◆□▶ ◆□▶ ▲□▶ ▲□▶ □ のQ@

Dynamic Coalgebraic Modalities

Raul Andres Leal¹ & Helle Hvid Hansen²

¹ILLC Universiteit van Amsterdam

²Eindhoven University of Technology, Centrum Wiskunde & Informatica, Amsterdam

CMCS 10, 2010 Cyprus

▲□▶ ▲□▶ ▲□▶ ▲□▶ = 三 のへで









The dark side of the moon







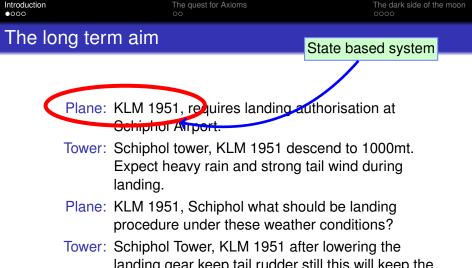


◆□▶ ◆□▶ ◆豆▶ ◆豆▶ □ のへで

(ロ) (同) (三) (三) (三) (○) (○)

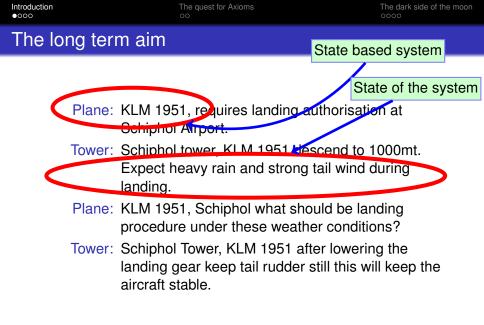
The long term aim

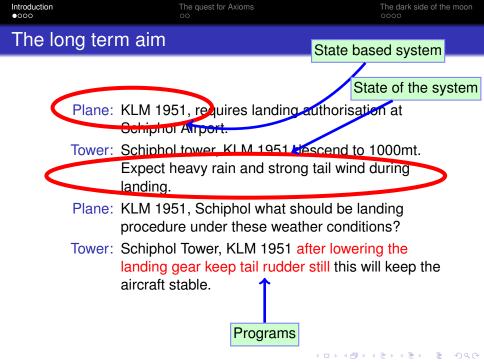
- Plane: KLM 1951, requires landing authorisation at Schiphol Airport.
- Tower: Schiphol tower, KLM 1951 descend to 1000mt. Expect heavy rain and strong tail wind during landing.
- Plane: KLM 1951, Schiphol what should be landing procedure under these weather conditions?
- Tower: Schiphol Tower, KLM 1951 after lowering the landing gear keep tail rudder still this will keep the aircraft stable.

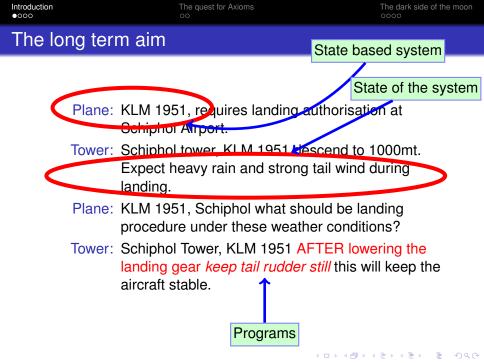


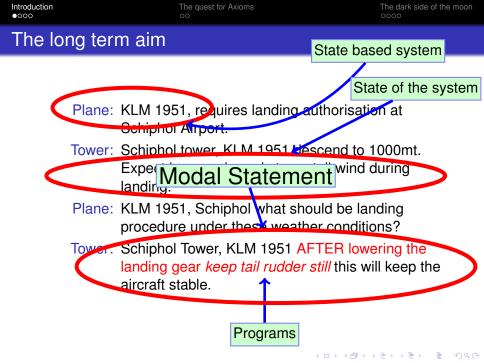
landing gear keep tail rudder still this will keep the aircraft stable.

◆□▶ ◆□▶ ▲□▶ ▲□▶ □ のQ@



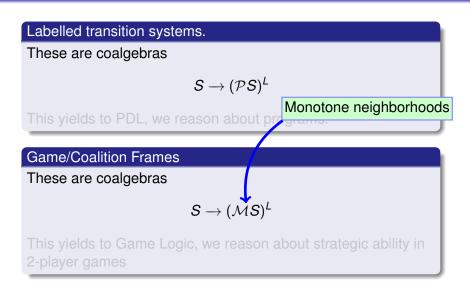






The dark side of the moon

From Planes to Kripke Frames



The dark side of the moon

From Planes to Kripke Frames

Labelled transition systems.

These are coalgebras

$$\mathcal{S} \to (\mathcal{P}\mathcal{S})^L$$

This yields to PDL, we reason about programs.

