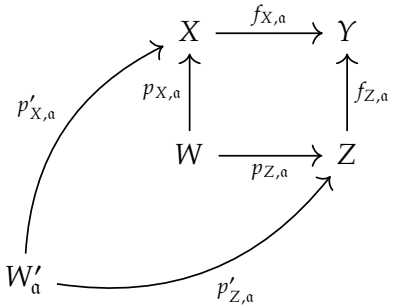
Lemma 3.2.26. *Let* $f_X : X \rightsquigarrow Y$*,* $f_Z : Z \rightsquigarrow Y$*,* $p_X : W \rightsquigarrow X$*, and* $p_Z : W \rightsquigarrow Z$ *be agent-indexed functions. Then the pair* $\langle p_X, p_Z \rangle$ *is a weak pullback of* $\langle f_X, f_Z \rangle$ *in* **ASet** *iff the pair* $\langle p_{X,\mathfrak{a}}, p_{Z,\mathfrak{a}} \rangle$ *is a weak pullback of* $\langle f_{X,\mathfrak{a}}, f_{Z,\mathfrak{a}} \rangle$ *in* **Set** *for all* $\mathfrak{a} \in \mathfrak{A}$*.*

Proof. Throughout this proof, we assume that $|\mathfrak{A}| > 1$. This is without loss of generality, since $\mathbf{ASet} \cong \mathbf{Set}$ otherwise, which would trivially prove the statement.

For the direction from left to right, suppose $\langle p_X, p_Z \rangle$ is a weak pullback of $\langle f_X, f_Z \rangle$ in **ASet**. Then by definition of **ASet**, the **Set**-diagram

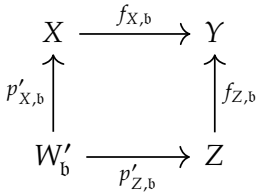


commutes for all $\mathfrak{a} \in \mathfrak{A}$. Fix an agent $\mathfrak{a} \in \mathfrak{A}$. Consider functions $p'_{X,\mathfrak{a}} : W'_{\mathfrak{a}} \to X$ and $p'_{Z,\mathfrak{a}} : W'_{\mathfrak{a}} \to Z$ making the diagram



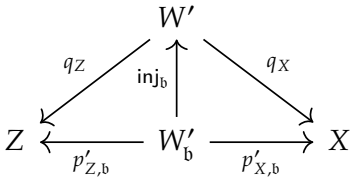
(3.5)

commute. For all $\mathfrak{b} \in \mathfrak{A} \setminus \{\mathfrak{a}\}$, take the pullback $\langle W'_{\mathfrak{b}}, p'_{X,\mathfrak{b}}, p'_{Z,\mathfrak{b}} \rangle$ of the functions $f_{X,\mathfrak{b}}$ and $f_{Z,\mathfrak{b}}$, making the diagram



(3.6)

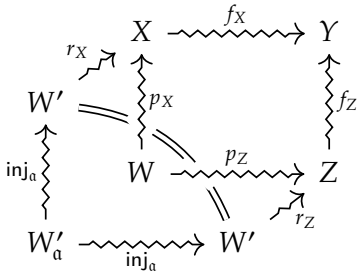
commute. Let W' be the coproduct $W' := \sum_{\mathfrak{b} \in \mathfrak{A}} W'_{\mathfrak{b}}$. By the universal property of the coproduct, there exists unique functions $q_X : W' \to X$ and $q_Z : W' \to Z$ such that the diagram



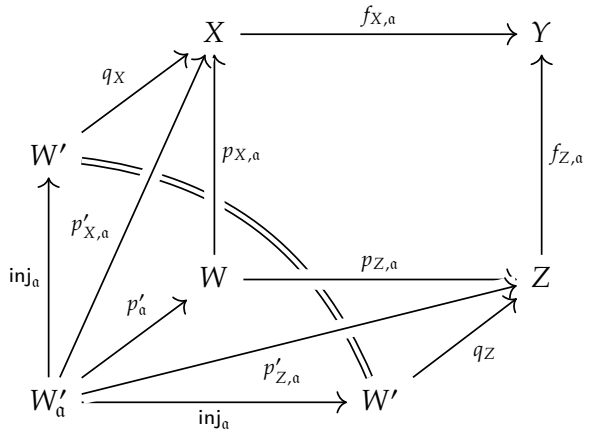
(3.7)

commutes for all $\mathfrak{b} \in \mathfrak{A}$. Take these functions q_X and q_Z, and trivially turn them into agent-indexed functions $r_X : W' \rightsquigarrow X$ and $r_Z : W' \rightsquigarrow Z$ by setting $r_{X,\mathfrak{b}} := q_X$ and $r_{Z,\mathfrak{b}} := q_Z$ for all $\mathfrak{b} \in \mathfrak{A}$. Similarly consider $\mathsf{inj}_{\mathfrak{a}}$ to be

an agent-indexed function by putting it in every slice. By commutativity of Diagrams (3.5) to (3.7), we then deduce that the diagram



commutes in **ASet**. Since we assumed p_X and p_Z form a weak pullback, there must be some agent-indexed function $p' : W'_\mathfrak{a} \rightsquigarrow W$ such that $r_X \circ \mathsf{inj}_\mathfrak{a} = p_X \circ p'$ and $r_Z \circ \mathsf{inj}_\mathfrak{a} = p_Z \circ p'$. Using the $\mathfrak{a}$-slice $p'_\mathfrak{a} : W'_\mathfrak{a} \to W$ of p', it then finally follows that the diagram



commutes, showing that $p_{X,\mathfrak{a}}$ and $p_{Z,\mathfrak{a}}$ form a weak pullback of $\langle f_{X,\mathfrak{a}}, f_{Z,\mathfrak{a}} \rangle$.

The proof of the direction from right to left is much simpler. Suppose $\langle p_{X,\mathfrak{a}}, p_{Z,\mathfrak{a}} \rangle$ is a weak pullback of $\langle f_{X,\mathfrak{a}}, f_{Z,\mathfrak{a}} \rangle$ for all $\mathfrak{a} \in \mathfrak{A}$. Then it follows immediately that $f_X \circ p_X = f_Z \circ p_Z$. For any agent-indexed functions $p'_X : W' \rightsquigarrow X$ and $p'_Z : W' \rightsquigarrow Z$ for which $f_X \circ p'_X = f_Z \circ p'_Z$, we construct the required $p' : W' \rightsquigarrow W$ by taking as its $\mathfrak{a}$-slices the functions $p'_\mathfrak{a} : W' \to W$ we get by applying the fact that $\langle p_{X,\mathfrak{a}}, p_{Z,\mathfrak{a}} \rangle$ is a weak pullback of $\langle f_{X,\mathfrak{a}}, f_{Z,\mathfrak{a}} \rangle$ to the $\mathfrak{a}$-slices $p'_{X,\mathfrak{a}}$ and $p'_{Z,\mathfrak{a}}$. □

Theorem 3.2.27. *For all set functors $\mathcal{T}$, it holds that $\mathcal{T}$ preserves weak pullbacks iff the agentization $\mathcal{T}_\mathfrak{A}$ preserves weak pullbacks.*

Proof. For the direction from left to right, take a weak pullback $\langle p_X, p_Z \rangle$ of agent-indexed functions f_X and f_Z. By Lemma 3.2.26, we get that $\langle p_{X,\mathfrak{a}}, p_{Z,\mathfrak{a}} \rangle$ is a weak pullback of $f_{X,\mathfrak{a}}$ and $f_{Z,\mathfrak{a}}$ for all agents $\mathfrak{a} \in \mathfrak{A}$. Since $\mathcal{T}$ preserves weak pullbacks, it follows that $\langle \mathcal{T}p_{X,\mathfrak{a}}, \mathcal{T}p_{Z,\mathfrak{a}} \rangle$ is a weak pullback

of $\mathcal{T}f_{X,\mathfrak{a}}$ and $\mathcal{T}f_{Z,\mathfrak{a}}$ for all agents $\mathfrak{a} \in \mathfrak{A}$. Since $\mathcal{T}g_{\mathfrak{a}} = (\mathcal{T}_{\mathfrak{A}}g)_{\mathfrak{a}}$ for any agent-indexed function g, we thus have that $\langle (\mathcal{T}_{\mathfrak{A}}p_X)_{\mathfrak{a}}, (\mathcal{T}_{\mathfrak{A}}p_Z)_{\mathfrak{a}} \rangle$ is a weak pullback of $(\mathcal{T}_{\mathfrak{A}}f_X)_{\mathfrak{a}}$ and $(\mathcal{T}_{\mathfrak{A}}f_Z)_{\mathfrak{a}}$ for all agents $\mathfrak{a} \in \mathfrak{A}$. So applying Lemma 3.2.26 again, we get that $\langle \mathcal{T}_{\mathfrak{A}}p_X, \mathcal{T}_{\mathfrak{A}}p_Z \rangle$ is a weak pullback of $\mathcal{T}_{\mathfrak{A}}f_X$ and $\mathcal{T}_{\mathfrak{A}}f_Z$. Thus $\mathcal{T}_{\mathfrak{A}}$ preserves weak pullbacks.

For the direction from right to left, take a weak pullback $\langle q_X, q_Z \rangle$ in **Set** of functions g_X and g_Z. Define the agent-indexed functions p_X and p_Z by taking q_X and q_Z respectively for each slice. Similarly, define agent-indexed functions f_X and f_Z in similar fashion for g_X and g_Z. By Lemma 3.2.26, we then have that p_X and p_Z form a weak pullback of g_X and g_Z. As $\mathcal{T}_{\mathfrak{A}}$ preserves weak pullbacks, it follows that $\mathcal{T}_{\mathfrak{A}}p_X$ and $\mathcal{T}_{\mathfrak{A}}p_Z$ form a weak pullback of $\mathcal{T}_{\mathfrak{A}}f_X$ and $\mathcal{T}_{\mathfrak{A}}f_Z$. Applying Lemma 3.2.26 again, we then finally find that $(\mathcal{T}_{\mathfrak{A}}p_X)_{\mathfrak{a}} = \mathcal{T}q_X$ and $(\mathcal{T}_{\mathfrak{A}}p_Z)_{\mathfrak{a}} = \mathcal{T}q_Z$ form a weak pullback of $(\mathcal{T}_{\mathfrak{A}}f_X)_{\mathfrak{a}} = \mathcal{T}g_X$ and $(\mathcal{T}_{\mathfrak{A}}f_Z)_{\mathfrak{a}} = \mathcal{T}g_Z$ for all $\mathfrak{a} \in \mathfrak{A}$. So $\mathcal{T}$ preserves weak pullbacks. □

3.2.4 Adequacy and Expressivity

We now finally turn our attention towards matters of adequacy and expressivity with respect to Boolean-valued coalgebraic modal logics. We state the results for an arbitrary endofunctor $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$ for greater generality. To do this, we first require a notion of logical equivalence for some agents.

Definition 3.2.28. Given a Boolean-valued coalgebraic modal logic $\mathbb{L}\mathrm{og} = \langle \mathbb{S}\mathrm{im}, \mathbf{Lift} \rangle$ over $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$ and $\mathcal{F}$-models $\mathbb{S}$ and $\mathbb{S}'$, we say states s in $\mathbb{S}$ and s' in $\mathbb{S}'$ are $\mathbb{L}\mathrm{og}$-*equivalent for* $\mathfrak{B} \subseteq \mathfrak{A}$ (and we write $\mathbb{S}, s \equiv^{\mathbb{L}\mathrm{og}}_{\mathfrak{B}} \mathbb{S}', s'$) if

$$\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{L}\mathrm{og}}_{\mathbb{S}}(s) \text{ iff } \mathfrak{a} \in [\![\varphi]\!]^{\mathbb{L}\mathrm{og}}_{\mathbb{S}'}(s')$$

for all formulas $\varphi \in \mathsf{Lang}_{\mathbb{S}\mathrm{im},\mathfrak{A}}$ and agents $\mathfrak{a} \in \mathfrak{A}$. ◁

Although we will always be working with $\mathcal{F}$-*models* throughout the following section, we will often pretend that $\mathcal{F}$ is itself a model functor for simplicity of notation.

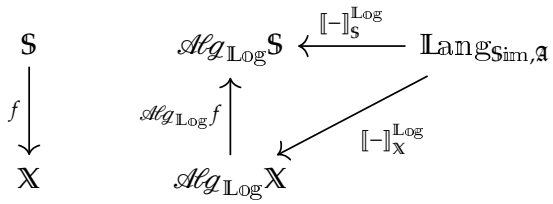
We state the adequacy and expressivity theorems for the single-agent versions of the behavioural and logical equivalence relations — the general versions for multiple agents then follow immediately by their definition. We start with adequacy, showing that behavioural equivalence implies logical equivalence. The proof of this theorem is largely identical to the proof in the two-valued setting, and proceeds through the following proposition showing that morphisms preserve the semantics.

Proposition 3.2.29. *Take some* $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$, *and let* $\mathbb{S}$ *and* $\mathbb{X}$ *be* $\mathcal{F}$-*models with an* $\mathcal{F}_{\mathsf{Prop}}$-*coalgebra morphism* $f : \mathbb{S} \to \mathbb{X}$, *and let* $\mathbb{L}\mathrm{og} = \langle \mathbb{S}\mathrm{im}, \mathbf{Lift} \rangle$ *be a Boolean-valued coalgebraic modal logic over* $\mathcal{F}$. *It holds that*

$$[\![\varphi]\!]^{\mathbb{L}\mathrm{og}}_{\mathbb{S}} = \left(\mathscr{P}\mathit{red}_{\mathfrak{A}} f\right)\left([\![\varphi]\!]^{\mathbb{L}\mathrm{og}}_{\mathbb{X}}\right)$$

for all formulas $\varphi \in \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$.

Proof. This follows immediately from our setup. As $\mathbb{Lang}_{\mathbb{Sim},\mathfrak{A}}$ is the initial $\mathcal{Sim}$-algebra, we have that the triangle in $\mathbf{Alg}_{\mathbf{BA}}(\mathcal{Sim})$ on the right of



commutes. Since $\mathcal{Alg}_{\mathbb{Log}} f = \mathcal{Pred}_{\mathfrak{A}} f$ (considered as a function), we get that it indeed holds that

$$[\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}} = (\mathcal{Pred}_{\mathfrak{A}} f)([\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{X}})$$

for any formula $\varphi \in \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$. □

Theorem 3.2.30. *Let* $\mathbb{S} = \langle S, \sigma\rangle$ *and* $\mathbb{S}' = \langle S', \sigma'\rangle$ *be* $\mathcal{F}$*-models and let* $\mathbb{Log}$ *be a Boolean-valued coalgebraic modal logic, all over some endofunctor* $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$*. For states* $s \in S$ *and* $s' \in S'$*, it holds that*

$$\mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s' \text{ implies } \mathbb{S}, s \equiv^{\mathbb{Log}}_{\mathfrak{a}} \mathbb{S}', s'$$

for all $\mathfrak{a} \in \mathfrak{A}$*.*

Proof. Suppose $\mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s'$ — i.e. that there is some $\mathcal{F}$-model $\mathbb{X}$ with $\mathcal{F}_{\mathsf{Prop}}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$ such that $f_{\mathfrak{a}}(s) = f'_{\mathfrak{a}}(s')$. By Proposition 3.2.29, we know that $[\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}} = (\mathcal{Pred}_{\mathfrak{A}} f)([\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{X}})$ and $[\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}'} = (\mathcal{Pred}_{\mathfrak{A}} f')([\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{X}})$ for any formula φ. Writing these equalities out, we get that $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s)$ iff $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{X}}(f_{\mathfrak{a}}(s))$, and that $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}'}(s')$ iff $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{X}}(f'_{\mathfrak{a}}(s'))$. Since $f_{\mathfrak{a}}(s) = f'_{\mathfrak{a}}(s')$, it thus follows that $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}}(s)$ iff $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}'}(s')$, showing that indeed $\mathbb{S}, s \equiv^{\mathbb{Log}}_{\mathfrak{a}} \mathbb{S}', s'$. □

For expressivity, we also proceed similarly to Section 2.2. We again require the two conditions required in the two-valued setting to also be satisfied: collections of predicate liftings need to be *separating*, and coalgebra type functors need to be *finitary*. But as we will see, our proof of expressivity for a *general* endofunctor on **ASet** requires the set $\mathfrak{A}$ of agents to be *finite* as well. We will come back to this after proving the general statement.

Let us start by defining separation. In the Boolean-valued setting, this concept needs to take the presence of agents into account. We require that different unfoldings can be differentiated by Boolean-valued predicates that differentiate them for *all* agents.

Definition 3.2.31. Given a modal similarity type $\langle \mathsf{Sym}, \mathsf{ar} \rangle$ and Boolean-valued predicate $\mathscr{F}$-liftings $\mathbf{Lift} = \langle \mathbf{lift}^{\heartsuit} \rangle_{\heartsuit \in \mathsf{Sym}}$ of the right arities, we say that **Lift** is *separating for* $\mathscr{F}$ if for all sets X and $U, U' \in \mathscr{F}X$ with $U \neq U'$, there exists some modality symbol $\heartsuit \in \mathsf{Sym}$ with Boolean-valued predicates $v_1, \ldots, v_{\mathsf{ar}\heartsuit} \in \mathscr{Pred}_{\mathfrak{A}} X$ such that for all agents $\mathfrak{a} \in \mathfrak{A}$ it holds that either $\mathfrak{a} \in \mathsf{lift}_X^{\heartsuit}(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U)$ or $\mathfrak{a} \in \mathsf{lift}_X^{\heartsuit}(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U')$, but not both. ◁

This definition truly generalizes that of separation of two-valued predicate liftings (Definition 2.2.8), as stated formally in the following theorem.

Theorem 3.2.32. *Let* $\langle \mathsf{Sym}, \mathsf{ar} \rangle$ *be a modal similarity type with two-valued predicate* $\mathscr{T}$*-liftings* $\mathbf{Lift} = \langle \mathbf{lift}^{\heartsuit} \rangle_{\heartsuit \in \mathsf{Sym}}$ *of the right arities. Then* **Lift** *is separating for* $\mathscr{T}$ *iff* $\mathbf{Lift}_{\mathfrak{A}}$ *(as defined in Definition 3.2.18) is separating for* $\mathscr{T}_{\mathfrak{A}}$*.*

Proof. For the direction from left to right, take some set X and $U, U' \in \mathscr{T}_{\mathfrak{A}} X = \mathscr{T}X$ with $U \neq U'$. Since **Lift** is separating for $\mathscr{T}$, there must be some $\heartsuit \in \mathsf{Sym}$ and $V_1, \ldots, V_{\mathsf{ar}\heartsuit} \in \mathscr{Pred}\, X$ such that either $U \in \mathsf{lift}_X^{\heartsuit}(V_1, \ldots, V_{\mathsf{ar}\heartsuit})$ or $U' \in \mathsf{lift}_X^{\heartsuit}(V_1, \ldots, V_{\mathsf{ar}\heartsuit})$, but not both. We assume without loss of generality that $U \in \mathsf{lift}_X^{\heartsuit}(V_1, \ldots, V_{\mathsf{ar}\heartsuit})$. We construct Boolean-valued predicates $v_1, \ldots, v_{\mathsf{ar}\heartsuit} \in \mathscr{Pred}_{\mathfrak{A}} X$ by putting $v_i(x) := \mathfrak{A}$ if $x \in V_i$, and $v_i(x) := \varnothing$ otherwise. By definition of Booleanized liftings (Definition 3.2.11), we find that

$$\begin{aligned}
(\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_X(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U) &= \left\{ \mathfrak{a} \in \mathfrak{A} \,;\, U \in \mathsf{lift}_X^{\heartsuit}(v_{1,\mathfrak{a}}, \ldots, v_{\mathsf{ar}\heartsuit,\mathfrak{a}}) \right\} \\
&= \left\{ \mathfrak{a} \in \mathfrak{A} \,;\, U \in \mathsf{lift}_X^{\heartsuit}(V_1, \ldots, V_{\mathsf{ar}\heartsuit}) \right\} \\
&= \mathfrak{A} \\
&\neq \varnothing \\
&= \left\{ \mathfrak{a} \in \mathfrak{A} \,;\, U' \in \mathsf{lift}_X^{\heartsuit}(V_1, \ldots, V_{\mathsf{ar}\heartsuit}) \right\} \\
&= (\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_X(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U'),
\end{aligned}$$

showing that indeed $\mathbf{Lift}_{\mathfrak{A}}$ is separating for $\mathscr{T}_{\mathfrak{A}}$.

For the direction from right to left, again take some set X and $U, U' \in \mathscr{T}X$ with $U \neq U'$. Since $\mathbf{Lift}_{\mathfrak{A}}$ is separating for $\mathscr{T}_{\mathfrak{A}}$, there must be some $\heartsuit \in \mathsf{Sym}$ and $v_1, \ldots, v_{\mathsf{ar}\heartsuit} \in \mathscr{Pred}_{\mathfrak{A}} X$ such that for all $\mathfrak{a} \in \mathfrak{A}$ either $\mathfrak{a} \in (\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_X(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U)$ or $\mathfrak{a} \in (\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_X(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U')$, but not both. We fix some agent $\mathfrak{a} \in \mathfrak{A}$, and assume (clearly without loss of generality) that $\mathfrak{a} \in (\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_X(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U)$. By definition of Booleanized liftings, we have that

$$(\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_X(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U) = \left\{ \mathfrak{b} \in \mathfrak{A} \,;\, U \in \mathsf{lift}_X^{\heartsuit}(v_{1,\mathfrak{b}}, \ldots, v_{\mathsf{ar}\heartsuit,\mathfrak{b}}) \right\}, \tag{3.8}$$

with a similar equality when we replace U by U'. So it holds that $U \in \mathsf{lift}_X^{\heartsuit}(v_{1,\mathfrak{a}}, \ldots, v_{\mathsf{ar}\heartsuit,\mathfrak{a}})$, while $U' \notin \mathsf{lift}_X^{\heartsuit}(v_{1,\mathfrak{a}}, \ldots, v_{\mathsf{ar}\heartsuit,\mathfrak{a}})$. Since $v_{i,\mathfrak{a}} \in \mathscr{Pred}\, X$, this already proves that **Lift** is separating for $\mathscr{T}$. □

Finitariness of set functors is usually defined in terms of ω-accessibility (cf. Worrell (2005)), which can be defined as stating that a functor preserves ω-filtered colimits. In our setting, we will work with a concrete and arguably simpler definition, which suffices for the proof of expressivity. First a quick note about notation: for sets A and B with $A \subseteq B$, we denote by $\mathsf{incl}_{A,B} : A \to B$ the obvious inclusion function. These inclusion functions can be treated as agent-indexed functions $\mathsf{incl}_{A,B} : A \rightsquigarrow B$ by placing the original inclusion function in each slice.

Definition 3.2.33. An endofunctor $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$ is *finitary* if for every set X and $U \in \mathcal{F}X$, there exists some finite subset $Y \subseteq X$ such that $U \in (\mathcal{F}\mathsf{incl}_{Y,X})_{\mathfrak{a}}[\mathcal{F}Y]$ for all $\mathfrak{a} \in \mathfrak{A}$. ◁

Remark 3.2.34. Note that we will assume that $\mathcal{F}$ preserves inclusions, in the following concrete sense. We say that $\mathcal{F}$ preserves inclusions if $\mathcal{F}\mathsf{incl}_{A,B} = \mathsf{incl}_{\mathcal{F}A,\mathcal{F}B}$. This also means that $X \subseteq Y$ implies $\mathcal{F}X \subseteq \mathcal{F}Y$. Under this assumption of inclusion preservation, we find that for finitary $\mathcal{F}$ we get that for every X and $U \in \mathcal{F}X$, there exists a finite subset $Y \subseteq X$ such that $U \in \mathcal{F}Y$.

This is quite an innocent assumption. We know (see e.g. Adámek and Trnková (1990)) that any set functor $\mathcal{T}$ is naturally isomorphic to a set functor $\mathcal{T}'$ that preserves inclusions, when restricted to nonempty sets. Given an endofunctor $\mathcal{F} : \mathbf{ASet} \to \mathbf{ASet}$, we can also define a naturally isomorphic (when restricted to nonempty sets) $\mathcal{F}' : \mathbf{ASet} \to \mathbf{ASet}$ that preserves inclusions. Note that for any agent $\mathfrak{a} \in \mathfrak{A}$, we can define a set functor $\mathcal{F}_{\mathfrak{a}}$ defined on sets X as $\mathcal{F}_{\mathfrak{a}}X$, and on functions f as $\mathcal{F}_{\mathfrak{a}}f := (\mathcal{F}f)_{\mathfrak{a}}$ (where we treat f as an agent-indexed function in the obvious way). As stated before, we then get a set functor $\mathcal{F}'_{\mathfrak{a}}$ preserving inclusions. Using these, we define the functor $\mathcal{F}'$. Since the $\mathcal{F}'_{\mathfrak{a}}$ are naturally isomorphic to $\mathcal{F}_{\mathfrak{a}}$ for all $\mathfrak{a} \in \mathfrak{A}$, and $\mathcal{F}_{\mathfrak{a}}X = \mathcal{F}X$, we can define $\mathcal{F}'X := \mathcal{F}X$ without problems. Similarly, for agent-indexed functions f we define $(\mathcal{F}'f)_{\mathfrak{a}} := \mathcal{F}'_{\mathfrak{a}}f_{\mathfrak{a}}$. It is not difficult to verify that $\mathcal{F}'$ must be naturally isomorphic to $\mathcal{F}$ (over nonempty sets), and preserves inclusions. ◁

Finitariness for endofunctors on **ASet** truly generalizes finitariness for set functors (Section 2.2), as stated in the following simple proposition for agentizations.

Proposition 3.2.35. *Let $\mathcal{T}$ be a set functor. Then $\mathcal{T}$ is finitary iff $\mathcal{T}_{\mathfrak{A}}$ is finitary.*

Proof. Assuming $\mathcal{T}$ preserves inclusions, it holds (as shown by Adámek, Milius, et al. (2019)) that $\mathcal{T}$ is finitary (i.e. preserves ω-filtered colimits) iff for every set X and $U \in \mathcal{T}X$, there exists finite $Y \subseteq X$ such that $U \in \mathcal{T}Y$. The statement follows trivially from this equivalence. □

We are now almost equipped to prove the general expressivity result. We need a simple lemma, intuitively stating that for the evaluation of a

Boolean-valued predicate lifting with predicates $v_1, \dots, v_n$ on some $U \in \mathcal{F}X$, one can safely restrict the predicates to the parts of X that are required to 'produce' U.

Lemma 3.2.36. *Let* **lift** *be an n-ary Boolean-valued predicate $\mathcal{F}$-lifting. Then for all sets $X, Y \subseteq X$, $U \in \mathcal{F}Y \subseteq \mathcal{F}X$, and $\langle v_1, \dots, v_n \rangle \in (\mathcal{P}red_{\mathfrak{A}}X)^n$ it holds that*

$$\mathsf{lift}_X(v_1, \dots, v_n)(U) = \mathsf{lift}_Y(v'_1, \dots, v'_n)(U),$$

where v'_i is defined by putting $v'_{i,\mathfrak{a}} := v_{i,\mathfrak{a}} \cap Y$ for all agents $\mathfrak{a} \in \mathfrak{A}$.

Proof. This follows quite simply from the naturality of **lift** and our assumption that $\mathcal{F}$ preserves inclusions. Considering the agent-indexed inclusion $\mathsf{incl}_{Y,X} : Y \rightsquigarrow X$, it follows from naturality of **lift** that the diagram

$$\begin{array}{ccc} (\mathcal{P}red_{\mathfrak{A}}Y)^n & \xrightarrow{\mathsf{lift}_Y} & \mathcal{P}red_{\mathfrak{A}}\mathcal{F}Y \\ {\scriptstyle (\mathcal{P}red_{\mathfrak{A}}\mathsf{incl}_{Y,X})^n}\uparrow & & \uparrow{\scriptstyle \mathcal{P}red_{\mathfrak{A}}\mathcal{F}\mathsf{incl}_{Y,X}} \\ (\mathcal{P}red_{\mathfrak{A}}X)^n & \xrightarrow[\mathsf{lift}_X]{} & \mathcal{P}red_{\mathfrak{A}}\mathcal{F}X \end{array} \tag{3.9}$$

commutes in **Set**, with us leaving out the forgetful functors for brevity of notation. Now for simplicity, we assume $n = 1$. Working out the top-left composition in Diagram (3.9), it follows from the definition of $\mathcal{P}red_{\mathfrak{A}}$ and $\mathsf{incl}_{Y,X}$ that

$$\begin{aligned} \mathsf{lift}_Y((\mathcal{P}red_{\mathfrak{A}}\mathsf{incl}_{Y,X})(v)) &= \mathsf{lift}_Y(y \in Y \mapsto \{\mathfrak{a} \in \mathfrak{A} \,;\, \mathfrak{a} \in v(\mathsf{incl}_{Y,X,\mathfrak{a}}(y))\}) \\ &= \mathsf{lift}_Y(y \in Y \mapsto \{\mathfrak{a} \in \mathfrak{A} \,;\, \mathfrak{a} \in v(y)\}) \\ &= \mathsf{lift}_Y(y \in Y \mapsto \{\mathfrak{a} \in \mathfrak{A} \,;\, y \in v_{\mathfrak{a}}\}) \\ &= \mathsf{lift}_Y(v') \end{aligned} \tag{3.10}$$

for $v \in \mathcal{P}red_{\mathfrak{A}}X$. Similarly working out the bottom-right composition in Diagram (3.9), it follows from $\mathcal{F}$ preserving inclusions and the definitions of $\mathcal{P}red_{\mathfrak{A}}$ and $\mathsf{incl}_{Y,X}$ that

$$\begin{aligned} &(\mathcal{P}red_{\mathfrak{A}}\mathcal{F}\mathsf{incl}_{Y,X})(\mathsf{lift}_X(v)) \\ &= (\mathcal{P}red_{\mathfrak{A}}\mathsf{incl}_{\mathcal{F}Y,\mathcal{F}X})(\mathsf{lift}_X(v)) \\ &= U \in \mathcal{F}Y \mapsto \{\mathfrak{a} \in \mathfrak{A} \,;\, \mathfrak{a} \in \mathsf{lift}_X(v)(\mathsf{incl}_{\mathcal{F}Y,\mathcal{F}X,\mathfrak{a}}(U))\} \\ &= U \in \mathcal{F}Y \mapsto \{\mathfrak{a} \in \mathfrak{A} \,;\, \mathfrak{a} \in \mathsf{lift}_X(v)(U)\} \\ &= U \in \mathcal{F}Y \mapsto \mathsf{lift}_X(v)(U). \end{aligned} \tag{3.11}$$

By commutativity of Diagram (3.9), we get that the expressions in Equations (3.10) and (3.11) are identical, which is precisely what we wished to prove. □

We prove the expressivity result given a finite set of agents, by considering the specific case in which we are considering states in the same model. The general result follows as a simple corollary. Our proof is derived from the proof by Schröder (2008, Theorem 41) of the expressivity of two-valued coalgebraic modal logic. Interestingly, our proof makes use of propositional constants $\ulcorner \mathfrak{B} \urcorner$, which are not present in the two-valued coalgebraic modal logics over which the result of Schröder (2008) is proven.

Theorem 3.2.37. *Let $\mathcal{F}$ be a finitary endofunctor on* **ASet**, *and let* $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ *be a Boolean-valued coalgebraic modal logic over $\mathcal{F}$ such that* **Lift** *is separating for $\mathcal{F}$. If $\mathfrak{A}$ is finite, then for all $\mathcal{F}$-models $\mathbb{S}$, it holds for all states s and t in $\mathbb{S}$ that*

$$\mathbb{S}, s \equiv_{\mathfrak{a}}^{\mathbb{Log}} \mathbb{S}, t \text{ implies } \mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}, t$$

for all $\mathfrak{a} \in \mathfrak{A}$.

Proof. Throughout the proof, we will drop the $\mathbb{Log}$-superscript for brevity of notation.

We will construct an $\mathcal{F}$-coalgebra $\mathbb{E}$ with an $\mathcal{F}$-coalgebra morphism $f : \mathbb{S} \to \mathbb{E}$ such that $f_{\mathfrak{a}}(s) = f_{\mathfrak{a}}(t)$ for all $s, t \in S$ and $\mathfrak{a} \in \mathfrak{A}$ with $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}, t$. This will suffice to prove the theorem.

Consider the set E defined as

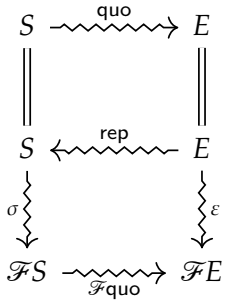
$$E := \sum_{\mathfrak{a} \in \mathfrak{A}} (S/\equiv_{\mathfrak{a}}),$$

where we treat $\equiv_{\mathfrak{a}}$ as a binary relation on S. Note that for each agent $\mathfrak{a} \in \mathfrak{A}$, there is a quotient function $\mathsf{quo}^{(\mathfrak{a})} : S \to S/\equiv_{\mathfrak{a}}$ sending a state s to its equivalence class $[s]_{\equiv_{\mathfrak{a}}}$. We can extend these quotient functions to an agent-indexed function $\mathsf{quo} : S \rightsquigarrow E$ by putting $\mathsf{quo}_{\mathfrak{a}} := \mathsf{inj}_{\mathfrak{a}} \circ \mathsf{quo}^{(\mathfrak{a})}$. We then immediately get that $\mathsf{quo}_{\mathfrak{a}}(s) = \mathsf{quo}_{\mathfrak{a}}(t)$ for all $s, t \in S$ and $\mathfrak{a} \in \mathfrak{A}$ with $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}, t$. So we only need to define an agent-indexed function $\varepsilon : E \rightsquigarrow \mathcal{F}E$ making quo an $\mathcal{F}$-coalgebra morphism from $\mathbb{S}$ to $\mathbb{E} := \langle E, \varepsilon \rangle$.

For each agent $\mathfrak{a} \in \mathfrak{A}$, fix a function $\mathsf{rep}^{(\mathfrak{a})} : S/\equiv_{\mathfrak{a}} \to S$ sending equivalence classes $C \in S/\equiv_{\mathfrak{a}}$ to a representative state $s \in C$.[11] By the universal property of the coproduct, there exists a unique function $g : E \to S$ such that $g \circ \mathsf{inj}_{\mathfrak{a}} = \mathsf{rep}^{(\mathfrak{a})}$ for all $\mathfrak{a} \in \mathfrak{A}$. We turn g into an agent-indexed function $\mathsf{rep} : E \rightsquigarrow S$ by putting $\mathsf{rep}_{\mathfrak{a}} := g$ for each $\mathfrak{a} \in \mathfrak{A}$. We then define $\varepsilon : E \rightsquigarrow \mathcal{F}E$

[11] Note that these functions are choice functions, and can generally only be proven to exist using the Axiom of Choice.

as the composition making the diagram



commute, i.e. $\varepsilon := \mathscr{F}\mathsf{quo} \circ \sigma \circ \mathsf{rep}$. To show that quo is an $\mathscr{F}$-coalgebra morphism $\mathsf{quo} : \mathbb{S} \to \mathbb{E}$, we thus need to show that

$$\mathscr{F}\mathsf{quo} \circ \sigma \circ \mathsf{rep} \circ \mathsf{quo} = \mathscr{F}\mathsf{quo} \circ \sigma. \tag{3.12}$$

Writing out what Equation (3.12) means, it follows from the definition of quo and rep that we need to show that for every agent $\mathfrak{a} \in \mathfrak{A}$ and states $s, t \in S$ such that $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}, t$, it holds that

$$(\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(s)) = (\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(t)). \tag{3.13}$$

For brevity, we write $U_{\mathfrak{a}} = (\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(s))$ and $U'_{\mathfrak{a}} = (\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(t))$.

Fix some agent $\mathfrak{a} \in \mathfrak{A}$, along with states $s, t \in S$ such that $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}, t$. Since **Lift** is separating for $\mathscr{F}$, considering the contrapositive of the statement in Definition 3.2.31, we find that we can show that Equation (3.13) holds by showing that for all modality symbols $\heartsuit \in \mathsf{Sym}$, Boolean-valued predicates $v_1, \ldots, v_{\mathsf{ar}\heartsuit} \in \mathscr{P}\mathit{red}_{\mathfrak{a}} E$, it holds that

$$\mathfrak{a} \in \mathsf{lift}^{\heartsuit}_E(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U_{\mathfrak{a}}) \text{ iff } \mathfrak{a} \in \mathsf{lift}^{\heartsuit}_E(v_1, \ldots, v_{\mathsf{ar}\heartsuit})(U'_{\mathfrak{a}}). \tag{3.14}$$

As $\mathscr{F}$ is finitary, there exist finite subsets $Z, Z' \subseteq S$ such that $\sigma_{\mathfrak{a}}(s) \in \mathscr{F}Z$ and $\sigma_{\mathfrak{a}}(t) \in \mathscr{F}Z'$. Taking the union $Y := Z \cup Z'$, it follows from $\mathscr{F}$ preserving inclusions that $\sigma_{\mathfrak{a}}(s), \sigma_{\mathfrak{a}}(t) \in \mathscr{F}Y \subseteq \mathscr{F}S$. Now take any number i with $1 \leqslant i \leqslant \mathsf{ar}\heartsuit$, and note that v_i is a Boolean-valued predicate $v_i \in \mathscr{P}\mathit{red}_{\mathfrak{A}} E$. Define a Boolean-valued predicate $b_i \in \mathscr{P}\mathit{red}_{\mathfrak{A}} Y$ by putting

$$b_{i,\mathfrak{b}} := Y \cap \mathsf{quo}_{\mathfrak{b}}^{-1}[v_{i,\mathfrak{b}}] \tag{3.15}$$

for all agents $\mathfrak{b} \in \mathfrak{A}$.

By definition of quo, we find that whenever $\mathbb{S}, y \equiv_{\mathfrak{b}} \mathbb{S}, y'$ for $y, y' \in Y$, it holds that $\mathfrak{b} \in b_i(y)$ iff $\mathfrak{b} \in b_i(y')$. So by contraposition we get that for all $y, y' \in Y$ such that $\mathfrak{b} \in b_i(y)$ but $\mathfrak{b} \notin b_i(y')$, it must be the case that $\mathbb{S}, y \not\equiv_{\mathfrak{b}} \mathbb{S}, y'$. By definition of logical $\mathfrak{b}$-equivalence, it then follows that for all $y, y' \in Y$ such that $\mathfrak{b} \in b_i(y)$ but $\mathfrak{b} \notin b_i(y')$, there exists some formula $\varphi_{i,\mathfrak{b},y,y'} \in \mathsf{Lang}_{\mathfrak{Sim},\mathfrak{A}}$ such that

$$\mathfrak{b} \in [\![\varphi_{i,\mathfrak{b},y,y'}]\!]_{\mathbb{S}}(y) \text{ and } \mathfrak{b} \notin [\![\varphi_{i,\mathfrak{b},y,y'}]\!]_{\mathbb{S}}(y').$$

As Y is the union of finite sets Z and Z', it follows that Y is itself finite. So the formula $\varphi_{i,\mathfrak{b}}$ defined as

$$\varphi_{i,\mathfrak{b}} := \bigwedge_{\substack{y' \in Y \\ \mathfrak{b} \notin b_i(y')}} \bigvee_{\substack{y \in Y \\ \mathfrak{b} \in b_i(y)}} \varphi_{i,\mathfrak{b},y,y'}$$

is truly a formula of $\mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$. We have for $z \in Y$ that

$$[\![\varphi_{i,\mathfrak{b}}]\!]_{\mathbb{S}}(z) = \bigcap_{\substack{y' \in Y \\ \mathfrak{b} \notin b_i(y')}} \bigcup_{\substack{y \in Y \\ \mathfrak{b} \in b_i(y)}} [\![\varphi_{i,\mathfrak{b},y,y'}]\!]_{\mathbb{S}}(z) \tag{3.16}$$

by Proposition 3.2.17.

