Coalgebraic Dynamic Logics

Clemens Kupke
University of Strathclyde, Glasgow

joint work with Helle Hvid Hansen (RU Groningen)

MSP101, 18 March 2021

Overview

- Preliminaries:
  - dynamic logics
  - coalgebraic logics
- coalgebraic dynamic logics
  - Syntax & Semantics
  - axiomatization
- iteration-free coalgebraic PDL: strong completeness
- Main result:

  weak completeness for coalgebraic dynamic logics

Why this talk?

- non-deterministic doctrines (and variants) from the MSP201
- quantitative equational theories
- games
- the selection monad

Part 1.1: Dynamic Logics

Motivation

- modal logics: versatile family of logics that allow to reason about state-based dynamical systems
- "robustly" decidable, e.g. adding recursion (fixpoint operators) to modal logic to reason about the ongoing, infinite behaviour of a system is possible (but "costly")
- dynamic logics offer balance between expressivity (limited recursion) and efficiency (tractable MC)

Propositional Dynamic Logic (PDL)

Fischer & Ladner, 1977. Reason about program correctness.

$[\alpha]\varphi$ "after all successful executions of program α, φ holds"

Propositional Dynamic Logic (PDL)

Fischer & Ladner, 1977. Reason about program correctness.

$[\alpha]\varphi$ "after all successful executions of program α, φ holds"

- Syntax: formulas φ $::= \; p \in P_0 \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\alpha\rangle\varphi$

  programs $\alpha \in A$ $::= \; a \in A_0 \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi?$

  composition (;), choice ($\cup$), iteration (*), tests (φ?)

Propositional Dynamic Logic (PDL)

Fischer & Ladner, 1977. Reason about program correctness.

$[\alpha]\varphi$ "after all successful executions of program α, φ holds"

- Syntax: formulas $\varphi \quad ::= \quad p \in P_0 \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\alpha\rangle\varphi$

  programs $\alpha \in A \quad ::= \quad a \in A_0 \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi?$

  composition (;), choice ($\cup$), iteration (*), tests (φ?)
- Multi-modal Kripke semantics: $M = (X, \{R_\alpha \mid \alpha \in A\}, V)$ where X is state space,
  - $R_\alpha : X \to \mathcal{P}(X)$ (relation, nondeterministic programs),
  - $V : P_0 \to \mathcal{P}(X)$ is a valuation.

$$M, x \models [\alpha]\varphi \quad \text{iff} \quad \forall y \in X.\, xR_\alpha y \to M, y \models \varphi.$$

Standard PDL Models

- Def. $M = (X, \{R_\alpha \mid \alpha \in A\}, V)$ is standard if

$$\begin{aligned} R_{\alpha;\beta} &= R_\alpha \circ R_\beta \text{ (relation composition)} \\ R_{\alpha\cup\beta} &= R_\alpha \cup R_\beta \\ R_{\alpha^*} &= R_\alpha^* \text{ (reflexive, transitive closure)} \\ R_{\varphi?} &= \{(x, x) \mid x \in [\![\varphi]\!]\} \end{aligned}$$

Standard PDL Models

- Def. $M = (X, \{R_\alpha \mid \alpha \in A\}, V)$ is standard if

$$\begin{array}{rcl} R_{\alpha;\beta} & = & R_\alpha \circ R_\beta \ \text{(relation composition)} \\ R_{\alpha\cup\beta} & = & R_\alpha \cup R_\beta \\ R_{\alpha^*} & = & R_\alpha^* \ \text{(reflexive, transitive closure)} \\ R_{\varphi?} & = & \{(x,x) \mid x \in [\![\varphi]\!]\} \end{array}$$

- Sound and (weakly) complete axiomatisation of standard models [Kozen & Parikh 1981]:

  PDL = Normal modal logic **K** (ML of Kripke frames) plus:

$$[\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \qquad [\alpha \cup \beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi$$

$$[\psi?]\varphi \leftrightarrow (\psi \rightarrow \varphi)$$

$$\varphi \wedge [\alpha][\alpha^*]\varphi \leftrightarrow [\alpha^*]\varphi \qquad \varphi \wedge [\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow [\alpha^*]\varphi$$

Game Logic (GL)

Parikh, 1985. Strategic ability in determined 2-player games.

