Coalgebraic semantics of modal logics

Wang Yunsong

Peking University, SMS

May 28th, 2019

Overview

Category

A *category* consists of the following data:

- *Objects*: $A, B, C, \ldots$
- *Arrow*(*morphism*): $f, g, h, \ldots$
- For each morphism f, there are given objects:
$$\mathrm{dom(f)},\ \mathrm{cod(f)}$$
called the *domain* and *codomain* of f. We write
$$f : A \to B$$
to indicate that A=dom(f) and B=cod(f).

Category

- Given arrows $f : A \to B$ and $g : B \to C$, that is, with
$$\text{cod(f)=dom(g)}$$
there is given an arrow
$$g \circ f : A \to C$$
called the *composite* of f and g.
- For each object A, there is given an arrow
$$1_A : A \to A$$
called the *identity* of A.

Category

- Given arrows $f : A \to B$ and $g : B \to C$, that is, with
$$\mathrm{cod(f)=dom(g)}$$
there is given an arrow
$$g \circ f : A \to C$$
called the *composite* of f and g.
- For each object A, there is given an arrow
$$1_A : A \to A$$
called the *identity* of A.

These data are required to satisfy the following laws:

- Associativity:
$$h \circ (g \circ f) = (h \circ g) \circ f$$
for all $f : A \to B$, $g : B \to C$, $h : C \to D$.
- Unit:
$$f \circ 1_A = f = 1_B \circ f$$
for all $f : A \to B$.

Examples of category

- **Sets**: the category of sets and functions.
  And we can also add some restrictions to sets and functions.
  e.g, finite sets and functions between them, sets and injective functions.

Examples of category

- **Sets**: the category of sets and functions.
  And we can also add some restrictions to sets and functions.
  e.g, finite sets and functions between them, sets and injective functions.
- groups and group homomorphisms
- vector spaces and linear mappings
- topological spaces and continuous mappings

Examples of category

- **Sets**: the category of sets and functions.
  And we can also add some restrictions to sets and functions.
  e.g, finite sets and functions between them, sets and injective functions.
- groups and group homomorphisms
- vector spaces and linear mappings
- topological spaces and continuous mappings
- category of proofs
- category of data types and computable functions

Further Definition

- A functor

$$F : \mathbf{C} \to \mathbf{D}$$

between categories **C** and **D** is a mapping of objects to objects and arrows to arrows, in such a way that:
(a) F($f : A \to B$)=F(f):$F(A) \to F(B)$,
(b) F(1_A)=$1_{F(A)}$,
(c) F($g \circ f$)=$F(g) \circ F(f)$.

So we have another example of a category, namely **Cat**, the category of all categories and functors.

Further Definition

- Opposite category $\mathbf{C}^{op}$
  The opposite category $\mathbf{C}^{op}$ of a category $\mathbf{C}$ has the same objects as $\mathbf{C}$, and an arrow $f : C \to D$ in $\mathbf{C}^{op}$ is an arrow $f : D \to C$ in $\mathbf{C}$. That is, $\mathbf{C}^{op}$ is just $\mathbf{C}$ with all of the arrows formally turned around.

Further Definition

- Opposite category $\mathbf{C}^{op}$
  The opposite category $\mathbf{C}^{op}$ of a category $\mathbf{C}$ has the same objects as $\mathbf{C}$, and an arrow $f : C \to D$ in $\mathbf{C}^{op}$ is an arrow $f : D \to C$ in $\mathbf{C}$. That is, $\mathbf{C}^{op}$ is just $\mathbf{C}$ with all of the arrows formally turned around.

- Contravariant functor
  A functor of the form $F : \mathbf{C}^{op} \to \mathbf{D}$ is called a contravariant functor on $\mathbf{C}$. Explicitly, such a functor takes $f : A \to B$ to $F(f) : F(B) \to F(A)$ and $F(g \circ f) = F(f) \circ F(g)$.

