Coalgebraic Dynamic Logic: Safety and Reducibility

Wolfgang Poiger
Institute of Computer Science of the Czech Academy of Sciences

Joint work with Helle H. Hansen (RUG, Groningen)

Czech Gathering of Logicians & Beauty of Logic

Prague - February 5, 2026

Motivation and Overview

Coalgebraic Modal Logic (CML)

- Modal logic for F-coalgebras
- Many-valued CML
  (cf. Kurz,P,Teheux 2023/24, Lin,Liau 2023, Bílková,Dostál 2013/16)
- Dynamic CML
  (cf. Hansen,Kupke,Leal 2014/15)

Motivation and Overview

Coalgebraic Modal Logic (CML)

- Modal logic for F-coalgebras
- Many-valued CML
  (cf. Kurz,P,Teheux 2023/24, Lin,Liau 2023, Bílková,Dostál 2013/16)
- Dynamic CML
  (cf. Hansen,Kupke,Leal 2014/15)

Aim: Combine many-valued and dynamic CML.

Motivation and Overview

Coalgebraic Modal Logic (CML)

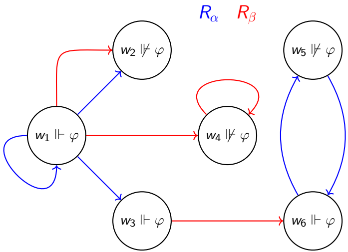
- Modal logic for F-coalgebras
- Many-valued CML
  (cf. Kurz,P,Teheux 2023/24, Lin,Liau 2023, Bílková,Dostál 2013/16)
- Dynamic CML
  (cf. Hansen,Kupke,Leal 2014/15)

Aim: Combine many-valued and dynamic CML.

Overview of this talk:

- Concrete many-valued dynamic logics
- Many-valued dynamic CMLs
- Safe operations
- Reducible operations and strong completeness

Propositional Dynamic Logic (PDL)

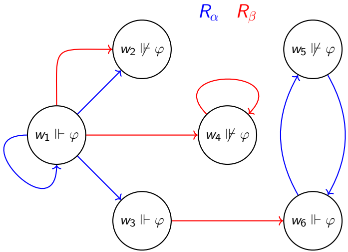


$w_1 \Vdash \langle\alpha\rangle\varphi \quad w_1 \nVdash \langle\beta\rangle\varphi$

'It is possible that φ holds after executing program α'.
'It is not possible that φ holds after executing program β'.

[Fischer and Ladner, 1979]

Propositional Dynamic Logic (PDL)



$w_1 \Vdash \langle\alpha\rangle\varphi \quad w_1 \not\Vdash \langle\beta\rangle\varphi$

'It is possible that φ holds after executing program α'.
'It is not possible that φ holds after executing program β'.

- $R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$
- $R_{\alpha;\beta} = R_\alpha ; R_\beta = \{(w, v) \mid \exists u\colon wR_\alpha uR_\beta v\}$
- $R_{\alpha^*} = \bigcup_{n\geq 0} R_\alpha^n$ (reflexive transitive closure)
- $R_{\varphi?} = \{(w, w) \mid w \Vdash \varphi\}$

[Fischer and Ladner, 1979]

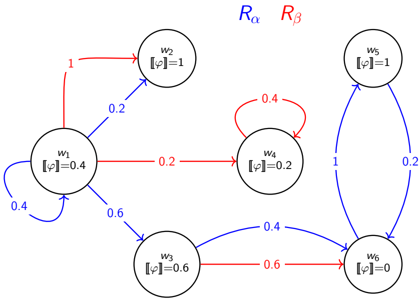
Beyond two

$$\mathbf{A} = \langle A, \wedge, \vee, \odot, \to, 0, 1 \rangle$$

Complete $\mathsf{FL}_{\mathrm{ew}}$-algebra
($\Leftrightarrow$ Complete commutative integral residuated lattice)
($\Leftrightarrow$ Commutative integral quantale)

Example. $[0, 1]$ with $\odot$ being the product of reals.

Many-valued PDL, labelled accessibility

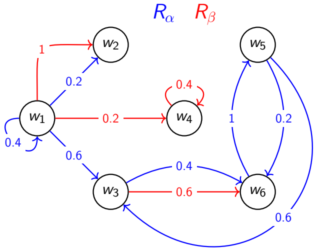


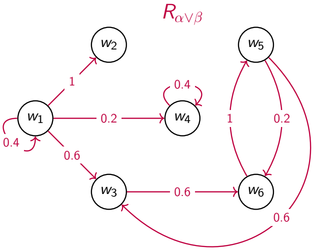
$$[\![\langle\alpha\rangle\varphi]\!](w_1) = \bigvee R_\alpha(w_1, w_i) \odot [\![\varphi]\!](w_i) = 0.36$$

The maximal chance to reach φ after executing α is 0.36'.

Similarly to [Sedlàr, 2020] and further references therein

Many-valued PDL, labelled accessibility

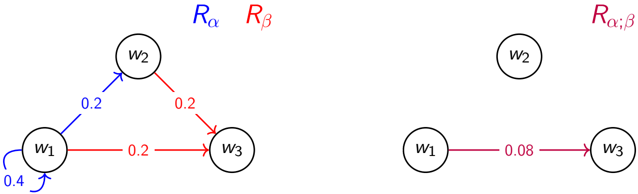




$$R_{\alpha\vee\beta} = R_\alpha \vee R_\beta$$

'*Maximal value to execute* α *or* β '

