An institutional approach to positive coalgebraic logic

Adriana Balan[1] Alexander Kurz[2]
Jiří Velebil[3]

[1]University Politehnica of Bucharest, Romania

[2]University of Leicester, UK

[3]Czech Technical University in Prague, Czech Republic

22nd International Workshop on Algebraic Development Techniques

September 2014 – Sinaia, Romania

Background

Modal logic is about Kripke frames

$$(X, X \stackrel{\mathbf{r}}{\nrightarrow} X)$$

These are coalgebras for the **powerset** functor

$$X \to \mathcal{P}X,\ x \mapsto \{x' \in X \mid \mathbf{r}(x', x)\}$$

More generally, replace P by any functor $T : \mathsf{Set} \to \mathsf{Set}$

T-coalgebras capture LTS, (non)deterministic automata, Mealy machines, probabilistic/stochastic transition systems, ...

Reasoning about T-coalgebras: coalgebraic (modal) logic (L, δ)

Logic $L : \mathsf{BA} \to \mathsf{BA}$ functor

Semantics $\delta : LP \to PT^{\mathsf{op}}$ natural transformation

Outline

Institution **Ins** of Set-based coalgebraic logic
[Kurz-Hennicker02, Pattinson03, Cîrstea06, ...]

Ins restricts to an institution $\mathbf{Ins}_{\text{wpb}}$ having

- signatures: Set-functors which preserve weak pullbacks
- morphisms between signatures: weakly cartesian natural transformations

Positive coalgebraic logic

Logic	Axiomatization of the positive fragment of modal logic [Dunn95] Dunn's result naturally **generalize** from modal logic to coalgebraic logic [B-Kurz-Velebil13]
Coalgebra	Looking at simulations instead of bisimulations? Posets provide the environment for that
Category Theory	Posets link universal coalgebra and domain theory

Technical issue: to ensure the monotonicity of modal operators, need to work in an ordered setting (Poset-enriched category theory)

ABC of Poset-enriched category theory

Poset-category: hom-sets are ordered and composition preserves this order

Poset-functor (locally monotone): functor preserving the order on the hom-(po)sets

Poset-natural transformation: natural transformation

What a Poset-enriched institution might be?

$\mathbf{Ins} = (\mathsf{Sign}, \mathsf{Mod}, \mathsf{Sen}, \models)$

- Sign Poset-category
- $\mathsf{Mod} : \mathsf{Sign}^{\mathsf{op}} \to$ Poset-Cat   locally monotone functor
- $\mathsf{Sen} : \mathsf{Sign} \to$ Poset   locally monotone functor
- For each signature T, a relation $|\mathsf{Mod}(T)| \overset{\models}{\nrightarrow} \mathsf{Sen}(T)$ such that

$$M \models \mathsf{Sen}(\sigma)(\varphi) \iff \mathsf{Mod}(\sigma)(M) \models \varphi$$

(for each $\sigma : T \to \hat{T}$, $M \in \mathsf{Mod}(\hat{T})$, $\varphi \in \mathsf{Sen}(T)$)

What a Poset-enriched institution might be?

Ins $= (\mathsf{Sign}, \mathsf{Mod}, \mathsf{Sen}, \models)$

- Sign category
- $\mathsf{Mod} : \mathsf{Sign}^{op} \to \text{Poset-Cat} \to \mathsf{Cat}$ functor
- $\mathsf{Sen} : \mathsf{Sign} \to \mathsf{Poset} \to \mathsf{Set}$ functor
- For each signature T, a relation $|\mathsf{Mod}(T)| \overset{\models}{\nrightarrow} \mathsf{Sen}(T)$ such that

$$M \models \mathsf{Sen}(\sigma)(\varphi) \iff \mathsf{Mod}(\sigma)(M) \models \varphi$$

(for each $\sigma : T \to \hat{T}$, $M \in \mathsf{Mod}(\hat{T})$, $\varphi \in \mathsf{Sen}(T)$)

Outline

Institution **Ins** of Set-based coalgebraic logic
[Kurz-Hennicker02, Pattinson03, Cîrstea06, . . .]

Ins restricts to an institution $\mathbf{Ins}_{\mathsf{wpb}}$ having

– signatures: Set-functors which preserve weak pullbacks

– morphisms between signatures: weakly cartesian natural transformations

Outline

Institution **Ins** of Set-based coalgebraic logic
[Kurz-Hennicker02, Pattinson03, Cîrstea06, . . .]

Ins restricts to an institution $\mathbf{Ins}_{\mathsf{wpb}}$ having

– signatures: Set-functors which preserve weak pullbacks

– morphisms between signatures: weakly cartesian natural transformations

Institution $\mathbf{Ins}'$ of Poset-based coalgebraic logic, using the contravariant adjunction Poset – DLat

$\mathbf{Ins}'$ restricts to an institution $\mathbf{Ins}_{\mathsf{ex\text{-}sq}}$ having

– signatures: locally monotone functors which preserve exact squares

– morphisms between signatures: weakly exact natural transformations

Exact square

Main result

Theorem

There is a (liberal) morphism of institutions between:

- *The institution of* Set*-functors which preserve weak pullbacks and their strongly finitary coalgebraic logic* $\mathbf{Ins}_{\mathsf{wpb}}$
- *The institution of* Poset*-functors which preserve exact squares and their strongly finitary coalgebraic logic* $\mathbf{Ins}'_{\mathsf{ex\text{-}sq}}$

sending a signature to its posetification, and assigning to each logic its positive fragment.

Two institutions of (positive) coalgebraic logic

Signatures

Category of signatures

$$\mathsf{Sign} = [\mathsf{Set}, \mathsf{Set}]^{\mathsf{op}}$$

- Signature: functor $T : \mathsf{Set} \to \mathsf{Set}$
- Morphism $T \to \hat{T}$ of signatures: natural transformation $\sigma : \hat{T} \to T$

(notice the change of direction!)

Poset-category of signatures

$$\mathsf{Sign}' = [\mathsf{Poset}, \mathsf{Poset}]^{\mathsf{op}}$$

- Signature: locally monotone functor $T' : \mathsf{Poset} \to \mathsf{Poset}$
- Morphism $T' \to \hat{T}'$ of signatures: monotone natural transformation $\sigma : \hat{T}' \to T'$

Two institutions of (positive) coalgebraic logic

Signatures

Discrete Poset-category

Category of signatures

$$\mathsf{Sign} = [\mathsf{Set}, \mathsf{Set}]^{\mathsf{op}}$$

- Signature: functor $T : \mathsf{Set} \to \mathsf{Set}$
- Morphism $T \to \hat{T}$ of signatures: natural transformation $\sigma : \hat{T} \to T$

(notice the change of direction!)