We claim that $\mathfrak{b} \in b_i(z)$ iff $\mathfrak{b} \in [\![\varphi_{i,\mathfrak{b}}]\!]_{\mathbb{S}}(z)$ for all $z \in Y$. For the direction from left to right, note that it must hold by definition of the formulas $\varphi_{i,\mathfrak{b},y,y'}$ that for all $y' \in Y$ with $\mathfrak{b} \notin b_i(y')$ it holds that $\mathfrak{b} \in [\![\varphi_{i,\mathfrak{b},z,y'}]\!]_{\mathbb{S}}(z)$. This corresponds to the semantics of $\varphi_{i,\mathfrak{b}}$ in Equation (3.16), and we thus get that $\mathfrak{b} \in [\![\varphi_{i,\mathfrak{b}}]\!]_{\mathbb{S}}(z)$. For the direction from right to left, suppose for the sake of contradiction that $\mathfrak{b} \notin b_i(z)$, then we would have that for all $y \in Y$ with $\mathfrak{b} \in b_i(y)$ it would hold that $\mathfrak{b} \notin [\![\varphi_{i,\mathfrak{b},y,z}]\!]_{\mathbb{S}}(z)$. So again looking at the semantics of $\varphi_{i,\mathfrak{b}}$ in Equation (3.16), we would get that $\mathfrak{b} \notin [\![\varphi_{i,\mathfrak{b}}]\!]_{\mathbb{S}}(z)$, which is contradictory. Thus $\mathfrak{b} \in b_i(z)$.

Since $\mathfrak{A}$ is finite, the formulas φ_i defined as

$$\varphi_i := \bigwedge_{\mathfrak{b} \in \mathfrak{A}} (\ulcorner \mathfrak{A} \setminus \{\mathfrak{b}\} \urcorner \vee \varphi_{i,\mathfrak{b}})$$

are also well-formed formulas of $\mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$. Working out the semantics of the φ_i-formulas, it follows from the semantics of the $\varphi_{i,\mathfrak{b}}$-formulas that

$$[\![\varphi_i]\!]_{\mathbb{S}}(z) = b_i(z) \tag{3.17}$$

for all $z \in Y$.

We now finally move on to the final part of the proof. Note that by the naturality of Boolean-valued predicate liftings, we have that the square on the right in the diagram

$$\begin{array}{ccc} S & (\mathscr{P}red_{\mathfrak{A}} S)^{\mathsf{ar}^\heartsuit} & \xrightarrow{\mathsf{lift}_S^\heartsuit} \mathscr{P}red_{\mathfrak{A}} \mathcal{F} S \\ \mathsf{quo} \downarrow & \uparrow{\scriptstyle (\mathscr{P}red_{\mathfrak{A}} \mathsf{quo})^{\mathsf{ar}^\heartsuit}} & \uparrow{\scriptstyle \mathscr{P}red_{\mathfrak{A}} \mathcal{F} \mathsf{quo}} \\ E & (\mathscr{P}red_{\mathfrak{A}} E)^{\mathsf{ar}^\heartsuit} & \xrightarrow[\mathsf{lift}_E^\heartsuit]{} \mathscr{P}red_{\mathfrak{A}} \mathcal{F} E \end{array} \tag{3.18}$$

commutes in **Set** — we have left out the forgetful functors for brevity of notation. Working the composition on the bottom-right of Diagram (3.18)

out, we find that

$$(\mathscr{Pred}_{\mathfrak{A}}\mathscr{F}\mathsf{quo})(\mathsf{lift}_E^{\heartsuit}(v)) = W \in \mathscr{F}S \mapsto \left\{\mathfrak{b} \in \mathfrak{A}\,;\, \mathfrak{b} \in \mathsf{lift}_E^{\heartsuit}(v)((\mathscr{F}\mathsf{quo})_{\mathfrak{b}}(W))\right\}, \tag{3.19}$$

where $v = \langle v_1, \ldots, v_{\mathsf{ar}_\heartsuit}\rangle$. And working the composition on the top-left of Diagram (3.18) out, we find that

$$\mathsf{lift}_S^{\heartsuit}((\mathscr{Pred}_{\mathfrak{A}}\mathsf{quo})^{\mathsf{ar}_\heartsuit}(v)) = \mathsf{lift}_S^{\heartsuit}(\ldots, s \in S \mapsto \{\mathfrak{b} \in \mathfrak{A}\,;\, \mathfrak{b} \in v_i(\mathsf{quo}_{\mathfrak{b}}(s))\}, \ldots). \tag{3.20}$$

Importantly, note that the Boolean-valued predicates

$$q_i = s \in S \mapsto \{\mathfrak{b} \in \mathfrak{A}\,;\, \mathfrak{b} \in v_i(\mathsf{quo}_{\mathfrak{b}}(s))\}$$

in Equation (3.20) are equal to

$$q_i = s \in S \mapsto \left\{\mathfrak{b} \in \mathfrak{A}\,;\, s \in \mathsf{quo}_{\mathfrak{b}}^{-1}[v_{i,\mathfrak{b}}]\right\}. \tag{3.21}$$

Given these equations, we find that

$$\begin{aligned}
\mathfrak{a} \in \mathsf{lift}_E^{\heartsuit}(v)((\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(s))) &\text{ iff } \mathfrak{a} \in (\mathscr{Pred}_{\mathfrak{A}}\mathscr{F}\mathsf{quo})(\mathsf{lift}_E^{\heartsuit}(v))(\sigma_{\mathfrak{a}}(s)) \\
&\qquad \text{(Equation (3.19))} \\
&\text{iff } \mathfrak{a} \in \mathsf{lift}_S^{\heartsuit}((\mathscr{Pred}_{\mathfrak{A}}\mathsf{quo})^{\mathsf{ar}_\heartsuit}(v))(\sigma_{\mathfrak{a}}(s)) \\
&\qquad \text{(Diagram (3.18) commutes)} \\
&\text{iff } \mathfrak{a} \in \mathsf{lift}_S^{\heartsuit}(q_1, \ldots, q_{\mathsf{ar}_\heartsuit})(\sigma_{\mathfrak{a}}(s)) \\
&\qquad \text{(Equation (3.20))} \\
&\text{iff } \mathfrak{a} \in \mathsf{lift}_Y^{\heartsuit}(b_1, \ldots, b_{\mathsf{ar}_\heartsuit})(\sigma_{\mathfrak{a}}(s)) \\
&\qquad \text{(Lemma 3.2.36 and Equations (3.15) and (3.21))} \\
&\text{iff } \mathfrak{a} \in \mathsf{lift}_S^{\heartsuit}([\![\varphi_1]\!]_{\mathbb{S}}, \ldots, [\![\varphi_{\mathsf{ar}_\heartsuit}]\!]_{\mathbb{S}})(\sigma_{\mathfrak{a}}(s)) \\
&\qquad \text{(Lemma 3.2.36 and Equation (3.17))} \\
&\text{iff } \mathfrak{a} \in (\mathscr{Pred}_{\mathfrak{A}}\sigma)(\mathsf{lift}_S^{\heartsuit}([\![\varphi_1]\!]_{\mathbb{S}}, \ldots, [\![\varphi_{\mathsf{ar}_\heartsuit}]\!]_{\mathbb{S}}))(s) \\
&\qquad \text{(Proposition 3.2.17)} \\
&\text{iff } \mathfrak{a} \in [\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}_\heartsuit})]\!]_{\mathbb{S}}(s).
\end{aligned}$$

Using identical reasoning, we also find that

$$\mathfrak{a} \in \mathsf{lift}_E^{\heartsuit}(v)((\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(t))) \text{ iff } \mathfrak{a} \in [\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}_\heartsuit})]\!]_{\mathbb{S}}(t).$$

Since $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}, t$, we know that

$$\mathfrak{a} \in [\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}_\heartsuit})]\!]_{\mathbb{S}}(s) \text{ iff } \mathfrak{a} \in [\![\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}_\heartsuit})]\!]_{\mathbb{S}}(t),$$

and thus we get

$$\mathfrak{a} \in \mathsf{lift}_E^{\heartsuit}(v)((\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(s))) \text{ iff } \mathfrak{a} \in \mathsf{lift}_E^{\heartsuit}(v)((\mathscr{F}\mathsf{quo})_{\mathfrak{a}}(\sigma_{\mathfrak{a}}(t)))$$

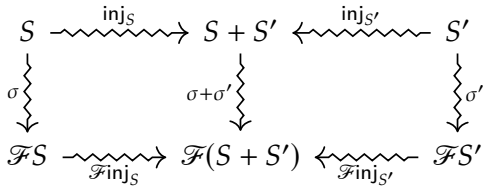
which is precisely what we set out to prove, namely Equation (3.14). Thus Equations (3.12) and (3.13) also hold, showing that $\mathsf{quo} : \mathbb{S} \to \mathbb{E}$ is indeed an $\mathscr{F}$-coalgebra morphism, and so $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}, t$ implies $\mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}, t$ for all $\mathfrak{a} \in \mathfrak{A}$. □

Corollary 3.2.38. *Let $\mathscr{F}$ be a finitary endofunctor on* **ASet**, *and let* $\mathbb{L}\mathrm{og} = \langle \mathbb{S}\mathrm{im}, \mathbf{Lift}\rangle$ *be a Boolean-valued coalgebraic modal logic over $\mathscr{F}$ such that* **Lift** *is separating for $\mathscr{F}$. If $\mathfrak{A}$ is finite, then for all $\mathscr{F}$-models $\mathbb{S}$ and $\mathbb{S}'$, it holds for all states s in $\mathbb{S}$ and s' in $\mathbb{S}'$ that*

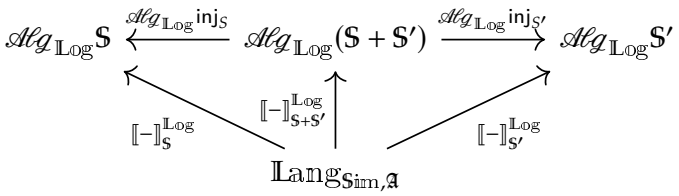
$$\mathbb{S}, s \equiv_{\mathfrak{a}}^{\mathbb{L}\mathrm{og}} \mathbb{S}', s' \text{ implies } \mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s'$$

for all $\mathfrak{a} \in \mathfrak{A}$.

Proof. Consider the coproduct $\mathbb{S} + \mathbb{S}'$ — i.e. the $\mathscr{F}$-model with state space $S + S'$ and coalgebra map $\sigma + \sigma'$ being the unique agent-indexed function making the diagram

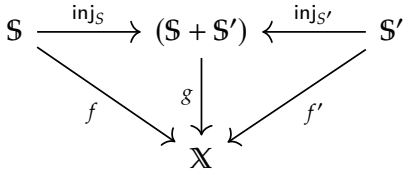


commute. By definition of the semantics of $\mathbb{L}\mathrm{og}$, we have that the diagram



commutes. Working this out, it follows that $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}+\mathbb{S}', \mathsf{inj}_S(s)$ and $\mathbb{S}', s' \equiv_{\mathfrak{a}} \mathbb{S} + \mathbb{S}', \mathsf{inj}_{S'}(s')$ for all $s \in S$ and $s' \in S'$. So $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}', s'$ iff $\mathbb{S} + \mathbb{S}', \mathsf{inj}_S(s) \equiv_{\mathfrak{a}} \mathbb{S} + \mathbb{S}', \mathsf{inj}_{S'}(s')$.

It follows from the universal property of the coproduct that whenever there are $\mathscr{F}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$, that there exists a unique $g : \mathbb{S} + \mathbb{S}' \to \mathbb{X}$ making the diagram



commute. Writing this out, it is easily verifiable that $\mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s'$ iff $\mathbb{S} + \mathbb{S}', \mathsf{inj}_S(s) \simeq_{\mathfrak{a}} \mathbb{S} + \mathbb{S}', \mathsf{inj}_{S'}(s')$.

The corollary now follows immediately through Theorem 3.2.37. □

While we are only capable of proving expressivity for general endofunctors on **ASet** in case there are finitely many agents, we can do away with this

requirement when we are considering Booleanizations of two-valued coalgebraic modal logics. For those, expressivity is an immediate consequence of expressivity in the two-valued setting. To see this, we require one final 'sliced' characterization: that of logical equivalence.

Proposition 3.2.39. *Let $\mathcal{T}$ be a set functor, and let $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ be a two-valued coalgebraic modal logic over $\mathcal{T}$. For all $\mathcal{T}_{\mathfrak{A}}$-models $\mathbb{S}$ and $\mathbb{S}'$, it holds that $\mathbb{S}, s \equiv_{\mathfrak{B}}^{\mathbb{Log}_{\mathfrak{A}}} \mathbb{S}', s'$ iff $\mathbb{S}_{\mathfrak{a}}, s \equiv^{\mathbb{Log}} \mathbb{S}'_{\mathfrak{a}}, s'$ for all $\mathfrak{a} \in \mathfrak{B}$.*

Proof. For the direction from left to right, take any $\varphi \in \mathsf{Lang}_{\mathbb{Sim}}$ (i.e. from the two-valued language) and $\mathfrak{a} \in \mathfrak{B}$. Suppose (clearly without loss of generality) that $s \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}_{\mathfrak{a}}}$. We will show that $s' \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}'_{\mathfrak{a}}}$ as well. Note that the translation $\mathsf{tr}_{\mathfrak{a}}$ used in the Coalgebraic Slicing Theorem (Theorem 3.2.19) is surjective: for any $\varphi \in \mathsf{Lang}_{\mathbb{Sim}}$, there is some $\psi \in \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$ such that $\mathsf{tr}_{\mathfrak{a}}(\psi) = \varphi$. So fix any such ψ for φ. Since $\mathbb{S}, s \equiv_{\mathfrak{B}}^{\mathbb{Log}_{\mathfrak{A}}} \mathbb{S}', s'$, we have that $\mathfrak{a} \in [\![\psi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}}(s)$ iff $\mathfrak{a} \in [\![\psi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}'}(s')$. We know from the Coalgebraic Slicing Theorem that $\mathfrak{a} \in [\![\psi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}}(s)$ iff $s \in [\![\mathsf{tr}_{\mathfrak{a}}(\psi)]\!]^{\mathbb{Log}}_{\mathbb{S}_{\mathfrak{a}}}$. As $\mathsf{tr}_{\mathfrak{a}}(\psi) = \varphi$ and $s \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}_{\mathfrak{a}}}$, it then follows that indeed $\mathfrak{a} \in [\![\psi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}}(s)$. And so we also find that $\mathfrak{a} \in [\![\psi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}'}(s')$. Again applying the Coalgebraic Slicing Theorem, we find that $s' \in [\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}'_{\mathfrak{a}}}$, which is what we needed. So $\mathbb{S}_{\mathfrak{a}}, s \equiv^{\mathbb{Log}} \mathbb{S}'_{\mathfrak{a}}, s'$.

For the direction from right to left, take any $\varphi \in \mathsf{Lang}_{\mathbb{Sim},\mathfrak{A}}$ and $\mathfrak{a} \in \mathfrak{B}$. Suppose (again, without loss of generality) that $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}}(s)$. We will show that $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}'}(s')$. By the Coalgebraic Slicing Theorem, it holds that $s \in [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]^{\mathbb{Log}}_{\mathbb{S}_{\mathfrak{a}}}$. As $\mathbb{S}_{\mathfrak{a}}, s \equiv^{\mathbb{Log}} \mathbb{S}'_{\mathfrak{a}}, s'$ for all $\mathfrak{a} \in \mathfrak{A}$, it then follows that $s' \in [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]^{\mathbb{Log}}_{\mathbb{S}'_{\mathfrak{a}}}$ as well. Again applying the Coalgebraic Slicing Theorem, we then immediately find that $\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{Log}_{\mathfrak{A}}}_{\mathbb{S}'}(s')$, which is what we needed to show. So $\mathbb{S}, s \equiv_{\mathfrak{B}}^{\mathbb{Log}_{\mathfrak{A}}} \mathbb{S}', s'$. □

Theorem 3.2.40. *Let $\mathcal{T}$ be a finitary set functor, and let $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ be a two-valued coalgebraic modal logic over $\mathcal{T}$ such that* **Lift** *is separating for $\mathcal{T}$. For all $\mathcal{T}_{\mathfrak{A}}$-models $\mathbb{S}$ and $\mathbb{S}'$, it holds for all states s in $\mathbb{S}$ and s' in $\mathbb{S}'$ that*

$$\mathbb{S}, s \equiv_{\mathfrak{a}}^{\mathbb{Log}_{\mathfrak{A}}} \mathbb{S}', s' \text{ implies } \mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s'$$

for all $\mathfrak{a} \in \mathfrak{A}$.

Proof. By Proposition 3.2.24, it holds that $\mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s'$ iff $\mathbb{S}_{\mathfrak{a}}, s \simeq \mathbb{S}'_{\mathfrak{a}}, s'$. And by Proposition 3.2.39, it holds that $\mathbb{S}, s \equiv_{\mathfrak{a}}^{\mathbb{Log}_{\mathfrak{A}}} \mathbb{S}', s'$ iff $\mathbb{S}_{\mathfrak{a}}, s \equiv^{\mathbb{Log}} \mathbb{S}'_{\mathfrak{a}}, s'$. As $\mathcal{T}$ is finitary and **Lift** is separating for $\mathcal{T}$, it follows from the expressivity result in the two-valued setting (Proposition 2.2.9) that $\mathbb{S}_{\mathfrak{a}}, s \equiv^{\mathbb{Log}} \mathbb{S}'_{\mathfrak{a}}, s'$ implies that $\mathbb{S}_{\mathfrak{a}}, s \simeq \mathbb{S}'_{\mathfrak{a}}, s'$. And thus we also have that $\mathbb{S}, s \equiv_{\mathfrak{a}}^{\mathbb{Log}_{\mathfrak{A}}} \mathbb{S}', s'$ implies $\mathbb{S}, s \simeq_{\mathfrak{a}} \mathbb{S}', s'$. □

Chapter 4

Multiplayer Game Logic

In this chapter, we will define and generalize multiplayer games and their logic as treated by Olde Loohuis and Venema (2010).[1] In Section 4.1, we give the basic definitions and properties of the game structures and their logic defined in *LAMP*. Afterwards, in Section 4.2 we generalize the aforementioned game structures in a way that is more amenable to coalgebraic generalization. Finally, in Section 4.3 we work out this coalgebraic generalization.

4.1 Deterministic Multiplayer Games

Throughout the chapter, we will be working with sets $\mathfrak{A}$ of *players*. Unlike in Chapter 3 however, we will usually not take $\mathfrak{A}$ to be fixed, and will instead parameterize our definitions by it.

In *LAMP*, Olde Loohuis and Venema propose a logic, based on earlier work by Tulenheimo and Venema (2008), of which the semantics is naturally defined through certain multiplayer games. These multiplayer games are games of perfect information played over directed graphs, with vertices in the graph corresponding to positions in the game, and outgoing edges corresponding to potential moves. The only difference between these games and those considered in the classical two-player evaluation game, is that any player $\mathfrak{a} \in \mathfrak{A}$ could get a turn at a given position, and that multiple players can win at a final position.

The semantics of the logic in *LAMP* is given similarly to the semantics of classical modal logic using two-player evaluation games, based on players having *winning strategies* at positions in games. But again different from the classical two-player setting, it is now possible for arbitrary sets of players to have winning strategies at a given position. So like in Chapter 3, the logic takes truth values from $\mathscr{P}\mathfrak{A}$.

[1] Hereafter, this article is referred to as *LAMP*.

4.1.1 Games, Syntax and Semantics

Let us first consider the basic game structure treated by *LAMP*. We define these in a manner that is more suited to our eventual generalizations, and is equivalent within the scope of both our logic and that of *LAMP*.

Definition 4.1.1. A *deterministic* $\mathfrak{A}$*-game* (or more generally, a *deterministic multiplayer game*) is a quintuple $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{Fin}, \mathsf{turn}, \mathsf{win} \rangle$, where Pos is a set of *positions*, Adm is a binary *admissibility relation* $\mathsf{Adm} \subseteq \mathsf{Pos} \times \mathsf{Pos}$ with no infinite Adm-chains[2], Fin is a set $\mathsf{Fin} \subseteq \mathsf{Pos}$ of *final positions* such that $\mathsf{Adm}[p] = \varnothing$ for all $p \in \mathsf{Fin}$, turn is a *turn function* $\mathsf{turn} : \mathsf{Pos} \setminus \mathsf{Fin} \to \mathfrak{A}$, and win is a *win function* $\mathsf{win} : \mathsf{Fin} \to \mathscr{P}\mathfrak{A}$. A *pointed deterministic* $\mathfrak{A}$*-game* is a pair $\mathbb{G}@p = \langle \mathbb{G}, p \rangle$ consisting of a deterministic $\mathfrak{A}$-game $\mathbb{G}$ and a position p in $\mathbb{G}$. ◁

Note that these games are never referred to as *deterministic* multiplayer games in *LAMP*. Our usage of the term 'deterministic' arises from the fact that the codomain of the turn function is the set $\mathfrak{A}$ of players, and hence, that fixing players' strategies (as we will shortly define) suffices to *determine* how matches proceed. But this only shows the terminology is *correct*. We will not be able to argue in favour of its *relevance* before we get to Section 4.2 and define *nondeterministic* multiplayer games, in which the codomain of the turn function is no longer the set $\mathfrak{A}$ of players, but instead the powerset $\mathscr{P}\mathfrak{A}$.

Let us expound upon the way matches proceed. A deterministic $\mathfrak{A}$-game $\mathbb{G}$ is played over a board represented by the directed graph $\langle \mathsf{Pos}, \mathsf{Adm} \rangle$. At each step of a match, the game will be in precisely one of the positions $p \in \mathsf{Pos}$. The admissibility relation Adm denotes the admissible positions which play can move to from a given position, with $p \ \mathsf{Adm} \ p'$ meaning that p' is admissible from p. If the game is at a nonfinal position $p \notin \mathsf{Fin}$, the player $\mathsf{turn}(p)$ chooses a position p' that is admissible from p, after which the game will be at position p'.

Note that contrary to what one might expect of nonfinal positions, it might be the case that $\mathsf{Adm}[p] = \varnothing$ for $p \notin \mathsf{Fin}$. A match ends if it reaches such positions, since the player $\mathsf{turn}(p)$ will not be able to choose an admissible position. In case a match ends in this manner, the players in $\mathfrak{A} \setminus \{\mathsf{turn}(p)\}$ are declared to be the winners of the match, with the player $\mathsf{turn}(p)$ intuitively losing because he got 'stuck'. Matches otherwise end if they reach a final position $p \in \mathsf{Fin}$, with the players in $\mathsf{win}(p)$ being declared to be the winners of the match. Since we require that there be no infinite Adm-chains, matches are guaranteed to end.

Formally, we define matches and strategies as follows.

[2] That is, we require that there exist no infinite sequences $\langle p_i \rangle_{i<\omega}$ of positions $p_i \in \mathsf{Pos}$ satisfying $p_i \ \mathsf{Adm} \ p_{i+1}$ for all $i < \omega$.

Definition 4.1.2. Fix a pointed deterministic $\mathfrak{A}$-game $\mathbb{G}@p_0$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{Fin}, \mathsf{turn}, \mathsf{win} \rangle$. A *match* (*of* $\mathbb{G}@p$) is a finite sequence $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$ of positions $p_i \in \mathsf{Pos}$ such that $p_i \ \mathsf{Adm} \ p_{i+1}$ for each $i < n$. The *last position of* $\boldsymbol{p}$ is the position $\mathsf{last}(\boldsymbol{p}) = p_n$.

A match $\boldsymbol{p}$ is said to be *complete* if $\mathsf{Adm}[\mathsf{last}(\boldsymbol{p})] = \varnothing$, and *incomplete* otherwise. The set of all complete matches is denoted $\mathsf{Comp}(\mathbb{G}@p)$, and we define a function $\overline{\mathsf{win}} : \mathsf{Comp}(\mathbb{G}@p) \to \mathcal{P}\mathfrak{A}$ by putting $\overline{\mathsf{win}}(\boldsymbol{p}) := \{\mathsf{win}(\mathsf{last}(\boldsymbol{p}))\}$ if $\mathsf{last}(\boldsymbol{p}) \in \mathsf{Fin}$, and $\overline{\mathsf{win}}(\boldsymbol{p}) := \mathfrak{A} \setminus \{\mathsf{turn}(\mathsf{last}(\boldsymbol{p}))\}$ otherwise. For players $\mathfrak{a} \in \mathfrak{A}$, $\mathfrak{a}$ *wins* the (complete) match $\boldsymbol{p}$ if $\mathfrak{a} \in \overline{\mathsf{win}}(\boldsymbol{p})$.

Given a player $\mathfrak{a} \in \mathfrak{A}$, the set of all incomplete matches $\boldsymbol{p}$ such that $\mathsf{turn}(\mathsf{last}(\boldsymbol{p})) = \mathfrak{a}$ is denoted $\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}@p)$. A *strategy* for $\mathfrak{a}$ is a function $\mathsf{strat} : \mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}@p) \to \mathsf{Pos}$ such that $\mathsf{last}(\boldsymbol{p}) \ \mathsf{Adm} \ \mathsf{strat}(\boldsymbol{p})$ for all $\boldsymbol{p} \in \mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}@p)$.[3] A *strategy profile* **strat** is a player-indexed collection $\mathbf{strat} = \langle \mathsf{strat}_{\mathfrak{a}} \rangle_{\mathfrak{a} \in \mathfrak{A}}$ such that $\mathsf{strat}_{\mathfrak{a}}$ is a strategy for $\mathfrak{a}$.

Given a strategy profile **strat**, a match $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$ is said to have been *played conform* **strat** if $p_{i+1} = \mathsf{strat}_{\mathsf{turn}(p_i)}(p_0, \ldots, p_i)$ for all $i < n$. ◁

The formal definition of when a match is played conform a strategy profile confirms what we stated earlier about the deterministic nature of these games. Given a strategy profile and an initial position, the course of the match is fully determined.

The notion of a *winning strategy* is defined in the most straightforward manner: a strategy is winning if it can force a win in any match.

Definition 4.1.3. Given a pointed deterministic $\mathfrak{A}$-game $\mathbb{G}@p$, a strategy strat for player $\mathfrak{a} \in \mathfrak{A}$ is a *winning strategy for* $\mathfrak{a}$ if for all strategy profiles **strat** with $\mathsf{strat}_{\mathfrak{a}} = \mathsf{strat}$ and all complete matches $\boldsymbol{p}$ played conform **strat**, it holds that $\mathfrak{a}$ wins $\boldsymbol{p}$. If there is a winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, we say that $\mathfrak{a}$ *has a winning strategy from* p *in* $\mathbb{G}$, or more concisely that $\mathfrak{a}$ *wins* $\mathbb{G}@p$. ◁

Having defined the game structures, we now move on to the logic. The syntax and semantics of *deterministic multiplayer logic*[4] are defined via a straightforward generalization of two-player evaluation games for basic modal logic to the multiplayer setting. In two-player evaluation games for basic modal logic, disjunctions $\varphi \vee \psi$ correspond to a binary choice of φ and ψ for the Verifier and modal formulas $\Diamond\varphi$ correspond to a choice of a successor state for the Verifier. As we have a set $\mathfrak{A}$ of players, it is natural to consider

[3] We define strategies as functions with sets of incomplete matches as their domain. It should be noted that we could in fact get away with a simpler definition, with us defining strategies for $\mathfrak{a}$ as functions $\mathsf{strat} : \mathsf{turn}^{-1}[\mathfrak{a}] \to \mathsf{Pos}$ (also respecting admissibility), which would intuitively correspond to *memoryless* strategies in which players only make local decisions at each step, with no regards to how they arrived at a certain position. For ease of comparison with the original definitions from *LAMP*, we refrain from pursuing this alternative definition any further.

[4] Note that we are not saying the logic itself is deterministic, but are only making explicit that we are considering multiplayer logic *interpreted* over deterministic multiplayer games.

binary choices and successor state choices for each player. Negation in two-player evaluation games corresponds to the Verifier and Falsifier switching roles. Generalizing this idea, we will allow arbitrary players to switch roles. To achieve this, we define the general notions of *role distributions*.

Definition 4.1.4. An $\mathfrak{A}$-*distribution* (or more generally, a *role distribution*) is a permutation $\mathsf{role} : \mathfrak{A} \to \mathfrak{A}$. The set of $\mathfrak{A}$-distributions is denoted $\mathsf{RD}_{\mathfrak{A}}$. For players $\mathfrak{a}, \mathfrak{b} \in \mathfrak{A}$, the $\mathfrak{A}$-distribution $[\mathfrak{a}, \mathfrak{b}]$ is defined by putting $[\mathfrak{a}, \mathfrak{b}](\mathfrak{a}) := \mathfrak{b}$, $[\mathfrak{a}, \mathfrak{b}](\mathfrak{b}) := \mathfrak{a}$, and $[\mathfrak{a}, \mathfrak{b}](\mathfrak{c}) := \mathfrak{c}$ for $\mathfrak{c} \in \mathfrak{A} \setminus \{\mathfrak{a}, \mathfrak{b}\}$. ◁

Intuitively, a role distribution role assigns to a player $\mathfrak{a}$ the role $\mathfrak{a}$ is taking, i.e. that of player $\mathsf{role}(\mathfrak{a})$.

We can now introduce the syntax of deterministic multiplayer logic.

Definition 4.1.5. The *language of deterministic* $\mathfrak{A}$*-logic* is the set $\mathsf{Lang}_{\mathrm{DET},\mathfrak{A}}$ inductively defined as

$$\mathsf{Lang}_{\mathrm{DET},\mathfrak{A}} \ni \varphi ::= p \mid \bot \mid \top \mid \bot_{\mathfrak{a}} \mid \top_{\mathfrak{a}} \mid (\varphi \vee_{\mathfrak{a}} \varphi) \mid (\Diamond_{\mathfrak{a}}\varphi) \mid (\overline{\mathsf{role}}\varphi),$$

where $p \in \mathsf{Prop}$, $\mathfrak{a} \in \mathfrak{A}$, and $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$. ◁

Intuitively, the atomic formulas $\bot$ and $\top$ represent positions at which nobody or everybody wins, respectively. To be able to express winning and losing for individual players, there are also player-indexed versions $\bot_{\mathfrak{a}}$ and $\top_{\mathfrak{a}}$: these are positions at which only $\mathfrak{a}$ loses or only $\mathfrak{a}$ wins, respectively. Formulas $\varphi \vee_{\mathfrak{a}} \psi$ are positions at which $\mathfrak{a}$ has to choose one of the positions corresponding to φ and ψ, and formulas $\Diamond_{\mathfrak{a}}\varphi$ are positions at which $\mathfrak{a}$ has to choose a position corresponding to φ and a successor state w.r.t. some given Kripke model.

Connectives $\overline{\mathsf{role}}$ are referred to as *role switches*. Some arbitrary fixed player $\mathfrak{x}$ has a turn at a formula $\overline{\mathsf{role}}\varphi$, and there is precisely one admissible position: the formula φ, but with players $\mathfrak{a} \in \mathfrak{A}$ now playing the 'role' of the player $\mathsf{role}(\mathfrak{a})$.[5]

We can now begin defining the semantics of deterministic multiplayer logic. The logic is defined over ordinary Kripke frames of which the colouring is not a function from states to sets of propositional variables, but instead an agent-indexed colouring (cf. Definition 3.1.2) $\mathsf{col} : S \to (\mathscr{P}\mathfrak{A})^{\mathsf{Prop}}$, intuitively assigning to each state s and $p \in \mathsf{Prop}$ precisely those $\mathfrak{a} \in \mathfrak{A}$ that win there. We will refer to these as *player-indexed colourings* in this chapter, and will refer to ordinary Kripke frames $\mathbb{F}$ equipped with player-indexed colourings col as *Kripke multiplayer* (*or* $\mathfrak{A}$-) *models* $\mathbb{M} = \langle \mathbb{F}, \mathsf{col} \rangle$.

Let us start by giving the evaluation games, which are instances of deterministic multiplayer games.

[5] In *LAMP*, role switches are indexed by two players, instead of an arbitrary role distribution. If $\mathfrak{A}$ is finite, these two approaches are equivalent. But when $\mathfrak{A}$ is infinite, indexing by role distributions gives more expressive power, which will be required in e.g. Section 4.2.4.

Table 4.1: Specification of a deterministic $\mathfrak{A}$-logic evaluation game.

Position	Admissible moves	Turn	Winners
$\langle p, s, \mathsf{role}\rangle$	$\varnothing$	-	$\mathsf{role}^{-1}[\mathsf{col}(s)(p)]$
$\langle \bot, s, \mathsf{role}\rangle$	$\varnothing$	-	$\varnothing$
$\langle \top, s, \mathsf{role}\rangle$	$\varnothing$	-	$\mathfrak{A}$
$\langle \bot_{\mathfrak{a}}, s, \mathsf{role}\rangle$	$\varnothing$	-	$\mathsf{role}^{-1}[\mathfrak{A} \smallsetminus \{\mathfrak{a}\}]$
$\langle \top_{\mathfrak{a}}, s, \mathsf{role}\rangle$	$\varnothing$	-	$\{\mathsf{role}^{-1}(\mathfrak{a})\}$
$\langle \varphi \vee_{\mathfrak{a}} \psi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{role}\rangle, \langle \psi, s, \mathsf{role}\rangle\}$	$\mathsf{role}^{-1}(\mathfrak{a})$	-
$\langle \Diamond_{\mathfrak{a}}\varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, t, \mathsf{role}\rangle ; t \in R[s]\}$	$\mathsf{role}^{-1}(\mathfrak{a})$	-
$\langle \overline{\mathsf{rd}}\varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{rd} \circ \mathsf{role}\rangle\}$	$\mathfrak{x}$	-

Definition 4.1.6. Given a Kripke $\mathfrak{A}$-model $\mathbb{M} = \langle\langle S, R\rangle, \mathsf{col}\rangle$, the *deterministic $\mathfrak{A}$-logic evaluation game over* $\mathbb{M}$ is the deterministic $\mathfrak{A}$-game $\mathscr{E}\mathit{val}_{\mathfrak{A}}^{\mathrm{DET}}\mathbb{M}$ with positions $\mathsf{Lang}_{\mathrm{DET},\mathfrak{A}} \times S \times \mathsf{RD}_{\mathfrak{A}}$, and other components specified in Table 4.1. ◁

Note that these evaluation games are always well-defined deterministic $\mathfrak{A}$-games: admissible moves always involve moving from a formula to a subformula, and so there are no infinite Adm-chains.

Having defined evaluation games, we are now able to define the semantics of deterministic $\mathfrak{A}$-logic fully analogously to the way it is defined in two-player evaluation games.

Definition 4.1.7. Given a Kripke $\mathfrak{A}$-model $\mathbb{M} = \langle\langle S, R\rangle, \mathsf{col}\rangle$, the *semantics of deterministic $\mathfrak{A}$-logic* is given by the function $[\![-]\!]_{\mathbb{M}}^{\mathrm{DET},\mathfrak{A}} : \mathsf{Lang}_{\mathrm{DET},\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$ defined by putting

$$[\![\varphi]\!]_{\mathbb{M}}^{\mathrm{DET},\mathfrak{A}}(s) := \{\mathfrak{a} \in \mathfrak{A} ; \mathfrak{a} \text{ wins } (\mathscr{E}\mathit{val}_{\mathfrak{A}}^{\mathrm{DET}}\mathbb{M})@\langle \varphi, s, \mathsf{id}_{\mathfrak{A}}\rangle\}$$

for $\varphi \in \mathsf{Lang}_{\mathrm{DET},\mathfrak{A}}$ and $s \in S$. ◁

We can give a more familiar compositional characterization of the semantics of deterministic $\mathfrak{A}$-logic. To do this, first note the following essential, general and easily proven property of deterministic $\mathfrak{A}$-games.

Proposition 4.1.8. *Let* $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{Fin}, \mathsf{turn}, \mathsf{win}\rangle$ *be a deterministic $\mathfrak{A}$-game, and* $p \in \mathsf{Pos}$ *a position with* $\mathsf{Adm}[p] \neq \varnothing$*. Then*

(*i*) $\mathsf{turn}(p)$ *has a winning strategy from* p *in* $\mathbb{G}$ *iff there is some position* $q \in \mathsf{Adm}[p]$ *such that* $\mathsf{turn}(p)$ *has a winning strategy from* q *in* $\mathbb{G}$*, and*

(*ii*) *any player* $\mathfrak{a} \in \mathfrak{A} \smallsetminus \{\mathsf{turn}(p)\}$ *has a winning strategy from* p *in* $\mathbb{G}$ *iff* $\mathfrak{a}$ *has a winning strategy from* q *in* $\mathbb{G}$ *for all positions* $q \in \mathsf{Adm}[p]$*.*

This is fully analogous to the situation in two-player evaluation games, where Verifier has a winning strategy at a position $\varphi \vee \psi$ iff Verifier either has a winning strategy at φ or at ψ, and Falsifier has a winning strategy at $\varphi \vee \psi$ iff Falsifier has a winning strategy at both φ and ψ.

Using Proposition 4.1.8, we can give a compositional characterization of the semantics of deterministic $\mathfrak{A}$-logic. This compositional characterization is most elegant and easy to interpret if we use the transpose of the semantics. Recall that given a function $f : X \to Y^Z$, we define its *transpose* to be the function $\widehat{f} : Z \to Y^X$ defined by putting $\widehat{f}(z)(x) := f(x)(z)$. Since $\mathcal{P}X \cong 2^X$ for all sets X, we also consider the transpose of a function $f : X \to \mathcal{P}Y$ to be a function $\widehat{f} : Y \to \mathcal{P}X$. Using these transposes, we can consider for any formula φ functions $\widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}} : \mathfrak{A} \to \mathcal{P}S$, with $\widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a})$ being the set of states s such that $\mathfrak{a}$ is in the truth value $[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}(s)$ at s.