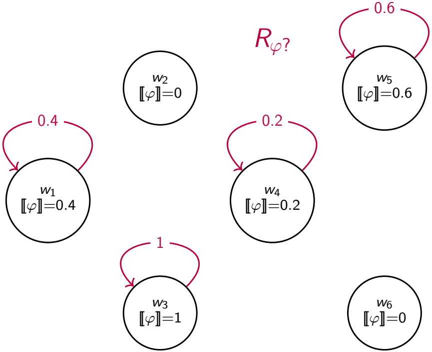
Many-valued PDL, labelled accessibility



$$R_{\alpha;\beta} = R_\alpha; R_\beta \qquad R_{\alpha;\beta}(x,z) = \bigvee\{R_\alpha(x,y) \odot R_\beta(y,z) \mid y \in X\}$$

'*Maximal value to execute first* α*, then* β '

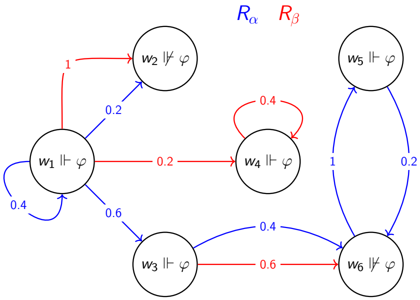
Many-valued PDL, labelled accessibility



$$R_{\varphi?}(x,x) = [\![\varphi]\!](x) \qquad R_{\varphi?}(x,y) = 0 \text{ for } x \neq y$$

'*Test φ at state x* '

Two-valued PDL, labelled accessibility

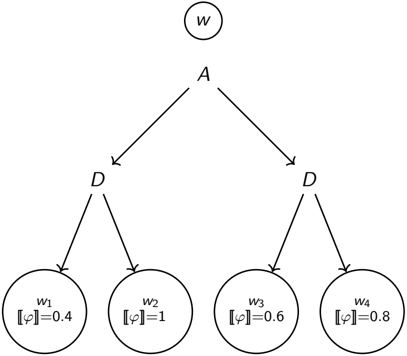


$$w_1 \Vdash \langle \alpha \rangle^{0.4} \varphi \qquad w_1 \nVdash \langle \beta \rangle^{0.6} \varphi$$

'The maximal chance to reach φ after executing α is ≥ 0.4'.
'All chances to reach φ after executing β are < 0.6'.

Many-valued Game Logic

Angel (A) and Demon (D) play the game α at state w.



$N_\alpha(w) = \{\{w_1, w_2\}, \{w_3, w_4\}\}\uparrow$ (Strategies of A)

$\nu_\alpha(w)(\llbracket\varphi\rrbracket) = \bigvee_{S \in N_\alpha(w)} \bigwedge_{z \in S} \llbracket\varphi\rrbracket(z) = 0.6$ (Best φ which A can ensure)

[Parikh, 1983] - Two-valued Game Logic

Game Logic

Game constructs:

$\alpha \cup \beta$	'*Angel chooses whether they play α or β*'
$\alpha;\beta$	'*First α is played, then β is played*'
α^*	'*Angel chooses how many (possibly zero) times α is played*'
α^∂	'*Game α with roles of Angel and Demon interchanged*'
φ?	'*If φ is false, Angel immediately loses*'

Coalgebras

Transition-systems are based on F-**coalgebras** $\gamma\colon X \to \mathsf{F}X$

Coalgebras

Transition-systems are based on F-**coalgebras** $\gamma \colon X \to \mathsf{F}X$

Examples (Coalgebras)

- Kripke frame is $\gamma \colon X \to \mathcal{P}(X) = \mathbf{2}^X$

Coalgebras

Transition-systems are based on F-**coalgebras** $\gamma\colon X \to \mathsf{F}X$

Examples (Coalgebras)

- Kripke frame is $\gamma\colon X \to \mathcal{P}(X) = \mathbf{2}^X$
- **A**-labelled frame is $\gamma\colon X \to \mathcal{P}_{\mathbf{A}}X = \mathbf{A}^X$

Coalgebras

Transition-systems are based on F-**coalgebras** $\gamma\colon X \to \mathsf{F}X$

Examples (Coalgebras)

- Kripke frame is $\gamma\colon X \to \mathcal{P}(X) = \mathbf{2}^X$
- **A**-labelled frame is $\gamma\colon X \to \mathcal{P}_{\mathbf{A}}X = \mathbf{A}^X$
- Monotone nbhd frame is $\gamma\colon X \to \mathcal{M}X = \{N \in \mathbf{2}^{\mathbf{2}^X} \text{ mon.}\}$

Coalgebras

Transition-systems are based on F-**coalgebras** $\gamma\colon X \to \mathsf{F}X$

Examples (Coalgebras)

- Kripke frame is $\gamma\colon X \to \mathcal{P}(X) = \mathbf{2}^X$
- **A**-labelled frame is $\gamma\colon X \to \mathcal{P}_{\mathbf{A}}X = \mathbf{A}^X$
- Monotone nbhd frame is $\gamma\colon X \to \mathcal{M}X = \{N \in \mathbf{2}^{\mathbf{2}^X} \text{ mon.}\}$
- Mon. '**A**-nbhd' frame is $\gamma\colon X \to \mathcal{M}_{\mathbf{A}}(X) = \{\nu \in \mathbf{A}^{\mathbf{A}^X} \text{ mon.}\}$

Predicate Liftings

Transition systems are based on F-coalgebras $\gamma\colon X \to \mathsf{F}X$
Modalities correspond to **predicate liftings** $\lambda^{\heartsuit}_X\colon A^X \to A^{\mathsf{F}X}$

$$[\![\heartsuit\varphi]\!] = \lambda^{\heartsuit}_X([\![\varphi]\!]) \circ \gamma$$