Poset-**category of signatures**

$$\mathsf{Sign}' = [\mathsf{Poset}, \mathsf{Poset}]^{\mathsf{op}}$$

- Signature: locally monotone functor $T' : \mathsf{Poset} \to \mathsf{Poset}$
- Morphism $T' \to \hat{T}'$ of signatures: monotone natural transformation $\sigma : \hat{T}' \to T'$

Two institutions of (positive) coalgebraic logic

A functor between signatures

$$\mathsf{Sign} = [\mathsf{Set}, \mathsf{Set}]^{\mathrm{op}} \overset{\Phi}{\dashrightarrow} \mathsf{Sign}' = [\mathsf{Poset}, \mathsf{Poset}]^{\mathrm{op}}$$

1. For $T : \mathsf{Set} \to \mathsf{Set}$, define

   $$\Phi(T) := \mathrm{Lan}_D(DT) : \mathsf{Poset} \to \mathsf{Poset}$$ ▸ Posetification

2. For $\sigma : \hat{T} \to T$, $\Phi(\sigma)$ is the unique monotone natural transformation such that

$$\begin{array}{ccc} D\hat{T} & \xrightarrow{D\sigma} & DT \\ \cong \downarrow & & \downarrow \cong \\ \Phi(\hat{T})D & \xrightarrow[\Phi(\sigma)]{} & \Phi(T)D \end{array}$$

Two institutions of (positive) coalgebraic logic

A functor between signatures

Discrete Poset-functor

$$\mathsf{Sign} = [\mathsf{Set}, \mathsf{Set}]^{op} \xrightarrow{\ \Phi\ } \mathsf{Sign}' = [\mathsf{Poset}, \mathsf{Poset}]^{op}$$

1. For $T : \mathsf{Set} \to \mathsf{Set}$, define

$$\Phi(T) := \mathrm{Lan}_D(DT) : \mathsf{Poset} \to \mathsf{Poset}$$

▸ Posetification

2. For $\sigma : \hat{T} \to T$, $\Phi(\sigma)$ is the unique monotone natural transformation such that

$$\begin{array}{ccc} D\hat{T} & \xrightarrow{D\sigma} & DT \\ \cong\downarrow & & \downarrow\cong \\ \Phi(\hat{T})D & \xrightarrow[\Phi(\sigma)]{} & \Phi(T)D \end{array}$$

Two institutions of (positive) coalgebraic logic

Models

Reduct functor

$\mathsf{Mod} : [\mathsf{Set}, \mathsf{Set}] \to \mathsf{Cat}$

Reduct Poset-functor

$\mathsf{Mod}' : [\mathsf{Set}, \mathsf{Set}] \to \mathsf{Poset} - \mathsf{Cat}$

1. Models: coalgebras

$$T \longmapsto \mathsf{Coalg}(T)$$

2. Morphisms between models: coalgebra morphisms

$$\begin{array}{ccc}
\hat{T} & \longmapsto & \mathsf{Mod}(\hat{T}) = \mathsf{Coalg}(\hat{T}) \\
\downarrow \sigma & & \vdots\ \mathsf{Mod}(\sigma) \\
T & \longmapsto & \mathsf{Mod}(T) = \mathsf{Coalg}(T)
\end{array}
\qquad
\begin{array}{c}
X \xrightarrow{\hat{c}} \hat{T}X \\
\downarrow \\
X \xrightarrow{\hat{c}} \hat{T}X \xrightarrow{\sigma} TX
\end{array}$$

Set-examples   Poset-examples

Two institutions of (positive) coalgebraic logic

Models

Discrete Poset-functor

Reduct functor

$\mathsf{Mod} : [\mathsf{Set}, \mathsf{Set}] \to \mathsf{Cat}$

Reduct Poset-**functor**

$\mathsf{Mod}' : [\mathsf{Set}, \mathsf{Set}] \to \mathsf{Poset}-\mathsf{Cat}$

1. Models: coalgebras

$$T \longmapsto \mathsf{Coalg}(T)$$

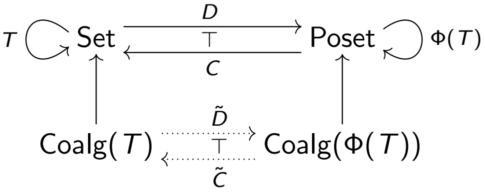
2. Morphisms between models: coalgebra morphisms

$$\begin{array}{ccc} \hat{T} & \longmapsto & \mathsf{Mod}(\hat{T}) = \mathsf{Coalg}(\hat{T}) \\ \sigma \downarrow & & \downarrow \mathsf{Mod}(\sigma) \\ T & \longmapsto & \mathsf{Mod}(T) = \mathsf{Coalg}(T) \end{array} \qquad \begin{array}{c} X \xrightarrow{\hat{c}} \hat{T}X \\ \downarrow \\ X \xrightarrow{\hat{c}} \hat{T}X \xrightarrow{\sigma} TX \end{array}$$

▸ Set-examples ▸ Poset-examples

Two institutions of (positive) coalgebraic logic

A transformation of models



The adjunction $C \dashv D$ lifts to coalgebras

There is a monotone-natural transformation

$$\beta : \mathsf{Mod} \longrightarrow \mathsf{Mod}' \circ \Phi$$

whose components $\beta_T : \mathsf{Coalg}(T) \to \mathsf{Coalg}(\Phi(T))$ are

$$X \stackrel{c}{\to} TX \longmapsto DX \stackrel{Dc}{\to} DTX \cong \Phi(T)DX$$

Notice that each component β_T has a left adjoint!