Proposition 4.1.9. *Let $\mathbb{M} = \langle\langle S, R\rangle, \mathsf{col}\rangle$ be a Kripke $\mathfrak{A}$-model. Then for any $p \in \mathsf{Prop}$, $\varphi, \psi \in \mathsf{Lang}_{\mathrm{DET},\mathfrak{A}}$, $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$, and $\mathfrak{a}, \mathfrak{b} \in \mathfrak{A}$, it holds that*

$$\widehat{[\![p]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \widehat{\mathsf{col}(p)}(\mathfrak{a}),$$

$$\widehat{[\![\bot]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \varnothing,$$

$$\widehat{[\![\top]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = S,$$

$$\widehat{[\![\bot_{\mathfrak{b}}]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \varnothing & \text{if } \mathfrak{a} = \mathfrak{b}, \\ S & \text{if } \mathfrak{a} \neq \mathfrak{b}, \end{cases}$$

$$\widehat{[\![\top_{\mathfrak{b}}]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} S & \text{if } \mathfrak{a} = \mathfrak{b}, \\ \varnothing & \text{if } \mathfrak{a} \neq \mathfrak{b}, \end{cases}$$

$$\widehat{[\![\varphi \vee_{\mathfrak{b}} \psi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cup \widehat{[\![\psi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} = \mathfrak{b}, \\ \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cap \widehat{[\![\psi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \neq \mathfrak{b}, \end{cases}$$

$$\widehat{[\![\Diamond_{\mathfrak{b}}\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \left\{ s \in S \,;\, R[s] \cap \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \neq \varnothing \right\} & \text{if } \mathfrak{a} = \mathfrak{b}, \\ \left\{ s \in S \,;\, R[s] \subseteq \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \right\} & \text{if } \mathfrak{a} \neq \mathfrak{b}, \text{ and} \end{cases}$$

$$\widehat{[\![\overline{\mathsf{role}}\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}}(\mathsf{role}(\mathfrak{a})).$$

Proof sketch. We can show this by induction on formulas $\varphi \in \mathsf{Lang}_{\mathrm{DET},\mathfrak{A}}$, in which we apply Proposition 4.1.8. The case for role switch formulas $\overline{\mathsf{role}}\varphi$ requires a lemma, stating that for all formulas φ, a player $\mathfrak{a}$ wins the pointed game $(\mathscr{Eval}^{\mathrm{DET}}_{\mathfrak{A}}\mathbb{M})@\langle\varphi, s, \mathsf{role}\rangle$ iff $\mathsf{role}(\mathfrak{a})$ wins $(\mathscr{Eval}^{\mathrm{DET}}_{\mathfrak{A}}\mathbb{M})@\langle\varphi, s, \mathsf{id}_{\mathfrak{A}}\rangle$. We can also show this lemma through induction on formulas. □

So we can consider e.g. formulas $\varphi \vee_{\mathfrak{a}} \psi$ to be 'behaving' like $\varphi \vee \psi$ for player $\mathfrak{a}$, while 'behaving' like $\varphi \wedge \psi$ for others. And similarly, $\Diamond_{\mathfrak{a}}\varphi$ 'behaves' like $\Diamond\varphi$ for $\mathfrak{a}$, while 'behaving' like $\Box\varphi$ for others.

Though the semantics of deterministic $\mathfrak{A}$-logic seems quite unintuitive, they in fact generalize the semantics of two-valued basic modal logic. To see this, note that we can consider all two-valued colourings to be player-indexed colourings with players $\mathfrak{A} = \{\mathfrak{v}, \mathfrak{f}\}$ corresponding to the Verifier and Falsifier respectively, by putting $\mathsf{col}(s)(p) := \{\mathfrak{v}\}$ if the two-valued colouring assigns p to s, and $\mathsf{col}(s)(p) := \{\mathfrak{f}\}$ otherwise. We can then give a translation $\mathsf{tr}_{\mathrm{DET}}$ from formulas of basic modal logic to formulas in $\mathsf{Lang}_{\mathrm{DET},\mathfrak{A}}$, defined inductively as

$$\begin{aligned}
\mathsf{tr}_{\mathrm{DET}}(\top) &:= \top_{\mathfrak{v}}, \\
\mathsf{tr}_{\mathrm{DET}}(\bot) &:= \bot_{\mathfrak{v}}, \\
\mathsf{tr}_{\mathrm{DET}}(p) &:= p, && (p \in \mathsf{Prop}) \\
\mathsf{tr}_{\mathrm{DET}}(\varphi \vee \psi) &:= \mathsf{tr}_{\mathrm{DET}}(\varphi) \vee_{\mathfrak{v}} \mathsf{tr}_{\mathrm{DET}}(\psi), \\
\mathsf{tr}_{\mathrm{DET}}(\neg\varphi) &:= \overline{[\mathfrak{v},\mathfrak{f}]}\mathsf{tr}_{\mathrm{DET}}(\varphi), \text{ and} \\
\mathsf{tr}_{\mathrm{DET}}(\Diamond\varphi) &:= \Diamond_{\mathfrak{v}}\mathsf{tr}_{\mathrm{DET}}(\varphi).
\end{aligned}$$

Then $\mathfrak{v} \in [\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}(s)$ iff φ holds at s in the original two-valued Kripke model, and $\mathfrak{f} \in [\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{M}}(s)$ iff φ does not hold at s in the two-valued Kripke model.

4.1.2 Undefinability of Connectives and Modalities

Even though deterministic $\mathfrak{A}$-logic generalizes two-valued logic in the sense described above, it has glaring limitations from the perspective of our purposes: as we will shortly argue, the framework of deterministic multiplayer games will not suffice for a proper coalgebraic generalization of multiplayer logic, capable of accounting for different kinds of modalities.

But first, we note that even before considering modalities, the framework of deterministic multiplayer games encounters problems when trying to define Boolean connectives like conjunction. Though this need not be a problem from the coalgebraic perspective, as there is no apparent reason *a priori* to require multiplayer logics to contain all of these Boolean connectives, it will in fact allow us to relate the multiplayer logic with the logic from Chapter 3, as we will do in Chapter 5.

For example, it is easily verified that it is not possible to define connectives $\wedge_{\mathfrak{a}}$ that, similarly to $\vee_{\mathfrak{a}}$, behave as conjunction for $\mathfrak{a}$, and as disjunction for others. And perhaps more pressingly, we observe that there is no true negation. While role switches are inspired by the way negation is defined in evaluation games for two-valued logic, they do not suffice on their own to be able to define negation in deterministic $\mathfrak{A}$-logic. Negation in two-valued evaluation games works because precisely one player has a turn or wins at

any given position. But in the setting of deterministic $\mathfrak{A}$-logic, an arbitrary set of players can be winning at a position. We can in fact verify quite easily by Proposition 4.1.9 that the semantics is monotone with respect to colourings, in the sense that $\mathsf{col}(s)(p) \subseteq \mathsf{col}'(s)(p)$ for all states s and $p \in \mathsf{Prop}$ implies that $\widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{F},\mathsf{col}}}(\mathfrak{a}) \subseteq \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{F},\mathsf{col}'}}(\mathfrak{a})$ for all players $\mathfrak{a}$. This monotonicity would obviously not hold if the logic allowed for negation, showing that indeed role switches do not suffice.

But for our purposes, it is most important that deterministic $\mathfrak{A}$-logic admits a natural coalgebraic generalization. When generalizing multiplayer logic to work over coalgebras other than Kripke frames (i.e. $\mathscr{P}$-coalgebras), the main challenge comes in how to approach *modalities*. We need to somehow fit these into the existing framework, with semantics based on the existence of winning strategies. Let us consider an example coalgebra type and modality, so we can see where the pitfalls lie.

Recall the distribution functor $\mathscr{D}\!\mathit{ist} : \mathbf{Set} \to \mathbf{Set}$ defined in Example 2.1.2, sending a set X to the set $\mathscr{D}\!\mathit{ist}\, X \subseteq [0,1]^X$ of discrete probability distributions over X. We have that $\mathscr{D}\!\mathit{ist}$-coalgebras are precisely Markov chains. We can define a unary predicate $\mathscr{D}\!\mathit{ist}$-lifting representing a modality $\Diamond^k$ for some $k \in [0,1]$, of which the semantics in coalgebraic modal logic will be defined for a $\mathscr{D}\!\mathit{ist}$-coalgebra $\mathbb{S} = \langle S, \sigma \rangle$ as

$$[\![\Diamond^k \varphi]\!]_{\mathbb{S}} = \left\{ s \in S \,;\, \sigma(s)([\![\varphi]\!]_{\mathbb{S}}) \geqslant k \right\},$$

where $\sigma(s)(T) := \sum_{t \in T} \sigma(s)(t)$ for $T \subseteq S$.

Suppose we were trying to include these modalities in ordinary multiplayer logic. We would have to construct a position in the resulting evaluation game corresponding to formulas $\Diamond^k \varphi$. In deterministic $\mathfrak{A}$-games, there needs to be exactly one player that gets a turn at this position. So to keep things natural, we create player-indexed versions $\Diamond^k_{\mathfrak{a}}$ of the modalities. If we interpret this indexed version as we did with the indexed $\Diamond$ in deterministic $\mathfrak{A}$-logic for Kripke $\mathfrak{A}$-models, then we want the formula $\Diamond^k_{\mathfrak{a}} \varphi$ to truly 'behave' as $\Diamond^k \varphi$ for $\mathfrak{a}$. In other words, we want our resulting semantics to satisfy

$$\widehat{[\![\Diamond^k_{\mathfrak{a}} \varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) = \left\{ s \in S \,;\, \sigma(s)(\, \widehat{[\![\varphi]\!]^{\mathrm{DET},\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a})) \geqslant k \right\}.$$

Keeping this goal in mind, we now ask ourselves: what actions can player $\mathfrak{a}$ take at the position corresponding to $\Diamond^k_{\mathfrak{a}} \varphi$ and state s? Unlike with the $\Diamond$-modality for Kripke models, the player needs to choose more than just a single state t, since the desired semantics of the modality seems to express something about *multiple* states. So let us have the player select a *set* of states. A natural option is to have the player select sets $T \subseteq S$ such that $\sum_{t \in T} \sigma(s)(t) \geqslant k$. This takes care of the probability-related part of the semantics.

But such sets T do not suffice. We need to somehow enforce that a winning strategy for $\mathfrak{a}$ from the original $\Diamond^k_{\mathfrak{a}}\varphi$ position corresponds to a winning strategy for $\mathfrak{a}$ from all positions corresponding to φ and states $t \in T$. If we can do this, then the resulting semantics for $\mathfrak{a}$ will check out. So after $\mathfrak{a}$ has chosen a set T, we require there to be a choice of state $t \in T$. If we let $\mathfrak{a}$ make this choice again, the resulting semantics will be wrong, since then $\mathfrak{a}$ has a winning strategy from the original position iff there is a set T as described earlier along with a state $t \in T$ such that $\mathfrak{a}$ has a winning strategy from the position corresponding to φ and t. Hence, some other player should get a turn after T is chosen. If we were to just let an arbitrary other player $\mathfrak{b}$ choose the state, then the semantics for $\mathfrak{a}$ will be correct. The semantics for $\mathfrak{b}$ will then behave like the dual $\Box^k$ of the $\Diamond^k$-modality. But for players other than $\mathfrak{a}$ and $\mathfrak{b}$, the semantics will become somewhat nonsensical, corresponding neither to $\Diamond^k$ nor to $\Box^k$. In fact, we can verify that the semantics for other players $\mathfrak{c}$ will correspond to a global, universal modality, with $\mathfrak{c}$ having a winning strategy from a $\Diamond^k_{\mathfrak{a}}\varphi$ position in state s iff $\mathfrak{c}$ has a winning strategy from φ in *all* states $t \in S$.

Thus, we arrive at the essential insight that trying to incorporate other modalities in evaluation games built on top of *deterministic* multiplayer games quickly gives rise to unnatural semantics. This is what motivates our move to nondeterministic multiplayer games in the next section.

4.2 Nondeterministic Multiplayer Games

Having discussed some problems with generalizing deterministic multiplayer logic based on deterministic multiplayer games, we now propose a solution that takes care both of the problems with including other modalities, and of the lack of usual connectives like conjunction and negation. We will reconsider the very base game structure, instead defining *nondeterministic* multiplayer games, in which players' strategies do not fully determine the course of matches.

4.2.1 Games and Matches

The definition of a nondeterministic multiplayer game is obtained by considering structures defined like deterministic multiplayer game, with the additional relaxation that arbitrary sets of players can be selected by the turn function at any position, instead of only a single player. This allows us to shorten the definition drastically, as in the following.

Definition 4.2.1. A *nondeterministic $\mathfrak{A}$-game* (or more generally, a *nondeterministic multiplayer game*) is a triple $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, where Pos is a set of *positions*, Adm is a binary *admissibility relation* $\mathsf{Adm} \subseteq \mathsf{Pos} \times \mathsf{Pos}$ with no infinite Adm-chains, and turn is a *turn function* $\mathsf{turn} : \mathsf{Pos} \to \mathscr{P}\mathfrak{A}$. A *pointed*

nondeterministic $\mathfrak{A}$*-game* is a pair $\mathbb{G}@p = \langle \mathbb{G}, p \rangle$ consisting of a nondeterministic $\mathfrak{A}$-game $\mathbb{G}$ and a position p in $\mathbb{G}$. $\lhd$

We will comment, after describing the way matches proceed, on why this shorter definition with no mention of final positions and win functions truly corresponds to the definition of a deterministic multiplayer game with the aforementioned relaxation of the turn function.

Let us expound upon the way matches proceed with this new definition. A nondeterministic $\mathfrak{A}$-game $\mathbb{G}$ is again played over a board represented by the directed graph $\langle \mathsf{Pos}, \mathsf{Adm} \rangle$, with the admissibility relation Adm denoting the admissible positions which play can move to from a given position. At any position $p \in \mathsf{Pos}$ with $\mathsf{turn}(p) \neq \varnothing$ and $\mathsf{Adm}[p] \neq \varnothing$, a nondeterministic choice of one of the players $\mathfrak{a} \in \mathsf{turn}(p)$ is made. Said player $\mathfrak{a}$ chooses a position p' that is admissible from p, after which the game will be at position p'. If $\mathsf{turn}(p) = \varnothing$, then a nondeterministic choice of an admissible position $p' \in \mathsf{Adm}[p]$ is made instead, with the game again moving to that position.

A match ends if it reaches a position p with no admissible moves (i.e. $\mathsf{Adm}[p] = \varnothing$). After ending, the players in $\mathfrak{A} \setminus \mathsf{turn}(p)$ are declared to be the winners of the match, with the players in $\mathsf{turn}(p)$ intuitively losing because they got stuck. Since we again require that there be no infinite Adm-chains, matches are guaranteed to end.

While deterministic multiplayer games had to make a distinction between final positions (over which the win function is defined) and nonfinal positions with no admissible moves, this distinction is not needed in nondeterministic multiplayer games. To see this, suppose we include in our definition of nondeterministic multiplayer games a subset $\mathsf{Fin} \subseteq \mathsf{Pos}$ of final positions such that $\mathsf{Adm}[p] = \varnothing$ for all $p \in \mathsf{Fin}$, and such that turn is defined only over nonfinal positions. We also include a win function win. We redefine the conditions for winning a match that ends at a position p by declaring the players in $\mathsf{win}(p)$ to be the winners if $p \in \mathsf{Fin}$, and those in $\mathfrak{A} \setminus \mathsf{turn}(p)$ otherwise. Now note that all such games correspond to ones without Fin and win, since if we just put $\mathsf{turn}(p) := \mathfrak{A} \setminus \mathsf{win}(p)$ for $p \in \mathsf{Fin}$, we would get that the same set of players win a match under either definition.

Keeping this in mind, we will often define the set $\mathsf{Fin} \subseteq \mathsf{Pos}$ of final positions in a nondeterministic $\mathfrak{A}$-game as an abbreviation $\mathsf{Fin} := \{p \in \mathsf{Pos}\,; \mathsf{Adm}[p] = \varnothing\}$. We similarly define a win function $\mathsf{win} : \mathsf{Fin} \to \mathscr{P}\mathfrak{A}$ by putting $\mathsf{win}(p) := \mathfrak{A} \setminus \mathsf{turn}(p)$.

Definition 4.2.2. Fix a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p_0$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$. A *match* (*of* $\mathbb{G}@p$) is a finite sequence $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$ of positions $p_i \in \mathsf{Pos}$ such that $p_i \mathsf{Adm} p_{i+1}$ for each $i < n$. The *last position of* $\boldsymbol{p}$ is the position $\mathsf{last}(\boldsymbol{p}) = p_n$.

A match $\boldsymbol{p}$ is said to be *complete* if $\mathsf{Adm}[\mathsf{last}(\boldsymbol{p})] = \varnothing$, and *incomplete* otherwise. The set of all complete matches is denoted $\mathsf{Comp}(\mathbb{G}@p)$, and we define

a function $\overline{\mathsf{win}} : \mathsf{Comp}(\mathbb{G}@p) \to \mathscr{P}\mathfrak{A}$ by putting $\overline{\mathsf{win}}(\boldsymbol{p}) := \mathsf{win}(\mathsf{last}(\boldsymbol{p}))$. For players $\mathfrak{a} \in \mathfrak{A}$, $\mathfrak{a}$ *wins* the (complete) match $\boldsymbol{p}$ if $\mathfrak{a} \in \overline{\mathsf{win}}(\boldsymbol{p})$.

Given a player $\mathfrak{a} \in \mathfrak{A}$, the set of all incomplete matches $\boldsymbol{p}$ such that $\mathfrak{a} \in \mathsf{turn}(\mathsf{last}(\boldsymbol{p}))$ is denoted $\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}@p)$. A *strategy* for $\mathfrak{a}$ is a function $\mathsf{strat} : \mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}@p) \to \mathsf{Pos}$ such that $\mathsf{last}(\boldsymbol{p})\ \mathsf{Adm}\ \mathsf{strat}(\boldsymbol{p})$ for all $\boldsymbol{p} \in \mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}@p)$. A *strategy profile* **strat** is a player-indexed collection $\mathbf{strat} = \langle \mathsf{strat}_{\mathfrak{a}} \rangle_{\mathfrak{a} \in \mathfrak{A}}$ such that $\mathsf{strat}_{\mathfrak{a}}$ is a strategy for $\mathfrak{a}$.

Given a strategy profile **strat**, a match $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$ is said to have been *played conform* **strat** if for all $i < n$ such that $\mathsf{turn}(p_i) \neq \varnothing$, it holds that $p_{i+1} = \mathsf{strat}_{\mathfrak{a}}(p_0, \ldots, p_i)$ for some $\mathfrak{a} \in \mathsf{turn}(p_i)$ ◁

Observe that this definition of matches and strategies matches our informal description of matches, and confirms our statement that the progress of matches of nondeterministic multiplayer games are not fully determined by the players' strategies. Given a strategy profile **strat** and a position p, there are distinct matches $\boldsymbol{p}$ and $\boldsymbol{p}'$ both starting from p and played conform **strat**.

Remark 4.2.3. As an aside before moving on, we wish to note that Başkent (2015) studies evaluation games of which the underlying games are similar in structure to the nondeterministic multiplayer game we are defining. Interestingly, he uses these games to give game-theoretic semantics for paraconsistent logics, of which e.g. Priest's Logic of Paradox (Priest 1979) can also be viewed as a multivalued logic. The games Başkent considers for the Logic of Paradox are also multiplayer games, in order to be able to define the multivalued semantics. ◁

4.2.2 Demonic and Angelic Winning Strategies

We have defined what strategies are in nondeterministic multiplayer games. But how do we define the notion of a winning strategy? As we will see, the most straightforward definition, in which we follow the usual definition of winning strategies on e.g. deterministic multiplayer games, will not be the most fruitful. A strategy for a player $\mathfrak{a} \in \mathfrak{A}$ in a (pointed) deterministic $\mathfrak{A}$-game is winning if no matter what strategies other players use, $\mathfrak{a}$ wins any complete match played conform the players' strategies. This definition makes no mention of the deterministic nature of the game, and so it is straightforward to generalize it to nondeterministic multiplayer games. We will refer to strategies defined thusly as *simple winning strategies*.

Definition 4.2.4. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$, a strategy strat for player $\mathfrak{a} \in \mathfrak{A}$ is a *simple winning strategy* for $\mathfrak{a} \in \mathfrak{A}$ if for all strategy profiles **strat** with $\mathsf{strat}_{\mathfrak{a}} = \mathsf{strat}$ and all complete matches $\boldsymbol{p}$ played conform **strat**, it holds that $\mathfrak{a}$ wins $\boldsymbol{p}$. If there is a simple winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, we say that $\mathfrak{a}$ *has a simple winning strategy from* p *in* $\mathbb{G}$, or more concisely that $\mathfrak{a}$ *simply wins* $\mathbb{G}@p$. ◁

In order to obtain a better understanding of the nature of simple winning strategies for nondeterministic multiplayer games, we first need to obtain a better understanding of the working of matches. Note that while the definition of matches and strategies in Definition 4.2.2 do capture the nondeterminism of the games (as explained), the definition makes no explicit mention of nondeterministic *choices* anywhere, even though these were part of our informal description of the workings of matches. We can make these choices explicit with the following definition.

Definition 4.2.5. Given a nondeterministic $\mathfrak{A}$-game $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, a *nondeterministic choice for* $\mathbb{G}$ (or simply a $\mathbb{G}$-*choice*) is a function[6] $\mathsf{choice} : \mathsf{Pos} \setminus \mathsf{Fin} \longrightarrow \mathfrak{A} \cup \mathsf{Pos}$ such that $\mathsf{choice}(p) \in \mathsf{turn}(p) \subseteq \mathfrak{A}$ for $p \in \mathsf{Pos} \setminus \mathsf{Fin}$ with $\mathsf{turn}(p) \neq \varnothing$, and $\mathsf{choice}(p) \in \mathsf{Adm}[p] \subseteq \mathsf{Pos}$ for $p \in \mathsf{Pos} \setminus \mathsf{Fin}$ with $\mathsf{turn}(p) = \varnothing$.

Given a $\mathbb{G}$-choice choice, the choice-*instantiation of* $\mathbb{G}$ is the deterministic $\mathfrak{A}$-game $\mathbb{G}_{\mathsf{choice}} := \langle \mathsf{Pos}, \mathsf{Adm}_{\mathsf{choice}}, \mathsf{Fin}, \mathsf{turn}_{\mathsf{choice}}, \mathsf{win} \rangle$, where Fin and win are defined precisely as they are for $\mathbb{G}$, the admissibility relation is defined as

$$\begin{aligned} \mathsf{Adm}_{\mathsf{choice}} := &\left\{ \langle p, q \rangle ; p \ \mathsf{Adm} \ q \text{ and } \mathsf{turn}(p) \neq \varnothing \right\} \\ &\cup \left\{ \langle p, \mathsf{choice}(p) \rangle ; p \in \mathsf{Pos} \setminus \mathsf{Fin} \text{ and } \mathsf{turn}(p) = \varnothing \right\}, \end{aligned}$$

and the turn function is defined as

$$\mathsf{turn}_{\mathsf{choice}}(p) := \begin{cases} \mathsf{choice}(p) & \text{if } \mathsf{turn}(p) \neq \varnothing, \text{ and} \\ \mathfrak{x} & \text{if } \mathsf{turn}(p) = \varnothing, \end{cases}$$

where $\mathfrak{x}$ is some arbitrary player.[7] We will generally refer to choice-instantiations of $\mathbb{G}$ as all being *instantiations of* $\mathbb{G}$. ◁

This definition fits the informal description stating that matches require nondeterministic choices of players who will take a turn at positions p such that $\mathsf{turn}(p) \neq \varnothing$, and choices of admissible positions at positions p such that $\mathsf{turn}(p) = \varnothing$. We can also make this relation precise: assuming that $\mathbb{G}@p$ is such that all $q \in \mathsf{Pos}$ are reachable from p through Adm, it is easily verified that given a strategy profile **strat**, the complete matches in $\mathbb{G}@p$ played conform **strat** correspond precisely to $\mathbb{G}$-choices.

These choices can be considered to make the game deterministic, as we capture through instantiations. Importantly, note that strategy profiles in a nondeterministic $\mathfrak{A}$-game $\mathbb{G}$ can be restricted to be strategy profiles in the choice-instantiation $\mathbb{G}_{\mathsf{choice}}$. This will allow us to define a strategy for a player in $\mathbb{G}$ to be winning, based on whether its restriction is winning in the instantiations of $\mathbb{G}$.

[6] Recall that we define $\mathsf{Fin} := \{ p \in \mathsf{Pos} ; \mathsf{Adm}[p] = \varnothing \}$, and $\mathsf{win}(p) := \mathfrak{A} \setminus \mathsf{turn}(p)$.

[7] It is easily verified that the choice of which player is used here is not important, since these positions have precisely one admissible move.

A natural first definition that is constructed in this manner is what we refer to as a *demonic winning strategy*. This is based on the term *demonic nondeterminism* used throughout the theory of programming, stated by Søndergaard and Sestoft (1992) to be due to Tony Hoare. In the theory of programming, demonic nondeterminism is roughly nondeterminism in which all nondeterministic choices are made by a 'demon' that will always choose the worst possible outcome.[8] In the context of the nondeterministic multiplayer games we defined, if the nondeterministic choices are made by a demon, then a player can only truly be guaranteed to win if he wins no matter which choices the demon makes.

Definition 4.2.6. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$, a strategy strat for player $\mathfrak{a} \in \mathfrak{A}$ is a *demonic winning strategy* for $\mathfrak{a} \in \mathfrak{A}$ if the restriction $\mathsf{strat}\restriction_{\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}_{\mathsf{choice}}@p)}$ is a winning strategy for $\mathfrak{a}$ in $\mathbb{G}_{\mathsf{choice}}@p$, for all $\mathbb{G}$-choices choice. If there is a demonic winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, we say that $\mathfrak{a}$ *has a demonic winning strategy from* p *in* $\mathbb{G}$, or more concisely that $\mathfrak{a}$ *demonically wins* $\mathbb{G}@p$. ◁

We consider the restriction $\mathsf{strat}\restriction_{\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}_{\mathsf{choice}}@p)}$, which is the strategy in $\mathbb{G}_{\mathsf{choice}}@p$ that is identical to strat. We require this restriction since there might be incomplete matches in $\mathbb{G}@p$ at which $\mathfrak{a}$ has a turn, but at which $\mathfrak{a}$ would not have a turn if we were considering $\mathbb{G}_{\mathsf{choice}}@p$. Even then, if we want to be fully precise, we can not just consider this restriction, since we defined instantiations of nondeterministic multiplayer games to require some fixed arbitrary player in the definition of $\mathsf{turn}_{\mathsf{choice}}(p)$ for $p \in \mathsf{Pos} \setminus \mathsf{Fin}$ with $\mathsf{turn}(p) = \varnothing$. If $\mathfrak{a}$ is said arbitrary player, the restriction would not truly be a strategy for $\mathfrak{a}$ in $\mathbb{G}_{\mathsf{choice}}@p$. But since $|\mathsf{Adm}_{\mathsf{choice}}[p]| = 1$ for such p, the restriction can be uniquely extended to be a proper strategy for $\mathfrak{a}$ in $\mathbb{G}_{\mathsf{choice}}@p$. So we can always assume without loss of generality that the player for whom we are considering a strategy is not the fixed arbitrary player used in the definition of $\mathsf{turn}_{\mathsf{choice}}$.

Perhaps unsurprisingly, demonic winning strategies correspond precisely to simple winning strategies.

Proposition 4.2.7. *Let* $\mathbb{G}@p$ *be a pointed nondeterministic* $\mathfrak{A}$*-game, and let* strat *be a strategy for a player* $\mathfrak{a} \in \mathfrak{A}$*. Then* strat *is a simple winning strategy iff it is a demonic winning strategy.*

Proof. Throughout the proof, we define the abbreviation

$$\mathsf{strat}^{\textsc{Rest}} := \mathsf{strat}\restriction_{\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}_{\mathsf{choice}}@p)}.$$

For the direction from left to right, take any $\mathbb{G}$-choice choice. We need to show that $\mathsf{strat}^{\textsc{Rest}}$ is a winning strategy for $\mathfrak{a}$ in $\mathbb{G}_{\mathsf{choice}}@p$. So consider

[8] To be precise, demonic nondeterminism in the theory of programming refers to a property of nondeterministic programs in which the possibility of nontermination implies certain nontermination, as stated by Berghammer and Zierer (1986).

a strategy profile **strat** in $\mathbb{G}_{\mathsf{choice}}@p$ with $\mathsf{strat}_{\mathfrak{a}} = \mathsf{strat}^{\text{REST}}$, and a complete match $\boldsymbol{p}$ in $\mathbb{G}_{\mathsf{choice}}@p$ played conform **strat**. Writing $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$ with $p_0 = p$, we then know that $\mathsf{Adm}_{\mathsf{choice}}[p_n] = \emptyset$ and $p_i \ \mathsf{Adm}_{\mathsf{choice}} \ p_{i+1}$ for all $i < n$. Since it is easily verified that $\mathsf{Adm}_{\mathsf{choice}} \subseteq \mathsf{Adm}$, and that $\mathsf{Adm}[q] = \emptyset$ iff $\mathsf{Adm}_{\mathsf{choice}}[q] = \emptyset$, it then follows that $\boldsymbol{p}$ is also a complete match in $\mathbb{G}@p$.

As $\mathsf{turn}_{\mathsf{choice}}(q) \in \mathsf{turn}(q)$ for all $q \in \mathsf{Pos} \setminus \mathsf{Fin}$ with $\mathsf{turn}(q) \neq \emptyset$, we also have that $\mathsf{Incomp}_{\mathfrak{b}}(\mathbb{G}_{\mathsf{choice}}@p) \subseteq \mathsf{Incomp}_{\mathfrak{b}}(\mathbb{G}@p)$. So we can safely fix a strategy profile **strat'** in $\mathbb{G}@p$ with $\mathsf{strat}'_{\mathfrak{a}} = \mathsf{strat}$, and strategies $\mathsf{strat}'_{\mathfrak{b}}$ such that

$$\mathsf{strat}'_{\mathfrak{b}} \restriction_{\mathsf{Incomp}_{\mathfrak{b}}(\mathbb{G}_{\mathsf{choice}}@p)} = \mathsf{strat}_{\mathfrak{b}}.$$

By definition of $\boldsymbol{p}$ being played conform **strat** in $\mathbb{G}_{\mathsf{choice}}@p$, we know that $p_{i+1} = \mathsf{strat}_{\mathsf{turn}_{\mathsf{choice}}(p_i)}(p_0, \ldots, p_i)$ for all $i < n$. Again since $\mathsf{turn}_{\mathsf{choice}}(q) \in \mathsf{turn}(q)$ for all $q \in \mathsf{Pos} \setminus \mathsf{Fin}$ with $\mathsf{turn}(q) \neq \emptyset$, it follows that $\boldsymbol{p}$ is also played conform **strat'** in $\mathbb{G}@p$.

Now since we assume strat is a simple winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, and $\mathsf{strat}'_{\mathfrak{a}} = \mathsf{strat}$, it must be that $\mathfrak{a}$ wins $\boldsymbol{p}$ in $\mathbb{G}@p$. And so by definition of the win function in $\mathbb{G}_{\mathsf{choice}}@p$, we have that $\mathfrak{a}$ also wins $\boldsymbol{p}$ in $\mathbb{G}_{\mathsf{choice}}@p$. Thus, strat is a demonic winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$.

For the direction from right to left, take any strategy profile **strat** with $\mathsf{strat}_{\mathfrak{a}} = \mathsf{strat}$, and any complete match $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$ with $p_0 = p$ in $\mathbb{G}@p$ played conform **strat**. Then for all $i < n$, it must hold that $p_i \ \mathsf{Adm} \ p_{i+1}$, and $p_{i+1} = \mathsf{strat}_{\mathfrak{b}_i}(p_0, \ldots, p_i)$ for some $\mathfrak{b}_i \in \mathsf{turn}(p_i)$ if $\mathsf{turn}(p_i) \neq \emptyset$. Let choice be a $\mathbb{G}$-choice such that for all $i < n$ it holds that $\mathsf{choice}(p_i) = \mathfrak{b}_i$ if $\mathsf{turn}(p_i) \neq \emptyset$, and $\mathsf{choice}(p_i) = p_{i+1}$ if $\mathsf{turn}(p_i) = \emptyset$.

Consider the strategy profile **strat'** in $\mathbb{G}_{\mathsf{choice}}@p$ with

$$\mathsf{strat}'_{\mathfrak{b}} := \mathsf{strat}_{\mathfrak{b}} \restriction_{\mathsf{Incomp}_{\mathfrak{b}}(\mathbb{G}_{\mathsf{choice}}@p)}$$

for all players $\mathfrak{b} \in \mathfrak{A}$. Since strat is a demonic winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, it follows that $\mathsf{strat}'_{\mathfrak{a}}$ is a winning strategy for $\mathfrak{a}$ in $\mathbb{G}_{\mathsf{choice}}@p$. By definition of choice, we have that $\boldsymbol{p}$ is a complete match played conform **strat'** in $\mathbb{G}_{\mathsf{choice}}@p$. So $\mathfrak{a}$ wins $\boldsymbol{p}$ in $\mathbb{G}_{\mathsf{choice}}@p$, and therefore by definition of the win function in $\mathbb{G}_{\mathsf{choice}}$, also wins $\boldsymbol{p}$ in $\mathbb{G}@p$. Therefore strat is a simple winning strategy for $\mathfrak{a} \in \mathfrak{A}$ in $\mathbb{G}@p$. □

Hence, we will be justified in just speaking of demonic winning strategies, instead of simple winning strategies. Whenever it aids understanding to do so, however, we will use the definition of simple winning strategies instead.

As already alluded to just before giving Definition 4.2.4, demonic winning strategies are not the winning strategies that are most fruitful in the study of nondeterministic multiplayer games. We will promptly show that one can essentially do away with most of the structure in nondeterministic multiplayer games differentiating them from deterministic multiplayer

games, when one only considers demonic winning strategies. To see this, we first define *demonizations* of nondeterministic multiplayer games. These modify the structure of a nondeterministic multiplayer game in such a way that we can determine the existence of a demonic winning strategy in the original game by determining their existence in the demonization.

Definition 4.2.8. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, the *demonization of* $\mathbb{G}$ is the pointed nondeterministic $\mathfrak{A}$-game $\mathscr{Dem}(\mathbb{G}@p) := \mathbb{G}_{\textsc{Demon}}@p$, where $\mathbb{G}_{\textsc{Demon}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\textsc{Demon}} \rangle$ is defined by putting

$$\mathsf{turn}_{\textsc{Demon}}(q) := \begin{cases} \mathsf{turn}(q) & \text{if } q \in \mathsf{Fin} \text{ or } |\mathsf{turn}(q)| = 1, \text{ and} \\ \varnothing & \text{otherwise,} \end{cases}$$

for $q \in \mathsf{Pos}$. ◁

In other words, in the demonization of a nondeterministic multiplayer game, players can only get a turn at a nonfinal positions if they are guaranteed to get a turn there. As a consequence, $|\mathsf{turn}(p)| \leqslant 1$ for all positions q, making demonizations more similar to deterministic multiplayer games than to general nondeterministic multiplayer games. Intuitively, it makes sense that the existence of demonic winning strategies is determined over demonizations: if the demon will always choose the worst possible outcome, players can safely assume they will only get a turn if the demon has no other choice but to grant them a turn.

Proposition 4.2.9. *Let $\mathbb{G}@p$ be a pointed nondeterministic $\mathfrak{A}$-game, and $\mathfrak{a} \in \mathfrak{A}$ a player. Then $\mathfrak{a}$ demonically wins $\mathbb{G}@p$ iff $\mathfrak{a}$ demonically wins $\mathscr{Dem}(\mathbb{G}@p)$.*

Proof. For the direction from left to right, let strat be a demonic winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$. Define a strategy $\mathsf{strat}^{\textsc{Demon}} := \mathsf{strat}{\restriction}_{\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}_{\textsc{Demon}}@p)}$ for $\mathfrak{a}$ in $\mathscr{Dem}(\mathbb{G}@p)$. We will show that $\mathsf{strat}^{\textsc{Demon}}$ is a demonic winning strategy in $\mathscr{Dem}(\mathbb{G}@p)$. Consider any strategy profile $\mathbf{strat}$ with $\mathsf{strat}_{\mathfrak{a}} = \mathsf{strat}^{\textsc{Demon}}$, and a complete match $\boldsymbol{p}$ (where $\boldsymbol{p} = \langle p_0, \ldots, p_n \rangle$) in $\mathscr{Dem}(\mathbb{G}@p)$ played conform $\mathbf{strat}$. By definition of demonizations, we have that $\boldsymbol{p}$ is also a complete match in $\mathbb{G}@p$.

Since $\boldsymbol{p}$ is played conform $\mathbf{strat}$, it holds that $p_{i+1} = \mathsf{strat}_{\mathfrak{b}}(p_0, \ldots, p_i)$ for some $\mathfrak{b} \in \mathsf{turn}_{\textsc{Demon}}(p_i)$ if $\mathsf{turn}_{\textsc{Demon}}(p_i) \neq \varnothing$, for all $i < n$. So for any $\mathfrak{b} \in \mathfrak{A} \setminus \{\mathfrak{a}\}$, we can safely consider an arbitrary strategy $\mathsf{strat}'_{\mathfrak{b}}$ in $\mathbb{G}@p$ with restriction $\mathsf{strat}'_{\mathfrak{b}}{\restriction}_{\mathsf{Incomp}_{\mathfrak{b}}(\mathbb{G}_{\textsc{Demon}}@p)} = \mathsf{strat}_{\mathfrak{b}}$ and $\mathsf{strat}'_{\mathfrak{b}}(p_0, \ldots, p_i) := p_{i+1}$ for all $i < n$ with $\mathfrak{b} \in \mathsf{turn}(p_i)$.

We gather these strategies $\mathsf{strat}'_{\mathfrak{b}}$ into a strategy profile $\mathbf{strat'}$ defined as $\mathsf{strat}'_{\mathfrak{a}} := \mathsf{strat}$. It follows from the definition of $\mathbf{strat'}$ and the turn function in the demonization that $\boldsymbol{p}$ is played conform $\mathbf{strat'}$. Since strat is a demonic winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, this means that $\mathfrak{a}$ wins $\boldsymbol{p}$ in $\mathbb{G}@p$, and so $\mathfrak{a}$

also wins p in $\mathscr{Dem}(\mathbb{G}@p)$. Thus, $\mathsf{strat}^{\textsc{Demon}}$ is a demonic winning strategy for $\mathfrak{a}$ in $\mathscr{Dem}(\mathbb{G}@p)$.

The proof of the direction from right to left is similar in spirit, with one extending demonic winning strategies for $\mathfrak{a}$ in $\mathscr{Dem}(\mathbb{G}@p)$ to demonic winning strategies for $\mathfrak{a}$ in $\mathbb{G}@p$ straightforwardly. □

Throughout the remainder of this section, we will usually give proof *sketches* instead of fully detailed proofs. This is because while the proofs are usually conceptually quite straightforward, they are still quite long, and do not offer much understanding beyond a sketch.

Note the following analogue to Proposition 4.1.8 from the deterministic setting, which states that a player $\mathfrak{a}$ has a winning strategy from a (nonfinal) position p if there is an admissible position at which $\mathfrak{a}$ has a winning strategy (in case $\mathsf{turn}(p) = \mathfrak{a}$), or if $\mathfrak{a}$ has a winning strategy at all admissible positions (in case $\mathsf{turn}(p) \neq \mathfrak{a}$). In the nondeterministic setting with demonic winning strategies, a distinction is made instead between the case where $\mathsf{turn}(p) = \{\mathfrak{a}\}$, and the case where $\mathsf{turn}(p) \neq \{\mathfrak{a}\}$.

Proposition 4.2.10. *Let* $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$ *be a nondeterministic* $\mathfrak{A}$*-game,* $p \in \mathsf{Pos} \setminus \mathsf{Fin}$ *a nonfinal position, and* $\mathfrak{a} \in \mathfrak{A}$ *a player. It holds that*

(*i*) *If* $\mathsf{turn}(p) = \{\mathfrak{a}\}$*, then* $\mathfrak{a}$ *demonically wins* $\mathbb{G}@p$ *iff there is some* $q \in \mathsf{Adm}[p]$ *such that* $\mathfrak{a}$ *demonically wins* $\mathbb{G}@q$*, and*

(*ii*) *if* $\mathsf{turn}(p) \neq \{\mathfrak{a}\}$ (*i.e. if* $\mathsf{turn}(p) = \varnothing$ *or* $\mathsf{turn}(p) \setminus \{\mathfrak{a}\} \neq \varnothing$)*, then* $\mathfrak{a}$ *demonically wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *demonically wins* $\mathbb{G}@q$ *for all* $q \in \mathsf{Adm}[p]$.