Predicate Liftings

Transition systems are based on F-coalgebras $\gamma \colon X \to \mathsf{F}X$
Modalities correspond to **predicate liftings** $\lambda^{\heartsuit}_X \colon A^X \to A^{\mathsf{F}X}$

$$[\![\heartsuit\varphi]\!] = \lambda^{\heartsuit}_X([\![\varphi]\!]) \circ \gamma$$

Examples (Predicate liftings)

- Modal logic $[\![\Diamond\varphi]\!] = \{x \mid \gamma(x) \cap [\![\varphi]\!] \neq \varnothing\}$

Predicate Liftings

Transition systems are based on F-coalgebras $\gamma\colon X \to \mathsf{F}X$
Modalities correspond to **predicate liftings** $\lambda_X^{\heartsuit}\colon A^X \to A^{\mathsf{F}X}$

$$[\![\heartsuit\varphi]\!] = \lambda_X^{\heartsuit}([\![\varphi]\!]) \circ \gamma$$

Examples (Predicate liftings)

- Modal logic $[\![\Diamond\varphi]\!] = \{x \mid \gamma(x) \cap [\![\varphi]\!] \neq \varnothing\}$
- **A**-valued modal logic $[\![\Diamond\varphi]\!](x) = \bigvee_{y \in X} \gamma(x)(y) \odot [\![\varphi]\!](y)$

Predicate Liftings

Transition systems are based on F-coalgebras $\gamma \colon X \to \mathsf{F}X$
Modalities correspond to **predicate liftings** $\lambda^{\heartsuit}_X \colon A^X \to A^{\mathsf{F}X}$

$$[\![\heartsuit\varphi]\!] = \lambda^{\heartsuit}_X([\![\varphi]\!]) \circ \gamma$$

Examples (Predicate liftings)

- Modal logic $[\![\Diamond\varphi]\!] = \{x \mid \gamma(x) \cap [\![\varphi]\!] \neq \varnothing\}$
- **A**-valued modal logic $[\![\Diamond\varphi]\!](x) = \bigvee_{y \in X} \gamma(x)(y) \odot [\![\varphi]\!](y)$
- Monotone nbhd logic $[\![\heartsuit\varphi]\!] = \{x \mid [\![\varphi]\!] \in \gamma(x)\}$

Predicate Liftings

Transition systems are based on F-coalgebras $\gamma\colon X \to \mathsf{F}X$
Modalities correspond to **predicate liftings** $\lambda^{\heartsuit}_X\colon A^X \to A^{\mathsf{F}X}$

$$[\![\heartsuit\varphi]\!] = \lambda^{\heartsuit}_X([\![\varphi]\!]) \circ \gamma$$

Examples (Predicate liftings)

- Modal logic $[\![\Diamond\varphi]\!] = \{x \mid \gamma(x) \cap [\![\varphi]\!] \neq \varnothing\}$
- **A**-valued modal logic $[\![\Diamond\varphi]\!](x) = \bigvee_{y\in X} \gamma(x)(y) \odot [\![\varphi]\!](y)$
- Monotone nbhd logic $[\![\heartsuit\varphi]\!] = \{x \mid [\![\varphi]\!] \in \gamma(x)\}$
- Monotone **A**-nbhd logic $[\![\heartsuit\varphi]\!](x) = \gamma(x)([\![\varphi]\!])$

Coalgebra Operations

Informal Definition (Coalgebra operation)

An n-ary *coalgebra operation* is an assignment

$$\vec{\gamma} = (\gamma_1, \ldots, \gamma_n) \colon X \to (\mathsf{F}X)^n \quad \mapsto \quad O(\vec{\gamma}) \colon X \to \mathsf{F}X.$$

Coalgebra Operations

Informal Definition (Coalgebra operation)

An *n*-ary *coalgebra operation* is an assignment

$$\vec{\gamma} = (\gamma_1, \ldots, \gamma_n) \colon X \to (\mathsf{F}X)^n \quad \mapsto \quad O(\vec{\gamma}) \colon X \to \mathsf{F}X.$$

For example in PDL:

$$\begin{aligned}
(\gamma_\alpha, \gamma_\beta) \colon X \to (\mathcal{P}X)^2 &\quad \mapsto \quad \gamma_{\alpha \cup \beta} \colon X \to \mathcal{P}X \\
(\gamma_\alpha, \gamma_\beta) \colon X \to (\mathcal{P}X)^2 &\quad \mapsto \quad \gamma_{\alpha ; \beta} \colon X \to \mathcal{P}X \\
\gamma_\alpha \colon X \to \mathcal{P}X &\quad \mapsto \quad \gamma_{\alpha^*} \colon X \to \mathcal{P}X
\end{aligned}$$

Coalgebra Operations

Informal Definition (Coalgebra operation)

An *n*-ary *coalgebra operation* is an assignment

$$\vec{\gamma} = (\gamma_1, \ldots, \gamma_n) \colon X \to (\mathsf{F}X)^n \quad \mapsto \quad O(\vec{\gamma}) \colon X \to \mathsf{F}X.$$

For example in PDL:

$$\begin{aligned} (\gamma_\alpha, \gamma_\beta) \colon X \to (\mathcal{P}X)^2 &\quad \mapsto \quad \gamma_{\alpha \cup \beta} \colon X \to \mathcal{P}X \\ (\gamma_\alpha, \gamma_\beta) \colon X \to (\mathcal{P}X)^2 &\quad \mapsto \quad \gamma_{\alpha ; \beta} \colon X \to \mathcal{P}X \\ \gamma_\alpha \colon X \to \mathcal{P}X &\quad \mapsto \quad \gamma_{\alpha^*} \colon X \to \mathcal{P}X \end{aligned}$$

In **A**-valued Game Logic:

$$\gamma_\alpha \colon X \to \mathcal{M}_{\mathbf{A}} \quad \mapsto \quad \gamma_{\alpha^\partial} \colon X \to \mathcal{M}_{\mathbf{A}}$$