The institution of Set-coalgebraic logic

Sentences

Context: standard contravariant adjunction of propositional logic

$$\mathsf{Set}^{\mathsf{op}} \underset{P}{\overset{S}{\rightleftarrows}} \mathsf{BA} \quad (S \dashv P)$$

- P maps a set to the BA of its subsets
- S maps a BA to the set of its ultrafilters

The institution of Set-coalgebraic logic

Sentences

Context: standard contravariant adjunction of propositional logic

$$T^{op} \circlearrowright \mathrm{Set}^{op} \underset{P}{\overset{S}{\underset{\perp}{\leftrightarrows}}} \mathrm{BA} \circlearrowleft L$$

Signature $T : \mathrm{Set} \to \mathrm{Set}$
T-models: T-coalgebras

- P maps a set to the BA of its subsets
- S maps a BA to the set of its ultrafilters

Coalgebraic logic, abstractly

Syntax: functor $L : \mathrm{BA} \to \mathrm{BA}$

Semantics: natural transformation $\delta : LP \to PT^{op}$

- $\mathrm{Alg}(L)$ is a variety
- L has a presentation by operations and equations
- L preserves sifted colimits
- L is determined by its restriction to the f. g. free BAs

The institution of Set-coalgebraic logic

Sentences

– Recall: predicate liftings of arity n are natural transformations

$$\mathsf{Set}(-, 2^n) \to \mathsf{Set}(T-, 2)$$

– Equivalently, elements of $\mathsf{Set}(T(2^n), 2) \cong UPT^{\mathsf{op}}SFn$

(here $F \dashv U : \mathsf{BA} \to \mathsf{Set}$ is the monadic adjunction between the free BA functor and the forgetful one)

– Define $LFn ::= PT^{\mathsf{op}}SFn$ on free finitely generated BA and extend continuously to all BA ($L = \mathrm{Lan}_J(PT^{\mathsf{op}}SJ)$, with $J : \mathsf{BA}_\mathsf{f} \to \mathsf{BA}$ the inclusion functor)

– The semantics $\delta : LP \to PT$ is the transpose of the canonical morphism $L \to PT^{\mathsf{op}}S$

The institution of Set-coalgebraic logic

Sentences

– The natural transformation δ provides one-step semantics

– To pass to the global semantics, have to iterate the one-step logic constructor L and form the initial L-algebra $LI \xrightarrow{\mathsf{in}} I$

The functor Sen : Sign = $[\mathsf{Set}, \mathsf{Set}]^{\mathsf{op}} \to \mathsf{Set}$

1. The set of T-sentences

$$T \longmapsto \mathsf{Sen}(T) = UI$$

2. Translation of sentences

$$\begin{array}{ccc} \hat{T} \\ \sigma \downarrow \\ T \end{array} \quad \mapsto \quad \begin{array}{ccc} LP & \xrightarrow{\delta} & PT^{\mathsf{op}} \\ \lambda P \downarrow & & \downarrow P\sigma \\ \hat{L}P & \xrightarrow{\hat{\delta}} & P\hat{T} \end{array} \qquad \begin{array}{ccccc} LI & & \xrightarrow{\mathsf{in}} & & I \\ \downarrow & & & & \downarrow \\ L\hat{I} & \xrightarrow{\lambda} & \hat{L}\hat{I} & \xrightarrow{\hat{\mathsf{in}}} & \hat{I} \end{array} \qquad \begin{array}{c} UI \\ \downarrow \mathsf{Sen}(\sigma) \\ U\hat{I} \end{array}$$

The institution of Set-coalgebraic logic

Satisfaction relation

- T-model (coalgebra) $\quad X \xrightarrow{c} TX$
- L-algebra of subsets $\quad LPX \xrightarrow{\delta} PT^{\mathrm{op}}X \xrightarrow{Pc} PX$
- Unique L-algebra morphism $\quad I \xrightarrow{[\![-]\!]_{(X,c)}} PX\ , \qquad \varphi \mapsto [\![\varphi]\!]_{(X,c)}$
- Satisfaction relation

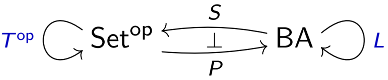
$$x \models_{(X,c)} \varphi \iff x \in [\![\varphi]\!]_{(X,c)} \qquad (X,c) \models \varphi \iff x \models_{(X,c)} \varphi,\ \forall x \in X$$

Theorem

The construction $\mathbf{Ins} = (\mathrm{Sign}, \mathrm{Mod}, \mathrm{Sen}, \models)$ *is an institution.*

Examples

Positive coalgebraic logic



Coalgebraic logic

Syntax: functor $L : \mathsf{BA} \to \mathsf{BA}$

Semantics: natural transformation $\delta : L\,P \to P\,T^{op}$

- $\mathsf{Alg}(L)$ is a variety
- L has a presentation by operations and equations
- L preserves sifted colimits
- L is determined by its restriction to free f. g. BAs

Positive coalgebraic logic

$$\mathsf{Set}^{\mathrm{op}} \underset{P}{\overset{S}{\rightleftarrows}} \mathsf{BA} \quad (\bot)$$

$$T'^{\mathrm{op}} \circlearrowright \mathsf{Poset}^{\mathrm{op}} \underset{P'}{\overset{S'}{\rightleftarrows}} \mathsf{DLat} \circlearrowleft L' \quad (\bot)$$

- P' maps a poset to the DLat of its upsets.
- S' associates to any DLat the poset of prime filters.

Poset-Coalgebraic logic

Syntax: locally monotone functor $L' : \mathsf{DLat} \to \mathsf{DLat}$

Semantics: monotone natural transformation $\delta' : L'P' \to P'T'^{\mathrm{op}}$

- $\mathsf{Alg}(L)$ is an ordered variety
- L has a presentation by monotone operations and equations
- L preserves Poset-sifted colimits
- L is determined by its restriction to free f. g. DLs on discrete posets

Positive coalgebraic logic

– Predicate liftings of arity p are monotone natural transformations

$$\mathsf{Poset}(-, [p, \mathbb{2}]) \to \mathsf{Poset}(T'-, \mathbb{2}), \quad (p \text{ finite poset})$$

– That is, elements of the poset $\mathsf{Poset}(T'([p, \mathbb{2}]), \mathbb{2}) \cong U'P'T'^{\mathrm{op}}S'F'p$

(here $[X, Y]$ is the poset of monotone maps $X \to Y$, and $F' \dashv U' : \mathsf{DLat} \to \mathsf{Poset}$ is the Poset-monadic adjunction between the free DL functor and the forgetful one)

– Define $L'F'Dn ::= P'T'^{\mathrm{op}}S'F'Dn$ on free finitely generated DL on discrete generators and extend continuously to all DL

– The semantics $\delta : L'P' \to P'T'$ is the transpose of the canonical morphism $L' \to P'T'^{\mathrm{op}}S'$

– Logic (L', δ') is expressive [Kapulkin-Kurz-Velebil12], for finitary T' which preserves embeddings

Positive coalgebraic logic

– Predicate liftings of arity p are monotone natural transformations

$$\mathsf{Poset}(-,[p,\mathbb{2}]) \to \mathsf{Poset}(T'-,\mathbb{2}), \quad (p \text{ finite poset})$$

– That is, elements of the poset $\mathsf{Poset}(T'([p,\mathbb{2}]),\mathbb{2}) \cong U'P'T'^{\mathsf{op}}S'F'p$

(here $[X,Y]$ is the p ... nd $F' \dashv U' : \mathsf{DLat} \to \mathsf{Poset}$ is the Poset-monadic adjunction between the free DL functor and the forgetful one)

Only discrete arities!