Proof sketch. This follows from application of Proposition 4.2.9. □

So having seen that demonic winning strategies do away with most of the structure of nondeterministic multiplayer games, we will instead consider another definition of winning strategies. Where demonic winning strategies are strategies that are winning under *all* possible nondeterministic choices, we will now consider strategies that are winning under *some* nondeterministic choices. This definition of winning strategies is in a sense dual to that of the demonic winning strategy.

Again analogously to nondeterminism in the theory of programming, we will refer to these as *angelic* winning strategies. In the theory of programming, angelic nondeterminism is roughly nondeterminism in which all nondeterministic choices are made by an 'angel' that will always choose the best possible outcome. In the context of nondeterministic multiplayer games, this will mean that a player need only win for some choices of the angel, and still be guaranteed to win.

Definition 4.2.11. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$, a strategy strat for player $\mathfrak{a} \in \mathfrak{A}$ is an *angelic winning strategy* for $\mathfrak{a} \in \mathfrak{A}$ if there is

some $\mathbb{G}$-choice choice for which the restriction $\mathsf{strat}{\restriction}_{\mathsf{Incomp}_{\mathfrak{a}}(\mathbb{G}_{\mathsf{choice}}@p)}$ is a winning strategy for $\mathfrak{a}$ in $\mathbb{G}_{\mathsf{choice}}@p$. If there is an angelic winning strategy for $\mathfrak{a}$ in $\mathbb{G}@p$, we say that $\mathfrak{a}$ *has an angelic winning strategy from* p *in* $\mathbb{G}$, or more concisely that $\mathfrak{a}$ *angelically wins* $\mathbb{G}@p$. ◁

While Proposition 4.2.9 intuitively shows that nondeterministic multiplayer games are 'almost' deterministic when one considers demonic winning strategies, considering angelic winning strategies requires one to preserve more structure. This is reflected in the definition of an *angelization*, analogous to demonizations. These angelizations are not the same for all players.

Definition 4.2.12. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}\rangle$ and a player $\mathfrak{a} \in \mathfrak{A}$, the $\mathfrak{a}$-*angelization of* $\mathbb{G}$ is the pointed nondeterministic $\mathfrak{A}$-game $\mathscr{Ang}_{\mathfrak{a}}(\mathbb{G}@p) := \mathbb{G}_{\text{ANGEL},\mathfrak{a}}@p$, where $\mathbb{G}_{\text{ANGEL},\mathfrak{a}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\text{ANGEL},\mathfrak{a}}\rangle$ is defined by putting

$$\mathsf{turn}_{\text{ANGEL},\mathfrak{a}}(q) := \begin{cases} \mathsf{turn}(q) & \text{if } q \in \mathsf{Fin} \text{ or } \mathfrak{a} \notin \mathsf{turn}(q), \text{ and} \\ \{\mathfrak{a}\} & \text{otherwise,} \end{cases}$$

for $q \in \mathsf{Pos}$. ◁

In other words, in the $\mathfrak{a}$-angelization of a nondeterministic multiplayer game, the player $\mathfrak{a}$ is guaranteed to get a turn whenever this is possible. Note that our definition of $\mathfrak{a}$-angelizations matches our intuitions about the angel: if the angel will always choose the best possible outcome for a player, the angel might as well choose to grant that player a turn as often as possible.

The way angelizations are bound to a specific player while demonizations were not also makes intuitive sense. Demons choosing the worst possible outcome for *everyone* can be considered by only granting players turns when there is no other option, while an angel always choosing the best possible outcome for a player will give that player a turn as often as possible, which is not compatible with that angel also always choosing the best possible outcome for another player as well, as there might be positions at which both players could get a turn.

Analogous to demonizations, the existence of angelic winning strategies is determined over angelizations. Noting that angelic winning strategies in angelizations are also demonic winning strategies, we can state this as follows.

Proposition 4.2.13. *Let* $\mathbb{G}@p$ *be a pointed nondeterministic* $\mathfrak{A}$*-game, and* $\mathfrak{a} \in \mathfrak{A}$ *a player. Then* $\mathfrak{a}$ *angelically wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *angelically* (*or demonically*) *wins* $\mathscr{Ang}_{\mathfrak{a}}(\mathbb{G}@p)$.

Proof sketch. If a strategy strat for $\mathfrak{a}$ is an angelic winning strategy in $\mathbb{G}@p$, then there is a $\mathbb{G}$-choice choice such that $\mathfrak{a}$ wins the choice-instantiation of $\mathbb{G}$

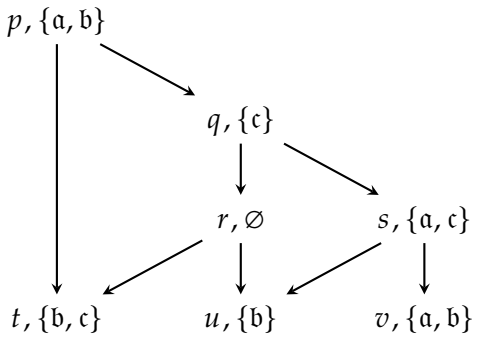


Figure 4.2.1: Nondeterministic multiplayer game considered in Example 4.2.15.

at p following strat. Note that if we modify choice to always select $\mathfrak{a}$ when possible, this will still hold. The strategy strat is also a strategy for $\mathfrak{a}$ in $\mathscr{Ang}_{\mathfrak{a}}(\mathbb{G}@p)$. Any complete match in $\mathscr{Ang}_{\mathfrak{a}}(\mathbb{G}@p)$ played conform a profile with $\mathfrak{a}$ following strat will then also be a complete match in the choice-instantiation of $\mathbb{G}$ at p, played conform a corresponding profile with $\mathfrak{a}$ following strat. And so $\mathfrak{a}$ wins the match.

If a strategy strat for $\mathfrak{a}$ is an angelic winning strategy in $\mathscr{Ang}_{\mathfrak{a}}(\mathbb{G}@p)$, then there is a $\mathbb{G}_{\text{ANGEL},\mathfrak{a}}$-choice choice such that $\mathfrak{a}$ wins the choice-instantiation of $\mathscr{Ang}_{\mathfrak{a}}(\mathbb{G}@p)$ following strat. We can easily extend choice to a $\mathbb{G}$-choice choice′ by always selecting $\mathfrak{a}$ whenever possible. Noting that strat is also a strategy in $\mathbb{G}@p$, it then follows quite simply that strat is a winning strategy in the choice′-instantiation of $\mathbb{G}@p$. □

We again have an analogue to Proposition 4.1.8 from the deterministic setting. A distinction is now made between the case where $\mathfrak{a} \in \mathsf{turn}(p)$ or not. As we will see in Section 4.2.3, it is this difference between angelic and demonic winning strategies that makes angelic winning strategies a more natural candidate for the semantics of our multiplayer logic.

Proposition 4.2.14. *Let* $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$ *be a nondeterministic* $\mathfrak{A}$*-game,* $p \in \mathsf{Pos} \setminus \mathsf{Fin}$ *a nonfinal position, and* $\mathfrak{a} \in \mathfrak{A}$ *a player. It holds that*

(*i*) *if* $\mathfrak{a} \in \mathsf{turn}(p)$*, then* $\mathfrak{a}$ *angelically wins* $\mathbb{G}@p$ *iff there is some* $q \in \mathsf{Adm}[p]$ *such that* $\mathfrak{a}$ *angelically wins* $\mathbb{G}@q$*, and it holds that*

(*ii*) *if* $\mathfrak{a} \notin \mathsf{turn}(p)$*, then* $\mathfrak{a}$ *angelically wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *angelically wins* $\mathbb{G}@q$ *for all* $q \in \mathsf{Adm}[p]$.

Proof sketch. This follows from application of Proposition 4.2.13. □

Example 4.2.15. Consider the nondeterministic $\{\mathfrak{a}, \mathfrak{b}, \mathfrak{c}\}$-game represented visually in Figure 4.2.1. The vertices represent positions, the edges represent

the admissibility relation, and the annotations at each vertex represent the turn function. This game shows that angelic winning strategies are truly more general than demonic winning strategies. In the bottom row (i.e. the final positions), the sets of winners are $\{\mathfrak{a}\}$ for t, $\{\mathfrak{a}, \mathfrak{c}\}$ for u, and $\{\mathfrak{c}\}$ for v.

Considering demonic winning strategies, it can be verified using Proposition 4.2.10 that $\mathfrak{a}$ has a demonic winning strategy at r but not at s, $\mathfrak{b}$ has a demonic winning strategy at neither r nor s, and that $\mathfrak{c}$ has a demonic winning strategy at s but not at r. And so, neither $\mathfrak{a}$ nor $\mathfrak{b}$ has a demonic winning strategy at q, while $\mathfrak{c}$ does (since $\mathsf{turn}(q) = \{c\}$). Finally, none of the players has a demonic winning strategy at p.

Considering angelic winning strategies, it can be verified using Proposition 4.2.14 that $\mathfrak{a}$ has an angelic winning strategy at both r and s, $\mathfrak{b}$ has an angelic winning strategy at neither r nor s, and that $\mathfrak{c}$ has an angelic winning strategy at s but not at r. And so, both $\mathfrak{a}$ and $\mathfrak{c}$ have an angelic winning strategy at q, while $\mathfrak{b}$ does not. Finally, only $\mathfrak{a}$ has an angelic winning strategy at p. ◁

Before moving on to defining multiplayer logic over nondeterministic multiplayer games, we wish to give some final comments on the relation between deterministic and nondeterministic multiplayer games. It should be clear that deterministic multiplayer games can be embedded within nondeterministic multiplayer games, in such a way that winning strategies correspond to demonic winning strategies, which end up being precisely angelic winning strategies in this setting.

Proposition 4.2.16. *Let* $\mathbb{G}@p$ *be a pointed deterministic* $\mathfrak{A}$*-game with* $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{Fin}, \mathsf{turn}, \mathsf{win} \rangle$*. Define the pointed nondeterministic* $\mathfrak{A}$*-game* $\mathscr{N}on(\mathbb{G}@p) := \langle \mathbb{G}_{\mathrm{NON}}, p \rangle$ *with* $\mathbb{G}_{\mathrm{NON}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\mathrm{NON}} \rangle$ *where* $\mathsf{turn}_{\mathrm{NON}}$ *is defined by putting*

$$\mathsf{turn}_{\mathrm{NON}}(q) := \begin{cases} \{\mathsf{turn}(q)\} & \text{if } q \in \mathsf{Pos} \setminus \mathsf{Fin}, \text{ and} \\ \mathfrak{A} \setminus \mathsf{win}(q) & \text{if } q \in \mathsf{Fin}, \end{cases}$$

for $q \in \mathsf{Pos}$*.*

For all $\mathfrak{a} \in \mathfrak{A}$*, it holds that* $\mathfrak{a}$ *wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *demonically wins* $\mathscr{N}on(\mathbb{G}@p)$*, or equivalently, iff* $\mathfrak{a}$ *angelically wins* $\mathscr{N}on(\mathbb{G}@p)$*.*

Given that deterministic multiplayer games always embed into nondeterministic multiplayer games in a way that preserves winning strategies, it is natural to ask whether converse embeddings preserving demonic or angelic winning strategies generally exist. For pointed nondeterministic multiplayer games $\mathbb{G}@p$ in which $|\mathsf{turn}(q)| = 1$ for all nonfinal positions q, there is a straightforward embedding. We will denote these deterministic games as $\mathscr{N}on^{-1}(\mathbb{G}@p)$, though $\mathscr{N}on^{-1}$ should not be strictly taken to denote the inverse of the $\mathscr{N}on$-operation. This $\mathscr{N}on^{-1}$-operation then preserves both demonic and angelic winning strategies.

But as we will start proving now, embeddings also exist in general, albeit being slightly more complicated. Essential for these embeddings are three ideas. First, when determining whether there exist demonic or angelic winning strategies for a player $\mathfrak{a}$, we can 'abstract away' from all other players, treating them as if they are part of the nondeterministic environment (i.e. the demon or angel). Second, we can make the role of the nondeterministic environment explicit by including it as a player $\mathfrak{n}$ in the game. Third, the existence of both demonic and angelic winning strategies is 'monotone' in a certain sense: if we increase the amount of nonfinal positions at which a player $\mathfrak{a}$ wins, the existence of demonic and angelic winning strategies is preserved.

We can capture the first two ideas in the following definition.

Definition 4.2.17. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, and a player $\mathfrak{a}$, the $\mathfrak{a}$-*abstraction of* $\mathbb{G}@p$ is the pointed nondeterministic $(\mathfrak{A} \cup \{\mathfrak{n}\})$-game $\mathscr{Abst}_{\mathfrak{a}}(\mathbb{G}@p) := \left\langle \mathbb{G}_{\text{Abst},\mathfrak{a}}, p \right\rangle$ with $\mathbb{G}_{\text{Abst},\mathfrak{a}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\text{Abst},\mathfrak{a}} \rangle$, where $\mathsf{turn}_{\text{Abst},\mathfrak{a}}$ is defined as

$$\mathsf{turn}_{\text{Abst},\mathfrak{a}}(q) := \begin{cases} \mathsf{turn}(q) & \text{if } q \in \mathsf{Fin}, \\ (\mathsf{turn}(q) \cap \{\mathfrak{a}\}) \cup \{\mathfrak{n}\} & \text{if } q \notin \mathsf{Fin} \text{ and } \mathsf{turn}(q) \neq \{\mathfrak{a}\}, \text{ and} \\ \{\mathfrak{a}\} & \text{otherwise,} \end{cases}$$

for $q \in \mathsf{Pos}$.

The *environment-only abstraction of* $\mathbb{G}@p$ is the pointed nondeterministic $(\mathfrak{A} \cup \{\mathfrak{n}\})$-game $\mathscr{Abst}_{\text{Env}}(\mathbb{G}@p) := \left\langle \mathbb{G}_{\text{Env}}, p \right\rangle$ with $\mathbb{G}_{\text{Env}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\text{Env}} \rangle$, where $\mathsf{turn}_{\text{Env}}$ is defined as

$$\mathsf{turn}_{\text{Env}}(q) := \begin{cases} \mathsf{turn}(q) & \text{if } q \in \mathsf{Fin} \text{ or } \mathsf{turn}(q) \neq \emptyset, \text{ and} \\ \{\mathfrak{n}\} & \text{otherwise,} \end{cases}$$

for $q \in \mathsf{Pos}$. ◁

Let us break down what this definition states. In the $\mathfrak{a}$-abstraction, we only allow $\mathfrak{a}$ and $\mathfrak{n}$ to actually make choices by replacing all other players by $\mathfrak{n}$ (though we do not modify turns/winners at nonfinal positions). In the environment-only abstraction, we give $\mathfrak{n}$ a turn at (nonfinal) positions where no other player can get a turn. This makes sense, since these are positions at which the next position is chosen nondeterministically.

The following is easily verified.

Proposition 4.2.18. *Let $\mathbb{G}@p$ be a pointed nondeterministic $\mathfrak{A}$-game, and $\mathfrak{a} \in \mathfrak{A}$ a player. Then*

(*i*) *$\mathfrak{a}$ demonically (resp. angelically) wins $\mathbb{G}@p$ iff $\mathfrak{a}$ demonically (resp. angelically) wins $\mathscr{Abst}_{\mathfrak{a}}(\mathbb{G}@p)$, and*

(*ii*) $\mathfrak{a}$ *demonically (resp. angelically) wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *demonically (resp. angelically) wins* $\mathscr{Abst}_{\textsc{Env}}(\mathbb{G}@p)$.

The third idea we mentioned will be used in the following form.

Definition 4.2.19. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, and a set $\mathfrak{B} \subseteq \mathfrak{A}$ of players, the $\mathfrak{B}$-*trivialization of* $\mathbb{G}@p$ is the pointed nondeterministic $\mathfrak{A}$-game $\mathscr{Triv}_{\mathfrak{B}}(\mathbb{G}@p) := \langle \mathbb{G}_{\textsc{Triv},\mathfrak{B}}, p \rangle$ with $\mathbb{G}_{\textsc{Triv},\mathfrak{B}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\textsc{Triv},\mathfrak{B}} \rangle$, where

$$\mathsf{turn}_{\textsc{Triv},\mathfrak{B}}(q) := \begin{cases} \mathsf{turn}(q) & \text{if } q \notin \mathsf{Fin}, \text{ and} \\ \mathsf{turn}(q) \setminus \mathfrak{B} & \text{if } q \in \mathsf{Fin}, \end{cases}$$

for $q \in \mathsf{Pos}$. ◁

Proposition 4.2.20. *Let* $\mathbb{G}@p$ *be a pointed nondeterministic* $\mathfrak{A}$*-game, and* $\mathfrak{B} \subseteq \mathfrak{A}$ *a set of players. Then for all players* $\mathfrak{a} \in \mathfrak{B}$*, it holds that* $\mathfrak{a}$ *both demonically and angelically wins* $\mathscr{Triv}_{\mathfrak{B}}(\mathbb{G}@p)$.

We are now equipped to show the embeddings. First, we give a simple embedding preserving demonic winning strategies. This embedding is quite simple, and relies on our intuition that demonizations as defined in Definition 4.2.8 are already 'almost' deterministic.

Definition 4.2.21. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, the *demonic determinization of* $\mathbb{G}@p$ is the pointed deterministic $(\mathfrak{A} \cup \{\mathfrak{n}\})$-game

$$\mathscr{Det}_{\textsc{Demon}}(\mathbb{G}@p) := \mathscr{Non}^{-1}\mathscr{Abst}_{\textsc{Env}}\mathscr{Dem}(\mathbb{G}@p). \qquad \triangleleft$$

Theorem 4.2.22. *Let* $\mathbb{G}@p$ *be a pointed nondeterministic* $\mathfrak{A}$*-game. For all players* $\mathfrak{a} \in \mathfrak{A}$*, it holds that* $\mathfrak{a}$ *demonically wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *wins* $\mathscr{Det}_{\textsc{Demon}}(\mathbb{G}@p)$.

Proof. This follows immediately from Propositions 4.2.9 and 4.2.18. □

Example 4.2.23. Denote the nondeterministic multiplayer game used in Example 4.2.15 and visualized in Figure 4.2.1 by $\mathbb{G}$. The demonic determinization of $\mathbb{G}@p$ is visualized in Figure 4.2.2. All vertices with no outgoing edges are final positions, and their annotations represent the win function. Note that indeed, none of the players win $\mathscr{Det}_{\textsc{Demon}}(\mathbb{G}@p)$. ◁

The embedding preserving angelic winning strategies will require us to consider angelizations for all players. To be able to gather these in a single game, we will use *coproducts of (not pointed) nondeterministic multiplayer games*. These are defined in the obvious manner, which is informally described by placing games alongside each other. We will also require a way to link each of these angelizations, for which we will use the following construction. Note that we slightly abuse notation, and will write e.g. $\mathscr{Ang}_{\mathfrak{a}}\mathbb{G}$ instead of $\mathbb{G}_{\textsc{Angel},\mathfrak{a}}$.

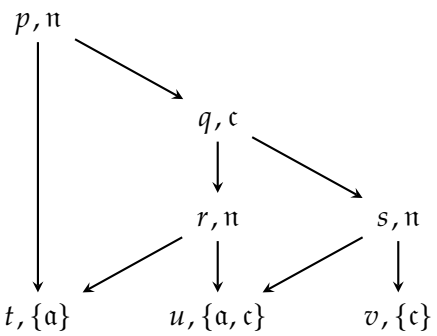


Figure 4.2.2: Demonic determinization considered in Example 4.2.23.

Definition 4.2.24. Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$, define

$$\mathbb{G}' := \sum_{\mathfrak{a}\in\mathfrak{A}} \mathscr{Abst}_{\mathfrak{a}} \mathscr{Triv}_{\mathfrak{A}\setminus\{\mathfrak{a}\}} \mathscr{Ang}_{\mathfrak{a}} \mathbb{G}.$$

Writing $\mathbb{G}' = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}\rangle$, we define the nondeterministic $(\mathfrak{A} \cup \{\mathfrak{n}\})$-game $\mathbb{G}'' := \left\langle \mathsf{Pos} \cup \left\{p_{\textsc{Det}}\right\}, \mathsf{Adm}_{\textsc{Det}}, \mathsf{turn}_{\textsc{Det}}\right\rangle$ by putting

$$\mathsf{Adm}_{\textsc{Det}} := \mathsf{Adm} \cup \left\{\left\langle p_{\textsc{Det}}, \mathsf{inj}_{\mathfrak{a}}(p)\right\rangle ; \mathfrak{a} \in \mathfrak{A}\right\}$$

and $\mathsf{turn}_{\textsc{Det}}(q) := \mathsf{turn}(q)$ for $q \in \mathsf{Pos}$, while $\mathsf{turn}_{\textsc{Det}}(p_{\textsc{Det}}) := \{\mathfrak{n}\}$.

The *angelic determinization of* $\mathbb{G}@p$ is the pointed deterministic $(\mathfrak{A} \cup \{\mathfrak{n}\})$-game

$$\mathscr{Det}_{\textsc{Angel}}(\mathbb{G}@p) := \mathscr{Non}^{-1}(\mathbb{G}''@p_{\textsc{Det}}). \qquad \triangleleft$$

Taken step by step, we construct the angelic determinization by taking the angelizations for all players. We make sure that in the $\mathfrak{a}$-angelization, all other players $\mathfrak{b} \in \mathfrak{A}\setminus\{\mathfrak{a}\}$ will trivially have angelic winning strategies. Then, we abstract away from these other players. Finally, we gather together the resulting nondeterministic $(\mathfrak{A}\cup\{\mathfrak{n}\})$-games for all players, and create a new position at which the nondeterministic environment (i.e. the angel) has to choose one of the component games.

Before stating the theorem that this embedding preserves angelic winning strategies, let us consider an example.

Example 4.2.25. As in Example 4.2.23, we denote the nondeterministic multiplayer game used in Example 4.2.15 and visualized in Figure 4.2.1 by $\mathbb{G}$. The angelic determinization of $\mathbb{G}@p$ is visualized in Figure 4.2.3. We remove all position names except those of the positions p in each angelization, which we subscript by the player to make clear in which angelization they lie, and the new position $p_{\textsc{Det}}$ — it should then be clear for other positions to which position in the original game they correspond. We can verify that indeed only $\mathfrak{a}$ wins $\mathscr{Det}_{\textsc{Angel}}(\mathbb{G}@p)$. $\triangleleft$

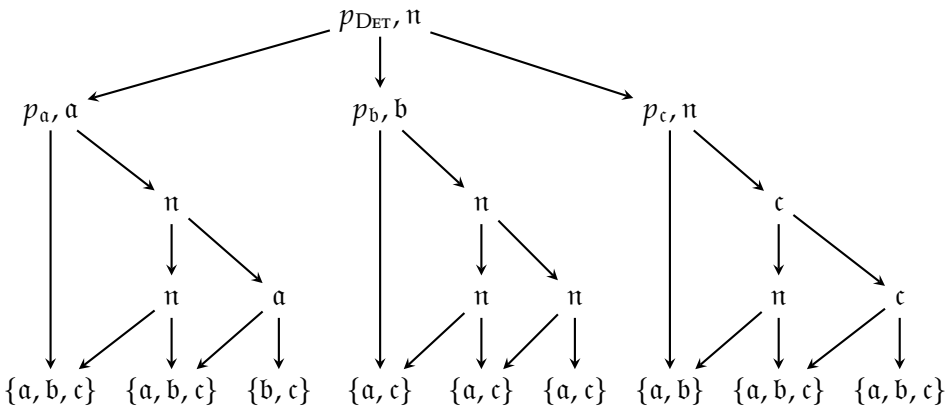


Figure 4.2.3: Angelic determinization considered in Example 4.2.25.

Theorem 4.2.26. *Let $\mathbb{G}@p$ be a pointed nondeterministic $\mathfrak{A}$-game. For all players $\mathfrak{a} \in \mathfrak{A}$, it holds that $\mathfrak{a}$ angelically wins $\mathbb{G}@p$ iff $\mathfrak{a}$ wins $\mathscr{Det}_{\text{ANGEL}}(\mathbb{G}@p)$.*

Proof. This follows from Propositions 4.1.8, 4.2.13, 4.2.18 and 4.2.20. □

Remark 4.2.27. We wish to note that introducing the nondeterministic environment as a player like we have done with abstractions (Definition 4.2.17) has interesting conceptual consequences. We could consider the nondeterministic environment to be a 'moderator' or an 'umpire' of sorts, tasked with choosing which player gets a turn, and with moving the game along whenever the players cannot (i.e. at positions p with $\mathsf{turn}(p) = \varnothing$).

Note that this is quite similar to the way nondeterminism is often treated in the theory of programming. As discussed by Harmer (1999), who considers game-theoretical semantics for nondeterministic computation, nondeterminism is often used in the theory of programming as an abstraction of certain aspects of a computational process, such as a machine's scheduler that is responsible for choosing which process to execute at what time. The role of the moderator in our setting is largely analogous to that of such a scheduler.

Considering moderators also offers a conceptual argument for the usage of angelic over demonic winning strategies. Consider debates (or if one wishes to stay within the realm of logic, medieval *obligationes*) as multiplayer games with a moderator. Positions correspond to states of the debate, while players' admissible moves correspond to statements they can utter. By Proposition 4.2.9, a demonic winning strategy for a player in this context would be a collection of arguments such that no matter whether the moderator actually gives the player a turn or not, the player is guaranteed to win the debate. This is quite a strict requirement, that would in this example mean that players have to collude with the moderator outside of the debate.

By Proposition 4.2.13 however, an angelic winning strategy for a player would be a collection of arguments that could win the debate if the moderator does not prevent the player from uttering it. One could then argue that angelic winning strategies better reflect the 'quality' of players' arguments, since they require no action outside of the debate.

Treating the nondeterministic environment as a moderator suggests avenues for further work, in which one could study games where the moderator has explicit goals of his own. It would be of interest to see what implications this could have for the logic we will be introducing in Section 4.2.3. ◁

4.2.3 Logic and Evaluation Games

We can now move on to define a more general version of the deterministic multiplayer logic of Section 4.1: *nondeterministic basic multiplayer logic* defined over nondeterministic multiplayer games and multiplayer Kripke models. Where deterministic multiplayer logic consists of instances of (a subset of) the connectives of two-valued basic modal logic for each individual player, the framework of nondeterministic multiplayer games naturally allows us to consider connectives for sets of players.

Definition 4.2.28. The *language of nondeterministic basic $\mathfrak{A}$-logic* is the inductively defined set

$$\mathsf{Lang}_{\mathrm{BASIC},\mathfrak{A}} \ni \varphi ::= p \mid \bot_{\mathfrak{B}} \mid (\varphi \vee_{\mathfrak{B}} \varphi) \mid (\Diamond_{\mathfrak{B}}\varphi) \mid (\overline{\mathsf{role}}\varphi),$$

where $p \in \mathsf{Prop}$, $\mathfrak{B} \subseteq \mathfrak{A}$, and $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$. If $\mathfrak{B} = \{\mathfrak{a}\}$, we will usually omit the brackets. ◁

In the way we will define the evaluation games, the atomic formulas $\bot_{\mathfrak{B}}$ will represent positions at which precisely those players in $\mathfrak{B}$ lose. Formulas $\varphi \vee_{\mathfrak{B}} \psi$ are positions at which some player in $\mathfrak{B}$ has to choose one of the positions corresponding to φ and ψ, and formulas $\Diamond_{\mathfrak{B}}\varphi$ are positions at which some player in $\mathfrak{B}$ has to choose a position corresponding to φ and a successor state in a given multiplayer Kripke model.

Definition 4.2.29. Given a Kripke $\mathfrak{A}$-model $\mathbb{M} = \langle \langle S, R \rangle, \mathsf{col} \rangle$, the *nondeterministic basic $\mathfrak{A}$-logic evaluation game over* $\mathbb{M}$ is the nondeterministic $\mathfrak{A}$-game $\mathscr{E}\mathit{val}_{\mathfrak{A}}^{\mathrm{BASIC}}\mathbb{M}$ with positions $\mathsf{Lang}_{\mathrm{BASIC},\mathfrak{A}} \times S \times \mathsf{RD}_{\mathfrak{A}}$, and other components specified in Table 4.2. ◁

Note that unlike in the evaluation games in the deterministic setting (Table 4.1), there is no need any more to fix an arbitrary player x who takes a turn at positions corresponding to role switches.

As should be evident from our discussion in the previous section, the semantics will be given in terms of angelic instead of demonic winning strategies, since demonic winning strategies do away with most nondeterministic

Table 4.2: Specification of a nondeterministic basic $\mathfrak{A}$-logic evaluation game.

Position	Admissible moves	Turn
$\langle p, s, \mathsf{role}\rangle$	$\varnothing$	$\mathfrak{A} \setminus \mathsf{role}^{-1}[\mathsf{col}(s)(p)]$
$\langle \bot_{\mathfrak{B}}, s, \mathsf{role}\rangle$	$\varnothing$	$\mathsf{role}^{-1}[\mathfrak{B}]$
$\langle \varphi \vee_{\mathfrak{B}} \psi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{role}\rangle, \langle \psi, s, \mathsf{role}\rangle\}$	$\mathsf{role}^{-1}[\mathfrak{B}]$
$\langle \Diamond_{\mathfrak{B}} \varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, t, \mathsf{role}\rangle ; t \in R[s]\}$	$\mathsf{role}^{-1}[\mathfrak{B}]$
$\langle \overline{\mathsf{rd}}\varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{rd} \circ \mathsf{role}\rangle\}$	$\varnothing$

structure. In particular, demonic winning strategies only consider games with at most one player that can get a turn at any position. This would not make sense for our logic, in which we have connectives for arbitrary sets of players.

Definition 4.2.30. Given a Kripke $\mathfrak{A}$-model $\mathbb{M} = \langle \langle S, R\rangle, \mathsf{col}\rangle$, the *semantics of nondeterministic basic $\mathfrak{A}$-logic* is given by the function

$$[\![-]\!]^{\textsc{Basic},\mathfrak{A}}_{\mathbb{M}} : \mathsf{Lang}_{\textsc{Basic},\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$$

defined by putting

$$[\![\varphi]\!]^{\textsc{Basic},\mathfrak{A}}_{\mathbb{M}}(s) := \{\mathfrak{a} \in \mathfrak{A} ; \mathfrak{a} \text{ angelically wins } (\mathscr{E}\mathit{val}^{\textsc{Basic}}_{\mathfrak{A}}\mathbb{M})@\langle \varphi, s, \mathsf{id}_{\mathfrak{A}}\rangle\}$$

for $\varphi \in \mathsf{Lang}_{\textsc{Basic},\mathfrak{A}}$ and $s \in S$. ◁

Example 4.2.31. For $\mathfrak{A} = \{\mathfrak{a}, \mathfrak{b}, \mathfrak{c}\}$, consider a Kripke $\mathfrak{A}$-model $\mathbb{M} = \langle \langle S, R\rangle, \mathsf{col}\rangle$ with $S = \{s, t, u\}$, $R = \{\langle s, t\rangle, \langle s, u\rangle\}$, $\mathsf{col}(s)(p) = \varnothing$, $\mathsf{col}(t)(p) = \{\mathfrak{a}\}$, and $\mathsf{col}(u)(p) = \{\mathfrak{c}\}$. Take the nondeterministic basic $\mathfrak{A}$-logic formula

$$\varphi = \bot_{\mathfrak{c}} \vee_{\{\mathfrak{a},\mathfrak{c}\}} \overline{[\mathfrak{a}, \mathfrak{b}]}\Diamond_{\{\mathfrak{b},\mathfrak{c}\}} p.$$

The nondeterministic basic $\mathfrak{A}$-logic evaluation game over $\mathbb{M}$ pointed at the position $\langle \varphi, s, \mathsf{id}_{\mathfrak{A}}\rangle$ is visualized in Figure 4.2.4. As in other examples, the annotations at each position represent the turn function. We can verify (using e.g. Proposition 4.2.14) that $[\![\varphi]\!]^{\textsc{Basic},\mathfrak{A}}_{\mathbb{M}}(s) = \{\mathfrak{c}\}$. ◁

Applying Proposition 4.2.14, we can give a compositional characterization of the semantics of nondeterministic basic multiplayer logic, similar to that of deterministic multiplayer logic (Proposition 4.1.9).

Proposition 4.2.32. *Let $\mathbb{M} = \langle \langle S, R\rangle, \mathsf{col}\rangle$ be a Kripke $\mathfrak{A}$-model. Then for any*

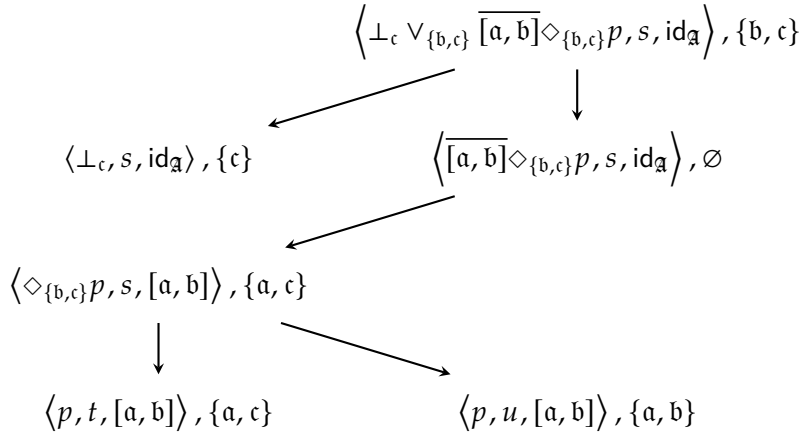


Figure 4.2.4: The nondeterministic basic multiplayer logic evaluation game considered in Example 4.2.31.

$p \in \mathsf{Prop}$, $\varphi, \psi \in \mathsf{Lang}_{\mathrm{BASIC},\mathfrak{A}}$, $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$, $\mathfrak{B} \subseteq \mathfrak{A}$, *and* $\mathfrak{a} \in \mathfrak{A}$, *it holds that*

$$\widehat{[\![p]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \widehat{\mathsf{col}}(p)(\mathfrak{a}),$$

$$\widehat{[\![\bot_{\mathfrak{B}}]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \varnothing & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ S & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\varphi \vee_{\mathfrak{B}} \psi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \widehat{[\![\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cup \widehat{[\![\psi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \widehat{[\![\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cap \widehat{[\![\psi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\Diamond_{\mathfrak{B}}\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \left\{ s \in S \,;\, R[s] \cap \widehat{[\![\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \neq \varnothing \right\} & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \left\{ s \in S \,;\, R[s] \subseteq \widehat{[\![\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \right\} & \text{if } \mathfrak{a} \notin \mathfrak{B}, \text{ and} \end{cases}$$

$$\widehat{[\![\overline{\mathsf{role}}\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \widehat{[\![\varphi]\!]^{\mathrm{BASIC},\mathfrak{A}}_{\mathbb{M}}}(\mathsf{role}(\mathfrak{a})).$$

Proof sketch. Like in Proposition 4.1.9, we can show this by induction on formulas $\varphi \in \mathsf{Lang}_{\mathrm{BASIC},\mathfrak{A}}$, in which we apply Proposition 4.2.14. The case for role switch formulas $\overline{\mathsf{role}}\varphi$ again requires a lemma, stating that for all formulas φ, a player $\mathfrak{a}$ wins the pointed game $(\mathscr{Eval}^{\mathrm{BASIC}}_{\mathfrak{A}}\mathbb{M})@\left\langle \varphi, s, \mathsf{role} \right\rangle$ iff $\mathsf{role}(\mathfrak{a})$ wins $(\mathscr{Eval}^{\mathrm{BASIC}}_{\mathfrak{A}}\mathbb{M})@\left\langle \varphi, s, \mathsf{id}_{\mathfrak{A}} \right\rangle$. This lemma is also shown through simple induction on formulas. □

4.2.4 Connectives and Negation

As discussed in Section 4.1.2 and afterwards mentioned in the beginning of Section 4.2, the usage of nondeterministic multiplayer games is a solution to the lack of usual connectives (like conjunction and negation) and alternative modalities in deterministic multiplayer logic. We are now equipped to address how nondeterministic multiplayer games solve the first of these two problems. The second problem will be addressed in Section 4.3.

Let us start with the positive connectives — i.e. conjunction and the $\Box$-modality. We can now very naturally define abbreviations

$$\varphi \wedge_{\mathfrak{B}} \psi := \varphi \vee_{\mathfrak{A} \setminus \mathfrak{B}} \psi \text{ and}$$
$$\Box_{\mathfrak{B}} \varphi := \Diamond_{\mathfrak{A} \setminus \mathfrak{B}} \varphi$$

for $\mathfrak{B} \subseteq \mathfrak{A}$. This is quite intuitive — e.g. conjunction for the players in $\mathfrak{B}$ is precisely disjunction for the others. By Proposition 4.2.32, we in fact immediately find that

$$\widehat{[\![\varphi \wedge_{\mathfrak{B}} \psi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \widehat{[\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cap \widehat{[\![\psi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \widehat{[\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cup \widehat{[\![\psi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\Box_{\mathfrak{B}} \varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \left\{ s \in S \,;\, R[s] \subseteq \widehat{[\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \right\} & \text{if } \mathfrak{a} \in \mathfrak{B}, \text{ and} \\ \left\{ s \in S \,;\, R[s] \cap \widehat{[\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \neq \varnothing \right\} & \text{if } \mathfrak{a} \notin \mathfrak{B}. \end{cases}$$

We can take this even further, by considering the connectives for $\mathfrak{B} = \mathfrak{A}$. These can be seen as 'absolute' versions of the connectives that behave the same for *all* players. For example, we have that

$$\widehat{[\![\Diamond_{\mathfrak{A}} \varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \left\{ s \in S \,;\, R[s] \cap \widehat{[\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \neq \varnothing \right\}.$$

By definition of the transpose, we then find that

$$\begin{aligned} [\![\Diamond_{\mathfrak{A}} \varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}(s) &= \left\{ \mathfrak{a} \in \mathfrak{A} \,;\, R[s] \cap \widehat{[\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \neq \varnothing \right\} \\ &= \bigcup_{t \in R[s]} [\![\varphi]\!]^{\text{Basic},\mathfrak{A}}_{\mathbb{M}}(t). \end{aligned}$$

Comparing this with the semantics of Boolean-valued basic modal logic (Definition 3.1.5), it quickly becomes apparent that nondeterministic basic multiplayer logic 'contains' Boolean-valued basic modal logic, at least over agent- indexed Kripke models with the same accessibility relation for all agents. We will revisit relations between Boolean-valued modal logics and nondeterministic multiplayer logics in Section 5.1.

Introducing negation into the logic is not as simple as defining it as an abbreviation. Recall the way negation works in two-player evaluation games for two-valued logic. Note that the underlying two-player games there are quite specific types of two-player games: the Verifier and Falsifier are *antagonistic* towards each other, in the sense that they are playing a zero-sum game in which precisely one of them will win. The antagonism is enforced by e.g. giving precisely one of the two players a turn at any position. Negation works in these games by way of this antagonism, since it allows one to express through role switches that one of the players does *not* have a winning strategy at a given position.