Coalgebra Operations

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation

$$O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$$

Coalgebra Operations

Examples (Induced Operations)

A collection of maps $\theta_X : (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation

$$O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$$

E.g., $\cup, \cap \colon \mathcal{P}^2 \Rightarrow \mathcal{P}$ $\quad \vee, \wedge, \odot \colon \mathcal{P}_{\mathbf{A}}^2 \Rightarrow \mathcal{P}_{\mathbf{A}}$ $\quad (\cdot)^\partial \colon \mathcal{M}_{\mathbf{A}} \Rightarrow \mathcal{M}_{\mathbf{A}}$

Coalgebra Operations

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation

$$O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$$

E.g., $\cup, \cap \colon \mathcal{P}^2 \Rightarrow \mathcal{P}$ $\quad \vee, \wedge, \odot \colon \mathcal{P}_{\mathbf{A}}^2 \Rightarrow \mathcal{P}_{\mathbf{A}}$ $\quad (\cdot)^\partial \colon \mathcal{M}_{\mathbf{A}} \Rightarrow \mathcal{M}_{\mathbf{A}}$

Examples (Sequential Composition and Iteration)

Collection of maps $\mu_X \colon \mathsf{FF}X \to \mathsf{F}X$ yields binary coalgebra operation

$$\gamma_1 * \gamma_2 \colon X \xrightarrow{\gamma_1} \mathsf{F}X \xrightarrow{\mathsf{F}\gamma_2} \mathsf{FF}X \xrightarrow{\mu_X} \mathsf{F}X$$

Coalgebra Operations

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation

$$O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$$

E.g., $\cup, \cap \colon \mathcal{P}^2 \Rightarrow \mathcal{P}$ $\quad \vee, \wedge, \odot \colon \mathcal{P}_{\mathbf{A}}^2 \Rightarrow \mathcal{P}_{\mathbf{A}}$ $\quad (\cdot)^\partial \colon \mathcal{M}_{\mathbf{A}} \Rightarrow \mathcal{M}_{\mathbf{A}}$

Examples (Sequential Composition and Iteration)

Collection of maps $\mu_X \colon \mathsf{FF}X \to \mathsf{F}X$ yields binary coalgebra operation

$$\gamma_1 * \gamma_2 \colon X \xrightarrow{\gamma_1} \mathsf{F}X \xrightarrow{\mathsf{F}\gamma_2} \mathsf{FF}X \xrightarrow{\mu_X} \mathsf{F}X$$

With $\eta_X \colon X \to \mathsf{F}X$ and $\bigsqcup_X \colon \mathcal{P}(\mathsf{F}X) \to \mathsf{F}X$ yields unary operation

$$\gamma^* = \bigsqcup_{k \in \omega} \gamma^{[k]} \quad \text{where } \gamma^{[0]} = \eta_X \text{ and } \gamma^{[k+1]} = \gamma^{[k]} * \gamma$$

Tests

Informal Definition (Test)

A *test* is an assignment

$$[\![\varphi]\!]\colon X \to A \quad \mapsto \quad \mathrm{test}([\![\varphi]\!])\colon X \to \mathsf{F}X$$

Tests

Informal Definition (Test)

A *test* is an assignment

$$[\![\varphi]\!]\colon X \to A \quad \mapsto \quad \mathsf{test}([\![\varphi]\!])\colon X \to \mathsf{F}X$$

For example in PDL:

$$[\![\varphi]\!]\colon X \to 2 \quad \mapsto \quad \gamma_{\varphi?}\colon X \to \mathcal{P}X$$

Tests

Informal Definition (Test)

A *test* is an assignment

$$[\![\varphi]\!] \colon X \to A \quad \mapsto \quad \mathsf{test}([\![\varphi]\!]) \colon X \to \mathsf{F}X$$

For example in PDL:

$$[\![\varphi]\!] \colon X \to 2 \quad \mapsto \quad \gamma_{\varphi?} \colon X \to \mathcal{P}X$$

In **A**-valued Game Logic:

$$[\![\varphi]\!] \colon X \to A \quad \mapsto \quad \gamma_{\varphi?} \colon X \to \mathcal{M}_{\mathbf{A}}X$$

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

- Algebra of *truth-degrees* **A**
  (Determines A-valued *propositional base*)

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

- Algebra of *truth-degrees* $\mathbf{A}$
  (Determines A-valued *propositional base*)
- Coalgebraic *signature functor* $\mathsf{F}\colon \mathsf{Set} \to \mathsf{Set}$
  (Determines type of *transition structures*)

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

- Algebra of *truth-degrees* **A**
  (Determines A-valued *propositional base*)
- Coalgebraic *signature functor* $\mathsf{F}\colon \mathsf{Set} \to \mathsf{Set}$
  (Determines type of *transition structures*)
- Countable collection of *predicate liftings* Λ
  (Determines interpretations of *modalities*)

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

- Algebra of *truth-degrees* **A**
  (Determines A-valued *propositional base*)
- Coalgebraic *signature functor* $\mathsf{F}\colon \mathsf{Set} \to \mathsf{Set}$
  (Determines type of *transition structures*)
- Countable collection of *predicate liftings* Λ
  (Determines interpretations of *modalities*)
- Finite collection of *coalgebra operations* Op
  (E.g. choice, sequential composition, iteration, dual, intersection ...)