– Define $L'F'Dn ::= P'T'^{\mathsf{op}}S'F'Dn$ on free finitely generated DL on discrete generators and extend continuously to all DL

– The semantics $\delta : L'P' \to P'T'$ is the transpose of the canonical morphism $L' \to P'T'^{\mathsf{op}}S'$

– Logic (L',δ') is expressive [Kapulkin-Kurz-Velebil12], for finitary T' which preserves embeddings

The institution of Poset-coalgebraic logic

Sentences and satisfaction relation

– $\mathsf{Sen}' : \mathsf{Sign}' \to$ Poset **locally monotone functor**

$$T' : \mathsf{Poset} \to \mathsf{Poset} \quad \longmapsto \quad \mathsf{Sen}(T') := U'I' \text{ poset of sentences}$$

(where $L'I' \longrightarrow I'$ is the initial L'-algebra)

– T'-coalgebra $\quad X \xrightarrow{c} T'X \quad \Longrightarrow \quad [\![-]\!]_{(X,c)} : I' \to P'X$

(a formula is sent to the upperset of states satisfying it)

– Satisfaction relation

$$x \vDash_{(X,c)} \varphi \Longleftrightarrow x \in [\![\varphi]\!]_{(X,c)}\ , \quad (X,c) \vDash \varphi \Longleftrightarrow \forall x \in X,\ x \vDash_{(X,c)} \varphi$$

↑

Monotone wrt simulations

▸ About simulations

The institution of Poset-based coalgebraic logic

Theorem

The construction $\mathbf{Ins}' = (\mathsf{Sign}', \mathsf{Mod}', \mathsf{Sen}', \models)$ *is an institution, where:*

1. $\mathsf{Sign}' = [\mathsf{Poset}, \mathsf{Poset}]^{\mathsf{op}}$
2. $\mathsf{Mod}' : \mathsf{Sign}'^{\mathsf{op}} \longrightarrow \mathsf{Poset}\text{-}\mathsf{Cat} \longrightarrow \mathsf{Cat}, \quad \mathsf{Mod}(T) = \mathsf{Coalg}(T)$
3. $\mathsf{Sen}' : \mathsf{Sign}' \longrightarrow \mathsf{Poset} \longrightarrow \mathsf{Set}, \quad \mathsf{Sen}'(T) = U'I'$
4. *The satisfaction relation* $\models \subseteq |\mathsf{Mod}'(T')| \times \mathsf{Sen}(T')$ *is defined as earlier*

The positive fragment of coalgebraic logic

Theorem (B-Kurz-Velebil13)

Let $T : \mathsf{Set} \to \mathsf{Set}$ *such that:*

- T *preserves weak pullbacks*
- $T' = \mathsf{Lan}_D(DT)$ *is the posetification of* T
- (L, δ) *and* (L', δ') *are the (strongly finitary) logics of* T *and* T'

Then L' *is the positive fragment of* L. *More precisely, there is an isomorphism*

$$\begin{array}{ccc} \mathsf{DLat} & \xrightarrow{L'} & \mathsf{DLat} \\ {\scriptstyle W}\downarrow & \cong & \downarrow{\scriptstyle W} \\ \mathsf{BA} & \xrightarrow[L]{} & \mathsf{BA} \end{array}$$

compatible with semantics $\delta : LP \to PT^{\mathsf{op}}$ *and* $\delta' : L'P' \to P'T'^{\mathsf{op}}$

Two institutions of (positive) coalgebraic logic

Relating the sentences

$$\begin{array}{ccc} \mathsf{DLat} & \xrightarrow{L'} & \mathsf{DLat} \\ {\scriptstyle W}\downarrow & \cong & \downarrow{\scriptstyle W} \\ \mathsf{BA} & \xrightarrow[L]{} & \mathsf{BA} \end{array}$$

Apply the isomorphism above to construct a monotone natural transformation between sentences

$$\alpha : \mathsf{Sen}' \circ \Phi \to \mathsf{Sen}$$

restricted to signature functors T which preserve weak pullbacks and weakly cartesian natural transformations

$$\begin{array}{ccccc} L'I' & & \xrightarrow{\mathsf{in}'} & & I' \\ \downarrow & & & & \vdots \downarrow \\ L'WI & \cong & WLI & \xrightarrow{W\mathsf{in}} & WI \end{array} \qquad \begin{array}{ccc} U'I' & & \\ \vdots\downarrow {\scriptstyle \alpha_T} & & \\ U'WI & \cong & UI \end{array}$$

▸ Example

Main result

Theorem

There is a (liberal) morphism of institutions between:

- *The institution of* Set*-functors which preserve weak pullbacks and their strongly finitary coalgebraic logic* $\mathbf{Ins}_{\mathsf{wpb}}$
- *The institution of* Poset*-functors which preserve exact squares and their strongly finitary coalgebraic logic* $\mathbf{Ins}'_{\mathsf{ex\text{-}sq}}$

sending a signature to its posetification, and assigning to each logic its positive fragment.

Some references

B-Kurz11 *Finitary functors: from Set to Preord and Poset*, CALCO2011

B-Kurz-Velebil13 *Positive fragments of coalgebraic logics*, CALCO2013, arXiv:1402.5922 (2014)

Cîrstea06 *An institution of modal logics for coalgebras*, J. Logic Alg. Progr. 67(2006)

Dunn95 *Positive Modal Logic*, Studia Logica 55(1995)

Kapulkin-Kurz-Velebil12 *Expressiveness of Positive Coalgebraic Logic*, Adv. Modal Logic 9(2012)

Kurz-Hennicker02 *On institutions for modular coalgebraic specifications*, TCS 280(2002)

Kurz-Pattinson00 *Coalgebras and Modal Logic for Parameterised Endofunctors*. CWI Tech. Rep. (2000)

Pattinson03 *Translating logics for coalgebras*, WADT2002

Thank you!