Further Definition

- Natural transformation
  For categories $\mathbf{C}$, $\mathbf{D}$ and functors
  $$F, G : \mathbf{C} \to \mathbf{D}$$
  a natural transformation $\vartheta : F \to G$ is a family of arrows in $\mathbf{D}$
  $$(\vartheta_C : FC \to GC)_{C \in \mathbf{C}_0}$$
  such that, for any $f : C \to C'$ in $\mathbf{C}$, one has $\vartheta_{C'} \circ F(f) = G(f) \circ \vartheta_C$.
  Given such a natural transformation $\vartheta : F \to G$, the D-arrow $\vartheta_C : FC \to GC$ is called the component of ϑ at C.

Coalgebra

- Let **C** be a category and T an endofunctor on **C**. A T-coalgebra is a pair (X, γ) where X is an object in **C** and $\gamma : X \to TX$ is a morphism in **C**.

Coalgebra

- Let **C** be a category and T an endofunctor on **C**. A T-coalgebra is a pair (X, γ) where X is an object in **C** and $\gamma : X \to TX$ is a morphism in **C**.

- A T-coalgebra morphism between two T-coalgebras (X, γ) and (X', γ') is a morphism $f : X \to X'$ in **C** satisfying $\gamma' \circ f = Tf \circ \gamma$.

Coalgebra

- Let **C** be a category and T an endofunctor on **C**. A T-coalgebra is a pair (X, γ) where X is an object in **C** and $\gamma : X \to TX$ is a morphism in **C**.

- A T-coalgebra morphism between two T-coalgebras (X, γ) and (X', γ') is a morphism $f : X \to X'$ in **C** satisfying $\gamma' \circ f = Tf \circ \gamma$.

- The collection of T-coalgebras and T-coalgebra morphisms forms a category, which we shall denote by **Coalg(T)**. The category **C** is called the base category of **Coalg(T)**.

Coalgebra

- Let **C** be a category and T an endofunctor on **C**. A T-coalgebra is a pair (X, γ) where X is an object in **C** and $\gamma : X \to TX$ is a morphism in **C**.
- A T-coalgebra morphism between two T-coalgebras (X, γ) and (X', γ') is a morphism $f : X \to X'$ in **C** satisfying $\gamma' \circ f = Tf \circ \gamma$.
- The collection of T-coalgebras and T-coalgebra morphisms forms a category, which we shall denote by **Coalg(T)**. The category **C** is called the base category of **Coalg(T)**.

For the most part, we restrict attention to coalgebras on sets and write Coalg(T) for the category of coalgebras induced by a set functor T.

Example 1

- Kripke frames
  Kripke frames correspond 1-1 with P-coalgebras where $\mathcal{P} : Set \to Set$ is the power set functor.

Example 1

- Kripke frames

  Kripke frames correspond 1-1 with P-coalgebras where $\mathcal{P} : Set \to Set$ is the power set functor.

  For a Kripke frame (X,R) define $\gamma_R : X \to PX : x \mapsto \{y|xRy\}$. Then (X, γ_R) is a P-coalgebra. Conversely, for a P-coalgebra (X, γ) define R_γ by $xR_\gamma y$ iff $y \in \gamma(x)$. Then (X, R_γ) is a Kripke frame. And this is a bijection between Kripke frames and P-coalgebras.

Example 1

- Kripke frames

  Kripke frames correspond 1-1 with P-coalgebras where $\mathcal{P} : Set \to Set$ is the power set functor.

  For a Kripke frame (X,R) define $\gamma_R : X \to PX : x \mapsto \{y | xRy\}$. Then (X, γ_R) is a P-coalgebra. Conversely, for a P-coalgebra (X, γ) define R_γ by $xR_\gamma y$ iff $y \in \gamma(x)$. Then (X, R_γ) is a Kripke frame. And this is a bijection between Kripke frames and P-coalgebras.

  Moreover, bounded morphisms between Kripke frames are precisely P-coalgebra morphisms. Thus, we have

  $$\mathbf{Krip} \cong \mathbf{Coalg(P)},$$

  where **Krip** is the category of Kripke frames and bounded morphisms.