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

- Algebra of *truth-degrees* $\mathbf{A}$
  (Determines A-valued *propositional base*)
- Coalgebraic *signature functor* $\mathsf{F}\colon \mathsf{Set} \to \mathsf{Set}$
  (Determines type of *transition structures*)
- Countable collection of *predicate liftings* Λ
  (Determines interpretations of *modalities*)
- Finite collection of *coalgebra operations* Op
  (E.g. choice, sequential composition, iteration, dual, intersection ...)
- Finite collection of *tests* Te
  (Nothing to add here)

Recipe: Coalgebraic Dynamic Logic

A highly parametrical dish for the whole logical family

Ingredients.

- Algebra of *truth-degrees* $\mathbf{A}$
  (Determines A-valued *propositional base*)
- Coalgebraic *signature functor* $\mathsf{F}\colon \mathsf{Set} \to \mathsf{Set}$
  (Determines type of *transition structures*)
- Countable collection of *predicate liftings* Λ
  (Determines interpretations of *modalities*)
- Finite collection of *coalgebra operations* Op
  (E.g. choice, sequential composition, iteration, dual, intersection ...)
- Finite collection of *tests* Te
  (Nothing to add here)

Generalises framework for Coalgebraic Dynamic Logic of [Hansen et al., 2014, Hansen and Kupke, 2015].

(Enjoy with or without monads)

Safety

A core feature of PDL [van Benthem, 1998]:

If $\mathfrak{M}$ and $\mathfrak{M}'$ are PDL models and $B \subseteq X \times X'$ is a bisimulation for all *atomic* programs, then B is a bisimulation for *all* programs.

Safety

A core feature of PDL [van Benthem, 1998]:

If $\mathfrak{M}$ and $\mathfrak{M}'$ are PDL models and $B \subseteq X \times X'$ is a bisimulation for all *atomic* programs, then B is a bisimulation for *all* programs.

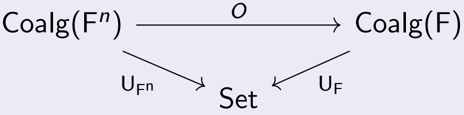
Smilarly for Game Logic [Pauly, 2000]:

If $\mathfrak{M}$ and $\mathfrak{M}'$ are game models and $B \subseteq X \times X'$ is a bisimulation for all *atomic* games, then B is a bisimulation for *all* games.

Safe Coalgebra Operation

Definition (Safe Coalgebra Operation)

A *safe coalgebra operation* is a functor $O\colon \mathrm{Coalg}(\mathsf{F}^n) \to \mathrm{Coalg}(\mathsf{F})$ for which the following commutes.



Safe Coalgebra Operation

Definition (Safe Coalgebra Operation)

A *safe coalgebra operation* is a functor $O\colon \mathsf{Coalg}(\mathsf{F}^n) \to \mathsf{Coalg}(\mathsf{F})$ for which the following commutes.

$$\begin{array}{ccccc} \mathsf{Coalg}(\mathsf{F}^n) & & \xrightarrow{\quad O \quad} & & \mathsf{Coalg}(\mathsf{F}) \\ & {\scriptstyle \mathsf{U}_{\mathsf{F}^n}} \searrow & & \swarrow {\scriptstyle \mathsf{U}_{\mathsf{F}}} & \\ & & \mathsf{Set} & & \end{array}$$

$$\begin{array}{ccc} (\gamma_1, \gamma_2, \ldots, \gamma_n)\colon X \to (\mathsf{F}X)^n & \mapsto & O(\vec{\gamma})\colon X \to \mathsf{F}X \\ f \big\downarrow \;\; f \big\downarrow \;\; \cdots \;\; f \big\downarrow & \Longrightarrow & f \big\downarrow \\ (\gamma_1', \gamma_2' \ldots, \gamma_n')\colon Y \to (\mathsf{F}Y)^n & \mapsto & O(\vec{\gamma'})\colon Y \to \mathsf{F}Y \end{array}$$

Safe Coalgebra Operations and Naturality

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation

$O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$

Safe Coalgebra Operations and Naturality

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation $O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$

O^θ is safe if and only if $\theta \colon \mathsf{F}^n \Rightarrow \mathsf{F}$ is a natural transformation.

Safe Coalgebra Operations and Naturality

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation
$O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$

O^θ is safe if and only if $\theta \colon \mathsf{F}^n \Rightarrow \mathsf{F}$ is a natural transformation.

Examples (Sequential Composition and Iteration)

Collection of maps $\mu_X \colon \mathsf{FF}X \to \mathsf{F}X$ yields binary coalgebra operation
$\gamma_1 * \gamma_2 \colon X \xrightarrow{\gamma_1} \mathsf{F}X \xrightarrow{\mathsf{F}\gamma_2} \mathsf{FF}X \xrightarrow{\mu_X} \mathsf{F}X$
With $\eta_X \colon X \to \mathsf{F}X$ and $\bigsqcup_X \colon \mathcal{P}(\mathsf{F}X) \to \mathsf{F}X$ yields unary operation
$\gamma^* = \bigsqcup_{k \in \omega} \gamma^{[k]}$ where $\gamma^{[0]} = \eta_X$ and $\gamma^{[k+1]} = \gamma^{[k]} * \gamma$

Safe Coalgebra Operations and Naturality

Examples (Induced Operations)

A collection of maps $\theta_X \colon (\mathsf{F}X)^n \to \mathsf{F}X$ *induces* coalgebra operation $O^\theta(\vec{\gamma}) \colon X \xrightarrow{\vec{\gamma}} (\mathsf{F}X)^n \xrightarrow{\theta_X} \mathsf{F}X$

O^θ is safe if and only if $\theta \colon \mathsf{F}^n \Rightarrow \mathsf{F}$ is a natural transformation.