Thank you!

Thank you!

Examples

1. $T = \mathcal{P}$ (finite) powerset functor

   Logic LA is the BA generated by $\Box a$, for $a \in A$, wrt $\Box$ preserving finite meets

   Semantics $\delta_X : LPX \to P\mathcal{P}^{\mathsf{op}}X, \quad \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\}$

2. $T = \mathcal{N}$ the neighbourhood functor.

   Logic LA is the BA generated by $\Box a$, for $a \in A$, no equations

   Semantics $\delta_X : LPX \to P\mathcal{N}^{\mathsf{op}}X, \Box a \mapsto \{s \in \mathcal{N}X \mid a \in s\}$

More examples...

1. $T = \mathcal{M}$ the (finite) multisets functor

   Logic: LA is the BA generated by $\Diamond_n a$, for $a \in A$, wrt $\Diamond_n$ preserving finite joins

   Semantics: $\delta_X : LPX \to P\mathcal{M}^{\mathrm{op}}X$,
   $\Diamond_n a \mapsto \{\varphi \in \mathcal{M}X \mid \operatorname*{card}_{x \in a} \varphi(x) \geq n\}$, for $n \in \mathbb{N}$

2. $T = \mathcal{D}$ (finite) probability functor

   Logic: LA is the BA generated by $\Diamond_q a$, for $a \in A$, wtr $\Diamond_q$ preserving finite joins

   Semantics: $\delta_X : LPX \to P\mathcal{D}^{\mathrm{op}}X$, $\Diamond_q a \mapsto \{d \in \mathcal{D}X \mid \sum_{x \in a} d(x) \geq q\}$
   for $q \in \mathbb{Q} \cap [0, 1]$

Posetifications - or how to extend functors from sets to posets

Functor $T : \mathsf{Set} \to \mathsf{Set}$

Extension Locally monotone functor
$T' : \mathsf{Poset} \to \mathsf{Poset}$

$$\begin{array}{ccc} \mathsf{Set} & \xrightarrow{D} & \mathsf{Poset} \\ {\scriptstyle T}\downarrow & \cong & \downarrow{\scriptstyle T'} \\ \mathsf{Set} & \xrightarrow{D} & \mathsf{Poset} \end{array}$$

Posetification Extension with universal property $T' = \mathrm{Lan}_D(DT)$
Poset-left Kan extension

Theorem (B-Kurz-Velebil13)

1. **Existence**

   Posetification exists for any functor $T : \mathsf{Set} \to \mathsf{Set}$

2. **A characterisation of left Kan extensions to posets**

   For locally monotone $T' : \mathsf{Poset} \to \mathsf{Poset}$, *TFAE*

   - T' *is* $\mathrm{Lan}_D(DT)$ *for some* $T : \mathsf{Set} \to \mathsf{Set}$
   - T' *preserves discrete posets and coinserters of simplicial resolutions*

3. **Taking posetifications is functorial**

$$[\mathsf{Set}, \mathsf{Set}] \longrightarrow [\mathsf{Poset}, \mathsf{Poset}], \quad T \mapsto \mathrm{Lan}_D(DT)$$

(proof technique: use a "simplicial representation" of posets)

Examples

Kripke functors

$$T ::= \mathrm{Id} \mid T_{X_0} \mid T_0 + T_1 \mid T_0 \times T_1 \mid T^A$$

Posetifications are as expected:

- Posetification of $\mathrm{Id}_{\mathsf{Set}}$ is $\mathrm{Id}_{\mathsf{Poset}}$
- Posetification of the constant functor at set X_0 is the constant functor at discrete poset $(X_0, =)$
- Posetification of (co)product functor is again the (co)product, this time in Poset
- Posetification of exponential functor $TX = X^A$ is again exponential in Poset

Examples (continued)

Motivating example: $T = \mathcal{P}$, the (finite) power-set functor

Posetification is the (finitely generated) convex power-set functor, with the Egli-Milner order.

Examples (continued)

Motivating example: $T = \mathcal{P}$, the (finite) power-set functor

Posetification is the (finitely generated) convex power-set functor, with the Egli-Milner order.

Distribution functor $\mathcal{D}X = \{d : X \to [0,1] \mid \sum_{x \in X} d(x) = 1\}$
Coalgebras: Markov chains
Posetification: $\mathcal{D}'(X, \leq)$ is $\mathcal{D}X$, with order given by

$$d \leq d' \Leftrightarrow \exists \omega \in \mathcal{D}(X \times X)\ .\ \begin{cases} \omega(x,y) > 0 \Rightarrow x \leq y \\ \sum_{y \in X} \omega(x,y) = d(x) \\ \sum_{x \in X} \omega(x,y) = d'(y) \end{cases}$$

Multiset functor $\mathcal{M}X = \{\varphi : X \to \mathbb{N} \mid \mathrm{supp}(\varphi) < \infty\}$
Coalgebras: multigraphs
Posetification: still to compute...

Simplicial representation of posets

X poset $\Longrightarrow$ diagram of (discrete po)sets: $X_1 \overset{d_0}{\underset{d_1}{\rightrightarrows}} X_0$

- X_0 is the underlying *set* of X
- X_1 is the *set* of comparable pairs $X_1 = \{(x, x') \in X \mid x \leq x'\}$
- $d_0, d_1 : X_1 \to X_0$ the usual projections

The coinserter of the diagram is X (coinserter = ordered analogue of a coequalizer)

The left Kan extension (posetification) of any $T : \mathsf{Set} \to \mathsf{Set}$

Put $T'X := \mathrm{coins}(Td_0, Td_1)$, for a poset X
The assignment $X \mapsto T'X$ is locally monotone, coincides with T on discrete posets and can be exhibited as left Kan extension of DT along D

Example

- $\hat{T} : \mathsf{Set} \to \mathsf{Set}, \quad \hat{T}X = 2 \times X^A$

  $\hat{T}$-coalgebras are deterministic automata with alphabet A and binary outputs, deciding if a state is accepting or not