Example $1'$

To capture labelled transition systems in the coalgebraic framework, we consider the functor $\mathcal{P}(\cdot)^{\mathcal{A}}$ where $\mathcal{A}$ is a set(of actions, or labels) and $\mathcal{P}(X)^{\mathcal{A}}$ is the set of all functions of type $\mathcal{A} \to \mathcal{P}(X)$: A labelled transition system is a pair (W, γ) where $\gamma : W \to \mathcal{P}(W)^{\mathcal{A}}$ is a function. This is again equivalent to the standard definition where a labelled transition system is understood as tuple (W,R) where W is the set of states and $R \subset W \times A \times W$ is a labelled transition relation.

Example 2

- Monotone neighbournood frames
  Let $D : Set \to Set$ be the functor given on objects by
  $$DX = \{W \subset PX | \text{if } a \in W \text{ and } a \subset b \text{ then } b \in W\},$$
  for X a set. For a morphism $f : X \to X'$ define
  $$Df : DX \to DX' : W \mapsto \{a' \in PX' | f^{-1}(a') \in W\}.$$
  Then the category of monotone frames and bounded morphisms is isomorphic to **Coalg(D)**.

Example 2′

Neighbourhood frames can be captured in the coalgebraic framework by means of the functor $\mathcal{N}X = 2^{2^X}$, technically the composition of the contravariant power set functor 2^- with itself.

Example $2'$

Neighbourhood frames can be captured in the coalgebraic framework by means of the functor $\mathcal{N}X = 2^{2^X}$, technically the composition of the contravariant power set functor 2^- with itself.

In other words, the action of $\mathcal{N}$ on maps is given by $\mathcal{N}(f) = (f^{-1})^{-1}$ where $g^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X)$ denotes the inverse image operation induced by a function $g : X \to Y$.

Example 2′

Neighbourhood frames can be captured in the coalgebraic framework by means of the functor $\mathcal{N}X = 2^{2^X}$, technically the composition of the contravariant power set functor 2^- with itself.

In other words, the action of $\mathcal{N}$ on maps is given by $\mathcal{N}(f) = (f^{-1})^{-1}$ where $g^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X)$ denotes the inverse image operation induced by a function $g : X \to Y$.

A neighbourhood frame is a pair (W, γ) where W is a set and $\gamma : W \to \mathcal{N}W$.

Example 3

- Probabilistic frames
  For a function $f : X \to \mathbb{R}$ we write $supp(f) = \{x \in X | f(x) \neq 0\}$ for the support of X and let $\mathcal{D}X = \{\mu : X \to [0, 1] | \text{ supp}(\mu) \text{ finite},$ $\Sigma_{x \in X} \mu(X) = 1\}$ be the set of finitely supported probability distributions on X.

Example 3

- Probabilistic frames

  For a function $f : X \to \mathbb{R}$ we write $supp(f) = \{x \in X | f(x) \neq 0\}$ for the support of X and let $\mathcal{D}X = \{\mu : X \to [0, 1] |$ supp(μ) finite, $\Sigma_{x \in X} \mu(X) = 1\}$ be the set of finitely supported probability distributions on X.

  A probabilistic frame is a pair (W, γ) where W is a set and $\gamma : W \to \mathcal{D}W$. Every probabilistic frame defines a discrete time Markov chain with transition probabilities given by the local probability distributions.

Predicate lifting

Definition

If Λ is a similarity type, a Λ-structure consists of an endofunctor $T : Set \to Set$, together with an assignment of an n-ary predicate lifting, that is, a natural transformation of type $[\![\heartsuit]\!] : (2^{-})^{n} \to 2^{-} \circ T$ where $2^{-} : Set \to Set^{op}$ is the contravariant power set functor, to every n-ary operator $\heartsuit \in \Lambda$.