Examples (Sequential Composition and Iteration)

Collection of maps $\mu_X \colon \mathsf{FF}X \to \mathsf{F}X$ yields binary coalgebra operation $\gamma_1 * \gamma_2 \colon X \xrightarrow{\gamma_1} \mathsf{F}X \xrightarrow{\mathsf{F}\gamma_2} \mathsf{FF}X \xrightarrow{\mu_X} \mathsf{F}X$

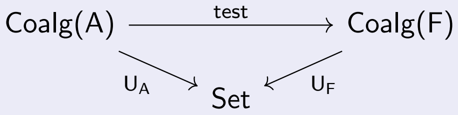
With $\eta_X \colon X \to \mathsf{F}X$ and $\bigsqcup_X \colon \mathcal{P}(\mathsf{F}X) \to \mathsf{F}X$ yields unary operation $\gamma^* = \bigsqcup_{k \in \omega} \gamma^{[k]}$ where $\gamma^{[0]} = \eta_X$ and $\gamma^{[k+1]} = \gamma^{[k]} * \gamma$

Composition $*$ is safe if $\mu \colon \mathsf{F} \circ \mathsf{F} \Rightarrow \mathsf{F}$ is a natural transformation. Iteration $(\cdot)^*$ is safe if $\eta \colon \mathrm{id} \Rightarrow \mathsf{F}$ and $\bigsqcup \colon \mathcal{P} \circ \mathsf{F} \Rightarrow \mathsf{F}$ are also natural.

Safe Tests and Naturality

Definition (Safe Test)

A *safe test* is a functor $\mathsf{test}\colon \mathsf{Coalg}(\mathsf{A}) \to \mathsf{Coalg}(\mathsf{F})$ for which the following commutes.



Safe Tests and Naturality

Definition (Safe Test)

A *safe test* is a functor $\mathsf{test}\colon \mathsf{Coalg}(\mathsf{A}) \to \mathsf{Coalg}(\mathsf{F})$ for which the following commutes.

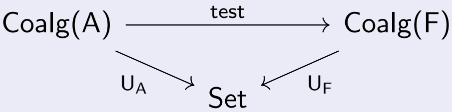
$$\begin{array}{ccc} \mathsf{Coalg}(\mathsf{A}) & \xrightarrow{\mathsf{test}} & \mathsf{Coalg}(\mathsf{F}) \\ & {\scriptstyle \mathsf{U_A}} \searrow \quad \swarrow {\scriptstyle \mathsf{U_F}} & \\ & \mathsf{Set} & \end{array}$$

A test is safe if and only if it is a natural transformation $\mathsf{id} \Rightarrow \mathsf{F}(\text{-})^{A^{(\text{-})}}$.

Safe Tests and Naturality

Definition (Safe Test)

A *safe test* is a functor $\mathsf{test}\colon \mathsf{Coalg}(\mathsf{A}) \to \mathsf{Coalg}(\mathsf{F})$ for which the following commutes.



A test is safe if and only if it is a natural transformation $\mathsf{id} \Rightarrow \mathsf{F}(\text{-})^{A^{(\text{-})}}$.

Examples

All the tests discussed in this talk are safe.

Safety is nice

Proposition

Let all operations and tests be safe. If $\mathfrak{M}$ and $\mathfrak{M}'$ are coalgebraic dynamic models and $B \subseteq X \times X'$ is a bisimulation for all *atomic* actions, then B is a bisimulation for *all* actions.

Reducible Operations & Tests (Examples)

A-valued PDL:

$$\langle\alpha;\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi, \quad \langle\alpha \vee \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi \quad \langle\psi?\rangle\varphi \leftrightarrow \psi \odot \varphi$$

Reducible Operations & Tests (Examples)

A-valued PDL:

$$\langle \alpha;\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi, \quad \langle \alpha \vee \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi \quad \langle\psi?\rangle\varphi \leftrightarrow \psi \odot \varphi$$

2-valued PDL, A-labelled accessibility (finite linear A):

$$\langle \alpha;\beta\rangle^{r}\varphi \leftrightarrow \bigvee_{r_1 \odot r_2 \geq r} \langle\alpha\rangle^{r_1}\langle\beta\rangle^{r_2}\varphi \quad \langle\psi?\rangle^{r}\varphi \leftrightarrow \psi \wedge \varphi$$

Reducible Operations & Tests (Examples)

A-valued PDL:

$$\langle \alpha;\beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \langle \beta \rangle \varphi, \quad \langle \alpha \vee \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi \quad \langle \psi ? \rangle \varphi \leftrightarrow \psi \odot \varphi$$

2-valued PDL, A-labelled accessibility (finite linear A):

$$\langle \alpha;\beta \rangle^{r} \varphi \leftrightarrow \bigvee_{r_1 \odot r_2 \geq r} \langle \alpha \rangle^{r_1} \langle \beta \rangle^{r_2} \varphi \quad \langle \psi ? \rangle^{r} \varphi \leftrightarrow \psi \wedge \varphi$$

A-valued Game Logic:

$$\langle \alpha^{\partial} \rangle \leftrightarrow \neg \langle \alpha \rangle \neg \varphi \quad \langle \psi ? \rangle \varphi \leftrightarrow \psi \odot \varphi$$

Reducibility is nice(r)

Proposition (Reducible $\Rightarrow$ Safe)

If collection of predicate liftings Λ is *separating*:

1. Every reducible coalgebra operation is safe.
2. Every reducible test is safe.

Axiomatizations

Let *every* $O \in \mathsf{Op}$ and test $\in$ Te be reducible.
Let $\mathcal{L}_{\mathsf{F}}$ be the *underlying coalgebraic logic* for (F, Λ).

Axiomatizations

Let *every* $O \in \mathsf{Op}$ and test $\in \mathsf{Te}$ be reducible.
Let $\mathcal{L}_\mathsf{F}$ be the *underlying coalgebraic logic* for (F, Λ).