- $T : \mathsf{Set} \to \mathsf{Set}, \quad TX = (\mathcal{P}X)^A$

  T-coalgebras are LTS, with label set A

- Natural transformation $\sigma : \hat{T} \to T, \quad \sigma_X : 2 \times X^A \to (\mathcal{P}X)^A$,
  $\sigma_X(i, f)(a) = \{f(a)\}$

  Then $\mathsf{Mod}(\sigma) : \mathsf{Coalg}(\hat{T}) \to \mathsf{Coalg}(T)$ transforms a deterministic automata into a LTS forgetting whether the resulting state is accepting outputs

Poset-functors and their coalgebras

T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$

Monotone transition map $\mathbb{X} \xrightarrow{c} T\mathbb{X}$

Monotone translation map $\mathbb{X} \xrightarrow{f} \mathbb{Y}$

$$\begin{array}{ccc} \mathbb{X} & \xrightarrow{f} & \mathbb{Y} \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle d} \\ T\mathbb{X} & \xrightarrow{Tf} & T\mathbb{Y} \end{array}$$

Poset-category Coalg(T)

Poset-functors and their coalgebras

T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$

Monotone transition map $\mathbb{X} \stackrel{c}{\to} T\mathbb{X}$

Monotone translation map $\mathbb{X} \stackrel{f}{\to} \mathbb{Y}$

$$\begin{array}{ccc} \mathbb{X} & \xrightarrow{f} & \mathbb{Y} \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle d} \\ T\mathbb{X} & \xrightarrow{Tf} & T\mathbb{Y} \end{array}$$

Poset-category Coalg(T)

Examples

- ordered automata: $X \to X^A \times \mathbb{2}$, with A the (discrete) input set

Poset-functors and their coalgebras

T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$
Monotone transition map $\mathbb{X} \xrightarrow{c} T\mathbb{X}$
Monotone translation map $\mathbb{X} \xrightarrow{f} \mathbb{Y}$

$$\begin{array}{ccc} \mathbb{X} & \xrightarrow{f} & \mathbb{Y} \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle d} \\ T\mathbb{X} & \xrightarrow{Tf} & T\mathbb{Y} \end{array}$$

Poset-category Coalg(T)

Examples

- ordered automata: $X \to X^A \times 2$, with A the (discrete) input set

$x \leq x'$

Poset-functors and their coalgebras

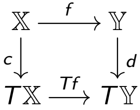
T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$

Monotone transition map $\mathbb{X} \xrightarrow{c} T\mathbb{X}$

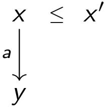
Monotone translation map $\mathbb{X} \xrightarrow{f} \mathbb{Y}$



Poset-category Coalg(T)

Examples

- ordered automata: $X \to X^A \times \mathbb{2}$, with A the (discrete) input set



Poset-functors and their coalgebras

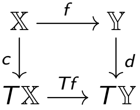
T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$

Monotone transition map $\mathbb{X} \xrightarrow{c} T\mathbb{X}$

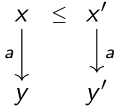
Monotone translation map $\mathbb{X} \xrightarrow{f} \mathbb{Y}$



Poset-category Coalg(T)

Examples

- ordered automata: $X \to X^A \times \mathbb{2}$, with A the (discrete) input set



Poset-functors and their coalgebras

T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$
Monotone transition map $\mathbb{X} \xrightarrow{c} T\mathbb{X}$
Monotone translation map $\mathbb{X} \xrightarrow{f} \mathbb{Y}$

$$\begin{array}{ccc} \mathbb{X} & \xrightarrow{f} & \mathbb{Y} \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle d} \\ T\mathbb{X} & \xrightarrow{Tf} & T\mathbb{Y} \end{array}$$

Poset-category Coalg(T)

Examples

- ordered automata: $X \to X^A \times \mathbb{2}$, with A the (discrete) input set

$$\begin{array}{ccc} x & \leq & x' \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle a} \\ y & \leq & y' \end{array}$$

Poset-functors and their coalgebras

T : Poset $\to$ Poset **locally monotone functor**

T-coalgebras

Partially ordered set of states $\mathbb{X} = (X, \leq)$

Monotone transition map $\mathbb{X} \xrightarrow{c} T\mathbb{X}$

Monotone translation map $\mathbb{X} \xrightarrow{f} \mathbb{Y}$

$$\begin{array}{ccc} \mathbb{X} & \xrightarrow{f} & \mathbb{Y} \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle d} \\ T\mathbb{X} & \xrightarrow{Tf} & T\mathbb{Y} \end{array}$$

Poset-category Coalg(T)

Examples

- ordered automata: $X \to X^A \times \mathbb{2}$, with A the (discrete) input set

$$\begin{array}{ccc} x & \leq & x' \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle a} \\ y & \leq & y' \end{array}$$

- ordered Kripke frames: $X \to \mathcal{P}_c X$, with $\mathcal{P}_c$ the convex powerset functor ordered by

$$U, V \in P_c X,\ U \sqsubseteq V \Leftrightarrow (\ \forall x \in U\ \exists y \in V.\ x \leq y\ )\ \wedge\ (\ \forall y \in V\ \exists x \in U.\ x \leq y\ )$$

Monotone relations

A monotone relation $X \overset{\mathbf{r}}{\nrightarrow} Y$ is a monotone map $\mathbf{r} : Y^{\mathsf{op}} \times X \to \mathbb{2}$

$$y' \leq y \quad \wedge \quad \mathbf{r}(y, x) \quad \wedge \quad x \leq x' \implies \mathbf{r}(y', x')$$

Each monotone map $f : X \to Y$ produce two (adjoint) monotone relations:

$$X \overset{f_\Diamond}{\nrightarrow} Y \qquad f_\Diamond(y, x) \iff y \leq f(x)$$

$$Y \overset{f^\Diamond}{\nrightarrow} X \qquad f^\Diamond(x, y) \iff f(x) \leq y$$