Predicate lifting

Definition

The language induced by a modal similarity type Λ is the set $\mathcal{F}(\Lambda)$ of formulae

$$\mathcal{F}(\Lambda) \ni A, B ::= p|A \wedge B|\neg A|\heartsuit(A_1, \ldots, A_n) \quad (p \in P, \heartsuit \in \Lambda n - ary)$$

where P is a fixed and denumerable set of propositional variables.

Predicate lifting

Definition

The language induced by a modal similarity type Λ is the set $\mathcal{F}(\Lambda)$ of formulae

$$\mathcal{F}(\Lambda) \ni A, B ::= p|A \wedge B|\neg A|\heartsuit(A_1, \dots, A_n) \quad (p \in P, \heartsuit \in \Lambda n - ary)$$

where P is a fixed and denumerable set of propositional variables.

A T-model is a triple $M = (W, \gamma, \pi)$ where $(W, \gamma) \in Coalg(T)$ and $\pi : P \to \mathcal{P}(W)$ is a valuation. Given a Λ-structure T and a T-model $M = (W, \gamma, \pi)$, the semantics of $A \in \mathcal{F}(\Lambda)$ is inductively given by

$$[\![p]\!]_M = \pi(p) \quad [\![A \wedge B]\!]_M = [\![A]\!]_M \cap [\![B]\!]_M \quad [\![\neg A]\!]_M = W \setminus [\![A]\!]_M$$

which gives the standard interpretation of the propositional connectives over the Boolean algebra $\mathcal{P}(W)$

Predicate lifting

For the modal operators we put

$$[\![\heartsuit(A_1, ...A_n)]\!]_M = \gamma^{-1} \circ [\![\heartsuit]\!]_W([\![A_1]\!]_M, ..., [\![A_n]\!]_M).$$

Predicate lifting

For the modal operators we put

$$[\![\heartsuit(A_1, ...A_n)]\!]_M = \gamma^{-1} \circ [\![\heartsuit]\!]_W([\![A_1]\!]_M, ..., [\![A_n]\!]_M).$$

Intuitively speaking, the above definition amounts to saying that a state $\omega \in W$ satisfies a formula $\heartsuit(A_1, ...A_n)$ if the transition function γ maps it to a successor $\gamma(\omega)$ that satisfies the property $\heartsuit$ that may depend on $A_1, ..., A_n$.

Predicate lifting

For the modal operators we put

$$[\![\heartsuit(A_1, ...A_n)]\!]_M = \gamma^{-1} \circ [\![\heartsuit]\!]_W([\![A_1]\!]_M, ..., [\![A_n]\!]_M).$$

Intuitively speaking, the above definition amounts to saying that a state $\omega \in W$ satisfies a formula $\heartsuit(A_1, ...A_n)$ if the transition function γ maps it to a successor $\gamma(\omega)$ that satisfies the property $\heartsuit$ that may depend on $A_1, ..., A_n$.

We write $M, \omega \models A$ if $\omega \in [\![A]\!]_M$ and $M \models A$ if $M, \omega \models A$ for all $\omega \in W$ and finally $Mod(T) \models A$ if for all $M \in Mod(T)$, where Mod(T) denotes the collection of all T-models.

Example 1

Kripke frames

If we take $TX = \mathcal{P}X$, we have seen that T-coalgebras are precisely Kripke frames. If we choose the similarity type $\Lambda = \{\Box\}$ we obtain the standard semantics of the modal logic K by associating $\Box$ with the lifting $[\![\Box]\!]_X(Z) = \{Y \in \mathcal{P}X | Y \subset Z\}$.

If (W, γ, π) is a $\mathcal{P}$-model (a Kripke model) and $A \in \mathcal{F}(\Lambda)$ is a formula with interpretation $[\![A]\!]$, we have that

$$[\![\Box A]\!] = \gamma^{-1} \circ [\![\Box]\!]_W([\![A]\!]) = \{\omega \in W | \gamma(\omega) \subset [\![A]\!]\}$$

so that $\omega \models \Box A$ iff $\omega' \models A$ for all $\omega' \in \gamma(\omega)$. This yields the standard Kripke semantics of modal logic.