Note that if we consider a (deterministic) two-player games in which the players are antagonistic in the sense just described, then it would suffice to know for just one of the players at which positions they get a turn and/or win, in order to be able to be able to specify the entire game's structure. Keeping this in mind, we could consider the evaluation games for two-valued logic to just be single-player games (with the player representing the Verifier) to which we add an antagonistic player.

These ideas can be naturally generalized to the multiplayer setting: we obtain negation by adding antagonists for each of the players. This also clarifies why the nondeterministic structure is necessary to introduce negation into multiplayer logic — adding antagonists for all players requires it to be possible for multiple players to get a turn at any given position.

Note that we can describe the process of adding antagonists for any general nondeterministic multiplayer game.

Definition 4.2.33. Given a set $\mathfrak{A}$ of players, its set of *antagonists* is a set $\mathfrak{A}^- := \{\mathfrak{a}^- ; \mathfrak{a} \in \mathfrak{A}\}$ of players disjoint from those in $\mathfrak{A}$. By slight abuse of notation, we will write $(\mathfrak{a}^-)^- := \mathfrak{a}$ for $\mathfrak{a} \in \mathfrak{A}$, and $\mathfrak{B}^- := \{\mathfrak{a}^- ; \mathfrak{a} \in \mathfrak{B}\}$. We also write $\mathfrak{B}_{\text{ANT}} := \mathfrak{B} \cup (\mathfrak{A} \setminus \mathfrak{B})^-$ for $\mathfrak{B} \subseteq \mathfrak{A}$. Additionally, we define for each $\mathfrak{B} \subseteq \mathfrak{A}$ the $(\mathfrak{A} \cup \mathfrak{A}^-)$-distribution $\mathsf{ant}_{\mathfrak{B}}$ defined by putting

$$\mathsf{ant}_{\mathfrak{B}}(\mathfrak{a}) := \begin{cases} \mathfrak{a}^- & \text{if } \mathfrak{a} \in \mathfrak{B} \cup \mathfrak{B}^-, \text{ and} \\ \mathfrak{a} & \text{otherwise.} \end{cases}$$

Finally, given an $\mathfrak{A}$-distribution role, we define the $(\mathfrak{A} \cup \mathfrak{A}^-)$-distribution $\mathsf{role}_{\text{ANT}}$ as $\mathsf{role}_{\text{ANT}}(\mathfrak{a}) := \mathsf{role}(\mathfrak{a})$ for $\mathfrak{a} \in \mathfrak{A}$, and $\mathsf{role}_{\text{ANT}}(\mathfrak{a}^-) := (\mathsf{role}(\mathfrak{a}))^-$ for $\mathfrak{a}^- \in \mathfrak{A}^-$.

Given a pointed nondeterministic $\mathfrak{A}$-game $\mathbb{G}@p$ with $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$, the *antagonization of* $\mathbb{G}@p$ is the pointed nondeterministic $(\mathfrak{A} \cup \mathfrak{A}^-)$-game $\mathscr{Ant}(\mathbb{G}@p) := \big\langle \mathbb{G}_{\text{ANT}}, p \big\rangle$, where $\mathbb{G}_{\text{ANT}} := \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn}_{\text{ANT}} \rangle$ is defined by putting $\mathsf{turn}_{\text{ANT}}(q) := (\mathsf{turn}(q))_{\text{ANT}}$ for all $q \in \mathsf{Pos}$. ◁

For clarity: the set $\mathfrak{B}_{\text{ANT}}$ consists of all players in $\mathfrak{B}$, along with the antagonists of all players *not* in $\mathfrak{B}$. And the role distribution $\mathsf{ant}_{\mathfrak{B}}$ swaps the players in $\mathfrak{B}$ with their antagonists.

Proposition 4.2.34. *Let* $\mathbb{G}@p$ *be a pointed nondeterministic* $\mathfrak{A}$*-game, and* $\mathfrak{a} \in \mathfrak{A}$ *a player. Then the following are equivalent:*

(*i*) $\mathfrak{a}$ *angelically wins* $\mathbb{G}@p$,

(*ii*) $\mathfrak{a}$ *angelically wins* $\mathscr{A}\!\mathit{nt}(\mathbb{G}@p)$, *and*

(*iii*) $\mathfrak{a}^-$ *does not angelically win* $\mathscr{A}\!\mathit{nt}(\mathbb{G}@p)$.

Proof sketch. The equivalence of (i) and (ii) can be proven using the insight that we can abstract away from other players (cf. Proposition 4.2.18). The equivalence of (ii) and (iii) is a bit more involved, and requires a lot of case distinctions (i.e. p is final, p is nonfinal with $\mathfrak{a} \in \mathsf{turn}(p)$, and $\mathfrak{a} \notin \mathsf{turn}(p)$). It can be shown by using the property that there exist no infinite Adm-chains, together with Proposition 4.2.14. The method for this is quite similar to the method we will use in the proof of Theorem 4.2.39 in Section 4.2.5. □

Applying the same ideas as used in general antagonizations, we can modify the evaluation games of nondeterministic $\mathfrak{A}$-logic to include a negation connective $\neg_{\mathfrak{B}}$ for $\mathfrak{B} \subseteq \mathfrak{A}$ as an abbreviation of a role switch $\overline{\mathsf{ant}_{\mathfrak{B}}}\varphi$.

Definition 4.2.35. The *language of nondeterministic basic* $\mathfrak{A}$*-logic with negation* is the inductively defined set

$$\mathsf{Lang}_{\neg\mathrm{Basic},\mathfrak{A}} \ni \varphi ::= p \mid \bot_{\mathfrak{B}} \mid (\varphi \vee_{\mathfrak{B}} \varphi) \mid (\Diamond_{\mathfrak{B}}\varphi) \mid (\overline{\mathsf{role}}\varphi) \mid (\neg_{\mathfrak{B}}\varphi),$$

where $p \in \mathsf{Prop}$, $\mathfrak{B} \subseteq \mathfrak{A}$, and $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$. If $\mathfrak{B} = \{\mathfrak{a}\}$, we will usually omit the brackets.

Given a Kripke $\mathfrak{A}$-model $\mathbb{M} = \langle\langle S, R\rangle, \mathsf{col}\rangle$, the *nondeterministic basic* $\mathfrak{A}$*-logic evaluation game over* $\mathbb{M}$ *with negation* is the nondeterministic $(\mathfrak{A} \cup \mathfrak{A}^-)$-game $\mathscr{E}\!\mathit{val}_{\mathfrak{A}}^{\neg\mathrm{Basic}}\mathbb{M}$ with positions taken from the set

$$\mathsf{Lang}_{\neg\mathrm{Basic},\mathfrak{A}} \times S \times \{\mathsf{role}_{\mathrm{Ant}} ; \mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}\},$$

and other components specified in Table 4.3.

The *semantics of nondeterministic basic* $\mathfrak{A}$*-logic with negation* is given by the function

$$[\![-]\!]_{\mathbb{M}}^{\neg\mathrm{Basic},\mathfrak{A}} : \mathsf{Lang}_{\neg\mathrm{Basic},\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$$

defined by putting

$$[\![\varphi]\!]_{\mathbb{M}}^{\neg\mathrm{Basic},\mathfrak{A}}(s) := \left\{\mathfrak{a} \in \mathfrak{A} ; \mathfrak{a} \text{ angelically wins } (\mathscr{E}\!\mathit{val}_{\mathfrak{A}}^{\neg\mathrm{Basic}}\mathbb{M})@\left\langle \varphi, s, \mathsf{id}_{\mathfrak{A}\cup\mathfrak{A}^-}\right\rangle\right\}$$

for $\varphi \in \mathsf{Lang}_{\neg\mathrm{Basic},\mathfrak{A}}$ and $s \in S$. ◁

Table 4.3: Specification of a nondeterministic basic $\mathfrak{A}$-logic evaluation game with negation.

Position	Admissible moves	Turn
$\langle p, s, \mathsf{role}\rangle$	$\varnothing$	$(\mathfrak{A} \setminus \mathsf{role}^{-1}[\mathsf{col}(s)(p)])_{\text{ANT}}$
$\langle \bot_{\mathfrak{B}}, s, \mathsf{role}\rangle$	$\varnothing$	$(\mathsf{role}^{-1}[\mathfrak{B}])_{\text{ANT}}$
$\langle \varphi \vee_{\mathfrak{B}} \psi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{role}\rangle, \langle \psi, s, \mathsf{role}\rangle\}$	$(\mathsf{role}^{-1}[\mathfrak{B}])_{\text{ANT}}$
$\langle \Diamond_{\mathfrak{B}} \varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, t, \mathsf{role}\rangle ; t \in R[s]\}$	$(\mathsf{role}^{-1}[\mathfrak{B}])_{\text{ANT}}$
$\langle \overline{\mathsf{rd}}\varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{rd}_{\text{ANT}} \circ \mathsf{role}\rangle\}$	$\varnothing_{\text{ANT}} = \mathfrak{A}^-$
$\langle \neg_{\mathfrak{B}} \varphi, s, \mathsf{role}\rangle$	$\{\langle \varphi, s, \mathsf{ant}_{\mathfrak{B}} \circ \mathsf{role}\rangle\}$	$\varnothing_{\text{ANT}} = \mathfrak{A}^-$

Theorem 4.2.36. *Let* $\mathbb{M} = \langle \langle S, R\rangle, \mathsf{col}\rangle$ *be a Kripke* $\mathfrak{A}$*-model. Then for any* $p \in \mathsf{Prop}$, $\varphi, \psi \in \mathsf{Lang}_{\neg\text{Basic},\mathfrak{A}}$, $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$, $\mathfrak{B} \subseteq \mathfrak{A}$, *and* $\mathfrak{a} \in \mathfrak{A}$, *it holds that*

$$\widehat{[\![p]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \widehat{\widehat{\mathsf{col}}(p)}(\mathfrak{a}),$$

$$\widehat{[\![\bot_{\mathfrak{B}}]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \varnothing & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ S & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\varphi \vee_{\mathfrak{B}} \psi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cup \widehat{[\![\psi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \cap \widehat{[\![\psi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\Diamond_{\mathfrak{B}} \varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} \left\{ s \in S ; R[s] \cap \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \neq \varnothing \right\} & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \left\{ s \in S ; R[s] \subseteq \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) \right\} & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\overline{\mathsf{role}}\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathsf{role}(\mathfrak{a})),$$

$$\widehat{[\![\neg_{\mathfrak{B}} \varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) = \begin{cases} S \setminus \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \in \mathfrak{B}, \text{ and} \\ \widehat{[\![\varphi]\!]^{\neg\text{Basic},\mathfrak{A}}_{\mathbb{M}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \notin \mathfrak{B}. \end{cases}$$

Proof sketch. Similar to Propositions 4.1.9 and 4.2.32, we require a lemma showing that for any $\varphi \in \mathsf{Lang}_{\neg\text{Basic},\mathfrak{A}}$, $s \in S$, $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$, and $\mathfrak{a} \in \mathfrak{A}$, it holds that $\mathfrak{a}$ has an angelic winning strategy from position $\langle \varphi, s, \mathsf{role}_{\text{ANT}}\rangle$ in $\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}^{\neg\text{Basic}}_{\mathfrak{A}}\mathbb{M}$ iff $\mathsf{role}_{\text{ANT}}(\mathfrak{a})$ has an angelic winning strategy from position $\langle \varphi, s, \mathsf{id}_{\mathfrak{A}\cup\mathfrak{A}^-}\rangle$ in $\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}^{\neg\text{Basic}}_{\mathfrak{A}}\mathbb{M}$. But we now also need a lemma showing that $\mathfrak{a}$ has an angelic winning strategy from position $\langle \varphi, s, \mathsf{role}_{\text{ANT}}\rangle$ in $\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}^{\neg\text{Basic}}_{\mathfrak{A}}\mathbb{M}$ iff $\mathfrak{a}^-$ does not have an angelic winning strategy from position $\langle \varphi, s, \mathsf{role}_{\text{ANT}}\rangle$ in $\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}^{\neg\text{Basic}}_{\mathfrak{A}}\mathbb{M}$. The theorem can then be proven by an induction in which we apply Proposition 4.2.14. □

4.2.5 Bisimulations and Adequacy

Having defined nondeterministic basic multiplayer logic, it is natural to ask whether it is adequate with respect to bisimulations on Kripke multiplayer models, which are defined in the obvious way. We will answer this question via a general property of nondeterministic multiplayer games.

Note that nondeterministic multiplayer games are structurally identical to Kripke models, with the turn function corresponding to a colouring, and the added requirement that the accessibility relation has no infinite chains. More precisely, nondeterministic $\mathfrak{A}$-games can be seen as specific instances of coalgebras for the set functor sending sets X to $\mathcal{P}X \times \mathcal{P}\mathfrak{A}$. This perspective is interesting, as we can then use general coalgebraic notions to study nondeterministic multiplayer games. We will make use of the coalgebraic definition of bisimulations, and obtain a definition of *nondeterministic multiplayer game bisimulations*. Unpacking these, we see that they can be defined as follows.

Definition 4.2.37. Given nondeterministic $\mathfrak{A}$-games $\mathbb{G} = \langle \mathsf{Pos}, \mathsf{Adm}, \mathsf{turn} \rangle$ and $\mathbb{G}' = \langle \mathsf{Pos}', \mathsf{Adm}', \mathsf{turn}' \rangle$, a binary relation $B \subseteq \mathsf{Pos} \times \mathsf{Pos}'$ is called a *nondeterministic $\mathfrak{A}$-game bisimulation* $B : \mathbb{G} \leftrightarrows \mathbb{G}'$ if the conditions

(turn) $\mathsf{turn}(p) = \mathsf{turn}'(p')$,

(forth) for all $q \in \mathsf{Adm}[p]$, there exists $q' \in \mathsf{Adm}'[p']$ such that qBq', and

(back) for all $q' \in \mathsf{Adm}'[p']$, there exists $q \in \mathsf{Adm}[p]$ such that qBq'

hold for all $p \in \mathsf{Pos}$ and $p' \in \mathsf{Pos}'$ such that pBp'.

If there is some B such that $B : \mathbb{G} \leftrightarrows \mathbb{G}'$, we write $\mathbb{G} \leftrightarrows \mathbb{G}'$. If pBp', we additionally write $\mathbb{G}, p \leftrightarrows \mathbb{G}', p'$. ◁

It is important to note that we are not making any claims about nondeterministic multiplayer game bisimulations being the *right* definition of bisimulations for nondeterministic multiplayer games. Instead, we consider these bisimulations to be simple definition that suffices for what we will state in Corollary 4.2.42: the semantics of deterministic multiplayer logic is invariant with respect to bisimilarity on Kripke multiplayer models. For this invariance, our definition of bisimulations is quite strong, as one can easily think of examples of games in which two positions have different sets of player that get a turn, but still have the same sets of angelically winning players. For more natural ways to approach bisimulations on games, we refer the interested reader to van Benthem, Bezhanishvili, and Enqvist (2019).

Remark 4.2.38. Using the coalgebraic view of nondeterministic $\mathfrak{A}$-games, we also obtain a natural notion of a nondeterministic $\mathfrak{A}$-game homomorphism, through which we can define a category $\mathbf{PND}_{\mathfrak{A}}$ of pointed nondeterministic multiplayer games and point-preserving nondeterministic $\mathfrak{A}$-game

homomorphisms. The astute reader might have noticed that we have typeset many of the constructions throughout this chapter like functors. All of these constructions can in fact be extended to become functors from and to (suitable variations on) $\mathbf{PND}_{\mathfrak{A}}$. ◁

The reason nondeterministic $\mathfrak{A}$-game bisimulations are of interest to us lies in the important fact that the existence of angelic winning strategies is invariant with respect to them.

Theorem 4.2.39. *Let* $\mathbb{G}$ *and* $\mathbb{G}'$ *be nondeterministic* $\mathfrak{A}$*-games with sets of positions* Pos *and* Pos'*, respectively. Then for all* $p \in \mathsf{Pos}$ *and* $p' \in \mathsf{Pos}'$*, it holds that* $\mathbb{G}, p \leftrightarrows \mathbb{G}', p'$ *implies that* $\mathfrak{a}$ *angelically wins* $\mathbb{G}@p$ *iff* $\mathfrak{a}$ *angelically wins* $\mathbb{G}'@p'$ *for all players* $\mathfrak{a} \in \mathfrak{A}$*.*

Proof. We show that for every nondeterministic $\mathfrak{A}$-game bisimulation $B : \mathbb{G} \leftrightarrows \mathbb{G}'$, it holds for all $\langle p, p' \rangle \in B$ and $\mathfrak{a} \in \mathfrak{A}$ that $\mathfrak{a}$ angelically wins $\mathbb{G}@p$ iff $\mathfrak{a}$ angelically wins $\mathbb{G}'@p'$. Fix such B and $\mathfrak{a}$.

Define $W \subseteq B$ to be the set of pairs $\langle p, p' \rangle \in B$ such that it is *not* the case that $\mathfrak{a}$ angelically wins $\mathbb{G}@p$ iff $\mathfrak{a}$ angelically wins $\mathbb{G}'@p'$. To prove the theorem, we assume for the sake of contradiction that $W \neq \varnothing$. Define a relation $\sqsubseteq \subseteq B \times B$ by putting $\langle p, p' \rangle \sqsubseteq \langle q, q' \rangle$ iff $p \ \mathsf{Adm}^* \ q$ and $p' (\mathsf{Adm}')^* q'$.[9] Since there exist neither infinite Adm- nor infinite Adm'-chains by definition of nondeterministic $\mathfrak{A}$-games, it is easily verified that $\sqsubseteq$ is a partial order on B. We denote the corresponding strict order by $\sqsubset$, defined as $\langle p, p' \rangle \sqsubset \langle q, q' \rangle$ iff $\langle p, p' \rangle \sqsubseteq \langle q, q' \rangle$ and $\langle p, p' \rangle \neq \langle q, q' \rangle$.

Again due to the absence of infinite Adm- and Adm'-chains, $\sqsubseteq$ can be seen to satisfy the ascending chain condition — i.e. there exists no infinite $\sqsubset$-chain. It follows from this that for all nonempty subsets $C \subseteq B$, there exist $\sqsubset$-maximal pairs $\langle p, p' \rangle \in C$ — i.e. there exist pairs $\langle p, p' \rangle \in C$ such that there is no $\langle q, q' \rangle \in C$ with $\langle p, p' \rangle \sqsubset \langle q, q' \rangle$. If this would not hold, then there would exist nonempty $C \subseteq B$ such that for all $\langle p, p' \rangle \in C$, there exists $\langle q, q' \rangle \in C$ with $\langle p, p' \rangle \sqsubset \langle q, q' \rangle$, contradicting the ascending chain condition.

Applying this to W, it follows that there is some $\sqsubset$-maximal pair $\langle p, p' \rangle \in W$. In other words, pBp' holds, but it is not the case that $\mathfrak{a}$ angelically wins $\mathbb{G}@p$ iff $\mathfrak{a}$ angelically wins $\mathbb{G}'@p'$. We can assume without loss of generality that $\mathfrak{a}$ angelically wins $\mathbb{G}@p$, while $\mathfrak{a}$ does *not* angelically win $\mathbb{G}'@p'$. Note that since p and p' are related by the nondeterministic $\mathfrak{A}$-game bisimulation B, precisely one of the following two conditions must hold: (i) $p \in \mathsf{Fin}$ and $p' \in \mathsf{Fin}'$, or (ii) $\mathsf{Adm}[p] \neq \varnothing \neq \mathsf{Adm}'[p']$.

First, suppose (i) holds. Then because B is a nondeterministic $\mathfrak{A}$-game bisimulation, it follows from the (turn) condition that $\mathsf{turn}(p) = \mathsf{turn}'(p')$.

[9] Given a binary relation $R \subseteq X \times X$, we inductively define relations $R^n \subseteq X \times X$ for all $n < \omega$ by defining $R^0 := \mathsf{Diag}_X = \{\langle x, x \rangle ; x \in X\}$, and putting $xR^{n+1}y$ iff there is some $z \in X$ such that $xR^n zRy$. We then put $R^* := \bigcup_{n<\omega} R^n$.

Since a player has a relative winning strategy from a final position iff that player is not in the set of players that can get a turn there, it therefore holds that $\mathfrak{a}$ angelically wins $\mathbb{G}@p$ iff $\mathfrak{a}$ angelically wins $\mathbb{G}'@p'$. This contradicts the assumption that $\langle p, p' \rangle \in W$.

Second, suppose (ii) holds. Assume first that $\mathfrak{a} \in \mathsf{turn}(p)$. Since $\mathfrak{a}$ angelically wins $\mathbb{G}@p$, it follows from Proposition 4.2.14 that there is some $q \in \mathsf{Adm}[p]$ such that $\mathfrak{a}$ angelically wins $\mathbb{G}@q$. As B is a nondeterministic $\mathfrak{A}$-game bisimulation, it follows from the (forth) condition that there is some $q' \in \mathsf{Adm}'[p']$ such that qBq'. So $\langle p, p' \rangle \sqsubseteq \langle q, q' \rangle$. As there exist neither infinite Adm- nor infinite Adm'-chains, it must be that $q \neq p$ and $q' \neq p'$. So $\langle p, p' \rangle \sqsubset \langle q, q' \rangle$. As $\langle p, p' \rangle$ is $\sqsubset$-maximal in W, it must be that $\langle q, q' \rangle \notin W$. So by definition of W, it holds that $\mathfrak{a}$ angelically wins $\mathbb{G}@q$ iff $\mathfrak{a}$ angelically wins $\mathbb{G}'@q'$. As we already derived that $\mathfrak{a}$ angelically wins $\mathbb{G}@q$, it therefore also holds that $\mathfrak{a}$ angelically wins $\mathbb{G}'@q'$. Since it follows from the (turn) condition that $\mathsf{turn}(p) = \mathsf{turn}'(p')$, and thus, that $\mathfrak{a} \in \mathsf{turn}'(p')$, we can then conclude using Proposition 4.2.14 again that $\mathfrak{a}$ also angelically wins $\mathbb{G}'@p'$. This contradicts our assumption that $\langle p, p' \rangle \in W$.

Still assuming that (ii) holds, we now assume that $\mathfrak{a} \notin \mathsf{turn}(p)$ (and hence, also $\mathfrak{a} \notin \mathsf{turn}'(p')$). Now take any $q' \in \mathsf{Adm}'[p']$. As B is a nondeterministic $\mathfrak{A}$-game bisimulation, it follows from the (back) condition that there is some $q \in \mathsf{Adm}[p]$ such that qBq'. As before, it holds that $\langle p, p' \rangle \sqsubset \langle q, q' \rangle$, and therefore $\langle q, q' \rangle \notin W$, which means that $\mathfrak{a}$ angelically wins $\mathbb{G}@q$ iff $\mathfrak{a}$ angelically wins $\mathbb{G}'@q'$. Since $\mathfrak{a}$ angelically wins $\mathbb{G}@p$, it follows from Proposition 4.2.14 that $\mathfrak{a}$ angelically wins $\mathbb{G}@r$ for *all* $r \in \mathsf{Adm}[p]$. So $\mathfrak{a}$ must also angelically win $\mathbb{G}@q$. Thus, it follows that $\mathfrak{a}$ angelically wins $\mathbb{G}'@q'$. As $q' \in \mathsf{Adm}'[p']$ was arbitrary, we can then conclude using Proposition 4.2.14 that $\mathfrak{a}$ angelically wins $\mathbb{G}'@p'$. This yet again contradicts our assumption that $\langle p, p' \rangle \in W$.

Thus, our initial assumption that $W \neq \varnothing$ holds is contradictory. □

Using Theorem 4.2.39, we can now show that nondeterministic basic multiplayer logic is adequate with respect to bisimulations on Kripke multiplayer models. We do this by showing that bisimulations on Kripke multiplayer models lift to nondeterministic multiplayer game bisimulations on the evaluation games in a suitable way. Note that we consider nondeterministic basic multiplayer logic *without* negation for simplicity — the statement and proof of these results can be modified to also work for the logic with negation, but this does not offer any conceptual insight.

Definition 4.2.40. Given Kripke $\mathfrak{A}$-models $\mathbb{M} = \langle S, R, \mathsf{col} \rangle$ and $\mathbb{M}' = \langle S', R', \mathsf{col}' \rangle$, along with a relation $T \subseteq S \times S'$, the *nondeterministic basic $\mathfrak{A}$-logic evaluation game lifting of T* is the binary relation

$$\mathscr{E}\mathit{val}_{\mathfrak{A}}^{\text{BASIC}} T \subseteq (\mathsf{Lang}_{\text{BASIC},\mathfrak{A}} \times S \times \mathsf{RD}_{\mathfrak{A}}) \times (\mathsf{Lang}_{\text{BASIC},\mathfrak{A}} \times S' \times \mathsf{RD}_{\mathfrak{A}})$$

defined by putting $\langle \varphi, s, \mathsf{role} \rangle \, (\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} T) \, \langle \psi, s', \mathsf{role}' \rangle$ iff $\varphi = \psi$, $\mathsf{role} = \mathsf{role}'$, and sTs'. ◁

Theorem 4.2.41. *Let $\mathbb{M} = \langle S, R, \mathsf{col} \rangle$ and $\mathbb{M}' = \langle S', R', \mathsf{col}' \rangle$ be Kripke $\mathfrak{A}$-models. Then a relation $B \subseteq S \times S'$ is a Kripke $\mathfrak{A}$-model bisimulation*

$$B : \mathbb{M} \leftrightarrows \mathbb{M}'$$

iff $\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} B$ is a nondeterministic $\mathfrak{A}$-game bisimulation

$$\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} B : \mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} \mathbb{M} \leftrightarrows \mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} \mathbb{M}'.$$

Proof sketch. The direction from left to right follows very simply from the definitions of the turn function and admissibility relation. For the direction from right to left, we can derive $\mathsf{col}(s)(p) = \mathsf{col}'(s')(p)$ for sBs' and $p \in \mathsf{Prop}$ by considering the turn functions at positions $\langle p, s, \mathsf{id}_{\mathfrak{A}} \rangle$ and $\langle p, s', \mathsf{id}_{\mathfrak{A}} \rangle$. And we can derive the (back) and (forth) conditions by considering positions $\langle \Diamond_{\mathfrak{B}} \varphi, s, \mathsf{id}_{\mathfrak{A}} \rangle$ and $\langle \Diamond_{\mathfrak{B}} \varphi, s', \mathsf{id}_{\mathfrak{A}} \rangle$ for any formula φ, and applying the (back) and (forth) conditions of nondeterministic $\mathfrak{A}$-game bisimulations. □

Corollary 4.2.42. *Let $\mathbb{M} = \langle S, R, \mathsf{col} \rangle$ and $\mathbb{M}' = \langle S', R', \mathsf{col}' \rangle$ be Kripke $\mathfrak{A}$-models, and consider states s in $\mathbb{M}$ and s' in $\mathbb{M}'$. Then $\mathbb{M}, s \leftrightarrows \mathbb{M}', s'$ implies that*

$$[\![\varphi]\!]_{\mathbb{M}}^{\text{BASIC},\mathfrak{A}}(s) = [\![\varphi]\!]_{\mathbb{M}'}^{\text{BASIC},\mathfrak{A}}(s')$$

for all $\varphi \in \mathsf{Lang}_{\text{BASIC},\mathfrak{A}}$.

Proof. This follows by applying Theorem 4.2.39 to Theorem 4.2.41, noting that $\mathbb{M}, s \leftrightarrows \mathbb{M}', s'$ implies that

$$\mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} \mathbb{M}, \langle \varphi, s, \mathsf{id}_{\mathfrak{A}} \rangle \leftrightarrows \mathscr{E}\mathscr{v}\mathscr{a}\mathscr{l}_{\mathfrak{A}}^{\text{BASIC}} \mathbb{M}', \langle \varphi, s', \mathsf{id}_{\mathfrak{A}} \rangle$$

for all formulas φ. □

4.3 Coalgebraic Multiplayer Logic

As discussed in Section 4.1.2, the main challenge in a proper coalgebraic generalization of multiplayer logic as treated in *LAMP*[10] lies in how we approach modalities. In this section, we will show how we can achieve a coalgebraic generalization with a uniform treatment of modalities, by using nondeterministic multiplayer games as the basic game structure.

[10] Recall that we use the term *LAMP* to refer to the article by Olde Loohuis and Venema (2010) on which this chapter is based.

4.3.1 Predicate Liftings and Neighbourhood Frames

Both deterministic and nondeterministic (basic) multiplayer logic are interpreted over Kripke (multiplayer) models, and only include (multiplayer variations of) the $\Diamond$- and $\Box$-modality. The behaviour of these modalities is quite simple, in the sense that the semantics of the $\Diamond$-modality can be defined through existential quantification over successor states, while the semantics of the $\Box$-modality is defined through universal quantification over successor states. Using Propositions 4.1.8 and 4.2.14, this fits nicely into the multiplayer game structures we have considered, as existential quantification can be captured through a choice for some player(s), and universal quantification can be captured through a choice for other players.

But in general, modalities need not only be restricted to either of these two patterns (i.e. existential or universal quantification over states). Instead, they can have richer choice patterns, as can be seen from the example of the $\Diamond^k$-modality for $\mathscr{D}ist$-coalgebras considered in Section 4.1.2. The way we proposed to game-theoretically capture the semantics of $\Diamond^k$ there consisted of a choice pattern consisting of existential quantification (over sets of states) followed by universal quantification (over states in those sets).

As we will show, the semantics of a large class of modalities in the coalgebraic setting can always be captured through this latter choice pattern: existential quantification follows by universal quantification, or in game-theoretic terms, a choice by some player(s) followed by a choice by all other players.[11] The way we show this makes use of an essential property of predicate liftings that allows us to view any coalgebras as (*polyadic*) *neighbourhood frames*.

Neighbourhood frames are standard structures used in the study of non-normal modal logics.[12] They generalize Kripke frames, in which states are related to others states, by considering states that are related to *sets* of states, referred to as *neighbourhoods*.

Definition 4.3.1. A *neighbourhood frame* is a pair $\mathbb{N} = \langle S, \mathsf{neigh}\rangle$, where S is a set of *states*, and $\mathsf{neigh} : S \to \mathscr{P}\mathscr{P}S$ is a *neighbourhood function*, mapping states s to the set $\mathsf{neigh}(s) \subseteq \mathscr{P}S$ of *neighbourhoods of* s. $\lhd$

Coalgebraically, neighbourhood frames are coalgebras for the functor $\mathscr{N} : \mathbf{Set} \to \mathbf{Set}$ defined as $\mathscr{N} := 2^{2^-}$. This coalgebraic perspective is essential, as we will now show.[13]

[11] We wish to briefly remark that this is quite similar to the way Venema (2006) defines evaluation games for coalgebraic fixed point logic, in which modal formulas are captured by two consecutive positions, with the Verifier (corresponding to existential quantification) moving first, followed by the Falsifier (corresponding to universal quantification).

[12] We refer the interested reader to Chellas (1980) and Hansen, Kupke, and Pacuit (2009) for an overview of the study of neighbourhood frames within modal logic.

[13] What we are about to show is well-known within the field of coalgebraic logic, albeit

We will begin in a simple setting with a single unary modality. Consider a unary predicate $\mathcal{T}$-lifting $\mathbf{lift} : \mathscr{Forg}_{\mathbf{BA},\mathbf{Set}} \circ \mathscr{Pred} \Rightarrow \mathscr{Forg}_{\mathbf{BA},\mathbf{Set}} \circ \mathscr{Pred} \circ \mathcal{T}$ for some set functor $\mathcal{T}$. Recalling that $\mathscr{Forg}_{\mathbf{BA},\mathbf{Set}} \circ \mathscr{Pred} = 2^{-}$, the lifting $\mathbf{lift}$ is more simply described as a natural transformation $\mathbf{lift} : 2^{-} \Rightarrow 2^{\mathcal{T}-}$. We have that the transposed collection of functions $\widehat{\mathsf{lift}_X} : \mathcal{T}X \to 2^{2^X}$ also defines a natural transformation $\widehat{\mathbf{lift}} : \mathcal{T} \Rightarrow \mathcal{N}$, as can be easily verified. Importantly, this natural transformation allows us to functorially transform $\mathcal{T}$-coalgebras into $\mathcal{N}$-coalgebras, i.e. neighbourhood frames.

Proposition 4.3.2. *Let $\mathcal{T}$ be a set functor, and $\mathbf{lift} : 2^{-} \Rightarrow 2^{\mathcal{T}-}$ be a unary predicate $\mathcal{T}$-lifting. Given a $\mathcal{T}$-coalgebra $\mathbb{S} = \langle S, \sigma \rangle$, define the neighbourhood frame $\mathbb{S}^{\mathbf{lift}} := \left\langle S, \widehat{\mathsf{lift}_S} \circ \sigma \right\rangle$. Given a function $f : S \to S'$, define $f^{\mathbf{lift}} := f$.*

The operation $(-)^{\mathbf{lift}}$ on $\mathcal{T}$-coalgebras and functions defines a functor

$$(-)^{\mathbf{lift}} : \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T}) \to \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{N}).$$

Proof. This follows immediately from the naturality of $\widehat{\mathbf{lift}}$. □

Predicate liftings are not always unary. General predicate liftings induce functorial transformations to so-called *polyadic neighbourhood frames*. These are quite naturally defined by associating to a state a set of *polyadic neighbourhoods*, each consisting of multiple sets of states.

Definition 4.3.3. For $n < \omega$, an *n-ary neighbourhood frame* (or more generally, a *polyadic neighbourhood frame*) is a pair $\mathbb{N} = \langle S, \mathsf{neigh} \rangle$, where S is a set of *states*, and $\mathsf{neigh} : S \to \mathcal{P}((\mathcal{P}S)^n)$ is an *n-ary neighbourhood function*, mapping states s to the set $\mathsf{neigh}(s) \subseteq (\mathcal{P}S)^n$ of *n-ary neighbourhoods of s*. ◁

We have that n-ary neighbourhood frames are coalgebras for the functor $\mathcal{N}_n : \mathbf{Set} \to \mathbf{Set}$ defined as $\mathcal{N}_n := 2^{(2^-)^n}$. Analogous to the situation before with Proposition 4.3.2, n-ary predicate $\mathcal{T}$-liftings $\mathbf{lift}$ induce natural transformations $\widehat{\mathbf{lift}} : \mathcal{T} \Rightarrow \mathcal{N}_n$, which again gives rise to a functor $(-)^{\mathbf{lift}} : \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T}) \to \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{N}_n)$ defined similarly.

Note that we have been considering a single predicate lifting so far. In the general setting, we will have a (two-valued) coalgebraic modal logic $\mathbb{L}\mathfrak{og} = \langle \langle \mathsf{Sym}, \mathsf{ar} \rangle, \mathbf{Lift} \rangle$ with $\mathbf{Lift} = \left\langle \mathbf{lift}^{\heartsuit} \right\rangle_{\heartsuit \in \mathsf{Sym}}$. By considering the transposes of each of these liftings, we can see that these logics still induce functorial transformations to a certain class of coalgebras.

largely folkloric. Part of the following is briefly mentioned by Hansen, Kupke, and Pacuit (2009, Remark 5.9). A much more restricted version of the following, however, was extensively treated by Pattinson (2001).

Definition 4.3.4. Given a set I and an I-indexed sequence $\boldsymbol{n} \in \omega^I$ of natural numbers, define the set functor $\mathcal{N}_{\boldsymbol{n}}$ as

$$\mathcal{N}_{\boldsymbol{n}} := \prod_{i \in I} \mathcal{N}_{n_i}.$$

Given a modal similarity type $\mathbb{Sim} = \langle \mathsf{Sym}, \mathsf{ar} \rangle$, we write $\mathcal{N}_{\mathbb{Sim}}$ for the functor $\mathcal{N}_{\mathsf{ar}}$ (where ar is treated as a Sym-indexed sequence of natural numbers).

Given a coalgebraic modal logic $\mathbb{Log} = \langle \mathbb{Sim}, \mathbf{Lift} \rangle$ over a set functor $\mathcal{T}$ and a $\mathcal{T}$-coalgebra $\mathbb{S} = \langle S, \sigma \rangle$, define the $\mathcal{N}_{\mathbb{Sim}}$-coalgebra $\mathbb{S}^{\mathbb{Log}}$ as

$$\mathbb{S}^{\mathbb{Log}} := \left\langle S, \left\langle \widehat{\mathsf{lift}^{\heartsuit}_S} \circ \sigma \right\rangle_{\heartsuit \in \mathsf{Sym}} \right\rangle,$$

and given a function $f : S \to S'$, define $f^{\mathbb{Log}}$ as $f^{\mathbb{Log}} := f$. ◁

Proposition 4.3.5. *For all coalgebraic modal logics $\mathbb{Log}$, the operation $(-)^{\mathbb{Log}}$ on $\mathcal{T}$-coalgebras and functions defines a functor*

$$(-)^{\mathbb{Log}} : \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T}) \to \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{N}_{\mathbb{Sim}}).$$

It will be of specific interest to us to consider a particular class of polyadic neighbourhood frames. Namely, the class of *polyadic monotone neighbourhood frames*. These are polyadic neighbourhood frames in which the set $\mathsf{neigh}(s)$ is closed upwards with respect to set inclusion for each of the n neighbourhoods. We give a coalgebraic definition of this class, by defining the coalgebra type functor.

Definition 4.3.6. For $n < \omega$, the *n-ary monotone neighbourhood functor* $\mathscr{M}_n : \mathbf{Set} \to \mathbf{Set}$ is defined on sets X as

$$\begin{aligned}\mathscr{M}_n X := \{ & W \subseteq (\mathcal{P}X)^n \,;\, \langle U_1, \ldots, U_i \cup V, \ldots, U_n \rangle \in W \\ & \text{for all } \boldsymbol{U} \in W, 1 \leqslant i \leqslant n, V \subseteq X\},\end{aligned}$$

and on functions $f : X \to Y$ as $(\mathcal{N}_n f) \restriction_{\mathscr{M}_n X}$. We refer to $\mathscr{M}_n$-coalgebras as *n-ary monotone neighbourhood frames*. ◁

Note that $\mathscr{M}_n$-coalgebras are indeed also $\mathcal{N}_n$-coalgebras.

The reason we are interested in polyadic monotone neighbourhoods is based on their relation to *monotone predicate liftings*. These are predicate liftings representing modalities of which the semantics is monotone with respect to the semantics of given argument formulas. For example, the $\Box$-modality of basic modal logic is monotone, with $[\![\Box\varphi]\!] \subseteq [\![\Box\psi]\!]$ if $[\![\varphi]\!] \subseteq [\![\psi]\!]$.

Definition 4.3.7. Given a set functor $\mathcal{T}$, an n-ary predicate $\mathcal{T}$-lifting **lift** is *monotone* if for all sets S and $A_1, \ldots, A_n, B \subseteq \mathcal{P}S$, it holds that

$$\mathsf{lift}_S(A_1, \ldots, A_i, \ldots, A_n) \subseteq \mathsf{lift}_S(A_1, \ldots, A_i \cup B, \ldots, A_n)$$

for all $1 \leqslant i \leqslant n$.