Game/Coalition Frames

These are coalgebras

$$\mathcal{S}
ightarrow (\mathcal{M}\mathcal{S})^L$$

This yields to Game Logic, we reason about strategic ability in 2-player games

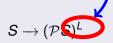
The dark side of the moon

From Planes to Kripke Frames

(free) algebra of regular expressions

Labelled transition systems.

These are coalgebras



This yields to PDL, we reason about programs.

Game/Coalition Frames

These are coalgebras

$$\mathcal{S}
ightarrow (\mathcal{M}\mathcal{S})^L$$

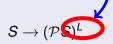
This yields to Game Logic, we reason about strategic ability in 2-player games

From Planes to Kripke Frames

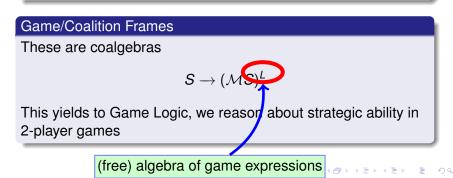
(free) algebra of regular expressions

Labelled transition systems.

These are coalgebras



This yields to PDL, we reason about programs.



The dark side of the moon

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

Double perspective

Algebraic Perspective

$$\sigma: L \to (GS)^S$$

Structure + Dynamics

Coalgebraic Perspective

$$\widehat{\sigma}: S \to (GS)^L$$

Behavior + Modalities

The dark side of the moon

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

Double perspective

Algebraic Perspective

$$\sigma: L \to (GS)^S$$

Structure + Dynamics

Coalgebraic Perspective

$$\widehat{\sigma}: S \to (GS)^L$$

Behavior + Modalities

▲□▶ ▲□▶ ▲三▶ ▲三▶ - 三 - のへで

Dynamic Modalities

Intuition

 $\boldsymbol{s} \Vdash \lambda^{\alpha} \varphi$ means "in state \boldsymbol{s} , after α , φ holds".

PDL

 $s \Vdash \Box^{\alpha} \varphi$ means "in state *s*, after transition α , φ holds".

Game Logic

 $s \Vdash \Diamond^{\alpha} \varphi$ means "in state *s*, player 1 has a strategy in game α to bring about φ ".

Dynamic Modalities

Intuition

 $\boldsymbol{s} \Vdash \lambda^{\alpha} \varphi$ means "in state \boldsymbol{s} , after α , φ holds".

PDL

 $\boldsymbol{s} \Vdash \Box^{\alpha} \varphi$ means "in state \boldsymbol{s} , after transition α , φ holds".

Game Logic

 $s \Vdash \Diamond^{\alpha} \varphi$ means "in state *s*, player 1 has a strategy in game α to bring about φ ".

◆□▶ ◆□▶ ◆□▶ ◆□▶ ● ● ● ●

Dynamic Modalities

Intuition

 $\boldsymbol{s} \Vdash \lambda^{\alpha} \varphi$ means "in state \boldsymbol{s} , after α , φ holds".

PDL

 $\boldsymbol{s} \Vdash \Box^{\alpha} \varphi$ means "in state \boldsymbol{s} , after transition α, φ holds".

Game Logic

 $s \Vdash \Diamond^{\alpha} \varphi$ means "in state *s*, player 1 has a strategy in game α to bring about φ ".

Dynamic Modalities

Intuition

 $s \Vdash \lambda^{\alpha} \varphi$ means "in state *s*, after α , φ holds".

Labelling

Given a predicate lifting $\lambda : Q \to QG$ and $\alpha \in L$, the α labelling of λ is a predicate lifting

$$\lambda^{\alpha}: \mathcal{Q} \to \mathcal{Q}G^{L}$$

given by

$$\lambda^{lpha}(\textit{\textit{U}}) = \{\delta \in \textit{\textit{G}}(\textit{\textit{S}})^{\textit{L}} \mid \delta(lpha) \in \lambda(\textit{\textit{U}})\}$$

・ロト・四ト・モート ヨー うへの











PDL

Take $\Box = \lambda$, why does

$$\lambda^{\alpha;\beta}\varphi \iff \lambda^\alpha\lambda^\beta\varphi$$

hold?

Predicate transformers

Given
$$\sigma: S \to (GS)^L$$
 consider

$$([\alpha]^{\sigma}) \quad \mathcal{QS} \xrightarrow{\lambda_{\mathcal{S}}^{\alpha}} \mathcal{Q}(GS)^{L} \xrightarrow{\sigma^{-1}} \mathcal{QS}$$

$$[\alpha;\beta]^{\sigma} = [\alpha]^{\sigma} \circ [\beta]^{\sigma}$$

PDL

Take $\Box = \lambda$, why does

$[\alpha;\beta]\varphi\iff [\alpha][\beta]\varphi$

hold?