Example 1′

For $TX = \mathcal{P}X^{\mathcal{A}}$ we have seen previously that T-coalgebras are in one-to-one correspondence with labelled transition systems. Here, we consider the similarity type $\Lambda = \{[a] | a \in \mathcal{A}\}$ where each [a] is a unary operator. We extend T to a Λ-structure by stipulating that

$$[\![[a]]\!]_X(Z) = \{f : \mathcal{A} \to \mathcal{P}(X) | f(a) \subset Z\}.$$

The coalgebraic semantics precisely coincides with the standard semantics of Hennessy-Milner logic.

Example 2

Neighbourhood frames

Neighbourhood frames can be seen as coalgebras for the functor $\mathcal{N}X = 2^{2^X}$. The modal logic of neighbourhood frames is induced by the similarity type $\Lambda = \{\Box\}$, and we obtain the standard semantics if we interpret $\Box$ by $[\![\Box]\!]_X(Z) = \{Y \in \mathcal{N}X | Z \in Y\}$.

Given a neighbourhood model $M = (W, \gamma, \pi)$ where $\gamma : W \to \mathcal{N}W$ we then obtain

$$\omega \models \Box A \text{ iff } [\![A]\!] \in \gamma(\omega)$$

where $[\![A]\!] \subset W$ is the interpretation of the formula $A \in \mathcal{F}(\Lambda)$. Again this gives the standard semantics. It can be seen easily that this correspondence restricts to monotone neighbourhood frames.

Example 3

Probabilistic frames

For probabilistic frames (that is, $\mathcal{D}$-coalgebras) there is a large variation of modal operators that we may wish to consider. The probabilistic modal logic of Heifetz and Mongin uses unary operators taken from $\Lambda = \{L_p | p \in [0,1] \cap \mathbb{Q}\}$ where a formula L_pA reads as 'A holds with probability at least p in the next state'. To capture the semantics of this logic, we use the interpretation

$$[\![\mathbb{L}_p]\!]_X(Y) = \{\mu \in \mathcal{D}(X) | \mu(Y) \geq p\}$$

where we have abbreviated $\mu(Y) = \sum_{y \in Y} \mu(y)$. Given a probabilistic model (W, γ, π) where now $\gamma : W \to \mathcal{D}W$, we obtain

$$\omega \models L_p A \quad \text{iff} \quad \gamma(\omega)([\![A]\!]) \geq p$$

which captures the semantics in a coalgebraic setting.

Example 3

The logic for reasoning about probability allows linear inequalities for reasoning about probabilities, and every formal rational linear inequality

$$a_1\mu(F_1) + \cdots + a_n\mu(F_n) \geq b$$

in (formula-valued) parameters $F_1, \ldots, F_k$ defines a k-ary modal operator. To express the semantics of these operators coalgebraically, we use the lifting

$$[\![\textstyle\sum_i a_i\mu(F_i) \geq b]\!]_X(Y_1, \ldots, Y_n) = \{\mu \in \mathcal{D}(X) \mid \textstyle\sum_i a_i\mu(Y_i) \geq b\}.$$

Predicate lifting

In summary, it seems fair to say that the predicate lifting approach to coalgebraic logics subsumes a large variety of structurally different modal logics. The strength of the coalgebraic approach becomes apparent once we establish properties (such as decidability or the Hennessy-Milner property) of coalgebraic logics in the abstract framework so that we readily obtain results about concretely given logics, once they have been recognised to admit a coalgebraic semantics.

Finitary ∇ -languages

Definition

The finitary part T_ω of a set functor is given by $T_\omega X = \bigcup_{X' \subseteq_\omega X} TX'$ for $X \in Set$ where the notation $X' \subseteq_\omega X$ means that X' is a finite subset of X. Intuitively, $T_\omega X$ contains those elements of TX that can be constructed using only finitely many elements of X.