Each monotone relation $X \overset{\mathbf{r}}{\nrightarrow} Y$ can be represented as a cospan

$$Y \xrightarrow{r_0} \mathsf{Coll}(\mathbf{r}) \xleftarrow{r_1} X \,, \qquad \mathbf{r} = r_0^\Diamond \circ r_{1\Diamond}$$

where the collage $\mathsf{Coll}(\mathbf{r})$ is $Y + X$, with order

$$y \leq y' \qquad x \leq x' \qquad \mathbf{r}(y, x)$$

Relation lifting

T : Poset $\to$ Poset locally monotone functor , $Y \overset{\mathbf{r}}{\nrightarrow} X$ monotone relation

$$Y \xrightarrow{r_0} \mathrm{Coll}(\mathbf{r}) \xleftarrow{r_1} X \qquad\qquad TY \xrightarrow{Tr_0} T\mathrm{Coll}(\mathbf{r}) \xleftarrow{Tr_1} TX$$

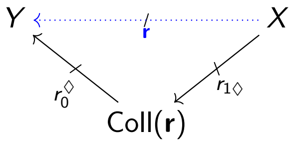
Relation lifting

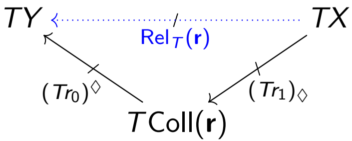
T : Poset $\to$ Poset locally monotone functor , $Y \overset{\mathbf{r}}{\nrightarrow} X$ monotone relation

$\mathbf{r} = r_0^\Diamond \circ r_{1\Diamond}$

Relation lifting

T : Poset $\to$ Poset locally monotone functor , $Y \overset{\mathbf{r}}{\nrightarrow} X$ monotone relation





$\mathbf{r} = r_0^\Diamond \circ r_{1\Diamond}$

Relation lifting $\mathsf{Rel}_T(\mathbf{r}) = (Tr_0)^\Diamond \circ (Tr_1)_\Diamond$

Relation lifting

T : Poset $\to$ Poset locally monotone functor , $Y \overset{\mathbf{r}}{\nrightarrow} X$ monotone relation

$\mathbf{r} = r_0^{\Diamond} \circ r_{1\Diamond}$ Relation lifting $\mathrm{Rel}_T(\mathbf{r}) = (Tr_0)^{\Diamond} \circ (Tr_1)_{\Diamond}$

$\mathrm{Rel}_T(\mathbf{r})(v, u) \iff Tr_0(v) \leq Tr_1(u)$ in $T\mathrm{Coll}(\mathbf{r})$

Relation lifting

T : Poset $\to$ Poset locally monotone functor , $Y \overset{\mathbf{r}}{\nrightarrow} X$ monotone relation

$\mathbf{r} = r_0^{\Diamond} \circ r_{1\Diamond}$ Relation lifting $\mathrm{Rel}_T(\mathbf{r}) = (Tr_0)^{\Diamond} \circ (Tr_1)_{\Diamond}$

$\mathrm{Rel}_T(\mathbf{r})(v, u) \iff Tr_0(v) \leq Tr_1(u)$ in $T\mathrm{Coll}(\mathbf{r})$

T preserves exact squares $\Longrightarrow$ well-behaved relation lifting

Exact squares

Exact square:

$$
\begin{array}{ccc}
E & \xrightarrow{p_1} & Y \\
{\scriptstyle p_0}\downarrow & \leq & \downarrow{\scriptstyle g} \\
X & \xrightarrow[f]{} & Z
\end{array}
\quad \text{and} \quad
\begin{array}{ccc}
E & \overset{p_{0\diamond}}{\nrightarrow} & Y \\
{\scriptstyle p_1^{\diamond}}\downarrow & = & \downarrow{\scriptstyle g^{\diamond}} \\
X & \underset{f_{\diamond}}{\nrightarrow} & Z
\end{array}
$$

(exact square = the ordered analogue of weak pullback)

Exact squares

Exact square: $\begin{array}{ccc} E & \xrightarrow{p_1} & Y \\ {\scriptstyle p_0}\downarrow & \leq & \downarrow{\scriptstyle g} \\ X & \xrightarrow[f]{} & Z \end{array}$ and

$$f(x) \leq g(y)$$
$$\Downarrow$$
$$\exists\, w \in E.\ (x \leq \alpha(w) \wedge \beta(w) \leq y)$$

(exact square = the ordered analogue of weak pullback)

Relation lifting - examples

1. $T = \mathcal{P}_c$ the convex powerset $\quad X \overset{\mathbf{r}}{\nrightarrow} Y$

   $\mathrm{Rel}_{\mathcal{P}_c}(\mathbf{r})(v, u) \iff (\forall y \in v\ \exists x \in u.\ \mathbf{r}(y, x)) \wedge (\forall x \in u\ \exists y \in v.\ \mathbf{r}(y, x))$

   (here $u \in \mathcal{P}_c(X)$, $v \in \mathcal{P}_c(Y)$ are convex subsets)

2. $T = (-)^A \times \mathbb{2}$

   $$\mathrm{Rel}_T(\mathbf{r})(v, u) \iff (\forall a \in A, \mathbf{r}(x_a, y_a)) \wedge (i \leq j)$$

   where $u = ((x_a)_{a \in A}, i) \in TX$, $v = ((y_a)_{a \in A}, j) \in TY$

Simulations

Locally monotone functor T' : Poset $\to$ Poset

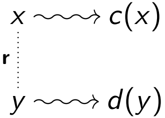
Coalgebras $X \stackrel{c}{\to} T'X$, $Y \stackrel{d}{\to} T'Y$

Simulations

Locally monotone functor $T' : \mathsf{Poset} \to \mathsf{Poset}$

Coalgebras $X \xrightarrow{c} T'X$, $Y \xrightarrow{d} T'Y$

A monotone relation $X \overset{\mathbf{r}}{\nrightarrow} Y$ is called a simulation if

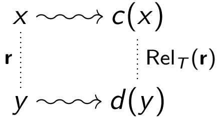


Simulations

Locally monotone functor $T' : \mathsf{Poset} \to \mathsf{Poset}$

Coalgebras $X \xrightarrow{c} T'X$, $Y \xrightarrow{d} T'Y$

A monotone relation $X \overset{\mathbf{r}}{\nrightarrow} Y$ is called a simulation if



Simulations

Locally monotone functor $T' : \mathsf{Poset} \to \mathsf{Poset}$

Coalgebras $X \xrightarrow{c} T'X$, $Y \xrightarrow{d} T'Y$

A monotone relation $X \overset{\mathbf{r}}{\nrightarrow} Y$ is called a simulation if