A coalgebraic modal logic $\mathbb{L}\mathbb{o}\mathbb{g} = \langle \langle \mathsf{Sym}, \mathsf{ar} \rangle, \mathbf{Lift} \rangle$ over $\mathcal{T}$ is *monotone* if $\mathbf{lift}^{\heartsuit}$ is monotone for all $\heartsuit \in \mathsf{Sym}$. ◁

Proposition 4.3.8. *Let $\mathcal{T}$ be a set functor, and* **lift** *be an n-ary monotone predicate $\mathcal{T}$-lifting. Then the operation $(-)^{\mathbf{lift}}$ defines a functor $(-)^{\mathbf{lift}} : \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T}) \to \mathbf{Coalg}_{\mathbf{Set}}(\mathscr{M}_n)$.*

Defining $\mathscr{M}_{\boldsymbol{n}}$ and $\mathscr{M}_{\mathbb{S}\mathbb{i}\mathbb{m}}$ analogously to the way $\mathscr{N}_{\boldsymbol{n}}$ and $\mathscr{N}_{\mathbb{S}\mathbb{i}\mathbb{m}}$ were defined in Definition 4.3.4, it holds for all coalgebraic modal logics $\mathbb{L}\mathbb{o}\mathbb{g} = \langle \mathbb{S}\mathbb{i}\mathbb{m}, \mathbf{Lift} \rangle$ that the operation $(-)^{\mathbb{L}\mathbb{o}\mathbb{g}}$ defines a functor $(-)^{\mathbb{L}\mathbb{o}\mathbb{g}} : \mathbf{Coalg}_{\mathbf{Set}}(\mathcal{T}) \to \mathbf{Coalg}_{\mathbf{Set}}(\mathscr{M}_{\mathbb{S}\mathbb{i}\mathbb{m}})$.

Proof. This follows easily from the naturality of $\widehat{\mathbf{lift}^{\heartsuit}}$ and the monotonicity of $\mathbf{lift}^{\heartsuit}$ for each $\heartsuit \in \mathsf{Sym}$. □

So monotone coalgebraic modal logics induce transformations from $\mathcal{T}$-coalgebras to (products of) polyadic monotone neighbourhood frames. This is of interest to us because, as we will shortly show, it allows us to capture the semantics of monotone modalities through certain modalities for monotone neighbourhood frames, in a way that is amenable to the eventual game-theoretic generalization. These certain modalities are the *polyadic monotone $\Box$-modalities*, denoted by the symbols $\boxdot^n$ for $n < \omega$, and given by the following predicate liftings.

Definition 4.3.9. For $n < \omega$, the *n-ary monotone $\Box$-lifting* is the n-ary predicate $\mathscr{M}_n$-lifting $\mathbf{lift}^{\boxdot,n}$ defined by putting

$$\mathsf{lift}^{\boxdot,n}_S(A_1, \ldots, A_n) := \{U \in \mathscr{M}_n S \,;\, \langle A_1, \ldots, A_n \rangle \in U\}$$

for sets S and $A_1, \ldots, A_n \subseteq S$. ◁

Given the general setting in which there is more than one modality, it will also be helpful to define the following version of the monotone $\Box$-modality, denoted by the symbols $\boxdot^{\boldsymbol{n},i}$ for sequences $\boldsymbol{n}$ of natural numbers, and indices i.

Definition 4.3.10. Given a set I, an I-indexed sequence $\boldsymbol{n} \in \omega^I$, and an index $i \in I$, the *monotone $\Box$-lifting for $(\boldsymbol{n}, i)$* is the n_i-ary predicate $\mathscr{M}_{\boldsymbol{n}}$-lifting $\mathbf{lift}^{\boxdot,\boldsymbol{n},i}$ defined by putting

$$\mathsf{lift}^{\boxdot,\boldsymbol{n},i}_S(A_1, \ldots, A_{n_i}) := \left\{\boldsymbol{U} \in \mathscr{M}_{\boldsymbol{n}} S \,;\, U_i \in \mathsf{lift}^{\boxdot,n_i}_S(A_1, \ldots, A_{n_i})\right\}$$

for sets S and $A_1, \ldots, A_{n_i} \subseteq S$. ◁

Recall that the semantics of a formula $\heartsuit(\varphi_1, \dots, \varphi_{\mathsf{ar}\heartsuit})$ in a coalgebraic modal logic $\mathbb{Log}$ is given as

$$[\![\heartsuit(\varphi_1, \dots, \varphi_{\mathsf{ar}\heartsuit})]\!]^{\mathbb{Log}}_{\mathbb{S}} = (\mathscr{Pred}\,\sigma)(\mathsf{lift}^{\heartsuit}_S([\![\varphi_1]\!]^{\mathbb{Log}}_{\mathbb{S}}, \dots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]^{\mathbb{Log}}_{\mathbb{S}})).$$

We show the semantics of modalities for $\mathcal{T}$-coalgebras can be captured by the semantics of monotone $\Box$-modalities for polyadic monotone neighbourhood frame obtained through the $(-)^{\mathbb{Log}}$-functor as follows.[14]

Proposition 4.3.11. *Let $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$ be a $\mathcal{T}$-model, and $\mathbb{Log} = \langle\langle \mathsf{Sym}, \mathsf{ar}\rangle, \mathbf{Lift}\rangle$ be a monotone coalgebraic modal logic over $\mathcal{T}$. Then for all $\heartsuit \in \mathsf{Sym}$, it holds that*

$$(\mathscr{Pred}\,\sigma)(\mathsf{lift}^{\heartsuit}_S(A_1, \dots, A_{\mathsf{ar}\heartsuit}))$$

is equal to

$$(\mathscr{Pred}\,\langle \widehat{\mathsf{lift}^{\clubsuit}_S} \circ \sigma\rangle_{\clubsuit\in\mathsf{Sym}})(\mathsf{lift}^{\boxdot,\mathsf{ar},\heartsuit}_S(A_1, \dots, A_{\mathsf{ar}\heartsuit}))$$

for all sets S and $A_1, \dots, A_{\mathsf{ar}\heartsuit} \subseteq S$.

Proof. For brevity, we will write $\boldsymbol{A} := \langle A_1, \dots, A_{\mathsf{ar}\heartsuit}\rangle$. We compute and find that

$$\begin{aligned}
&(\mathscr{Pred}\,\langle \widehat{\mathsf{lift}^{\clubsuit}_S} \circ \sigma\rangle_{\clubsuit\in\mathsf{Sym}})(\mathsf{lift}^{\boxdot,\mathsf{ar},\heartsuit}_S(\boldsymbol{A})) \\
&= \left\{ s \in S \,;\, \left(\langle \widehat{\mathsf{lift}^{\clubsuit}_S} \circ \sigma\rangle_{\clubsuit\in\mathsf{Sym}}\right)(s) \in \mathsf{lift}^{\boxdot,\mathsf{ar},\heartsuit}_S(\boldsymbol{A}) \right\} && \text{(definition } \mathscr{Pred}\text{)} \\
&= \left\{ s \in S \,;\, \langle \widehat{\mathsf{lift}^{\clubsuit}_S}(\sigma(s))\rangle_{\clubsuit\in\mathsf{Sym}} \in \mathsf{lift}^{\boxdot,\mathsf{ar},\heartsuit}_S(\boldsymbol{A}) \right\} \\
&= \left\{ s \in S \,;\, \langle \left\{ \boldsymbol{T} \in (\mathscr{P}S)^{\mathsf{ar}\clubsuit} \,;\, \sigma(s) \in \mathsf{lift}^{\clubsuit}_S(\boldsymbol{T}) \right\}\rangle_{\clubsuit\in\mathsf{Sym}} \in \mathsf{lift}^{\boxdot,\mathsf{ar},\heartsuit}_S(\boldsymbol{A}) \right\} \\
& && \text{(definition transpose)} \\
&= \left\{ s \in S \,;\, \left\{ \boldsymbol{T} \in (\mathscr{P}S)^{\mathsf{ar}\heartsuit} \,;\, \sigma(s) \in \mathsf{lift}^{\heartsuit}_S(\boldsymbol{T}) \right\} \in \mathsf{lift}^{\boxdot,\mathsf{ar}\heartsuit}_S(\boldsymbol{A}) \right\} \\
& && \text{(definition } \mathbf{lift}^{\boxdot,\mathsf{ar},\heartsuit}\text{)} \\
&= \left\{ s \in S \,;\, \boldsymbol{A} \in \left\{ \boldsymbol{T} \in (\mathscr{P}S)^{\mathsf{ar}\heartsuit} \,;\, \sigma(s) \in \mathsf{lift}^{\heartsuit}_S(\boldsymbol{T}) \right\} \right\} && \text{(definition } \mathbf{lift}^{\boxdot,\mathsf{ar}\heartsuit}\text{)} \\
&= \left\{ s \in S \,;\, \sigma(s) \in \mathsf{lift}^{\heartsuit}_S(\boldsymbol{A}) \right\} \\
&= (\mathscr{Pred}\,\sigma)(\mathsf{lift}^{\heartsuit}_S(\boldsymbol{A})), && \text{(definition } \mathscr{Pred}\text{)}
\end{aligned}$$

which is what we set out to prove. □

Definition 4.3.12. Given a monotone coalgebraic modal logic $\mathbb{Log}$ with modal similarity type $\mathbb{Sim}$ over a set functor $\mathcal{T}$, its *monotone neighbourhood*

[14] The following proposition would also hold if $\mathbb{Log}$ were not monotone. The reason we state it for monotone logics specifically, is for ease of comparison with our eventual game-theoretic definitions.

translation is the coalgebraic modal logic $\mathbb{Log}_{\boxdot}$ over $\mathscr{M}_{\mathbb{Sim}}$ with modal similarity type

$$\left\langle \left\{ \boxdot^{\mathsf{ar},\heartsuit} ; \heartsuit \in \mathsf{Sym} \right\}, \boxdot^{\mathsf{ar},\heartsuit} \mapsto \mathsf{ar}\heartsuit \right\rangle$$

and the predicate $\mathscr{M}_{\mathbb{Sim}}$-lifting $\mathbf{lift}^{\boxdot,\mathsf{ar},\heartsuit}$ for the symbol $\boxdot^{\mathsf{ar},\heartsuit}$. We inductively define a translation $\mathsf{tr}_{\mathbb{Log},\boxdot}$ from the language of $\mathbb{Log}$ to the language of $\mathbb{Log}_{\boxdot}$ by putting

$$\begin{aligned}
\mathsf{tr}_{\mathbb{Log},\boxdot}(p) &:= p, \\
\mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi \vee \psi) &:= \mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi) \vee \mathsf{tr}_{\mathbb{Log},\boxdot}(\psi), \\
\mathsf{tr}_{\mathbb{Log},\boxdot}(\neg\varphi) &:= \neg\mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi), \text{ and} \\
\mathsf{tr}_{\mathbb{Log},\boxdot}(\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})) &:= \boxdot^{\mathsf{ar},\heartsuit}(\mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi_1), \ldots, \mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi_{\mathsf{ar}\heartsuit}))
\end{aligned}$$

for $p \in \mathsf{Prop}$ and $\heartsuit \in \mathsf{Sym}$. ◁

Corollary 4.3.13. *Let* $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$ *be a* $\mathcal{T}$*-model, and* $\mathbb{Log} = \langle \langle \mathsf{Sym}, \mathsf{ar}\rangle, \mathbf{Lift}\rangle$ *be a monotone coalgebraic modal logic over* $\mathcal{T}$*. Then it holds that*

$$[\![\varphi]\!]^{\mathbb{Log}}_{\mathbb{S}} = [\![\mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi)]\!]^{\mathbb{Log}_{\boxdot}}_{\mathbb{S}^{\mathbb{Log}}}$$

for all formulas φ *in the language of* $\mathbb{Log}$*.*[15]

Proof. Follows from the definition of $(-)^{\mathbb{Log}}$ and Proposition 4.3.11. □

4.3.2 Polyadic Monotone Neighbourhood Games

We now have all that we need to define nondeterministic multiplayer logics for arbitrary coalgebra types. The idea is to start from a monotone coalgebraic modal logic $\mathbb{Log}$ over a set functor $\mathcal{T}$ with modal similarity type $\mathbb{Sim}$, and to instead work in the logic $\mathbb{Log}_{\boxdot}$ over $\mathscr{M}_{\mathbb{Sim}}$. By Corollary 4.3.13, it holds that the semantics of $\mathbb{Log}$ is precisely captured by $\mathbb{Log}_{\boxdot}$, and so it will suffice to give a proper multiplayer version of the coalgebraic modal logic over functors $\mathscr{M}_{\boldsymbol{n}}$ for sequences $\boldsymbol{n} \in \omega^I$.

Definition 4.3.14. Given a sequence $\boldsymbol{n} \in \omega^I$, the *language of nondeterministic monotone neighbourhood* $\mathfrak{A}$*-logic w.r.t.* $\boldsymbol{n}$ is the inductively defined set

$$\mathsf{Lang}_{\mathscr{M}_{\boldsymbol{n}},\mathfrak{A}} \ni \varphi ::= p \mid \bot_{\mathfrak{B}} \mid (\varphi \vee_{\mathfrak{B}} \varphi) \mid (\boxdot^{\boldsymbol{n},i}_{\mathfrak{B}}(\underbrace{\varphi, \ldots, \varphi}_{n_i \text{ times}})) \mid (\overline{\mathsf{role}}\varphi),$$

where $p \in \mathsf{Prop}$, $\mathfrak{B} \subseteq \mathfrak{A}$, $i \in I$, and $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$. If $\mathfrak{B} = \{\mathfrak{a}\}$, we will usually omit the brackets. ◁

[15] Note that we extend the $(-)^{\mathbb{Log}}$-functor to operate on $\mathcal{T}$-models in the obvious way, by keeping colourings intact and only operating on the actual $\mathcal{T}$-coalgebra map.

Note that we can give an alternative characterization of the predicate liftings $\mathbf{lift}^{\boxdot,\boldsymbol{n},i}$ that makes the pattern of existential quantification followed by universal quantification explicit. By definition of polyadic monotone neighbourhood frames, it can be verified that $\mathsf{lift}_S^{\boxdot,\boldsymbol{n},i}(A_1,\dots,A_{n_i})$ is equal to the set

$$\{\boldsymbol{U} \in \mathscr{M}_{\boldsymbol{n}} S\,;\exists\,\langle X_1,\dots,X_{n_i}\rangle \in U_i\,(\forall(1 \leqslant k \leqslant n_i)(\forall s \in X_k : s \in A_k))\}\,. \quad (4.1)$$

We can capture this in nondeterministic monotone neighbourhood $\mathfrak{A}$-logic as follows. Given an $\mathscr{M}_{\boldsymbol{n}}$-coalgebra $\mathbb{S} = \langle S, \sigma\rangle$ (i.e. with $\sigma : S \to \prod_{i\in I} \mathscr{M}_{n_i} S$), we will construct the evaluation games in such a way that at a position corresponding to formula $\boxdot^{\boldsymbol{n},i}_{\mathfrak{B}}(\varphi_1,\dots,\varphi_{n_i})$ and state s, players in $\mathfrak{B}$ select an n_i-ary neighbourhood $\langle X_1,\dots,X_{n_i}\rangle \in \mathsf{proj}_i(\sigma(s))$, after which the other players (i.e. those in $\mathfrak{A}\setminus\mathfrak{B}$) select one of the sets X_k for $1 \leqslant k \leqslant n_i$, and a state $t \in X_k$. Play then continues from φ_k and t.

In order to formally define this, we will add another component to the positions in the evaluation games representing the n_i-ary neighbourhoods.

Definition 4.3.15. Given a multiplayer $\mathscr{M}_{\boldsymbol{n}}$-model[16] $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$ for some $\boldsymbol{n} \in \omega^I$, the *nondeterministic monotone neighbourhood $\mathfrak{A}$-logic evaluation game over* $\mathbb{S}$ is the nondeterministic $\mathfrak{A}$-game $\mathscr{E}\mathscr{V}\mathscr{A}\mathscr{L}^{\mathscr{M}_{\boldsymbol{n}}}_{\mathfrak{A}}\mathbb{S}$ with positions taken from the greatest subset

$$\mathsf{Pos} \subseteq \mathsf{Lang}_{\mathscr{M}_{\boldsymbol{n}},\mathfrak{A}} \times S \times \mathsf{RD}_{\mathfrak{A}} \times \left(\{*\} \cup \bigcup_{i\in I} (\mathscr{P}S)^{n_i}\right)$$

such that

$$\big\langle \varphi, s, \mathsf{role}, \boldsymbol{X}\big\rangle \in \mathsf{Pos}$$

only if $\varphi = \boxdot^{\boldsymbol{n},i}_{\mathfrak{B}}(\varphi_1,\dots,\varphi_{n_i})$ and $\boldsymbol{X} \in (\mathscr{P}S)^{n_i}$ for some $i \in I$, and with other components specified in Table 4.4.

The *semantics of nondeterministic monotone neighbourhood $\mathfrak{A}$-logic* is given by the function

$$[\![-]\!]^{\mathscr{M}_{\boldsymbol{n}},\mathfrak{A}}_{\mathbb{S}} : \mathsf{Lang}_{\mathscr{M}_{\boldsymbol{n}},\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$$

defined by putting

$$[\![\varphi]\!]^{\mathscr{M}_{\boldsymbol{n}},\mathfrak{A}}_{\mathbb{S}}(s) := \left\{\mathfrak{a} \in \mathfrak{A}\,;\,\mathfrak{a} \text{ angelically wins } (\mathscr{E}\mathscr{V}\mathscr{A}\mathscr{L}^{\mathscr{M}_{\boldsymbol{n}}}_{\mathfrak{A}}\mathbb{S})@\big\langle \varphi, s, \mathsf{id}_{\mathfrak{A}}, *\big\rangle\right\}$$

for $\varphi \in \mathsf{Lang}_{\mathscr{M}_{\boldsymbol{n}},\mathfrak{A}}$ and $s \in S$. ◁

[16] Analogously to how Kripke multiplayer models are Kripke models with an agent-indexed colouring, we consider multiplayer $\mathcal{T}$-models for a set functor $\mathcal{T}$ to be $\mathcal{T}$-models with an agent-indexed colouring.

Table 4.4: Specification of a nondeterministic monotone neighbourhood multiplayer logic evaluation game.

Position	Admissible moves	Turn
$\langle p, s, \mathsf{role}, * \rangle$	$\varnothing$	$\mathfrak{A} \setminus \mathsf{role}^{-1}[\mathsf{col}(s)(p)]$
$\langle \bot_{\mathfrak{B}}, s, \mathsf{role}, * \rangle$	$\varnothing$	$\mathsf{role}^{-1}[\mathfrak{B}]$
$\langle \varphi \vee_{\mathfrak{B}} \psi, s, \mathsf{role}, * \rangle$	$\{\langle \varphi, s, \mathsf{role}, * \rangle, \langle \psi, s, \mathsf{role}, * \rangle\}$	$\mathsf{role}^{-1}[\mathfrak{B}]$
$\langle \boxdot^{n,i}_{\mathfrak{B}}(\boldsymbol{\varphi}), s, \mathsf{role}, * \rangle$	$\{\langle \boxdot^{n,i}_{\mathfrak{B}}(\boldsymbol{\varphi}), s, \mathsf{role}, \boldsymbol{X} \rangle ; \boldsymbol{X} \in \mathsf{proj}_i(\sigma(s))\}$	$\mathsf{role}^{-1}[\mathfrak{B}]$
$\langle \boxdot^{n,i}_{\mathfrak{B}}(\boldsymbol{\varphi}), s, \mathsf{role}, \boldsymbol{X} \rangle$	$\{\langle \varphi_k, t, \mathsf{role}, * \rangle ; 1 \leqslant k \leqslant n_i \text{ and } t \in X_k\}$	$\mathsf{role}^{-1}[\mathfrak{A} \setminus \mathfrak{B}]$
$\langle \overline{\mathsf{rd}}\varphi, s, \mathsf{role}, * \rangle$	$\{\langle \varphi, s, \mathsf{rd} \circ \mathsf{role}, * \rangle\}$	$\varnothing$

Proposition 4.3.16. *Let $\mathbb{S} = \langle S, \sigma, \mathsf{col} \rangle$ be a multiplayer $\mathscr{M}_{\boldsymbol{n}}$-model for some $\boldsymbol{n} \in \omega^I$. Then for any $p \in \mathsf{Prop}$, $\varphi, \psi \in \mathsf{Lang}_{\mathscr{M}_n, \mathfrak{A}}$, $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$, $\mathfrak{B} \subseteq \mathfrak{A}$, and $\mathfrak{a} \in \mathfrak{A}$, it holds that*

$$\widehat{[\![p]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) = \widehat{\mathsf{col}(p)}(\mathfrak{a}),$$

$$\widehat{[\![\bot_{\mathfrak{B}}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) = \begin{cases} \varnothing & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ S & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\varphi \vee_{\mathfrak{B}} \psi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) = \begin{cases} \widehat{[\![\varphi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) \cup \widehat{[\![\psi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \widehat{[\![\varphi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) \cap \widehat{[\![\psi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\boxdot^{n,i}_{\mathfrak{B}}(\boldsymbol{\varphi})]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) = \begin{cases} (\mathscr{P}\!\mathit{red}\,\sigma)\left(\mathsf{lift}^{\boxdot,\boldsymbol{n},i}_{S}\left(\widehat{[\![\boldsymbol{\varphi}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a})\right)\right) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ S \setminus (\mathscr{P}\!\mathit{red}\,\sigma)\left(\mathsf{lift}^{\boxdot,\boldsymbol{n},i}_{S}\left(S \setminus \widehat{[\![\boldsymbol{\varphi}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a})\right)\right) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\overline{\mathsf{role}}\varphi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) = \widehat{[\![\varphi]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathsf{role}(\mathfrak{a})),$$

where we use abbreviations $\boldsymbol{\varphi} = \langle \varphi_1, \ldots, \varphi_{n_i} \rangle$,

$$\widehat{[\![\boldsymbol{\varphi}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) := \left\langle \widehat{[\![\varphi_1]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}), \ldots, \widehat{[\![\varphi_{n_i}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) \right\rangle, \text{ and}$$

$$S \setminus \widehat{[\![\boldsymbol{\varphi}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) := \left\langle S \setminus \widehat{[\![\varphi_1]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}), \ldots, S \setminus \widehat{[\![\varphi_{n_i}]\!]^{\mathscr{M}_n,\mathfrak{A}}_{\mathbb{S}}}(\mathfrak{a}) \right\rangle.$$

Proof sketch. This can be shown using Proposition 4.2.14. □

Remark 4.3.17. We can now briefly comment on why we required our coalgebraic modal logic to be *monotone*. Note that Equation (4.1) would not

hold if the coalgebraic modal logic was not monotone. As a result, we would need to define evaluation games in which at a position corresponding to formula $\boxdot^{n,i}_{\mathfrak{B}}(\varphi_1, \dots, \varphi_{n_i})$ and state s, players in $\mathfrak{B}$ choose an n_i-ary neighbourhood $\langle X_1, \dots, X_{n_i}\rangle \in \mathsf{proj}_i(\sigma(s))$, such that $X_i = \widehat{[\![\varphi_i]\!]_{\mathbb{S}}}(\mathfrak{a})$ for all $1 \leqslant i \leqslant n_i$. To guarantee these equalities using the machinery of angelic winning strategies, we would need to also capture that all states in $\widehat{[\![\varphi_i]\!]_{\mathbb{S}}}(\mathfrak{a})$ lie in X_i as well. The only way we know of to achieve this, is to allow the players not in $\mathfrak{B}$ to either select a state $x \in X_i$ for some i, from which play then continues using formula φ_i, or to select some i and a state x that is *not* in the set X_i, from which play would then continue using some sort of negation of the formula φ_i.

But these negations are not part of the logic we have just defined. Though we are able to modify evaluation games to also allow negation in Section 4.2.4, we still refrain from pursuing nondeterministic multiplayer logic for nonmonotone coalgebraic modal logic any further. This is mainly by virtue of how unnatural such a logic is: note that we would no longer even have the property that play in evaluation games always proceeds to subformulas, since play (as described earlier) could potentially move to the negation of a formula φ_i. But beyond this, monotonicity is a property we will also rely on in Section 5.1 when proving Proposition 5.1.4, showing nondeterministic $\mathfrak{A}$-logic and Boolean-valued coalgebraic modal logic are equiexpressive. ◁

Having defined the multiplayer variant of the coalgebraic modal logic over $\mathscr{M}_n$ with modalities $\boxdot^{n,i}$, we can now finally define the multiplayer variant of any arbitrary monotone coalgebraic modal logic.

Definition 4.3.18. Given a modal similarity type $\mathbb{Sim} = \langle \mathsf{Sym}, \mathsf{ar}\rangle$, the $\mathbb{Sim}$-*language of nondeterministic* $\mathfrak{A}$*-logic* is the inductively defined set

$$\mathsf{Lang}_{\mathrm{ND},\mathbb{Sim},\mathfrak{A}} \ni \varphi ::= p \mid \bot_{\mathfrak{B}} \mid (\varphi \vee_{\mathfrak{B}} \varphi) \mid (\heartsuit_{\mathfrak{B}}(\underbrace{\varphi, \dots, \varphi}_{\mathsf{ar}\heartsuit\ \text{times}})) \mid (\overline{\mathsf{role}}\varphi),$$

where $p \in \mathsf{Prop}$, $\mathfrak{B} \subseteq \mathfrak{A}$, $\heartsuit \in \mathsf{Sym}$, and $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$. If $\mathfrak{B} = \{\mathfrak{a}\}$, we will usually omit the brackets.

Given a monotone coalgebraic modal logic $\mathbb{Log}$ over $\mathscr{T}$ with modal similarity type $\mathbb{Sim}$, the *semantics of nondeterministic* $\mathfrak{A}$*-logic w.r.t.* $\mathbb{Log}$ for a multiplayer $\mathscr{T}$-model $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$ is given by the function

$$[\![-]\!]^{\mathbb{Log},\mathfrak{A}}_{\mathbb{S}} : \mathsf{Lang}_{\mathrm{ND},\mathbb{Sim},\mathfrak{A}} \to (\mathscr{P}\mathfrak{A})^S$$

defined by putting

$$[\![\varphi]\!]^{\mathbb{Log},\mathfrak{A}}_{\mathbb{S}} := [\![\mathsf{tr}_{\mathbb{Log},\boxdot}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathbb{S}^{\mathbb{Log}}}$$

for $\varphi \in \mathsf{Lang}_{\mathrm{ND},\mathbb{Sim},\mathfrak{A}}$, where $\mathsf{tr}_{\mathbb{Log},\boxdot}$ is the translation defined analogously to Definition 4.3.12, extended to preserve the $\mathfrak{B}$-subscripts. ◁

Theorem 4.3.19. *Let $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$ be a multiplayer $\mathcal{T}$-model for some set functor $\mathcal{T}$, and let $\mathbb{L}\mathbb{o}\mathbb{g}$ be a monotone coalgebraic modal logic over $\mathcal{T}$ with modal similarity type $\mathbb{S}\mathbb{i}\mathbb{m}$. Then for any $p \in \mathsf{Prop}$, $\varphi, \psi \in \mathsf{Lang}_{\mathrm{ND},\mathbb{S}\mathbb{i}\mathbb{m},\mathfrak{A}}$, $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$, $\mathfrak{B} \subseteq \mathfrak{A}$, and $\mathfrak{a} \in \mathfrak{A}$, it holds that*

$$\widehat{[\![p]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) = \widehat{\widehat{\mathsf{col}}(p)}(\mathfrak{a}),$$

$$\widehat{[\![\bot_{\mathfrak{B}}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) = \begin{cases} \varnothing & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ S & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\varphi \vee_{\mathfrak{B}} \psi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) = \begin{cases} \widehat{[\![\varphi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) \cup \widehat{[\![\psi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ \widehat{[\![\varphi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) \cap \widehat{[\![\psi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\heartsuit_{\mathfrak{B}}(\boldsymbol{\varphi})]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) = \begin{cases} (\mathscr{P}\!\mathit{red}\,\sigma)\left(\mathsf{lift}_S^{\heartsuit}\left(\widehat{[\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a})\right)\right) & \text{if } \mathfrak{a} \in \mathfrak{B}, \\ S \setminus (\mathscr{P}\!\mathit{red}\,\sigma)\left(\mathsf{lift}_S^{\heartsuit}\left(S \setminus \widehat{[\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a})\right)\right) & \text{if } \mathfrak{a} \notin \mathfrak{B}, \end{cases}$$

$$\widehat{[\![\overline{\mathsf{role}}\varphi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) = \widehat{[\![\varphi]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathsf{role}(\mathfrak{a})),$$

where we use abbreviations $\boldsymbol{\varphi} = \langle \varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit} \rangle$,

$$\widehat{[\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) := \left\langle \widehat{[\![\varphi_1]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}), \ldots, \widehat{[\![\varphi_{\mathsf{ar}\heartsuit}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) \right\rangle, \text{ and}$$

$$S \setminus \widehat{[\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) := \left\langle S \setminus \widehat{[\![\varphi_1]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}), \ldots, S \setminus \widehat{[\![\varphi_{\mathsf{ar}\heartsuit}]\!]_{\mathbb{S}}^{\mathbb{L}\mathbb{o}\mathbb{g},\mathfrak{A}}}(\mathfrak{a}) \right\rangle.$$

Proof sketch. Follows from Proposition 4.3.16 and corollary 4.3.13. □

Thus, we have obtained a coalgebraic generalization of the multiplayer logic in *LAMP*, at least when restricting ourselves to monotone coalgebraic modal logics. Though we do not pursue this any further for the sake of brevity, we note that many of the ideas considered in Sections 4.2.3 to 4.2.5 can also be applied to and proven for the coalgebraic multiplayer logic (e.g. defining additional logical connectives and duals of modalities, introducing negation, adequacy).

Chapter 5

Multiplayer Games and Multiagent-Valued Logic

In this chapter, we will consider two ways to bring together the Boolean-valued coalgebraic modal logics from Chapter 3 and the nondeterministic coalgebraic multiplayer logics from Chapter 4. In Section 5.1, we begin by showing that under some restrictions, nondeterministic coalgebraic multiplayer logics is equiexpressive to Boolean-valued coalgebraic modal logic. We then finish in Section 5.2 by showing how enriching Boolean-valued coalgebraic modal logic with the tools of nondeterministic coalgebraic multiplayer logic enables it to fully express more standard ways of defining multiagent versions of coalgebraic modal logics.

5.1 Equiexpressivity and Deagentization

Throughout this section, we fix a monotone two-valued coalgebraic modal logic $\mathbb{L}\mathrm{og}$ over a set functor $\mathcal{T}$ with modal similarity type $\langle \mathsf{Sym}, \mathsf{ar} \rangle$. We will assume that there exists at least one nonnullary modality — i.e. some $\clubsuit \in \mathsf{Sym}$ with $\mathsf{ar}\clubsuit \geqslant 1$.

The semantics of the Booleanization $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ is fully determined by the semantics of $\mathbb{L}\mathrm{og}$ applied to slices, as in the Coalgebraic Slicing Theorem (Theorem 3.2.19). Considering the semantics of nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ (cf. Theorem 4.3.19), we can informally observe that a similar property almost holds there, with role switches forming the only source of interaction between agents. We will make this observation precise: the nondeterministic $\mathfrak{A}$-logic without (arbitrary) role switches is equiexpressive to $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$, given suitable assumptions.

We briefly note that we will be working with a version of nondeterministic $\mathfrak{A}$-logic *with* negation, which can be defined analogously to the way we defined nondeterministic basic $\mathfrak{A}$-logic with negation (cf. Section 4.2.4).

When we speak of role switches, we will not mean it to include negation (which is implemented as role switches between players and their antagonists), but only role switches between players in $\mathfrak{A}$.

We need to be careful in defining when we consider these logics to be equiexpressive, since they are not interpreted over the same type of structure. The Booleanization $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ is interpreted over $\mathscr{T}_{\mathfrak{A}}$-models, while the nondeterministic $\mathfrak{A}$-logic is interpreted over multiplayer $\mathscr{T}$-models (i.e. $\mathscr{T}$-coalgebras with agent-indexed colourings). We will informally say one logic *is expressed by* another logic, if there is a transformation from the structures of the former logic to the structures of the latter logic, as well as a translation of the formulas of the former to the formulas of the latter, such that the semantics of the former logic are preserved by said transformation and translation.[1]

Using this informal definition, we will first tackle the matter of $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ expressing nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$. We will define the transformation between *pointed* versions of the structures in question, and show that the semantics is preserved for the specific given points. Generally speaking, a pointed $\mathscr{T}$-coalgebra for a set functor $\mathscr{T}$ is a pair $\langle \mathbb{S}, s\rangle$ such that s is a state in $\mathbb{S}$, referred to as a *basepoint*. The morphisms in the category $\mathbf{Coalg}^{*}_{\mathbf{Set}}(\mathscr{T})$ of pointed $\mathscr{T}$-coalgebras are $\mathscr{T}$-coalgebra morphisms $f : \mathbb{S} \to \mathbb{S}'$ that preserve basepoints. It is obvious how we can extend this notion of pointedness to also work with multiplayer models and coalgebras on $\mathbf{ASet}$ — note that for the latter, morphisms preserve basepoints in *all* slices.

We denote by $\mathscr{T}_{\mathrm{MP}}$ the functor defined as $\mathscr{T}_{\mathrm{MP}}X := \mathscr{T}X \times (\mathscr{P}\mathfrak{A})^{\mathsf{Prop}}$. Then pointed multiplayer $\mathscr{T}$-models will precisely be pointed $\mathscr{T}_{\mathrm{MP}}$-coalgebras. We can define an operation

$$\mathsf{Tr} : \mathsf{Obj}\big(\mathbf{Coalg}^{*}_{\mathbf{Set}}(\mathscr{T}_{\mathrm{MP}})\big) \to \mathsf{Obj}\big(\mathbf{Coalg}^{*}_{\mathbf{ASet}}(\mathscr{T}_{\mathfrak{A},\mathsf{Prop}})\big)$$

(where $\mathscr{T}_{\mathfrak{A},\mathsf{Prop}}$ is the type of $\mathscr{T}_{\mathfrak{A}}$-models) in the obvious manner, by keeping the colouring intact, and having the same transition structure for each agent. That is, we define

$$\mathsf{Tr}\,\langle S, \sigma, \mathsf{col}, s\rangle := \langle S, \mathfrak{a} \in \mathfrak{A} \mapsto \sigma, \mathsf{col}, s\rangle\,.$$

We inductively define a translation tr from the formulas of nondeterministic

[1] Though it can be natural to require functoriality of the transformations, we will not do so. Our definition is intended as a simple to understand definition of logics expressing one another, and we make no claims to this being *the* optimal definition.

$\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ (minus role switches) to the formulas of $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$, by putting

$$\begin{aligned}
\mathsf{tr}(p) &:= p, && (p \in \mathsf{Prop})\\
\mathsf{tr}(\bot_{\mathfrak{B}}) &:= \ulcorner \mathfrak{A} \setminus \mathfrak{B} \urcorner,\\
\mathsf{tr}(\varphi \vee_{\mathfrak{B}} \psi) &:= \ulcorner \mathfrak{B} \urcorner \to (\mathsf{tr}(\varphi) \vee \mathsf{tr}(\psi)) \wedge \ulcorner \mathfrak{A} \setminus \mathfrak{B} \urcorner \to (\mathsf{tr}(\varphi) \wedge \mathsf{tr}(\psi)),\\
\mathsf{tr}(\neg_{\mathfrak{B}} \varphi) &:= \ulcorner \mathfrak{B} \urcorner \to \neg \mathsf{tr}(\varphi) \wedge \ulcorner \mathfrak{A} \setminus \mathfrak{B} \urcorner \to \mathsf{tr}(\varphi), \text{ and}\\
\mathsf{tr}(\heartsuit_{\mathfrak{B}}(\boldsymbol{\varphi})) &:= \ulcorner \mathfrak{B} \urcorner \to \heartsuit(\mathbf{tr}(\boldsymbol{\varphi})) \wedge \ulcorner \mathfrak{A} \setminus \mathfrak{B} \urcorner \to \neg\heartsuit(\neg \mathbf{tr}(\boldsymbol{\varphi})),
\end{aligned}$$

where we use abbreviations $\boldsymbol{\varphi} = \langle \varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit} \rangle$,

$$\mathbf{tr}(\boldsymbol{\varphi}) = \langle \mathsf{tr}(\varphi_1), \ldots, \mathsf{tr}(\varphi_{\mathsf{ar}\heartsuit}) \rangle, \text{ and}$$

$$\neg\mathbf{tr}(\boldsymbol{\varphi}) := \langle \neg\mathsf{tr}(\varphi_1), \ldots, \neg\mathsf{tr}(\varphi_{\mathsf{ar}\heartsuit}) \rangle.$$

This translation makes use of bounding formulas, which were briefly mentioned in Section 3.1.2.

Theorem 5.1.1. *Let $\langle \mathbb{S}, s \rangle$ be a pointed multiplayer $\mathcal{T}$-model with $\mathbb{S} = \langle S, \sigma, \mathsf{col} \rangle$. Denoting the basepoint of $\mathsf{Tr}\,\langle \mathbb{S}, s \rangle$ by s^*, it holds that*

$$[\![\varphi]\!]^{\mathbb{L}\mathrm{og},\mathfrak{A}}_{\mathbb{S}}(s) = [\![\mathsf{tr}(\varphi)]\!]^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}}_{\mathsf{Tr}\mathbb{S}}(s^*)$$

for all formulas φ of nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$, minus role switches.

Proof sketch. We can show this by induction on φ, in which we apply Proposition 3.2.17 and theorem 4.3.19. Note that for any $\mathfrak{B} \subseteq \mathfrak{A}$ and formulas ψ and χ of $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$, we have that $\mathfrak{a} \in \mathfrak{B}$ implies that $\ulcorner \mathfrak{B} \urcorner \to \psi \wedge \ulcorner \mathfrak{A} \setminus \mathfrak{B} \urcorner \to \chi$ holds for $\mathfrak{a}$ (at some state) iff ψ holds for $\mathfrak{a}$, while $\mathfrak{a} \notin \mathfrak{B}$ implies that the formula holds for $\mathfrak{a}$ iff χ holds for $\mathfrak{a}$. □

It makes conceptual sense that the Booleanization $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ only expresses nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ as long as there are no role switches. By the Coalgebraic Slicing Theorem (Theorem 3.2.19), there is no interaction between different agents, which is precisely what role switches require.

Showing that nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ (minus role switches) expresses the Booleanization $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ is more difficult. This difficulty lies in the difference between the structures over which the two logics are interpreted. For nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ to express $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$, we need to find a proper way to transform arbitrary $\mathcal{T}_{\mathfrak{A}}$-models (which are *agent-indexed* coalgebras) to multiplayer $\mathcal{T}$-models, which are just $\mathcal{T}_{\mathfrak{A}}$-models in which agents share the same transition structure.[2]

We will define the transformation of structures and the translation in three steps.

[2]Note that the way $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ expresses nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ minus role switches implies that in a sense, the latter logic is just the former logic, interpreted over agent-indexed models in which each agent has the same transition structure, but potentially different colourings. So in essence, we will be showing a way to express Boolean-valued coalgebraic modal logic over coalgebras with 'two-valued' transition structure.