$\langle\gamma\rangle\varphi$ "player 1 has strategy in γ to ensure outcome satisfies φ"

("player 1 is effective for φ")

Game Logic (GL)

Parikh, 1985. Strategic ability in determined 2-player games.

$\langle\gamma\rangle\varphi$ "player 1 has strategy in γ to ensure outcome satisfies φ"

("player 1 is effective for φ")

- Syntax: PDL syntax extended with dual operation on games:
  - $\gamma_1;\gamma_2$: play γ_1 then γ_2,
  - $\gamma_1 \cup \gamma_2$: player 1 chooses to play γ_1 or γ_2,
  - γ^*: player 1 chooses when to stop.
  - γ^d: players switch roles.

Game Logic (GL)

Parikh, 1985. Strategic ability in determined 2-player games.

$\langle\gamma\rangle\varphi$ "player 1 has strategy in γ to ensure outcome satisfies φ"

("player 1 is effective for φ")

- Syntax: PDL syntax extended with dual operation on games:
  - $\gamma_1;\gamma_2$: play γ_1 then γ_2,
  - $\gamma_1 \cup \gamma_2$: player 1 chooses to play γ_1 or γ_2,
  - γ^*: player 1 chooses when to stop.
  - γ^d: players switch roles.
- Semantics: Game model $M = (X, \{E_\gamma \mid \gamma \in \Gamma\}, V)$ where $E_\gamma : X \to \mathcal{PP}(X)$ is monotonic neighbourhood function:

  If $U \in E_\gamma(x)$ and $U \subseteq U'$ then $U' \in E_\gamma(x)$.

  $U \in E_\gamma(x)$ iff player 1 is effective for U in γ starting in x.

  Modal semantics: $M, x \models \langle\gamma\rangle\varphi \quad \text{iff} \quad [\![\varphi]\!] \in E_\gamma(x)$

Standard GL Models

- Standard GL model: similar to PDL notion,

$$U \in E_{\gamma^d}(x) \text{ iff } X \setminus U \notin E_\gamma(x).$$

Standard GL Models

- Standard GL model: similar to PDL notion,

$$U \in E_{\gamma^d}(x) \text{ iff } X \setminus U \notin E_\gamma(x).$$

- **GL** = monotonic modal logic **M** (ML of mon. nbhd. frames) plus

$$\langle\gamma;\delta\rangle\varphi \leftrightarrow \langle\gamma\rangle\langle\delta\rangle\varphi \qquad \langle\gamma\cup\delta\rangle\varphi \leftrightarrow \langle\gamma\rangle\varphi \vee \langle\delta\rangle\varphi$$

$$\langle\psi?\rangle\varphi \leftrightarrow (\psi \wedge \varphi) \qquad \langle\gamma^d\rangle\varphi \leftrightarrow \neg\langle\gamma\rangle\neg\varphi$$

$$\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi \to \langle\gamma^*\rangle\varphi \qquad \frac{\varphi \vee \langle\gamma\rangle\varphi \to \psi}{\langle\gamma^*\rangle\varphi \to \psi}$$

Standard GL Models

- Standard GL model: similar to PDL notion,

$$U \in E_{\gamma^d}(x) \text{ iff } X \setminus U \notin E_\gamma(x).$$

- **GL** = monotonic modal logic **M** (ML of mon. nbhd. frames) plus

$$\langle\gamma;\delta\rangle\varphi \leftrightarrow \langle\gamma\rangle\langle\delta\rangle\varphi \qquad \langle\gamma\cup\delta\rangle\varphi \leftrightarrow \langle\gamma\rangle\varphi \vee \langle\delta\rangle\varphi$$

$$\langle\psi?\rangle\varphi \leftrightarrow (\psi \wedge \varphi) \qquad \langle\gamma^d\rangle\varphi \leftrightarrow \neg\langle\gamma\rangle\neg\varphi$$

$$\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi \to \langle\gamma^*\rangle\varphi \qquad \frac{\varphi \vee \langle\gamma\rangle\varphi \to \psi}{\langle\gamma^*\rangle\varphi \to \psi}$$

- Without dual: sound and (weakly) complete [Parikh 1985].
- Without iteration: sound and strongly complete [Pauly 2001].
- Completeness of full GL [Enqvist, Hansen, K, Marti, Venema 2019]

Basic observation:

- $\mathcal{P}$ is monad $(\mathcal{P}, \eta, \mu)$ with:

  $\eta_X(x) = \{x\}, \quad \mu_X(\{U_i \mid i \in I\}) = \bigcup_{i \in I} U_i.$
- $\mathcal{M}$ is a monad $(\mathcal{M}, \eta, \mu)$ with:

  $$\begin{array}{rcl} \eta_X(x) & = & \{U \subseteq X \mid x \in U\} \\ \mu_X(W) & = & \{U \subseteq X \mid \eta_{\mathcal{P}(X)}(U) \in W\} \end{array}$$
- Composition of programs and games is Kleisli composition.