Finitary ∇ -languages

Finitary ∇ -languages now take the following form:

Definition

Let T be a set functor. The set $\mathcal{L}^T$ of formulae of coalgebraic ∇-logic is inductively defined as the smallest set closed under the following rules:

$$\frac{}{T \in \mathcal{L}^T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}^T}{\wedge\Phi \in \mathcal{L}^T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}^T}{V\Phi \in \mathcal{L}^T} \quad \frac{A \in \mathcal{L}^T}{\neg A \in \mathcal{L}^T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}^T}{\nabla\alpha \in \mathcal{L}^T} \quad \alpha \in T\Phi$$

where $X \subseteq_\omega Y$ denotes that X is a finite subset of Y.

The modal depth d(A) of a formula is defined as usually by induction on the structure of the formula. We only mention the ∇-case of the definition:

$$\mathrm{d}(\nabla\alpha) = \min\{\max\{\mathrm{d}(A) | A \in \Phi\} | \alpha \in T\Phi\} + 1$$

Finally, we write $\mathcal{L}_n^T$ for the collection of formulae with modal depth n.

And this definition ensures that each formula has a finite set of subformulas. This is the justification for calling $\mathcal{L}^T$ the finitary ∇-language for T.

Relation lifting

The key for defining the semantics of formulae in the ∇-language is the so-called relation lifting associated with a given functor.

Definition

Let $T : \text{Set} \to \text{Set}$ be a functor and let $R \subseteq X_1 \times X_2$ be a binary relation. The $(T-)$ lifted relation $\overline{T}R \subseteq TX_1 \times TX_2$ is given by

$$\overline{T}R = \{(t_1, t_2) | \exists z \in TR(T\pi_i(z) = t_i \text{ for } i = 1, 2)\}$$

where $\pi_i : R \to X_i$ is the ith projection map.

The relation lifting is well-defined for an arbitrary set functor.

Relation lifting

Nevertheless, in order to ensure that the semantics of the ∇-language is well-behaved, we make one more assumption on the functor T : we require the functor to preserve weak pullbacks. This ensures that T can be seen as a functor on the category Rel of sets and relations.

Proposition

Let T be a set functor and $\overline{T}$ its associated relation lifting. We have $\overline{T}(R \circ S) = \overline{T}R \circ \overline{T}S$ for all relations $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ iff T preserves weak pullbacks.

The semantics of ∇-formulae

From now on, when dealing with the ∇-language, we fix a standard and weak pullback preserving set functor. The following proposition lists important properties of the relation lifting for such functors.

Proposition

Let $T : Set \to Set$ be a standard, weak pullback preserving set functor and let $\overline{T}$ the corresponding relation lifting. Then (1) $\overline{T}$ is an endofunctor on the category Rel of sets and relations, (2) for any two relations $R, S \subseteq X \times Y$ we have $R \subseteq S$ implies $\overline{T}R \subseteq \overline{T}S$, and (3) $\overline{T}$ commutes with taking restrictions: $\overline{T}(R|_{Y_1 \times Y_2}) = (\overline{T}R)|_{TY_1 \times TY_2}$ for any relation $R \subseteq X_1 \times X_2$ and sets $Y_1 \subseteq X_1$, $Y_2 \subseteq X_2$.

The semantics of ∇-formulae

The semantics of ∇-formulae hinges on the preliminaries above, and takes the following form:

Definition

Let $T : Set \to Set$ be a standard, weak pullback preserving set functor and let (W, γ) be a T -coalgebra. We define the satisfaction relation $\models W \times \mathcal{L}^T$ by induction as follows:

- $\omega \models \top$ for all $\omega \in W$
- $\omega \models \wedge\Phi$ if $\omega \models A$ for all $A \in \Phi$
- $\omega \models \vee\Phi$ if there is $A \in \Phi$ with $\omega \models A$
- $\omega \models \neg A$ if not $\omega \models A$
- $\omega \models \nabla\alpha$ if $(\gamma(\omega), \alpha) \in \overline{T}(\models |_{W \times \mathcal{L}_n^T})$ for $\nabla\alpha \in \mathcal{L}_{n+1}^T$.