First Step

We begin the first step by noting that like in Definition 4.3.4, we can define a transformation[3]

$$(-)^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}} : \mathsf{Obj}\big(\mathbf{Coalg}^{*}_{\mathbf{ASet}}(\mathscr{T}_{\mathfrak{A},\mathsf{Prop}})\big) \to \mathsf{Obj}\big(\mathbf{Coalg}^{*}_{\mathbf{ASet}}(\mathscr{M}_{\mathbb{Sim},\mathfrak{A},\mathsf{Prop}})\big)$$

by putting

$$\langle S, \sigma, \mathsf{col}, s\rangle^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}} := \left\langle S, \mathfrak{a} \in \mathfrak{A} \mapsto \left\langle \widehat{\mathsf{lift}^{\heartsuit}_{S}} \circ \sigma_{\mathfrak{a}} \right\rangle_{\heartsuit \in \mathsf{Sym}}, \mathsf{col}, s \right\rangle.$$

We can verify, like in Proposition 4.3.11, that this transformation preserves the semantics, in the sense that

$$[\![\varphi]\!]^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}}_{\mathbb{S}}(s) = [\![\mathsf{tr}_{\mathbb{L}\mathrm{og},\boxdot,\mathfrak{A}}(\varphi)]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}}}(s) \tag{5.1}$$

for all φ in $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$, where $\mathsf{tr}_{\mathbb{L}\mathrm{og},\boxdot,\mathfrak{A}}$ is the translation from Definition 4.3.12 extended to include constants $\ulcorner \mathfrak{B} \urcorner$, and $\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}$ is the Booleanization of the coalgebraic modal logic $\mathbb{L}\mathrm{og}_{\boxdot}$ over $\mathscr{M}_{\mathbb{Sim}}$ also defined in Definition 4.3.12.

Second Step

Now we consider the second step. Given an agent-indexed polyadic monotone neighbourhood model, as is produced by the functor $(-)^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}}$ defined above, we will proceed in somewhat similar fashion to the way we defined angelic determinizations of nondeterministic $\mathfrak{A}$-games in Definition 4.2.24. There, we took the coproduct of angelizations for each agent, after which we introduced a new position from which one of the angelizations needed to be chosen. Here, we will take the coproduct of the slices of each agent, after which we will introduce a new state from which one of the slices needs to be chosen. We refer to the resulting multiplayer polyadic monotone neighbourhood model as a *deagentization*.

Definition 5.1.2. We fix an enumeration $\mathsf{Prop} = \{p_i \,;\, i < \omega\}$.

Given a pointed $\mathscr{M}_{\mathbb{Sim},\mathfrak{A}}$-model $\langle \mathbb{S}, s\rangle$ with $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$, define the multiplayer $\mathscr{M}_{\mathbb{Sim}}$-model $\langle S', \sigma', \mathsf{col}'\rangle$ by letting $\langle S', \sigma'\rangle$ be the coproduct

$$\sum_{\mathfrak{a}\in\mathfrak{A}} \langle S, \sigma_{\mathfrak{a}}\rangle,$$

and letting col′ be the unique function induced from col by the universal property of the coproduct $\sum_{\mathfrak{a}} S$ of sets.

The *deagentization* of $\langle \mathbb{S}, s\rangle$ is the pointed multiplayer $\mathscr{M}_{\mathbb{Sim}}$-model

$$\mathsf{Deag}\,\langle \mathbb{S}, s\rangle := \left\langle S' \cup \{s_{\mathrm{DEAG}}\}, \sigma^{\mathrm{DEAG}}, \mathsf{col}^{\mathrm{DEAG}}, s_{\mathrm{DEAG}}\right\rangle,$$

[3] We write $\mathsf{Obj}(\mathbf{C})$ for the collection of objects of a category $\mathbf{C}$.

where σ^{DEAG} is defined for $\heartsuit \in \mathsf{Sym}$ by putting $\sigma^{\mathrm{DEAG}}_{\heartsuit}(s') := \sigma'_{\heartsuit}(s')$ for $s' \in S'$, and

$$\sigma^{\mathrm{DEAG}}_{\heartsuit}(s_{\mathrm{DEAG}}) := (\mathscr{P}(S' \cup \{s_{\mathrm{DEAG}}\}) \setminus \{\varnothing\})^{\mathsf{ar}\heartsuit}.$$

The colouring is defined by putting

$$\begin{aligned}
\mathsf{col}^{\mathrm{DEAG}}(s^{\mathrm{DEAG}})(p_i) &:= \varnothing,\\
\mathsf{col}^{\mathrm{DEAG}}(\mathsf{inj}_{\mathfrak{a}}(s))(p_0) &:= \{\mathfrak{a}\},\\
\mathsf{col}^{\mathrm{DEAG}}(\mathsf{inj}_{\mathfrak{a}}(s))(p_{i+1}) &:= \mathsf{col}(s)(p_i),\\
\mathsf{col}^{\mathrm{DEAG}}(s')(p_0) &:= \varnothing,\\
\mathsf{col}^{\mathrm{DEAG}}(s')(p_{i+1}) &:= \mathsf{col}(s')(p_i),
\end{aligned}$$

where $s' \neq \mathsf{inj}_{\mathfrak{a}}(s)$ for all $\mathfrak{a} \in \mathfrak{A}$, and $s' \neq s^{\mathrm{DEAG}}$. ◁

Intuitively, the deagentization 'flattens' the multiagent structure in a polyadic monotone neighbourhood model by separating the slices through the use of the coproduct. The state s_{DEAG} is conceptually similar to the position p_{DET} in the angelic determinization (Definition 4.2.24). Where we could then consider p_{DET} to be a position from which a choice of one of the relativizations could be made by way of the admissibility relation, we are now working in a setting where we have a monotone neighbourhood function. So to enforce that one of the states $\mathsf{inj}_{\mathfrak{a}}(s)$ for $\mathfrak{a} \in \mathfrak{A}$ is 'chosen' from s_{DEAG}, we fix the propositional variable p_0, and will enforce through our translation of formulas that a neighbourhood is chosen from s_{DEAG} that satisfies p_0. In order to deal with formulas that contained p_0, we 'move forward' all propositional variables one place.

Though this is all quite informal, the corresponding translation of formulas should make things clearer. We will be translating to nondeterministic monotone neighbourhood $\mathfrak{A}$-logic.

Definition 5.1.3. Inductively define a function $\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}$ from the language of $\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}$ to the language of nondeterministic monotone neighbourhood $\mathfrak{A}$-logic w.r.t. $\mathbb{S}\mathrm{im}$ as

$$\begin{aligned}
\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(p_i) &:= p_{i+1}, && (i < \omega)\\
\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\ulcorner \mathfrak{B} \urcorner) &:= \bot_{\mathfrak{A} \setminus \mathfrak{B}},\\
\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi \vee \psi) &:= \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi) \vee_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\psi),\\
\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\neg\varphi) &:= \neg_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi), \text{ and}\\
\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\boxdot^{\mathsf{ar},\heartsuit}(\varphi_1,\dots,\varphi_{\mathsf{ar}\heartsuit})) &:= \boxdot^{\mathsf{ar},\heartsuit}_{\mathfrak{A}}(\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_1),\dots,\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_{\mathsf{ar}\heartsuit}))
\end{aligned}$$

Fix some $\clubsuit \in \mathsf{Sym}$ with $\mathsf{ar}\clubsuit \geqslant 1$. The *translation from the language of* $\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}$ *to the language of nondeterministic monotone neighbourhood* $\mathfrak{A}$*-logic w.r.t.* $\mathbb{S}\mathrm{im}$ is

then given by the function $\mathsf{tr}_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}$ defined as

$$\mathsf{tr}_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi) := \boxdot^{\mathsf{ar},\clubsuit}_{\mathfrak{A}}(\underbrace{p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi), \ldots, p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)}_{\mathsf{ar}\clubsuit \text{ times}})$$

for all φ in the language of $\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}$. ◁

Proposition 5.1.4. *Let $\langle \mathbb{S}, s\rangle$ be a pointed $\mathscr{M}_{\mathbb{S}\mathrm{im},\mathfrak{A}}$-model with $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$, and write* $\mathsf{Deag}\,\langle \mathbb{S}, s\rangle = \langle \mathsf{Deag}\mathbb{S}, s_{\mathrm{DEAG}}\rangle$ *and* $\mathsf{Deag}\mathbb{S} = \left\langle S_{\mathrm{DEAG}}, \sigma^{\mathrm{DEAG}}, \mathsf{col}^{\mathrm{DEAG}}\right\rangle$. *Then it holds that*

$$[\![\varphi]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}(s) = [\![\mathsf{tr}_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}(s_{\mathrm{DEAG}})$$

for all formulas φ of $\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}$.

Proof. We first prove by induction on φ that

$$\mathfrak{a} \in [\![\varphi]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}(t) \text{ iff } \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}(\mathsf{inj}_{\mathfrak{a}}(t)) \tag{5.2}$$

for all $t \in S$ and $\mathfrak{a} \in \mathfrak{A}$.

For the case of p_i for some $i < \omega$, we have that

$$\begin{aligned} \mathfrak{a} \in [\![p_i]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}(t) &\text{ iff } \mathfrak{a} \in \mathsf{col}(t)(p_i) \\ &\text{iff } \mathfrak{a} \in \mathsf{col}^{\mathrm{DEAG}}(\mathsf{inj}_{\mathfrak{a}}(t))(p_{i+1}) \\ &\text{iff } \mathfrak{a} \in [\![p_{i+1}]\!]^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}(\mathsf{inj}_{\mathfrak{a}}(t)), \end{aligned}$$

and so the statement follows from $\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(p_i) = p_{i+1}$. The case of $\ulcorner \mathfrak{B} \urcorner$ is immediate.

The cases for disjunction and negation follow using routine Boolean reasoning. Take a modal formula $\boxdot^{\mathsf{ar},\heartsuit}(\boldsymbol{\varphi})$ with $\boldsymbol{\varphi} = \langle \varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit}\rangle$, and write the abbreviations

$$[\![\boldsymbol{\varphi}]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}} := \left\langle [\![\varphi_1]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}, \ldots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}\right\rangle,$$

$$\widehat{[\![\boldsymbol{\varphi}]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}}(\mathfrak{a}) := \left\langle \widehat{[\![\varphi_1]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}}(\mathfrak{a}), \ldots, \widehat{[\![\varphi_{\mathsf{ar}\heartsuit}]\!]^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}_{\mathbb{S}}}(\mathfrak{a})\right\rangle, \text{ and}$$

$$\mathbf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\boldsymbol{\varphi}) := \left\langle \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_1), \ldots, \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_{\mathsf{ar}\heartsuit})\right\rangle.$$

We have that $\mathfrak{a} \in [\![\boxdot^{\mathsf{ar},\heartsuit}(\boldsymbol{\varphi})]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}(s)$ holds

$$\begin{aligned}
&\text{iff } \mathfrak{a} \in (\mathsf{lift}_{\mathfrak{A}}^{\boxdot,\mathsf{ar},\heartsuit})_S([\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}(\sigma_{\mathfrak{a}}(s))) && \text{(semantics } \mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}\text{)}\\
&\text{iff } \mathfrak{a} \in (\mathsf{lift}_{\mathfrak{A}}^{\boxdot,\mathsf{ar}\heartsuit})_S([\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}})(\sigma_{\mathfrak{a},\heartsuit}(s)) && \text{(definition } \mathbf{lift}^{\boxdot,\mathsf{ar},\heartsuit}\text{)}\\
&\text{iff } \widehat{[\![\boldsymbol{\varphi}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}}\,(\mathfrak{a}) \in \sigma_{\mathfrak{a},\heartsuit}(s) && \text{(definition } \mathbf{lift}^{\boxdot,\mathsf{ar}\heartsuit}\text{)}\\
&\text{iff } \left\{t \in S\,;\, \mathfrak{a} \in [\![\varphi_i]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\boxdot,\mathfrak{A}}}(t)\right\} \in \{U_i\,;\, \boldsymbol{U} \in \sigma_{\mathfrak{a},\heartsuit}(s)\}\\
&\qquad\qquad \text{(definition transpose, for all } 1 \leqslant i \leqslant \mathsf{ar}\heartsuit\text{)}\\
&\text{iff } \left\{t \in S\,;\, \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_i)]\!]_{\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(\mathsf{inj}_{\mathfrak{a}}(t))\right\} \in \{U_i\,;\, \boldsymbol{U} \in \sigma_{\mathfrak{a},\heartsuit}(s)\}\\
&\qquad\qquad \text{(IH, for all } 1 \leqslant i \leqslant \mathsf{ar}\heartsuit\text{)}\\
&\text{iff } \left\{\mathsf{inj}_{\mathfrak{a}}(t)\,;\, \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_i)]\!]_{\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(\mathsf{inj}_{\mathfrak{a}}(t))\right\} \in \{\mathsf{inj}_{\mathfrak{a}}[U_i]\,;\, \boldsymbol{U} \in \sigma_{\mathfrak{a},\heartsuit}(s)\}\\
&\qquad\qquad \text{(}\mathsf{inj}_{\mathfrak{a}}\text{ is injective, for all } 1 \leqslant i \leqslant \mathsf{ar}\heartsuit\text{)}\\
&\text{iff } \left\{\mathsf{inj}_{\mathfrak{a}}(t)\,;\, \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_i)]\!]_{\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(\mathsf{inj}_{\mathfrak{a}}(t))\right\} \in \mathsf{proj}_i[(\mathscr{M}_{\mathsf{ar}\heartsuit}\mathsf{inj}_{\mathfrak{a}})(\sigma_{\mathfrak{a},\heartsuit}(s))]\\
&\qquad\qquad \text{(definition } \mathscr{M}_{\mathsf{ar}\heartsuit}\text{, for all } 1 \leqslant i \leqslant \mathsf{ar}\heartsuit\text{)}\\
&\text{iff } \left\{\mathsf{inj}_{\mathfrak{a}}(t)\,;\, \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_i)]\!]_{\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(\mathsf{inj}_{\mathfrak{a}}(t))\right\} \in \mathsf{proj}_i[\sigma_{\heartsuit}^{\mathrm{DEAG}}(\mathsf{inj}_{\mathfrak{a}}(s))]\\
&\qquad\qquad \text{(coproduct in Definition 5.1.2, for all } 1 \leqslant i \leqslant \mathsf{ar}\heartsuit\text{)}\\
&\text{iff } \left\{t \in S_{\mathrm{DEAG}}\,;\, \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi_i)]\!]_{\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(t)\right\} \in \mathsf{proj}_i[\sigma_{\heartsuit}^{\mathrm{DEAG}}(\mathsf{inj}_{\mathfrak{a}}(s))]\\
&\qquad\qquad \text{(monotonicity, for all } 1 \leqslant i \leqslant \mathsf{ar}\heartsuit\text{)}\\
&\text{iff } \widehat{[\![\mathbf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\boldsymbol{\varphi})]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}}\,(\mathfrak{a}) \in \sigma_{\heartsuit}^{\mathrm{DEAG}}(\mathsf{inj}_{\mathfrak{a}}(s)) && \text{(definition transpose)}\\
&\text{iff } \mathfrak{a} \in [\![\boxdot_{\mathfrak{A}}^{\mathsf{ar},\heartsuit}(\mathbf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\boldsymbol{\varphi}))]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(\mathsf{inj}_{\mathfrak{a}}(s)),\\
&\qquad\qquad \text{(semantics nondeterministic } \mathfrak{A}\text{-logic)}
\end{aligned}$$

completing the induction.

Having shown Equation (5.2), we now show that

$$\mathfrak{a} \in [\![\mathsf{tr}_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(s_{\mathrm{DEAG}}) \text{ iff } \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(\mathsf{inj}_{\mathfrak{a}}(s)) \tag{5.3}$$

for all $\mathfrak{a} \in \mathfrak{A}$ and formulas φ. By the semantics of nondeterministic $\mathfrak{A}$-logic, we find that $\mathfrak{a} \in [\![\mathsf{tr}_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(s_{\mathrm{DEAG}})$ holds

$$\begin{aligned}
&\text{iff } \mathfrak{a} \in [\![\boxdot_{\mathfrak{A}}^{\mathsf{ar},\clubsuit}(p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi), \ldots, p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi))]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}(s_{\mathrm{DEAG}})\\
&\text{iff } \left\{\widehat{[\![p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}}\,(\mathfrak{a})\right\}^{\mathsf{ar}\clubsuit} \in \sigma_{\clubsuit}^{\mathrm{DEAG}}(s_{\mathrm{DEAG}})\\
&\qquad\qquad \text{(semantics nondeterministic } \mathfrak{A}\text{-logic)}\\
&\text{iff } \left\{\widehat{[\![p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{S}\mathrm{im}}}(\varphi)]\!]_{\mathsf{Deag}\mathbb{S}}^{\mathscr{M}_{\mathbb{S}\mathrm{im}},\mathfrak{A}}}\,(\mathfrak{a})\right\}^{\mathsf{ar}\clubsuit} \in (\mathscr{P}S_{\mathrm{DEAG}} \setminus \{\varnothing\})^{\mathsf{ar}\clubsuit}\\
&\qquad\qquad \text{(definition } \sigma^{\mathrm{DEAG}}\text{)}
\end{aligned}$$

$$\text{iff } \overline{[\![p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}}(\mathfrak{a}) \neq \varnothing$$

$$\text{iff } \overline{[\![p_0]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}}(\mathfrak{a}) \cap \overline{[\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}}(\mathfrak{a}) \neq \varnothing$$

(semantics nondeterministic $\mathfrak{A}$-logic)

$$\text{iff } \overline{\overline{\mathsf{col}_{\mathrm{DEAG}}}(p_0)}(\mathfrak{a}) \cap \overline{[\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}}(\mathfrak{a}) \neq \varnothing$$

(semantics nondeterministic $\mathfrak{A}$-logic)

$$\text{iff } \{\mathsf{inj}_{\mathfrak{a}}(s)\} \cap \overline{[\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}}(\mathfrak{a}) \neq \varnothing \qquad \text{(definition } \mathsf{col}_{\mathrm{DEAG}})$$

$$\text{iff } \mathsf{inj}_{\mathfrak{a}}(s) \in \overline{[\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}}(\mathfrak{a})$$

$$\text{iff } \mathfrak{a} \in [\![\mathsf{tr}'_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\varphi)]\!]^{\mathscr{M}_{\mathbb{Sim}},\mathfrak{A}}_{\mathsf{Deag}\mathbb{S}}(\mathsf{inj}_{\mathfrak{a}}(s)). \qquad \text{(definition transpose)}$$

Putting together Equations (5.2) and (5.3), we obtain the statement we set out to prove. □

Third Step

Starting with a pointed $\mathscr{T}_{\mathfrak{A}}$-model $\langle \mathbb{S}, s\rangle$, we can go to a pointed $\mathscr{M}_{\mathbb{Sim},\mathfrak{A}}$-model $\langle \mathbb{S}, s\rangle^{\mathbb{Log}_{\mathfrak{A}}}$, which we can then deagentize to get the multiplayer $\mathscr{M}_{\mathbb{Sim}}$-model $\mathsf{Deag}(\langle \mathbb{S}, s\rangle^{\mathbb{Log}_{\mathfrak{A}}})$. Similarly, starting with a formula φ of $\mathbb{Log}_{\mathfrak{A}}$, we translate it to a formula $\mathsf{tr}_{\mathbb{Log},\boxdot,\mathfrak{A}}(\varphi)$ of $\mathbb{Log}_{\boxdot,\mathfrak{A}}$, which we then translate to a formula $\mathsf{tr}_{\boxdot,\mathscr{M}_{\mathbb{Sim}}}(\mathsf{tr}_{\mathbb{Log},\boxdot,\mathfrak{A}}(\varphi))$ of nondeterministic monotone neighbourhood $\mathfrak{A}$-logic w.r.t. $\mathbb{Sim}$.

But in order to obtain a proof of equiexpressivity, we now need to transform pointed $\mathscr{T}_{\mathfrak{A}}$-models to pointed multiplayer $\mathscr{T}$-models, and translate formulas of $\mathbb{Log}_{\mathfrak{A}}$ to formulas of nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{Log}$ (minus role switches). We will aim to transform pointed $\mathscr{T}_{\mathfrak{A}}$-models $\langle \mathbb{S}, s\rangle$ to pointed multiplayer $\mathscr{T}$-models $\mathsf{Deag}_{\mathscr{T}}\langle \mathbb{S}, s\rangle$, which we will refer to as *$\mathscr{T}$-deagentizations*. Informally, we define $\mathsf{Deag}_{\mathscr{T}}$ such that the diagram

$$\begin{array}{ccc}
\mathsf{Obj}\left(\mathbf{Coalg}^{*}_{\mathbf{ASet}}(\mathscr{T}_{\mathfrak{A},\mathsf{Prop}})\right) & & \\
\Big\downarrow{\scriptstyle (-)^{\mathbb{Log}_{\mathfrak{A}}}} & \searrow^{\mathsf{Deag}_{\mathscr{T}}} & \\
\mathsf{Obj}\left(\mathbf{Coalg}^{*}_{\mathbf{ASet}}(\mathscr{M}_{\mathbb{Sim},\mathfrak{A},\mathsf{Prop}})\right) & & \mathsf{Obj}\left(\mathbf{Coalg}^{*}_{\mathbf{Set}}(\mathscr{T}_{\mathrm{MP}})\right) \\
\Big\downarrow{\scriptstyle \mathsf{Deag}} & \swarrow_{(-)^{\mathbb{Log}}} & \\
\mathsf{Obj}\left(\mathbf{Coalg}^{*}_{\mathbf{Set}}(\mathscr{M}_{\mathbb{Sim},\mathrm{MP}})\right) & &
\end{array} \tag{5.4}$$

almost commutes. Let us make precise what we mean by this. Recall that we have fixed some nonnullary $\clubsuit \in \mathsf{Sym}$. We will write $\mathsf{Deag}(\langle \mathbb{S}, s\rangle^{\mathbb{Log}_{\mathfrak{A}}}) =$

$\langle \mathbb{S}', s'_* \rangle$ with $\mathbb{S}' = \langle S', \sigma', \mathsf{col}' \rangle$ and $(\mathsf{Deag}_{\mathscr{T}} \langle \mathbb{S}, s \rangle)^{\mathbb{L}\mathrm{og}} = \langle \mathbb{S}'', s''_* \rangle$ with $\mathbb{S}'' = \langle S'', \sigma'', \mathsf{col}'' \rangle$. We will define $\mathsf{Deag}_{\mathscr{T}}$ in such a way that $S' = S''$, $s'_* = s''_*$, $\mathsf{col}' = \mathsf{col}''$, $\sigma'_\heartsuit(s') = \sigma''_\heartsuit(s')$ for all $s' \in S' \setminus \{s'_*\}$ and $\heartsuit \in \mathsf{Sym}$, and $\sigma'_\clubsuit(s'_*) = \sigma''_\clubsuit(s'_*)$. Thus we justify the usage of the term *almost*: Diagram (5.4) would commute, were it not for the possibility that $\sigma'_\heartsuit(s'_*) \neq \sigma''_\heartsuit(s'_*)$ for $\heartsuit \neq \clubsuit$.

Let us first define the corresponding translation $\mathsf{tr}_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}$ from $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ to nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$. This is completely analogous to the composition $\mathsf{tr}_{\boxdot, \mathscr{M}_{\mathfrak{Sim}}} \circ \mathsf{tr}_{\mathbb{L}\mathrm{og}, \boxdot, \mathfrak{A}}$. We first inductively define

$$
\begin{aligned}
\mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(p_i) &:= p_{i+1}, && (i < \omega)\\
\mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\ulcorner \mathfrak{B} \urcorner) &:= \bot_{\mathfrak{A} \setminus \mathfrak{B}},\\
\mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi \vee \psi) &:= \mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi) \vee_{\mathfrak{A}} \mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\psi),\\
\mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\neg\varphi) &:= \neg_{\mathfrak{A}} \mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi), \text{ and}\\
\mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\heartsuit(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})) &:= \heartsuit_{\mathfrak{A}}(\mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi_1), \ldots, \mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi_{\mathsf{ar}\heartsuit})).
\end{aligned}
$$

Then we can define

$$
\mathsf{tr}_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi) := \clubsuit_{\mathfrak{A}}(\underbrace{p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi), \ldots, p_0 \wedge_{\mathfrak{A}} \mathsf{tr}'_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}(\varphi)}_{\mathsf{ar}\clubsuit \text{ times}})
$$

for all φ in $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$. Using this translation $\mathsf{tr}_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}$ along with a transformation $\mathsf{Deag}_{\mathscr{T}}$ making Diagram (5.4) almost commute in the sense described earlier, it will follow immediately from the semantics of nondeterministic $\mathfrak{A}$-logic w.r.t. $\mathbb{L}\mathrm{og}$ and Equation (5.1) and proposition 5.1.4 that the semantics gets preserved using said transformation and translation. To understand this better, observe from the definition of $\mathsf{tr}_{\mathscr{M}_{\mathfrak{Sim}}, \mathbb{L}\mathrm{og}}$ that intuitively we will only be interested in the $\clubsuit$-neighbourhood function, at least when we are considering the basepoint of a multiplayer $\mathscr{T}$-model. This is again quite informal, and will become more precise when we prove Proposition 5.1.7.

To define a transformation $\mathsf{Deag}_{\mathscr{T}}$ satisfying our criteria, we unfortunately require an additional assumption about the logic $\mathbb{L}\mathrm{og}$. Besides merely assuming there to be some $\clubsuit \in \mathsf{Sym}$ with $\mathsf{ar}\clubsuit \geqslant 1$, we also require that for all nonempty sets S, there exists some $U_S \in \mathscr{T}S$ such that

$$
U_S \in \mathsf{lift}^{\clubsuit}_S(A_1, \ldots, A_{\mathsf{ar}\clubsuit}) \text{ iff there exists } 1 \leqslant i \leqslant \mathsf{ar}\clubsuit \text{ such that } A_i \neq \emptyset,
$$

for all $A_1, \ldots, A_{\mathsf{ar}\clubsuit} \in \mathscr{P}S$. We will refer to this assumption as $(\star)$.

Remark 5.1.5. Our assumption of $(\star)$ is quite a strong assumption to make. In general, not every set functor admits a predicate lifting $\mathbf{lift}^{\clubsuit}$ (of any arity) such that $(\star)$ holds. Considering some of the classic examples of coalgebra functors (cf. Example 2.1.2), the interested reader can verify that the functors $\mathscr{Id}_{\mathbf{Set}}$, $C \times (-)$ for some set C, and $2 \times (-)^C$ do not admit predicate liftings satisfying $(\star)$, no matter the arity. Using the well-known bijective

correspondence between n-ary predicate $\mathcal{T}$-liftings and subsets $C \subseteq \mathcal{T}(2^n)$ (see e.g. Leal (2008)), it is not difficult to verify for e.g. the identity functor $\mathcal{Id}_{\mathbf{Set}}$ that $(\star)$ being satisfied would entail that $A_i \neq \emptyset$ for some i (as in the definition of $(\star)$) implies that there is some j such that $U_S \in A_j$, which obviously does not hold in general.

But other classic examples like $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{M}$, as well as the bag functor $\mathcal{B}\mathcal{a}\mathcal{g}$[4], all admit (unary) predicate liftings satisfying $(\star)$. Consdering e.g. $\mathcal{P}$, the lifting is in fact the one corresponding to the $\Diamond$-modality: taking $U_S := S$ works. Similarly, considering $\mathcal{B}\mathcal{a}\mathcal{g}$, the lifting is the one corresponding to the unary modality $\Diamond^{>0}$, with $\Diamond^{>0}\varphi$ holding at a state $s \in S$ given transition map σ iff there is some $t \in S$ at which φ holds such that $\sigma(s)(t) > 0$.

Though we do not pursue this further or try to argue formally for it, it informally seems to be the case that $(\star)$ holding for some coalgebra type $\mathcal{T}$ requires that states in $\mathcal{T}$-coalgebras can potentially 'reach' all other states. All examples of functors we gave for which $(\star)$ holds are of this kind, while the counterexamples are in fact not. $\triangleleft$

Definition 5.1.6. Given a pointed $\mathcal{T}_{\mathfrak{A}}$-model $\langle \mathbb{S}, s\rangle$ with $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$, define the multiplayer $\mathcal{T}$-model $\langle S', \sigma', \mathsf{col}'\rangle$ by letting $\langle S', \sigma'\rangle$ be the coproduct

$$\sum_{\mathfrak{a}\in\mathfrak{A}} \langle S, \sigma_{\mathfrak{a}}\rangle\,,$$

and letting col' be the unique function induced from col by the universal property of the coproduct $\sum_{\mathfrak{a}} S$ of sets.

The *$\mathcal{T}$-deagentization* of $\langle \mathbb{S}, s\rangle$ is the pointed multiplayer $\mathcal{T}$-model

$$\mathsf{Deag}_{\mathcal{T}}\langle \mathbb{S}, s\rangle := \left\langle S' \cup \{s_{\mathrm{DEAG}}\}\,, \sigma^{\mathrm{DEAG},\mathcal{T}}, \mathsf{col}^{\mathrm{DEAG}}, s_{\mathrm{DEAG}}\right\rangle,$$

where $\sigma^{\mathrm{DEAG},\mathcal{T}}$ is defined for $s' \in S'$ as $\sigma^{\mathrm{DEAG},\mathcal{T}}(s') := \sigma'(s')$, and for s_{DEAG} as $\sigma^{\mathrm{DEAG},\mathcal{T}}(s_{\mathrm{DEAG}}) := U_{S'\cup\{s_{\mathrm{DEAG}}\}}$, where $U_{S'\cup\{s_{\mathrm{DEAG}}\}}$ is given by our assumption of $(\star)$. The colouring $\mathsf{col}^{\mathrm{DEAG}}$ is defined identically to the colouring of the same name in Definition 5.1.2. $\triangleleft$

Proposition 5.1.7. *Let $\langle \mathbb{S}, s\rangle$ be a pointed $\mathcal{T}_{\mathfrak{A}}$-model with $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$, and write $\mathsf{Deag}(\langle \mathbb{S}, s\rangle^{\mathbb{L}\mathbb{o}\mathbb{g}_{\mathfrak{A}}}) = \left\langle S_{\mathrm{DEAG}}, \sigma^{\mathrm{DEAG}}, \mathsf{col}^{\mathsf{Deag}}, s_{\mathrm{DEAG}}\right\rangle$ and $(\mathsf{Deag}_{\mathcal{T}}\langle \mathbb{S}, s\rangle)^{\mathbb{L}\mathbb{o}\mathbb{g}} = \left\langle S_{\mathrm{DEAG}}, \sigma^{\mathrm{DEAG},\mathcal{T}}, \mathsf{col}^{\mathsf{Deag}}, s_{\mathrm{DEAG}}\right\rangle$. It holds that*

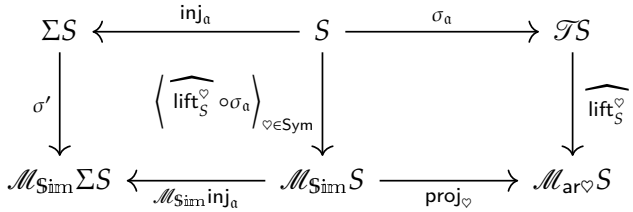
(i) $\sigma^{\mathrm{DEAG}}_{\heartsuit}(\mathsf{inj}_{\mathfrak{a}}(s)) = \sigma^{\mathrm{DEAG},\mathcal{T}}_{\heartsuit}(\mathsf{inj}_{\mathfrak{a}}(s))$ *for all* $s \in S$ *and* $\heartsuit \in \mathsf{Sym}$,

(ii) $\sigma^{\mathrm{DEAG}}_{\clubsuit}(s_{\mathrm{DEAG}}) = \sigma^{\mathrm{DEAG},\mathcal{T}}_{\clubsuit}(s_{\mathrm{DEAG}})$.

[4] Also known as the multiset functor, this functor sends sets X to the set $\mathcal{B}\mathcal{a}\mathcal{g}X := (\omega{+}1)^X$.

Proof. First of all, note that the theorem is stated correctly: the state spaces, colourings and basepoints of $\mathsf{Deag}(\langle \mathbb{S}, s\rangle^{\mathbb{L}\mathfrak{og}_{\mathfrak{A}}})$ and $(\mathsf{Deag}_{\mathcal{T}}\langle \mathbb{S}, s\rangle)^{\mathbb{L}\mathfrak{og}}$ are indeed equal.

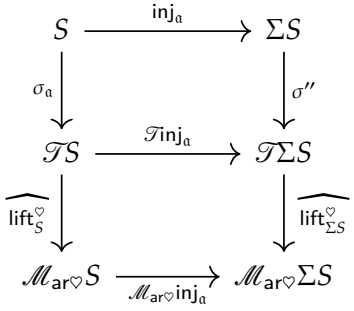
We begin by showing (i). Expanding the definition of $\mathsf{Deag}(\langle \mathbb{S}, s\rangle^{\mathbb{L}\mathfrak{og}_{\mathfrak{A}}})$, we denote the transition map of the coproduct obtained through the deagentization (Definition 5.1.2) by σ'. We can (partially) summarize the construction of $\mathsf{Deag}(\langle \mathbb{S}, s\rangle^{\mathbb{L}\mathfrak{og}_{\mathfrak{A}}})$ by stating that the diagram



(5.5)

commutes for all $\mathfrak{a} \in \mathfrak{A}$ and $\heartsuit \in \mathsf{Sym}$. We write Σ instead of $\sum_{\mathfrak{a}\in\mathfrak{A}}$ for simplicity.

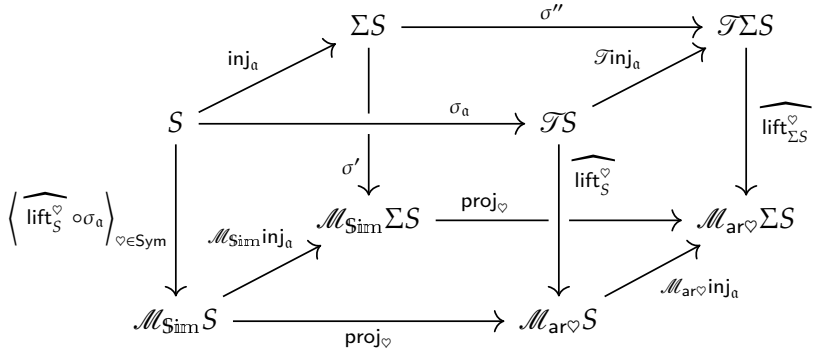
Denoting the transition map of the coproduct in the construction of the $\mathcal{T}$-deagentization $(\mathsf{Deag}_{\mathcal{T}}\langle \mathbb{S}, s\rangle)^{\mathbb{L}\mathfrak{og}}$ by σ'', we similarly find that the diagram



(5.6)

commutes for all $\mathfrak{a} \in \mathfrak{A}$ and $\heartsuit \in \mathsf{Sym}$.

Gluing Diagrams (5.5) and (5.6) together and adding the projection map $\mathsf{proj}_{\heartsuit} : \mathscr{M}_{\mathbb{S}\mathrm{im}}\Sigma S \to \mathscr{M}_{\mathsf{ar}\heartsuit}\Sigma S$, we obtain the diagram



(5.7)

for each $\mathfrak{a}$ and $\heartsuit$. It follows from Diagrams (5.5) and (5.6) commuting that Diagram (5.7) commutes if the bottom and back faces of the cube commute.

The back face of the cube commutes by definition of σ' and σ''. The commutativity of the bottom face can be easily verified by direct computation.

As a particular instance of the commutativity of Diagram (5.7), we then get that

$$\widehat{\mathsf{lift}^{\heartsuit}_{\Sigma S}} \circ \sigma'' \circ \mathsf{inj}_{\mathfrak{a}} = \mathsf{proj}_{\heartsuit} \circ \sigma' \circ \mathsf{inj}_{\mathfrak{a}},$$

from which (i) follows immediately by definition of $\sigma^{\textsc{Deag}}$ and $\sigma^{\textsc{Deag},\mathscr{T}}$.

It finally follows immediately from the definition of the operation $(-)^{\mathbb{L}\mathbb{O}\mathbb{G}}$ and the assumption of $(\star)$ that (ii) also holds. □

Theorem 5.1.8. *Let $\langle \mathbb{S}, s\rangle$ be a pointed $\mathscr{T}_{\mathfrak{A}}$-model with $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$, and write $\mathsf{Deag}_{\mathscr{T}}\langle \mathbb{S}, s\rangle = \langle \mathsf{Deag}_{\mathscr{T}}\mathbb{S}, s_{\textsc{Deag}}\rangle$. Then it holds that*

$$[\![\varphi]\!]^{\mathbb{L}\mathbb{O}\mathbb{G}_{\mathfrak{A}}}_{\mathbb{S}}(s) = [\![\mathsf{tr}_{\mathscr{M}_{\mathbf{Sim}},\mathbb{L}\mathbb{O}\mathbb{G}}(\varphi)]\!]^{\mathbb{L}\mathbb{O}\mathbb{G},\mathfrak{A}}_{\mathsf{Deag}_{\mathscr{T}}\mathbb{S}}(s_{\textsc{Deag}})$$

for all formulas φ of $\mathbb{L}\mathbb{O}\mathbb{G}_{\mathfrak{A}}$.

Proof. This follows from Equation (5.1) and propositions 5.1.4 and 5.1.7. □

5.2 Multiagent Modal Logic and Role Switches

In *How True*[5], Fitting gives examples showing how Boolean-valued basic modal logic "can facilitate the natural expression of things." These are examples of multiagent situations which would usually be expressed using propositional variables and modalities for each agent, like the motivating example we gave in Chapter 1, in which propositional variables $p_{\mathfrak{a}}$ and $p_{\mathfrak{b}}$ are replaced by a single propositional variable p. Fitting further alludes to such applications of Boolean-valued basic modal logic, by stating that e.g. using an epistemic interpretation of the $\Box$-modality, it holds that "a truth value for $\Box\varphi$, in $\mathscr{P}\{\mathfrak{a}, \mathfrak{b}\}$, is the set of [agents] who know φ."

But as the Extended Slicing Theorem (Theorem 3.1.8) and the more general Coalgebraic Slicing Theorem (Theorem 3.2.19) show, Boolean-valued modal logic is more limited in this regard than one would reasonably expect. This limitation arises from the fact that Boolean-valued modal logic allows no interactions between agents' slices, which corresponds to e.g. a lack of nested modalities for different agents. In this section, we will rigorously show that there are even further limitations, and propose role switches, from the nondeterministic $\mathfrak{A}$-logic of Section 4.3, as a way to circumvent these limitations.

We begin by making formal a *desideratum* stating that Boolean-valued coalgebraic modal logic "facilitates the natural expression of things." We interpret this as meaning that Boolean-valued coalgebraic modal logics over a

[5]Recall that we use the term *How True* to refer to the article by Fitting (2009) on which Chapter 3 is based.

set functor $\mathcal{T}$ (more specifically, Booleanizations $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$) express two-valued coalgebraic modal logics over the set functor $(\mathcal{T}(-))^{\mathfrak{A}}$, in which propositional variables and modalities are all indexed by agents. We refer to such two-valued coalgebraic modal logics as *multiagent coalgebraic modal logics*.