Basic observation:

- $\mathcal{P}$ is monad $(\mathcal{P}, \eta, \mu)$ with:
  $\eta_X(x) = \{x\}, \quad \mu_X(\{U_i \mid i \in I\}) = \bigcup_{i \in I} U_i$.
- $\mathcal{M}$ is a monad $(\mathcal{M}, \eta, \mu)$ with:
  $$\begin{array}{lcl} \eta_X(x) & = & \{U \subseteq X \,|\, x \in U\} \\ \mu_X(W) & = & \{U \subseteq X \mid \eta_{\mathcal{P}(X)}(U) \in W\} \end{array}$$
- Composition of programs and games is Kleisli composition.

Basic setup:

- Action/program $X \to TX$ where T a Set-monad
  (T describes computation type, side-effects, ...)
- Sequential composition as Kleisli composition $*_T$.
- Multi-program setting: $X \to (TX)^A$ where A is a set of program labels.

Part 1.2: Coalgebraic Logics (in 4 slides)

Coalgebraic Modal Logic & PDL

- Observation: Kripke models are $\mathcal{P}$-coalgebras, ie, pairs (X, γ) with

$$\gamma : X \to \mathcal{P}X$$

- in this logical context X is usually a set (or some concrete category)
- Idea: Develop modal logic for T-coalgebras, where T is an endofunctor. Development should be parametric in T.

Coalgebraic Logic: Syntax

Given a collection of modal operators Λ and a set P_0 of propositional variables.

Definition

The set $\mathcal{F}(\Lambda)$ of formulas over Λ is defined a follows:

$$\mathcal{F}(\Lambda) \ni \varphi ::= p \in P_0 \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \heartsuit\varphi, \heartsuit \in \Lambda$$

Note

In this talk the (basic) similarity type will consist of one unary modality only!

Coalgebraic Logic: Semantics

In order to be able to interpret modal formulas we need

- a set functor T
- for every modal operator $\heartsuit \in \Lambda$ a natural transformation

$$\heartsuit : P \longrightarrow PT,$$

  where P denotes the contravariant power set functor.

Formulas are then interpreted over T-models (X, γ, V) consisting of

$$\gamma : X \longrightarrow TX \qquad \text{and} \qquad V : \mathsf{Var} \longrightarrow \mathcal{P}(X).$$

$$\begin{array}{rcll} [\![p]\!] & = & V(p) & \text{for } p \in \mathsf{Var} \\ & \vdots & & \\ [\![\heartsuit\varphi]\!] & = & P\gamma(\heartsuit([\![\varphi]\!])) = \gamma^{-1}(\heartsuit([\![\varphi]\!])) & \end{array}$$

Equivalently

$\heartsuit : P \longrightarrow PT$ is in one-to-one correspondence to

- $\widehat{\heartsuit} : T \longrightarrow P^{\mathrm{op}}P$ (T-coalgebras to neighbourhood frames)

$$x \models \heartsuit\varphi \quad \text{iff} \quad [\![\varphi]\!] \in (\widehat{\heartsuit} \circ \gamma)(x).$$

- $\check{\heartsuit} : T2 \longrightarrow 2$ ("allowed 0-1 patterns")

$$\begin{array}{ccccc} X & \xrightarrow{\chi_{[\![\varphi]\!]}} & 2 & & \\ {\scriptstyle \gamma}\downarrow & & & & \\ T(X) & \xrightarrow{T(\chi_{[\![\varphi]\!]})} & T(2) & \xrightarrow{\check{\heartsuit}} & 2 \end{array}$$

$$(X, \gamma, V), x \models \heartsuit\varphi \quad \text{iff} \quad \check{\heartsuit}(T(\chi_{[\![\varphi]\!]})(\gamma(x)) = 1.$$