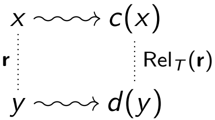
$$\begin{array}{ccc} x & \rightsquigarrow & c(x) \\ \mathbf{r}\ \vdots & & \vdots\ \mathsf{Rel}_T(\mathbf{r}) \\ y & \rightsquigarrow & d(y) \end{array} \qquad\qquad \begin{array}{ccc} X & \xrightarrow[]{c_\Diamond}\!\!\!\!\!\!\!\!\!\nrightarrow & TX \\ \mathbf{r}\downarrow\!\!\!\!+ & \leq & +\!\!\!\!\downarrow \mathsf{Rel}_T(\mathbf{r}) \\ Y & \xrightarrow[d_\Diamond]{}\!\!\!\!\!\!\!\!\!\nrightarrow & TY \end{array}$$

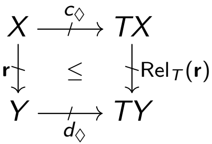
Simulations

Locally monotone functor $T' : \mathsf{Poset} \to \mathsf{Poset}$

Coalgebras $X \xrightarrow{c} T'X$, $Y \xrightarrow{d} T'Y$

A monotone relation $X \overset{\mathbf{r}}{\nrightarrow} Y$ is called a simulation if





A coalgebra morphism $X \xrightarrow{f} Y$ induces simulations $X \overset{f_\diamond}{\nrightarrow} Y$ and $Y \overset{f^\diamond}{\nrightarrow} X$

Simulations are closed under composition if T' preserves exact squares

Satisfaction relation is monotone wrt simulation order on states:

$$\mathbf{r}(y, x) \ \wedge \ (x \models \varphi) \ \wedge \ (\varphi \le \psi) \Longrightarrow (y \models \psi)$$

for all simulations $X \overset{\mathbf{r}}{\nrightarrow} Y$, states $x \in X$, $y \in Y$ and formulae $\varphi, \psi \in I'$

Motivating example

Signature $T = \mathcal{P}$ (finite) powerset functor

Logic LA is the BA generated by $(\Box a)_{a \in A}$ such that
$\Box(a \wedge b) = \Box a \wedge \Box b$

Semantics $\delta_X : LPX \to P\mathcal{P}^{\mathrm{op}}X, \quad \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\}$

Posetification $T' = \mathcal{P}_c$ (finitely generated) convex powerset functor

Logic $L'A$ is the DLat generated by $(\Box a, \Diamond a)_{a \in A}$b such that
$\Box(a \wedge b) = \Box a \wedge \Box b, \quad \Diamond(a \vee b) = \Diamond a \vee \Diamond b$
$\Box a \wedge \Diamond b \leq \Diamond(a \wedge b), \quad \Box(a \vee b) \leq \Diamond a \vee \Box b$

Semantics

$$\delta'_X : L'P'X \to P'\mathcal{P}'^{\mathrm{op}}X, \begin{cases} \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\} \\ \Diamond a \mapsto \{b \in \mathcal{P}X \mid b \cap a \neq \emptyset\} \end{cases}$$

Translation $L'W \cong WL$ induce the DL morphism $\alpha_{\mathcal{P}}$

$$\alpha_{\mathcal{P}}(\Diamond\varphi) = \neg\Box\neg\alpha_{\mathcal{P}}(\varphi) \qquad \alpha_{\mathcal{P}}(\Box\varphi) = \Box\alpha_{\mathcal{P}}(\varphi)$$

Motivating example

Signature $T = \mathcal{P}$ (finite) powerset functor

Logic LA is the BA generated by $(\Box a)_{a \in A}$ such that $\Box(a \wedge b) = \Box a \wedge \Box b$

Semantics $\delta_X : LPX \to P\mathcal{P}^{op}X, \quad \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\}$

Posetification $T' = \mathcal{P}_c$ (finitely generated) convex powerset functor

Logic $L'A$ is the DLat generated by $(\Box a, \Diamond a)_{a \in A}$b such that

$$\Box(a \wedge b) = \Box a \wedge \Box b, \quad \Diamond(a \vee b) = \Diamond a \vee \Diamond b$$

$$\Box a \wedge \Diamond b \leq \Diamond(a \wedge b), \quad \Box(a \vee b) \leq \Diamond a \vee \Box b$$

Semantics

$$\delta'_X : L'P'X \to P'\mathcal{P}'^{op}X, \begin{cases} \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\} \\ \Diamond a \mapsto \{b \in \mathcal{P}X \mid b \cap a \neq \emptyset\} \end{cases}$$

Translation $L'W \cong WL$ induce the DL morphism $\alpha_{\mathcal{P}}$

$$\alpha_{\mathcal{P}}(\Diamond\varphi) = \neg\Box\neg\alpha_{\mathcal{P}}(\varphi) \qquad \alpha_{\mathcal{P}}(\Box\varphi) = \Box\alpha_{\mathcal{P}}(\varphi)$$

Motivating example

Signature $T = \mathcal{P}$ (finite) powerset functor

Logic LA is the BA generated by $(\Box a)_{a \in A}$ such that
$\Box(a \wedge b) = \Box a \wedge \Box b$

Semantics $\delta_X : LPX \to P\mathcal{P}^{\mathrm{op}}X, \quad \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\}$

Posetification $T' = \mathcal{P}_c$ (finitely generated) convex powerset functor

Logic $L'A$ is the DLat generated by $(\Box a, \Diamond a)_{a \in A}$b such that
$\Box(a \wedge b) = \Box a \wedge \Box b, \quad \Diamond(a \vee b) = \Diamond a \vee \Diamond b$
$\Box a \wedge \Diamond b \leq \Diamond(a \wedge b), \quad \Box(a \vee b) \leq \Diamond a \vee \Box b$

Semantics

$$\delta'_X : L'P'X \to P'\mathcal{P}'^{\mathrm{op}}X, \begin{cases} \Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\} \\ \Diamond a \mapsto \{b \in \mathcal{P}X \mid b \cap a \neq \emptyset\} \end{cases}$$

Translation $L'W \cong WL$ induce the DL morphism $\alpha_{\mathcal{P}}$

$$\alpha_{\mathcal{P}}(\Diamond\varphi) = \neg\Box\neg\alpha_{\mathcal{P}}(\varphi) \qquad \alpha_{\mathcal{P}}(\Box\varphi) = \Box\alpha_{\mathcal{P}}(\varphi)$$