Definition. *Dynamic coalgebraic logic* $\mathcal{L}_\mathsf{F}^{\mathsf{Dyn}}$ is obtained as follows.

- Take $\mathcal{L}_\mathsf{F}$ instantiated at every $\alpha \in \mathsf{Act}$.
- Add all instances of reduction axioms for coalgebra operations.
- Add all instances of reduction axioms for tests.

Example: Iteration-free $\mathbf{A}$-valued Game Logic

For *finite* $\mathbf{A}$ combine:

- $\mathbf{A}$-valued propositional logic (*e.g.*, finite Łukasiewicz or Gödel)

Example: Iteration-free A-valued Game Logic

For *finite* **A** combine:

- **A**-valued propositional logic (*e.g.*, finite Łukasiewicz or Gödel)
- Coalgebraic logic $\mathcal{L}_{\mathcal{M}_{\mathbf{A}}}$, *e.g.*,

$$\langle\alpha\rangle p \to \langle\alpha\rangle(p \vee q) \text{ for all actions } \alpha.$$

Example: Iteration-free A-valued Game Logic

For *finite* **A** combine:

- **A**-valued propositional logic (*e.g.*, finite Łukasiewicz or Gödel)
- Coalgebraic logic $\mathcal{L}_{\mathcal{M}_{\mathbf{A}}}$, *e.g.*,

$$\langle \alpha \rangle p \to \langle \alpha \rangle (p \vee q) \text{ for all actions } \alpha.$$

- Reduction axioms

$$\langle \alpha ; \beta \rangle p \leftrightarrow \langle \alpha \rangle \langle \beta \rangle p$$

$$\langle \alpha \vee \beta \rangle p \leftrightarrow \langle \alpha \rangle p \vee \langle \beta \rangle p$$

$$\langle \alpha \wedge \beta \rangle p \leftrightarrow \langle \alpha \rangle p \wedge \langle \beta \rangle p$$

$$\langle \alpha^{\partial} \rangle p \leftrightarrow \neg \langle \alpha \rangle \neg p$$

$$\langle \varphi ? \rangle p \leftrightarrow (\varphi \odot p)$$

Strong completeness (Sketch)

Canonical models: Replace maximally consistent sets with *non-modal homomorphisms* [Bou, et al., 2011]

Strong completeness (Sketch)

Canonical models: Replace maximally consistent sets with *non-modal homomorphisms* [Bou, et al., 2011]

Theorem

Given a canonical model on *atomic actions*, it stays canonical on *all actions* (and this suffices to obtain strong completeness).

Idea: *Induction on the structure of reduction axioms.*

Strong completeness (Sketch)

Canonical models: Replace maximally consistent sets with *non-modal homomorphisms* [Bou, et al., 2011]

Theorem

Given a canonical model on *atomic actions*, it stays canonical on *all actions* (and this suffices to obtain strong completeness).

Idea: *Induction on the structure of reduction axioms*.

Theorem

For **A** finite, Λ separating, F *weakly preserving inverse limits of surjective ω-cochains*, $\mathcal{L}_{\mathsf{F}}$ *strongly one-step complete over finite sets*: There exists a canonical model on atomic actions.

Idea: Similarly to [Schröder and Pattinson, 2009], approach set of all formulas as inverse limit of finite sets of formulas, then apply one-step completeness of $\mathcal{L}_{\mathsf{F}}$.

Conclusion

Presented a coalgebraic framework for many-valued dynamic logics.

- Parametric in $\mathsf{FL}_{\mathrm{ew}}$-algebra $\mathbf{A}$ of truth-degrees.

Conclusion

Presented a coalgebraic framework for many-valued dynamic logics.

- Parametric in $\mathsf{FL}_{\mathrm{ew}}$-algebra $\mathbf{A}$ of truth-degrees.
- Extends [Hansen et al., 2014] (which requires monad).
    - *E.g.* includes Instantial PDL [van Benthem et al., 2019] ($\mathsf{F} = \mathcal{P}\mathcal{P}$ not monad).

Conclusion

Presented a coalgebraic framework for many-valued dynamic logics.

- Parametric in $\mathsf{FL}_{\mathrm{ew}}$-algebra $\mathbf{A}$ of truth-degrees.
- Extends [Hansen et al., 2014] (which requires monad).
    - *E.g.* includes Instantial PDL [van Benthem et al., 2019] ($\mathsf{F} = \mathcal{P}\mathcal{P}$ not monad).
- Coalgebraic definitions of safety and reducibility.

Conclusion

Presented a coalgebraic framework for many-valued dynamic logics.

- Parametric in $\mathsf{FL}_{\mathrm{ew}}$-algebra **A** of truth-degrees.
- Extends [Hansen et al., 2014] (which requires monad).
  - *E.g.* includes Instantial PDL [van Benthem et al., 2019] ($\mathsf{F} = \mathcal{P}\mathcal{P}$ not monad).
- Coalgebraic definitions of safety and reducibility.
- Coalgebraic strong completeness for iteration-free logics with reducible operations and tests.
  - *E.g.* new instances for iteration-free **A**-valued game logic, two-valued PDL with **A**-labelled accessibility.

Future Research

1. Aim for (weak) completeness including *non-reducible operations*, in particular *including iteration constructs*.
    - Reducibility $\rightsquigarrow$ General definition of *Fischer-Ladner closure*?
    - Strong completeness proof $\rightsquigarrow$ Canonical model (unfiltered)

Future Research

1. Aim for (weak) completeness including *non-reducible operations*, in particular *including iteration constructs*.
   - Reducibility $\rightsquigarrow$ General definition of *Fischer-Ladner closure*?
   - Strong completeness proof $\rightsquigarrow$ Canonical model (unfiltered)
2. Find a basis for all *safe first-order definable* coalgebra operations (as for PDL [van Benthem, 1998], Game Logic [Pauly, 2000]).