Examples

- $T = \mathcal{P}$, $\heartsuit = \Box$:

$$\begin{aligned} \heartsuit(U) &= \{V \subseteq X \mid V \subseteq U\}, \\ \widehat{\heartsuit}(V) &= \{U \subseteq X \mid V \subseteq U\} \text{ and} \\ \breve{\heartsuit}(V \subseteq \mathcal{P}2) &= 1 \quad \text{iff} \quad 0 \notin V \end{aligned}$$

- $T = \mathcal{M}$, $\heartsuit = \Box$:

$$\begin{aligned} \heartsuit(U) &= \{N \in \mathcal{M}X \mid U \in N\} \\ \widehat{\heartsuit}(N) &= N \\ \breve{\heartsuit}(N \in \mathcal{M}2) &= 1 \quad \text{iff} \quad 1 \in N \end{aligned}$$

⋮

Overview articles

- Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, Yde Venema: Modal Logics are Coalgebraic. The Computer Journal (2011)

- CK, Dirk Pattinson: Coalgebraic semantics of modal logics: An overview. (2011)

Part 2.1: Coalgebraic PDL - Syntax and Semantics

Coalgebra-Algebra

Two perspectives:

$$\frac{\xi \colon X \to (TX)^A \quad T^A\text{-coalgebra, modal logic}}{\widehat{\xi} \colon A \to (TX)^X \quad \text{algebra homomorphism, program operations}}$$

Questions:

- What are "program" operations like $\cup$ and ^{d}?
- What is a standard model?
- Which compositionality axioms?
- How to prove soundness and completeness?

Dynamic Syntax

Given

- Σ, a signature (functor).
- P_0, a countable set of atomic propositions.
- A_0, a countable set of atomic programs.

we define

$$\begin{aligned} \text{formulas } \mathcal{F}(P_0, A_0, \Sigma) \ni \varphi \quad &::= \quad p \in P_0 \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\alpha\rangle\varphi \\ \text{programs } A(P_0, A_0, \Sigma) \ni \alpha \quad &::= \quad a \in A_0 \mid \alpha;\alpha \mid \sigma(\alpha_1, \ldots, \alpha_n) \\ &\qquad \mid ?\varphi \mid \alpha^* \end{aligned}$$

where $\sigma \in \Sigma$ is n-ary.

Pointwise Program Operations via Natural Operations

- An n-ary natural operation on T is a natural transformation $\sigma \colon T^n \longrightarrow T$

Pointwise Program Operations via Natural Operations

- An n-ary natural operation on T is a natural transformation $\sigma\colon T^n \longrightarrow T$
- $\sigma\colon T^n \longrightarrow T$ yields pointwise operation on $(TX)^X$, e.g.,

$$\sigma_X^X(\gamma_1, \gamma_2)(x) = \sigma_X(\gamma_1(x), \gamma_2(x))$$

Pointwise Program Operations via Natural Operations

- An n-ary natural operation on T is a natural transformation $\sigma\colon T^n \longrightarrow T$

- $\sigma\colon T^n \longrightarrow T$ yields pointwise operation on $(TX)^X$, e.g.,

$$\sigma_X^X(\gamma_1, \gamma_2)(x) = \sigma_X(\gamma_1(x), \gamma_2(x))$$

- Given finitary signature functor Σ,
  a natural Σ-algebra is natural transformation $\theta\colon \Sigma T \longrightarrow T$,
  and yields pointwise Σ-algebra $\theta_X^X\colon \Sigma((TX)^X) \longrightarrow (TX)^X$.

Natural and Pointwise Operations: Examples

Natural operations on $\mathcal{P}$:

- Union $\cup \colon \mathcal{P} \times \mathcal{P} \longrightarrow \mathcal{P}$ is a natural operation, since

  $$f[U \cup U'] = f[U] \cup f[U'] \quad (\mathcal{P}f(U) = f[U])$$

  The pointwise extension of $\cup \colon \mathcal{P} \times \mathcal{P} \longrightarrow \mathcal{P}$ is union of relations $(R_1 \cup R_2)(x) = R_1(x) \cup R_2(x)$.
- Observation: Intersection and complement are not natural operations on $\mathcal{P}$.