Definition 5.2.1. Let $\mathbb{L}\mathrm{og} = \langle \mathbb{S}\mathrm{im}, \mathbf{Lift}\rangle$ be a two-valued coalgebraic modal logic over a set functor $\mathcal{T}$, and Prop a set of propositional variables. The *multiagent coalgebraic modal logic* $\mathbb{L}\mathrm{og}_{\mathrm{MA}}$, over the set functor $(\mathcal{T}(-))^{\mathfrak{A}}$ and set

$$\mathsf{Prop}_{\mathrm{MA}} := \left\{p_{\mathfrak{a}}\,;\, p \in \mathsf{Prop}, \mathfrak{a} \in \mathfrak{A}\right\}$$

of propositional variables, is defined as $\mathbb{L}\mathrm{og}_{\mathrm{MA}} := \langle \mathbb{S}\mathrm{im}_{\mathrm{MA}}, \mathbf{Lift}_{\mathrm{MA}}\rangle$, where

$$\mathsf{Sym}_{\mathrm{MA}} := \{\heartsuit_{\mathfrak{a}}\,;\, \heartsuit \in \mathsf{Sym}, \mathfrak{a} \in \mathfrak{A}\}\,,$$

and

$$\mathbf{Lift}_{\mathrm{MA}} := \left\langle \mathbf{lift}^{\heartsuit_{\mathfrak{a}}}\right\rangle_{\heartsuit\in\mathsf{Sym},\mathfrak{a}\in\mathfrak{A}}$$

is defined by putting

$$\mathsf{lift}_S^{\heartsuit_{\mathfrak{a}}}(A_1,\ldots,A_{\mathsf{ar}\heartsuit}) := \left\{U \in (\mathcal{T}S)^{\mathfrak{A}}\,;\, U(\mathfrak{a}) \in \mathsf{lift}_S^{\heartsuit}(A_1,\ldots,A_{\mathsf{ar}\heartsuit})\right\}$$

for $A_1,\ldots,A_{\mathsf{ar}\heartsuit} \in \mathcal{P}S$. ◁

We can view $(\mathcal{T}(-))^{\mathfrak{A}}$-models (over $\mathsf{Prop}_{\mathrm{MA}}$) as $\mathcal{T}_{\mathfrak{A}}$-models (over Prop), and vice versa, giving the transformation between structures. This should be immediately clear for the transition maps, but is also easy for the colourings, since we can consider a colouring assigning $p_{\mathfrak{a}}$ and $p_{\mathfrak{b}}$ to a state s to be an agent-indexed colouring assigning $\{\mathfrak{a},\mathfrak{b}\}$ to p at s. Keeping this in mind, we will no longer distinguish these two types of models, and will refer to both as $\mathcal{T}_{\mathfrak{A}}$-models for brevity of notation.

Note that we need to be careful in defining when we consider the Booleanization $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ to express the multiagent coalgebraic modal logic $\mathbb{L}\mathrm{og}_{\mathrm{MA}}$, since their truth values differ. We will say the Booleanized coalgebraic modal logic $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ expresses the multiagent coalgebraic modal logic $\mathbb{L}\mathrm{og}_{\mathrm{MA}}$ if there is a translation tr from the formulas of $\mathbb{L}\mathrm{og}_{\mathrm{MA}}$ to those of $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$, such that for all $\mathcal{T}_{\mathfrak{A}}$-models $\mathbb{S} = \langle S, \sigma, \mathsf{col}\rangle$ and states $s \in S$, it holds that

$$s \in [\![\varphi]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}} \text{ iff } [\![\mathsf{tr}(\varphi)]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}}}(s) = \mathfrak{A}$$

for all formulas φ in $\mathbb{L}\mathrm{og}_{\mathrm{MA}}$. This is a natural way to define $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ expressing $\mathbb{L}\mathrm{og}_{\mathrm{MA}}$, as the truth value $\top$ in the two-element Boolean algebra corresponds to the truth value $\mathfrak{A}$ in the powerset $\mathcal{P}\mathfrak{A}$. In fact, Fitting also considers the assertion of a formula φ to mean that its truth value is $\mathfrak{A}$.

Let us first treat some examples in which the translation is easily found.

Example 5.2.2. Consider $\mathfrak{A} = \{\mathfrak{a}, \mathfrak{b}\}$. For single propositional variables $p_{\mathfrak{a}}$, the translation is easily given, with $\mathsf{tr}(p_{\mathfrak{a}}) := \ulcorner \mathfrak{a} \urcorner \to p$ doing the trick. Conjunctions $p_{\mathfrak{a}} \wedge p_{\mathfrak{b}}$ are also easily dealt with, by defining $\mathsf{tr}(p_{\mathfrak{a}} \wedge p_{\mathfrak{b}}) := (\ulcorner \mathfrak{a} \urcorner \to p) \wedge (\ulcorner \mathfrak{b} \urcorner \to p)$. Negations are very simple, with $\mathsf{tr}(\neg p_{\mathfrak{a}}) := \ulcorner \mathfrak{a} \urcorner \to \neg p$. We can even work with some nested modalities and more complex formulas. Considering a unary modality $\Diamond$ for simplicity, we can e.g. use the translation

$$\mathsf{tr}(\Diamond_{\mathfrak{a}} \Diamond_{\mathfrak{a}} p_{\mathfrak{a}} \wedge \neg p_{\mathfrak{b}}) := (\ulcorner \mathfrak{a} \urcorner \to \Diamond\Diamond p) \wedge (\ulcorner \mathfrak{b} \urcorner \to \neg p). \qquad \lhd$$

Unfortunately, we can also give counterexamples showing that suitable translations do not generally exist.

Example 5.2.3. Consider $\mathfrak{A} = \{\mathfrak{a}, \mathfrak{b}\}$. As said before, the Coalgebraic Slicing Theorem shows that we cannot deal with e.g. formulas $\Diamond_{\mathfrak{a}} \Diamond_{\mathfrak{b}} p_{\mathfrak{a}}$. Interestingly, there is an even simpler and more essential part of $\mathbb{L}\mathsf{og}_{\mathrm{MA}}$ that $\mathbb{L}\mathsf{og}_{\mathfrak{A}}$ cannot deal with. Consider the formula $p_{\mathfrak{a}} \vee p_{\mathfrak{b}}$. Intuition obtained from the other translations would suggest that a suitable translation could be something like $(\ulcorner \mathfrak{a} \urcorner \to p) \vee (\ulcorner \mathfrak{b} \urcorner \to p)$. But by the semantics of $\mathbb{L}\mathsf{og}_{\mathfrak{A}}$, this translation is always trivially valid with truth value $\mathfrak{A}$.

There is in fact no suitable translation $\mathsf{tr}(p_{\mathfrak{a}} \vee p_{\mathfrak{b}})$. Consider three $\mathscr{T}_{\mathfrak{A}}$-models $\mathbb{S} = \langle \{s\}, \sigma, \mathsf{col} \rangle$, $\mathbb{S}' = \langle \{s\}, \sigma, \mathsf{col}' \rangle$ and $\mathbb{S}'' = \langle \{s\}, \sigma, \mathsf{col}'' \rangle$, where σ is irrelevant and the colourings are defined as $\mathsf{col}(s)(p) = \varnothing$, $\mathsf{col}'(s)(p) = \{\mathfrak{a}\}$, and $\mathsf{col}''(s)(p) = \{\mathfrak{b}\}$. Then clearly, $\mathbb{S}, s \leftrightarrows_{\mathfrak{b}} \mathbb{S}', s$ and $\mathbb{S}, s \leftrightarrows_{\mathfrak{a}} \mathbb{S}'', s$ (where we are using bisimilarity as defined in Definition 3.1.9), and so by adequacy (Theorem 3.1.12) it follows that $\mathbb{S}, s \equiv_{\mathfrak{b}} \mathbb{S}', s$ and $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}'', s$.

Obviously, $s \in [\![p_{\mathfrak{a}} \vee p_{\mathfrak{b}}]\!]^{\mathbb{L}\mathsf{og}_{\mathrm{MA}}}_{\mathbb{S}'}$ and $s \in [\![p_{\mathfrak{a}} \vee p_{\mathfrak{b}}]\!]^{\mathbb{L}\mathsf{og}_{\mathrm{MA}}}_{\mathbb{S}''}$. Supposing the semantics-preserving formula $\varphi = \mathsf{tr}(p_{\mathfrak{a}} \vee p_{\mathfrak{b}})$ exists, we would then have that $[\![\varphi]\!]^{\mathbb{L}\mathsf{og}_{\mathfrak{A}}}_{\mathbb{S}'}(s) = \mathfrak{A}$ and $[\![\varphi]\!]^{\mathbb{L}\mathsf{og}_{\mathfrak{A}}}_{\mathbb{S}''}(s) = \mathfrak{A}$. By $\mathbb{S}, s \equiv_{\mathfrak{b}} \mathbb{S}', s$ and $\mathbb{S}, s \equiv_{\mathfrak{a}} \mathbb{S}'', s$, it then follows that $[\![\varphi]\!]^{\mathbb{L}\mathsf{og}_{\mathfrak{A}}}_{\mathbb{S}}(s) = \mathfrak{A}$ as well. But $s \notin [\![p_{\mathfrak{a}} \vee p_{\mathfrak{b}}]\!]^{\mathbb{L}\mathsf{og}_{\mathrm{MA}}}_{\mathbb{S}}$, contradicting the assumption that $\mathsf{tr}(p_{\mathfrak{a}} \vee p_{\mathfrak{b}})$ preserves semantics. $\lhd$

As we will now show, enriching $\mathbb{L}\mathsf{og}_{\mathfrak{A}}$ with role switches allows us to define tr. Furthermore, we can even restrict to role switches of the form $\overline{[\mathfrak{a}, \mathfrak{b}]}$ (where $[\mathfrak{a}, \mathfrak{b}]$ is the role distribution only switching $\mathfrak{a}$ and $\mathfrak{b}$), as opposed to general role switches $\overline{\mathsf{role}}$ for arbitrary $\mathsf{role} \in \mathsf{RD}_{\mathfrak{A}}$.

Definition 5.2.4. The *language of* $\mathbb{L}\mathsf{og}_{\mathfrak{A}}$ *with role switches* is the inductively defined set

$$\mathsf{Lang}_{\mathbb{L}\mathsf{og}_{\mathfrak{A}},\mathrm{RS}} \ni \varphi ::= p \mid \ulcorner \mathfrak{B} \urcorner \mid (\varphi \vee \varphi) \mid (\neg\varphi) \mid (\heartsuit(\underbrace{\varphi, \ldots, \varphi}_{\mathsf{ar}\heartsuit \text{ times}})) \mid (\overline{[\mathfrak{a}, \mathfrak{b}]}\varphi),$$

where $p \in \mathsf{Prop}$, $\mathfrak{B} \subseteq \mathfrak{A}$, $\heartsuit \in \mathsf{Sym}$, and $\mathfrak{a}, \mathfrak{b} \in \mathfrak{A}$.

Given a $\mathscr{T}_{\mathfrak{A}}$-model $\mathbb{S} = \langle S, \sigma, \mathsf{col} \rangle$, the *semantics of* $\mathbb{L}\mathsf{og}_{\mathfrak{A}}$ *with role switches* is given by the function $[\![-]\!]^{\mathbb{L}\mathsf{og}_{\mathfrak{A}},\mathrm{RS}}_{\mathbb{S}} : \mathsf{Lang}_{\mathbb{L}\mathsf{og}_{\mathfrak{A}},\mathrm{RS}} \to (\mathscr{P}\mathfrak{A})^S$, inductively defined

as

$$\begin{aligned}
[\![p]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) &:= \mathsf{col}(s)(p),\\
[\![\ulcorner \mathfrak{B} \urcorner]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) &:= \mathfrak{B},\\
[\![\varphi \vee \psi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) &:= [\![\varphi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) \cup [\![\psi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s),\\
[\![\neg\varphi]\!]_{S}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) &:= \mathfrak{A} \smallsetminus [\![\psi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s),\\
[\![\heartsuit(\varphi_1,\ldots,\varphi_{\mathsf{ar}\heartsuit})]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}} &:= (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma)\left(\mathsf{lift}_S^{\heartsuit}([\![\varphi_1]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}},\ldots,[\![\varphi_{\heartsuit}]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}})\right),\\
[\![\overline{[\mathfrak{a},\mathfrak{b}]}]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}} &:= \mathscr{P}[\mathfrak{a},\mathfrak{b}] \circ [\![\varphi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}
\end{aligned}$$

for $\varphi \in \mathsf{Lang}_{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}$. ◁

The translation we will now define is based on the informal idea that we can consider the semantics of formulas in $\mathbb{L}\mathrm{og}_{\mathrm{MP}}$ for 'one agent at a time', with role switches then allowing us to capture the semantics for the entire set of agents.

Definition 5.2.5. The $\mathfrak{a}$-*translation* $\mathsf{tr}_{\mathfrak{a}}$ for $\mathfrak{a} \in \mathfrak{A}$ from formulas of $\mathbb{L}\mathrm{og}_{\mathrm{MP}}$ to formulas of $\mathbb{L}\mathrm{og}_{\mathfrak{A}}$ with role switches is defined by simultaneous induction over all $\mathfrak{a} \in \mathfrak{A}$ as

$$\begin{aligned}
\mathsf{tr}_{\mathfrak{a}}(p_{\mathfrak{a}}) &:= p,\\
\mathsf{tr}_{\mathfrak{a}}(p_{\mathfrak{b}}) &:= \overline{[\mathfrak{a},\mathfrak{b}]}p, && (\mathfrak{a} \neq \mathfrak{b})\\
\mathsf{tr}_{\mathfrak{a}}(\varphi \vee \psi) &:= \mathsf{tr}_{\mathfrak{a}}(\varphi) \vee \mathsf{tr}_{\mathfrak{a}}(\psi),\\
\mathsf{tr}_{\mathfrak{a}}(\neg\varphi) &:= \neg\mathsf{tr}_{\mathfrak{a}}(\varphi),\\
\mathsf{tr}_{\mathfrak{a}}(\heartsuit_{\mathfrak{a}}(\varphi_1,\ldots,\varphi_{\mathsf{ar}\heartsuit})) &:= \heartsuit(\mathsf{tr}_{\mathfrak{a}}(\varphi_1),\ldots,\mathsf{tr}_{\mathfrak{a}}(\varphi_{\mathsf{ar}\heartsuit})), \text{ and}\\
\mathsf{tr}_{\mathfrak{a}}(\heartsuit_{\mathfrak{b}}(\varphi_1,\ldots,\varphi_{\mathsf{ar}\heartsuit})) &:= \overline{[\mathfrak{a},\mathfrak{b}]}\heartsuit(\mathsf{tr}_{\mathfrak{b}}(\varphi_1),\ldots,\mathsf{tr}_{\mathfrak{b}}(\varphi_{\mathsf{ar}\heartsuit})). && (\mathfrak{a} \neq \mathfrak{b})
\end{aligned}$$

Fixing an $\mathfrak{a} \in \mathfrak{A}$, the translation tr is then defined as

$$\mathsf{tr}(\varphi) := \ulcorner \mathfrak{a} \urcorner \rightarrow \mathsf{tr}_{\mathfrak{a}}(\varphi)$$

for φ in the language of $\mathbb{L}\mathrm{og}_{\mathrm{MP}}$. ◁

Theorem 5.2.6. *Let* $\mathfrak{S} = \langle S, \sigma, \mathsf{col}\rangle$ *be a* $\mathscr{T}_{\mathfrak{A}}$*-model over* $\mathsf{Prop}_{\mathrm{MP}}$*. Then it holds that*

$$s \in [\![\varphi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}} \textit{ iff } [\![\mathsf{tr}(\varphi)]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) = \mathfrak{A}$$

for all φ *in the language of* $\mathbb{L}\mathrm{og}_{\mathrm{MP}}$*.*

Proof. We first show by induction on φ that

$$s \in [\![\varphi]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}} \text{ iff } \mathfrak{b} \in [\![\mathsf{tr}_{\mathfrak{b}}(\varphi)]\!]_{\mathfrak{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) \tag{5.8}$$

for all $\mathfrak{b} \in \mathfrak{A}$. First consider a variable $p_{\mathfrak{b}}$. Then we have that

$$\begin{aligned} s \in [\![p_{\mathfrak{b}}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}} &\text{ iff } \mathfrak{b} \in \mathsf{col}(s)(p) \\ &\text{iff } \mathfrak{b} \in [\![p]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s). \end{aligned}$$

Now consider a variable $p_{\mathfrak{c}}$ with $\mathfrak{b} \neq \mathfrak{c}$. Then we have that

$$\begin{aligned} s \in [\![p_{\mathfrak{c}}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}} &\text{ iff } \mathfrak{c} \in \mathsf{col}(s)(p) \\ &\text{iff } \mathfrak{c} \in [\![p]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) \\ &\text{iff } \mathfrak{b} \in [\![\overline{[\mathfrak{b},\mathfrak{c}]}p]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s). \end{aligned}$$

The inductive steps for disjunction and negation only require routine Boolean reasoning, so we only consider modalities. Starting with the formula $\heartsuit_{\mathfrak{b}}(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})$, we find that $s \in [\![\heartsuit_{\mathfrak{b}}(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}}$

$$\begin{aligned} &\text{iff } s \in (\mathscr{P}\!\mathit{red}\,\sigma)(\mathsf{lift}_S^{\heartsuit_{\mathfrak{b}}}([\![\varphi_1]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}}, \ldots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}})) \\ &\text{iff } \sigma_{\mathfrak{b}}(s) \in \mathsf{lift}_S^{\heartsuit}([\![\varphi_1]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}}, \ldots, [\![\varphi_{\mathsf{ar}\heartsuit}]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}}) \\ &\text{iff } \sigma_{\mathfrak{b}}(s) \in \mathsf{lift}_S^{\heartsuit}(\overbrace{[\![\mathsf{tr}_{\mathfrak{b}}(\varphi_1)]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}}(\mathfrak{b}), \ldots, \overbrace{[\![\mathsf{tr}_{\mathfrak{b}}(\varphi_{\mathsf{ar}\heartsuit})]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}}(\mathfrak{b})) \qquad \text{(IH)} \\ &\text{iff } \mathfrak{b} \in (\mathscr{P}\!\mathit{red}_{\mathfrak{A}}\sigma)((\mathsf{lift}_{\mathfrak{A}}^{\heartsuit})_S([\![\mathsf{tr}_{\mathfrak{b}}(\varphi_1)]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}, \ldots, [\![\mathsf{tr}_{\mathfrak{b}}(\varphi_{\mathsf{ar}\heartsuit})]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}))(s) \\ &\text{iff } \mathfrak{b} \in [\![\heartsuit(\mathsf{tr}_{\mathfrak{b}}(\varphi_1), \ldots, \mathsf{tr}_{\mathfrak{b}}(\varphi_{\mathsf{ar}\heartsuit}))]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) \\ &\text{iff } \mathfrak{b} \in [\![\mathsf{tr}_{\mathfrak{b}}(\heartsuit_{\mathfrak{b}}(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit}))]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s). \end{aligned}$$

Finally, taking the formula $\heartsuit_{\mathfrak{c}}(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})$ for $\mathfrak{c} \neq \mathfrak{b}$, we find that

$$s \in [\![\heartsuit_{\mathfrak{c}}(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit})]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathrm{MA}}} \text{ iff } \mathfrak{b} \in [\![\mathsf{tr}_{\mathfrak{b}}(\heartsuit_{\mathfrak{c}}(\varphi_1, \ldots, \varphi_{\mathsf{ar}\heartsuit}))]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s)$$

by the semantics of $\overline{[\mathfrak{b},\mathfrak{c}]}$ and the previous inductive step.

Having shown Equation (5.8), the theorem then follows trivially from the simple observation that $[\![\mathsf{tr}(\varphi)]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s) = \mathfrak{A}$ holds if and only if $\mathfrak{a} \in [\![\mathsf{tr}_{\mathfrak{a}}(\varphi)]\!]_{\mathbb{S}}^{\mathbb{L}\mathrm{og}_{\mathfrak{A}},\mathrm{RS}}(s)$. □

Chapter 6

Conclusion

In this thesis, we have studied coalgebraic generalizations of two logics in which truth values are identified as sets of agents. We have generalized the multiagent-valued logic of Fitting (2009) by applying the theory of coalgebraic modal logic to a new base category of sets and agent-indexed functions, and give proofs of the coincidence of bisimilarity and behavioural equivalence, as well as of adequacy and expressivity. We have generalized the multiplayer logic of Olde Loohuis and Venema (2010) by considering a new, nondeterministic game structure, on which we defined multiplayer evaluation games by transforming coalgebras into monotone neighbourhood frames. Finally, we have proven, under some assumptions on the coalgebra type, that the coalgebraic multiagent-valued logic is equiexpressive to the fragment of coalgebraic multiplayer logic without role switches, and that adding role switches to coalgebraic multiagent-valued logic allows the resulting logic to naturally express situations with multiple agents.

Our work lends itself to several interesting avenues for future work.

- Coalgebraic modal logic offers provides general approaches to soundness and completeness proofs on the basis of so-called *one-step derivation systems* (see e.g. Kupke and Pattinson (2011)). Similar to how we lifted two-valued coalgebraic modal logics to multiagent-valued coalgebraic modal logics, one could ask whether sound and complete one-step derivation systems for a two-valued coalgebraic modal logic also lift to sound and complete derivation systems for multiagent-valued coalgebraic modal logics.

  Furthermore, Fitting (2009) provides tableaux for his logic. Tableaux have also been considered in the literature for (two-valued) coalgebraic logics (Cîrstea, Kupke, and Pattinson 2009; Goré, Kupke, and Pattinson 2010; Goré, Kupke, Pattinson, and Schröder 2010) – it is of interest whether we can similarly find and study tableaux for multiagent-valued coalgebraic modal logics.

- The way we considered coalgebraic modal logics for a functor $\mathcal{T}$ as requiring collections of predicate $\mathcal{T}$-liftings, is not fully parametric in the coalgebra type $\mathcal{T}$. But the original coalgebraic modal logic of Moss (1999) — who founded the field — was defined in a way that *is* fully parametric in $\mathcal{T}$, albeit syntactically a bit more unintuitive, and requiring $\mathcal{T}$ to preserve weak pullbacks.[1] The semantics of Moss' logic are defined using *relation lifting*, mapping relations $R \subseteq X \times Y$ to lifted relations $\overline{\mathcal{T}}R \subseteq \mathcal{T}X \times \mathcal{T}Y$ in a certain way. For a multiagent-valued version of Moss' logic, it is of interest to see what the corresponding relation liftings look like.

- In Chapter 1, we mentioned that there has been other work relating coalgebraic modal logic and games. In particular, Venema (2006) considers game-theoretic semantics for a coalgebraic generalization of the modal μ-calculus. The games considered by Venema (2006) are parity games, in which matches are generally *infinite*, unlike in our nondeterministic games with finite matches. It is of interest to see whether suitable (nondeterministic) multiplayer analogues of these parity games can be given, and whether they provide game-theoretic semantics for some multiagent-valued version of the coalgebraic μ-calculus.

- The multiagent-valued logic of Fitting (2009) logic was in fact historically preceded by the multiagent-valued logic of Fitting (1992). That logic is built using a partial ordering $\leqslant$ over agents, instead of merely a set of agents, with $\mathfrak{a} \leqslant \mathfrak{b}$ being interpreted as stating that $\mathfrak{a}$ *dominates* $\mathfrak{b}$, meaning that all basic facts considered to be true by $\mathfrak{a}$ are also considered to be true by $\mathfrak{b}$. Using this partial ordering, Fitting (1992) then constructs an *intuitionistic* multiagent-valued modal logic in which the space of truth values corresponds to the Heyting algebra of upwards closed sets of agents (with respect to the dominance ordering), as opposed to the powerset Boolean algebra. This gives a richer logic, as the property of dominance between agents becomes built into the truth values.

  It is of interest to see whether this intuitionistic multiagent-valued logic can also be coalgebraically generalized. This would most likely require a logical connection with the category **HA** of Heyting algebras, and a suitable multiagent version of the category **IntKF** of intuitionistic Kripke frames, analogously to how **ASet** is a multiagent version of **Set**.

  If a coalgebraic generalization of the logic of Fitting (1992) is achieved, it is then natural to ask whether there is a corresponding

[1] Though not relevant to our recommendation for future work, there is work (see e.g. Marti and Venema (2015)) on Moss-style coalgebraic modal logic without the requirement of weak pullback preservation.

generalization of multiplayer logic that also takes the dominance ordering into account. This could potentially arise through interesting implementations of the multiplayer games, with e.g. turn functions being monotone with respect to the dominance ordering.

- Finally, we close with the following suggestion. There is work by Baltag (2003), Carreiro, Gorín, and Schröder (2013), and Cîrstea and Sadrzadeh (2007), in which coalgebraic dynamic epistemic systems and logics are considered. Since dynamic epistemic logics are generally studied in a multiagent setting, it is natural to study how multiagent-valued logic relates to the aforementioned coalgebraic dynamic epistemic logics.

Bibliography

Peter Aczel and Nax Mendler (1989). "A Final Coalgebra Theorem." In: *Category Theory and Computer Science*. Ed. by David H. Pitt, David E. Rydeheard, Peter Dybjer, Andrew M. Pitts, and Axel Poigné. Berlin, Heidelberg: Springer, pp. 357–365. ISBN: 978-3-540-46740-3. DOI: 10.1007/BFb0018361.

Jiří Adámek, Stefan Milius, Lurdes Sousa, and Thomas Wißmann (2019). "On Finitary Functors." In: *Theory and Applications of Categories* 34.35, pp. 1134–1164.

Jiří Adámek and Vera Trnková (1990). "F-Automata." In: *Automata and Algebras in Categories*. 1st ed. USA: Kluwer Academic Publishers. Chap. 3. ISBN: 0792300106.

Octavian Babus and Alexander Kurz (2016). "On the Logic of Generalised Metric Spaces." In: *Proceedings of the 13th International Workshop on Coalgebraic Methods in Computer Science*. Ed. by Ichiro Hasuo. Berlin, Heidelberg: Springer, pp. 136–155. ISBN: 978-3-319-40369-4. DOI: 10.1007/978-3-319-40370-0_9.

Alexandru Baltag (2000). "A Logic for Coalgebraic Simulation." In: *Electronic Notes in Theoretical Computer Science* 33. CMCS'2000, Coalgebraic Methods in Computer Science, pp. 42–60. ISSN: 1571-0661. DOI: 10.1016/S1571-0661(05)80343-3.

— (2003). "A Coalgebraic Semantics for Epistemic Programs." In: *Electronic Notes in Theoretical Computer Science* 82.1. CMCS'03, Coalgebraic Methods in Computer Science (Satellite Event for ETAPS 2003), pp. 17–38. ISSN: 1571-0661. DOI: 10.1016/S1571-0661(04)80630-3.

Can Başkent (2015). "Game Theoretical Semantics for Paraconsistent Logics." In: *Logic, Rationality, and Interaction. Lecture Notes in Computer Science*. Ed. by Wiebe van der Hoek, Wesley H. Holliday, and Wen-fang Wang. Berlin, Heidelberg: Springer, pp. 14–26. DOI: 10.1007/978-3-662-48561-3_2.

Johan van Benthem, Nick Bezhanishvili, and Sebastian Enqvist (2019). "A New Game Equivalence, its Logic and Algebra." In: *Journal of Philosophical Logic* 48, pp. 649–684. DOI: 0.1007/s10992-018-9489-7.

Rudolf Berghammer and Hans Zierer (1986). "Relational Algebraic Semantics of Deterministic and Nondeterministic Programs." In: *Theoretical*

Computer Science 43, pp. 123–147. ISSN: 0304-3975. DOI: 10.1016/0304-3975(86)90172-6.

Nick Bezhanishvili, Marcello Bonsangue, Helle Hvid Hansen, Dexter Kozen, Clemens Kupke, Prakash Panangaden, and Alexandra Silva (2020). *Minimisation in Logical Form*. arXiv: 2005.11551 (cs.FL).

Marta Bílková and Matěj Dostál (2013). "Many-Valued Relation Lifting and Moss' Coalgebraic Logic." In: *Proceedings of the 5th International Conference on Algebra and Coalgebra in Computer Science*. Ed. by Reiko Heckel and Stefan Milius. Berlin, Heidelberg: Springer, pp. 66–79. ISBN: 978-3-642-40205-0. DOI: 10.1007/978-3-642-40206-7_7.

— (2016). "Expressivity of Many-Valued Modal Logics, Coalgebraically." In: *Proceedings of the 23rd International Workshop on Logic, Language, Information, and Computation*. Ed. by Jouko Väänänen, Åsa Hirvonen, and Ruy de Queiroz. Berlin, Heidelberg: Springer, pp. 109–124. ISBN: 978-3-662-52920-1. DOI: 10.1007/978-3-662-52921-8_8.

Marta Bílková, Alexander Kurz, Daniela Petrişan, and Jiří Velebil (2013). "Relation Lifting, With an Application to the Many-Valued Cover Modality." In: *Logical Methods in Computer Science* 9 (4). DOI: 10.2168/LMCS-9(4:8)2013.

Patrick Blackburn, Maarten de Rijke, and Yde Venema (2002). *Modal Logic*. 1st ed. Cambridge University Press. 578 pp. ISBN: 9780521527149. DOI: 10.1017/CBO9781107050884.

Marcello Bonsangue and Alexander Kurz (2005). "Duality for Logics of Transition Systems." In: *Proceedings of the International Conference on Foundations of Software Science and Computational Structures*. Ed. by Vladimiro Sassone. Berlin, Heidelberg: Springer, pp. 455–469. ISBN: 978-3-540-25388-4. DOI: 10.1007/978-3-540-31982-5_29.

— (2006). "Presenting Functors by Operations and Equations." In: *Proceedings of the 9th International Conference on Foundations of Software Science and Computation Structures*. Ed. by Luca Aceto and Anna Ingólfsdóttir. Berlin, Heidelberg: Springer, pp. 172–186. ISBN: 978-3-540-33045-5. DOI: 10.1007/11690634_12.

Facundo Carreiro, Daniel Gorín, and Lutz Schröder (2013). "Coalgebraic Announcement Logics." In: *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming*. Ed. by Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg. Berlin, Heidelberg: Springer, pp. 101–112. ISBN: 978-3-642-39211-5. DOI: 10.1007/978-3-642-39212-2_12.

Brian F. Chellas (1980). *Modal Logic: An Introduction*. Cambridge University Press. ISBN: 9780511621192. DOI: 10.1017/CBO9780511621192.

Corina Cîrstea, Clemens Kupke, and Dirk Pattinson (2009). "EXPTIME Tableaux for the Coalgebraic μ-Calculus." In: *Proceedings of the 23rd International Workshop on Computer Science Logic*. Ed. by Erich Grädel and

Reinhard Kahle. Berlin, Heidelberg: Springer, pp. 179–193. ISBN: 978-3-642-04026-9. DOI: 10.1007/978-3-642-04027-6_15.

Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema (2009). "Modal Logics are Coalgebraic." In: *The Computer Journal* 54.1, pp. 31–41. ISSN: 0010-4620. DOI: 10.1093/comjnl/bxp004.

Corina Cîrstea and Mehrnoosh Sadrzadeh (2007). "Coalgebraic Epistemic Update Without Change of Model." In: *Proceedings of the 2nd International Conference on Algebra and Coalgebra in Computer Science*. Ed. by Till Mossakowski, Ugo Montanari, and Magne Haveraaen. Berlin, Heidelberg: Springer, pp. 158–172. ISBN: 978-3-540-73857-2. DOI: 10.1007/978-3-540-73859-6_11.

— (2008). "Modular Games for Coalgebraic Fixed Point Logics." In: *Electronic Notes in Theoretical Computer Science* 203.5. Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science (CMCS 2008), pp. 71–92. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2008.05.020.

Brian A. Davey and Hilary A. Priestley (2002). *Introduction to Lattices and Order*. 2nd ed. Cambridge University Press. DOI: 10.1017/CBO9780511809088.

Melvin Fitting (1992). "Many-Valued Modal Logics II." In: *Fundamenta Informaticae* 17, pp. 55–73.

— (2003). "Bisimulations and Boolean Vectors." In: *Advances in Modal Logic* 4. Ed. by Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev, pp. 97–125.

— (2009). "How True It Is = Who Says It's True." In: *Studia Logica: An International Journal for Symbolic Logic* 91.3, pp. 335–366. ISSN: 00393215, 15728730. DOI: 10.1007/s11225-009-9178-1.

Rajeev Goré, Clemens Kupke, and Dirk Pattinson (2010). "Optimal Tableau Algorithms for Coalgebraic Logics." In: *Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by Javier Esparza and Rupak Majumdar. Berlin, Heidelberg: Springer, pp. 114–128. ISBN: 978-3-642-12001-5. DOI: 10.1007/978-3-642-12002-2_9.

Rajeev Goré, Clemens Kupke, Dirk Pattinson, and Lutz Schröder (2010). "Global Caching for Coalgebraic Description Logics." In: *Proceedings of the 5th International Joint Conference on Automated Reasoning*. Ed. by Jürgen Giesl and Reiner Hähnle. Berlin, Heidelberg: Springer, pp. 46–60. ISBN: 978-3-642-14202-4. DOI: 10.1007/978-3-642-14203-1_5.

Helle Hvid Hansen and Bartek Klin (2011). "Pointwise Extensions of GSOS-Defined Operations." In: *Mathematical Structures in Computer Science* 21.2, pp. 321–361. ISSN: 0960-1295. DOI: 10.1017/S096012951000054X.

Helle Hvid Hansen, Clemens Kupke, and Eric Pacuit (2009). "Neighbourhood Structures: Bisimilarity and Basic Model Theory." In: *Logical Methods in Computer Science* 5.2, pp. 1–38. ISSN: 1860-5974. DOI: 10.2168/LMCS-5(2:2)2009.

Russell S. Harmer (1999). "Games and Full Abstraction for Nondeterministic Languages." PhD thesis. University of London.

Jaakko Hintikka (1983). *The Game of Language: Studies in Game-Theoretical Semantics and its Applications*. 1st ed. Springer Netherlands. 356 pp. ISBN: 978-90-277-1950-8. DOI: 10.1007/978-94-010-9847-2.

Bart Jacobs (2016). *Introduction to Coalgebra: Towards Mathematics of States and Observation*. 1st ed. Cambridge University Press. 494 pp. ISBN: 97811071-77895. DOI: 10.1017/CBO9781316823187.

Bartek Klin (2007). "Coalgebraic Modal Logic Beyond Sets." In: *Electronic Notes in Theoretical Computer Science* 173. Proceedings of the 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII), pp. 177–201. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2007.02.034.

Barbara König, Christina Mika-Michalski, and Lutz Schröder (2020). "Explaining Non-bisimilarity in a Coalgebraic Approach: Games and Distinguishing Formulas." In: *Proceedings of the 15th International Workshop on Coalgebraic Methods in Computer Science*. Ed. by Daniela Petrişan and Jurriaan Rot. Berlin, Heidelberg: Springer, pp. 133–154. ISBN: 978-3-030-57200-6. DOI: 10.1007/978-3-030-57201-3_8.

Clemens Kupke, Alexander Kurz, and Dirk Pattinson (2004). "Algebraic Semantics for Coalgebraic Logics." In: *Electronic Notes in Theoretical Computer Science* 106. Proceedings of the Workshop on Coalgebraic Methods in Computer Science (CMCS), pp. 219–241. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2004.02.037.

Clemens Kupke and Dirk Pattinson (2011). "Coalgebraic Semantics of Modal Logics: An Overview." In: *Theoretical Computer Science* 412.38. CMCS Tenth Anniversary Meeting, pp. 5070–5094. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2011.04.023.

Alexander Kurz (2017). "Boolean-Valued Coalgebraic Logic." Unpublished.

Alexander Kurz and Raul Andres Leal (2012). "Modalities in the Stone Age: A Comparison of Coalgebraic Logics." In: *Theoretical Computer Science* 430. Mathematical Foundations of Programming Semantics (MFPS XXV), pp. 88–116. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2012.03.027.

Raul Andres Leal (2008). "Predicate Liftings Versus Nabla Modalities." In: *Electronic Notes in Theoretical Computer Science* 203.5. Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science (CMCS 2008), pp. 195–220. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2008.05.026.

Johannes Marti and Yde Venema (2015). "Lax Extensions of Coalgebra Functors and Their Logic." In: *Journal of Computer and System Sciences* 81.5. 11th International Workshop on Coalgebraic Methods in Computer Science, CMCS 2012 (Selected Papers), pp. 880–900. ISSN: 0022-0000. DOI: 10.1016/j.jcss.2014.12.006.

Lawrence S. Moss (1999). "Coalgebraic Logic." In: *Annals of Pure and Applied Logic* 96.1, pp. 277–317. ISSN: 0168-0072. DOI: 10.1016/S0168-0072(98)00042-6.

Loes Olde Loohuis and Yde Venema (2010). "Logics and Algebras for Multiple Players." In: *The Review of Symbolic Logic* 3.3, pp. 485–519. DOI: 10.1017/S1755020310000079.

Dirk Pattinson (2001). "Semantical Principles in the Modal Logic of Coalgebras." In: *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*. Berlin, Heidelberg: Springer-Verlag, pp. 514–526. ISBN: 3540416951. DOI: 10.1007/3-540-44693-1_45.

Dusko Pavlovic, Michael Mislove, and James Worrell (2006). "Testing Semantics: Connecting Processes and Process Logics." In: *Proceedings of the International Conference on Algebraic Methodology and Software Technology*. Ed. by Michael Johnson and Varmo Vene. Berlin, Heidelberg: Springer, pp. 308–322. ISBN: 978-3-540-35633-2. DOI: 10.1007/11784180_24.

Graham Priest (1979). "The Logic of Paradox." In: *Journal of Philosophical Logic* 8.1, pp. 219–241. ISSN: 00223611, 15730433.

Jan J.M.M. Rutten (2000). "Universal Coalgebra: A Theory of Systems." In: *Theoretical Computer Science* 249.1. Modern Algebra, pp. 3–80. ISSN: 0304-3975. DOI: 10.1016/S0304-3975(00)00056-6.

— (2019). *The Method of Coalgebra: Exercises in Coinduction*. Amsterdam, The Netherlands: CWI. ISBN: 978-90-6196-568-8.

Lutz Schröder (2008). "Expressivity of Coalgebraic Modal Logic: The Limits and Beyond." In: *Theoretical Computer Science* 390.2. Foundations of Software Science and Computational Structures, pp. 230–247. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2007.09.023.

Harald Søndergaard and Peter Sestoft (1992). "Non-Determinism in Functional Languages." In: *The Computer Journal* 35.5, pp. 514–523. ISSN: 0010-4620. DOI: 10.1093/comjnl/35.5.514.

Sam Staton (2009). "Relating Coalgebraic Notions of Bisimulation." In: *Algebra and Coalgebra in Computer Science*. Ed. by Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki. Berlin, Heidelberg: Springer, pp. 191–205. ISBN: 978-3-642-03741-2. DOI: 10.1007/978-3-642-03741-2_14.

Alfred Tarski (1935). "Zur Grundlegung der Boole'schen Algebra. I." German. In: *Fundamenta Mathematicae* 24.1, pp. 177–198.

Tero Tulenheimo and Yde Venema (2008). "Propositional Logics for Three." In: *Dialogues, Logics and Other Strange Things. Essays in Honour of Shahid Rahman*. London, UK: College Publications, pp. 399–429. DOI: 11245/1.299852.

Yde Venema (2006). "Automata and Fixed Point Logic: A Coalgebraic Perspective." In: *Information and Computation* 204.4. Seventh Workshop on Coalgebraic Methods in Computer Science 2004, pp. 637–678. ISSN: 0890-5401. DOI: 10.1016/j.ic.2005.06.003.

James Worrell (2005). "On the Final Sequence of a Finitary Set Functor." In: *Theoretical Computer Science* 338.1, pp. 184–199. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2004.12.009.