Future Research

1. Aim for (weak) completeness including *non-reducible operations*, in particular *including iteration constructs*.
   - Reducibility $\rightsquigarrow$ General definition of *Fischer-Ladner closure*?
   - Strong completeness proof $\rightsquigarrow$ Canonical model (unfiltered)
2. Find a basis for all *safe first-order definable* coalgebra operations (as for PDL [van Benthem, 1998], Game Logic [Pauly, 2000]).
3. Extend completeness proofs to *infinite* $\mathbf{A}$?

Future Research

1. Aim for (weak) completeness including *non-reducible operations*, in particular *including iteration constructs*.
    - Reducibility $\rightsquigarrow$ General definition of *Fischer-Ladner closure*?
    - Strong completeness proof $\rightsquigarrow$ Canonical model (unfiltered)
2. Find a basis for all *safe first-order definable* coalgebra operations (as for PDL [van Benthem, 1998], Game Logic [Pauly, 2000]).
3. Extend completeness proofs to *infinite* $\mathbf{A}$?
4. Other base categories than Set, for example measurable spaces/quantale-enriched/poset-enriched/...

Future Research

1. Aim for (weak) completeness including *non-reducible operations*, in particular *including iteration constructs*.
   - Reducibility $\rightsquigarrow$ General definition of *Fischer-Ladner closure*?
   - Strong completeness proof $\rightsquigarrow$ Canonical model (unfiltered)
2. Find a basis for all *safe first-order definable* coalgebra operations (as for PDL [van Benthem, 1998], Game Logic [Pauly, 2000]).
3. Extend completeness proofs to *infinite* **A**?
4. Other base categories than Set, for example measurable spaces/quantale-enriched/poset-enriched/...

Thanks!

More information found in the papers:
Conference version here. Extended version here.

References I

Bílková, M. and Dostál, M. (2016).
Expressivity of many-valued modal logics, coalgebraically.
In Väänänen, J., Hirvonen, Å., and de Queiroz, R., editors, *Logic, Language, Information, and Computation (WoLLIC'16)*, pages 109–124. Springer.
doi:10.1007/978-3-662-52921-8_8

Bou, F., Esteva, F., Godo, L., and Rodríguez, R. (2011).
On the minimum many-valued modal logic over a finite residuated lattice.
Journal of Logic and Computation, 21(5):739–790.
doi:10.1093/logcom/exp062

Fischer, M. J. and Ladner, R. E. (1979).
Propositional dynamic logic of regular programs.
Journal of Computer and System Sciences, 18(2):194–211.
doi:10.1016/0022-0000(79)90046-1

References II

Hansen, H. and Kupke, C. (2015).
Weak Completeness of Coalgebraic Dynamic Logics.
In Matthes, R. and Mio, M., editors, *Tenth International Workshop on Fixed Points in Computer Science (FICS'15)*, volume 191 of *EPTCS*, pages 90–104.
doi:10.4204/EPTCS.191.9

Hansen, H., Kupke, C., and Leal, R. (2014).
Strong completeness for iteration-free coalgebraic dynamic logics.
In Díaz, J., Lanese, I., and Sangiorgi, D., editors, *8th International Conference on Theoretical Computer Science (TCS'14)*, pages 281–295. Springer.
doi:10.1007/978-3-662-44602-7_22

Kurz, A., Poiger, W., and Teheux, B. (2024).
Many-valued coalgebraic logic over semi-primal varieties
Logical Methods in Computer Science, 2(3): 6:1–6:31.
doi:10.46298/lmcs-20(3:6)2024

References III

Lin, C.-Y. and Liau, C.-J. (2023).
Many-valued coalgebraic modal logic: One-step completeness and finite model property.
Fuzzy Sets and Systems, 467:article no. 108564.
doi:10.1016/j.fss.2023.108564

Parikh, R. (1983).
Propositional game logic.
In *24th Annual Symposium on Foundations of Computer Science*, pages 195–200. IEEE Computer Society.
doi:10.1109/SFCS.1983.47

Pauly, M. (2000).
From programs to games: Invariance and safety for bisimulation.
In Clote, P. and Schwichtenberg, H., editors, *Computer Science Logic*, volume 1862 of *Lecture Notes in Computer Science*, pages 485–496. Springer.
doi:10.1007/3-540-44622-2_33

References IV

Schröder, L. and Pattinson, D. (2009).
Strong completeness of coalgebraic modal logics.
In Albers, S. and Marion, J.-Y., editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS'09)*, volume 3 of *Leibniz International Proceedings in Informatics*, pages 673–684.

Sedlàr, I. (2020).
Finitely-valued Propositional Dynamic Logic.
In Olivetti, N., Verbrugge, R., Negri, S., and Sandu, G., editors, *Advances in Modal Logic (AiML'20)*, volume 13, pages 561–579. College Publications.

van Benthem, J. (1998).
Program constructions that are safe for bisimulation.
Studia Logica, 60:311–330.
doi:10.1023/A:1005072201319

van Benthem, J., Bezhanishvili, N., and Enqvist, S. (2019).
A propositional dynamic logic for instantial neighborhood semantics.
Studia Logica, 107(4):719–751.
doi:10.1007/s11225-018-9825-5

„Biography of Fake News with a Touch of AI: Dangerous Phenomenon through the Prism of Modern Human Sciences" project no.: CZ.02.01.01/00/23_025/0008724 via the Operational Programme Jan Amos Komenský.

Co-funded by the European Union

MŠMT
MINISTRY OF EDUCATION, YOUTH AND SPORTS