Natural and Pointwise Operations: Examples

Natural operations on $\mathcal{P}$:

- Union $\cup\colon \mathcal{P} \times \mathcal{P} \longrightarrow \mathcal{P}$ is a natural operation, since

$$f[U \cup U'] = f[U] \cup f[U'] \quad (\mathcal{P}f(U) = f[U])$$

  The pointwise extension of $\cup\colon \mathcal{P} \times \mathcal{P} \longrightarrow \mathcal{P}$ is union of relations $(R_1 \cup R_2)(x) = R_1(x) \cup R_2(x)$.
- Observation: Intersection and complement are not natural operations on $\mathcal{P}$.

Natural operations on $\mathcal{M}$:

- $\cup$ and $\cap$ (since preserved by f^{-1}).
- Dual operation ${}^d\colon \mathcal{M} \longrightarrow \mathcal{M}$ where for all $N \in \mathcal{M}(X)$, and $U \subseteq X$, $U \in N^d$ iff $X \setminus U \notin N$.

  Dual game operation is the pointwise extension.

Standard dynamic models

A (P_0, A_0, θ)-*dynamic* T-*model* $\mathfrak{M} = (X, \gamma_0, \heartsuit, V)$ consists of

- a set X,
- an interpretation of atomic actions $\widehat{\gamma}_0\colon A_0 \longrightarrow (TX)^X$,
- a unary predicate lifting $\heartsuit\colon P \longrightarrow P \circ T$ whose transpose $\widehat{\heartsuit}\colon T \longrightarrow P^{\mathrm{op}} P$ is a monad morphism, and
- a valuation $V\colon P_0 \longrightarrow \mathcal{P}(X)$.

Semantics

Let $A = \Sigma \cup \{;\} \cup \{*\}$-terms over A_0. We define the truth set $[\![\varphi]\!]^{\mathfrak{M}}$ of dynamic formulas and the semantics $\widehat{\gamma}\colon A \longrightarrow (TX)^X$ of complex actions in $\mathfrak{M}$ by mutual induction:

$$[\![p]\!]^{\mathfrak{M}} = V(p), \quad [\![\varphi \wedge \psi]\!]^{\mathfrak{M}} = [\![\varphi]\!]^{\mathfrak{M}} \cap [\![\psi]\!]^{\mathfrak{M}}, \quad [\![\neg\varphi]\!]^{\mathfrak{M}} = X \setminus [\![\varphi]\!]^{\mathfrak{M}},$$

$$[\![\langle\alpha\rangle\varphi]\!]^{\mathfrak{M}} = (\widehat{\gamma}(\alpha)^{-1} \circ \heartsuit_X)([\![\varphi]\!]^{\mathfrak{M}})$$

$$\begin{array}{lcll}
\widehat{\gamma}(\underline{\sigma}(\alpha_1, \ldots, \sigma_n)) & = & \sigma_X^X(\widehat{\gamma}(\alpha_1), \ldots, \widehat{\gamma}(\alpha_n)) & \\
\widehat{\gamma}(\alpha; \beta) & = & \widehat{\gamma}(\alpha) * \widehat{\gamma}(\beta) & \text{(Kleisli composition)} \\
\widehat{\gamma}(\varphi?)(x) & = & ? & \\
\widehat{\gamma}(\alpha^*) & = & \widehat{\gamma}(\alpha)^* & \text{(Kleisli iteration)}
\end{array}$$

(red parts later)

Standardness as a property of a T^A-coalgebra

Some terminology:

- Given natural algebra $\theta\colon \Sigma T \longrightarrow T$ then $\gamma\colon X \longrightarrow (TX)^A$ is θ-standard iff

$$\widehat{\gamma}\colon A \longrightarrow (TX)^X \quad \text{is a} \quad \Sigma\text{-algebra homomorphism.}$$

- If T is a monad, then $\gamma\colon X \longrightarrow (TX)^A$ is ;-standard iff

$$\text{for all } \alpha, \beta \in A, \quad \widehat{\gamma}(\alpha;\beta) = \widehat{\gamma}(\alpha) * \widehat{\gamma}(\beta).$$

Part II of this talk will discuss the axiomatisation in detail.

Conclusions

- generic completeness result for dynamic logics (PDL, dual-free GL)
- currently not enough examples: $\mathcal{P}/\mathcal{M}/\mathcal{F}$
- need extend to a quantitative setting
- model-checking rather than completeness?
- automata (partial result: automata for game logic)
- What about logics for doctrines? Other game/strategy logics?