Finally we write $A \models B$ for two formulae $A, B \in \mathcal{L}^T$ if for all T-coalgebras (W, γ) and all states $\omega \in W$ we have $\omega \models A$ implies $\omega \models B$.

The semantics of ∇-formulae

Remark

Note that for $\nabla\alpha \in \mathcal{L}^T_{n+1}$ we have $\alpha \in T\mathcal{L}^T_n$ and hence

$$\begin{aligned}(\gamma(\omega),\alpha) \in \overline{T}(\models|_{W\times\mathcal{L}^T_n}) \quad &\text{iff} \quad (\gamma(\omega),\alpha) \in \overline{T}(\models)|_{TW\times T\mathcal{L}^T_n} \\ &\text{iff} \quad (\gamma(\omega),\alpha) \in \overline{T}(\models)\end{aligned}$$

where the first and the second equivalence follow from item (3) and item (2) of Proposition, respectively. Therefore we have $\omega \models \nabla\alpha$ iff $(\gamma(\omega),\alpha) \in \overline{T}(\models)$, which is precisely Moss'original definition of the semantics of the ∇-operator.

The semantics of ∇-formulae

Remark

We do not include propositional variables in the ∇-language $\mathcal{L}^T$. Variables can be treated by moving to a coloured version of the endofunctor under consideration: we put $T'X = \mathcal{P}(P) \times TX$ for a set P of propositional variables so that T-models are in one to one correspondence to T'-coalgebras. Concretely, in order to obtain a ∇-language for Kripke models, one considers the functor $T = \mathcal{P}(P) \times \mathcal{P}_{\_}$ where P denotes the set of propositional variables. A ∇-formula in $\mathcal{L}^T$ is then of the form $\nabla(C, \Phi)$ with $C \subseteq P$ and $\Phi \subseteq \mathcal{P}_\omega \mathcal{L}^T$. Translated to the syntax of normal modal logic, the formula $\nabla(C, \Phi)$ corresponds to the formula

$$\bigwedge_{p \in C} p \wedge \bigwedge_{p \notin C} \neg p \wedge \Box \bigvee \Phi \wedge \bigwedge_{A \in \Phi} \Diamond A.$$

Example 1

Let $T = C \times \_$ for some set C. In this case ∇-formulae are of the form $\nabla(c, A)$ where $c \in C$ (a "colour") and $A \in \mathcal{L}$ is another formula. Let $(W, \gamma : W \to C \times W)$ be a T-coalgebra. Then $\nabla(c, A)$ is true at a state $\omega \in W$ with $\gamma(\omega) = (c', \omega')$ if $c = c'$ and $\omega' \models A$.

Example 2

If we consider the power set functor $T = \mathcal{P}$, we obtain ∇-formulae of the form $\nabla\{A_1, ..., A_n\}$ where $A_1, ..., A_n$ are formulae in $\mathcal{L}$. Note that the argument of the ∇-operator is a finite set of formulae. The semantics of ∇ can be nicely expressed using the $\{\Box, \Diamond\}$-syntax of "standard" modal logic:

$$\omega \models \nabla\{A_1, ...A_n\} \text{ if } \omega \models \bigwedge_{1\leq i\leq n} \Diamond A_i \wedge \Box \bigvee_{1\leq i\leq n} A_i.$$

More formally we have that a state ω in some T-coalgebra (W, γ) makes $\nabla\{A_1, ..., A_n\}$ true if

$$\begin{array}{ll} (i) & \forall A \in \{A_1, ..., A_n\} \quad \exists \omega' \in \gamma(\omega) \quad \omega' \models A \\ (ii) & \forall \omega' \in \gamma(\omega) \quad \exists A \in \{A_1, ..., A_n\} \quad \omega' \models A. \end{array}$$

Summary

Thank you!