Predicate transformers

Given
$$\sigma: S \to (GS)^L$$
 consider

$$([\alpha]^{\sigma}) \quad \mathcal{QS} \xrightarrow{\lambda_{\mathcal{S}}^{\alpha}} \mathcal{Q}(GS)^{L} \xrightarrow{\sigma^{-1}} \mathcal{QS}$$

$$[\alpha;\beta]^{\sigma} = [\alpha]^{\sigma} \circ [\beta]^{\sigma}$$

PDL

Take $\Box = \lambda$, why does

$$\lambda^{\alpha;\beta}\varphi \iff \lambda^\alpha\lambda^\beta\varphi$$

hold?

Predicate transformers

Given
$$\sigma: S \to (GS)^L$$
 consider

$$([\alpha]^{\sigma}) \quad \mathcal{QS} \xrightarrow{\lambda_{\mathcal{S}}^{\alpha}} \mathcal{Q}(GS)^{L} \xrightarrow{\sigma^{-1}} \mathcal{QS}$$

$$[\alpha;\beta]^{\sigma} = [\alpha]^{\sigma} \circ [\beta]^{\sigma}$$

PDL

Take $\Box = \lambda$, why does

$$\lambda^{\alpha;\beta}\varphi \iff \lambda^{\alpha}\lambda^{\beta}\varphi$$

hold?

Predicate transformers

Given
$$\sigma: S \to (GS)^L$$
 consider

$$([\alpha]^{\sigma}) \quad \mathcal{QS} \xrightarrow{\lambda_{\mathcal{S}}^{\alpha}} \mathcal{Q}(GS)^{L} \xrightarrow{\sigma^{-1}} \mathcal{QS}$$

$$[\alpha;\beta]^{\sigma} = [\alpha]^{\sigma} \circ [\beta]^{\sigma}$$

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

$$\lambda^{lpha;eta}\varphi\iff\lambda^{lpha}\lambda^{eta}\varphi$$

- the transpose $\widehat{\lambda}$: $\mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: G2 \rightarrow 2 is a G-algebra (monads).

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

 $[\alpha;\beta]^{\sigma} = [\alpha]^{\sigma} \circ [\beta]^{\sigma}$

- the transpose $\widehat{\lambda}$: $\mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: G2 \rightarrow 2 is a G-algebra (monads).

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

$$\lambda^{lpha;eta}\varphi\iff\lambda^{lpha}\lambda^{eta}\varphi$$

- the transpose $\widehat{\lambda}$: $\mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: G2 \rightarrow 2 is a G-algebra (monads).

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

$$\lambda^{lpha;eta}\varphi\iff\lambda^{lpha}\lambda^{eta}\varphi$$

- the transpose $\widehat{\lambda} : \mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: $G2 \rightarrow 2$ is a G-algebra (monads).

◆□▶ ◆□▶ ◆□▶ ◆□▶ ● ● ● ●

Monads for composition

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

$$\lambda^{lpha;eta}\varphi\iff\lambda^{lpha}\lambda^{eta}\varphi$$

holds if one of the following conditions hold

- the transpose $\widehat{\lambda} : \mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: G2 \rightarrow 2 is a G-algebra (monads).

Yoneda

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

$$\lambda^{lpha;eta}\varphi\iff\lambda^{lpha}\lambda^{eta}\varphi$$

- the transpose $\widehat{\lambda} : \mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: G2 \rightarrow 2 is a G-algebra (monads).

Theorem

Let λ be a predicate lifting. If sequential composition is interpreted as Kleisli composition, then

$$\lambda^{lpha;eta}\varphi\iff\lambda^{lpha}\lambda^{eta}\varphi$$

- the transpose $\widehat{\lambda} : \mathbf{G} \to \mathcal{Q}\mathcal{Q}$ is a monad morphism.
- The algebra $Y(\lambda)$: G2 \rightarrow 2 is a G-algebra (monads).









▲口▶▲圖▶▲≣▶▲≣▶ ■ のみの

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 のへで

Towards Planes; more complex operations

Other Operations

How do we obtain axioms like

$$\lambda^{\alpha \cup \beta} \varphi \iff \lambda^{\alpha} \varphi \wedge \lambda^{\beta} \varphi?$$

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

Towards Planes; more complex operations

Other Operations

How do we obtain axioms like

$$\lambda^{\alpha \cup \beta} \varphi \iff \lambda^{\alpha} \varphi \wedge \lambda^{\beta} \varphi?$$

Answer 1: There is an enriched functor

$$\widehat{\lambda} \circ - : \mathcal{K}(\mathcal{G}) \to \mathcal{K}(\mathcal{Q}\mathcal{Q})$$

◆□▶ ◆□▶ ◆□▶ ◆□▶ ● ● ● ●

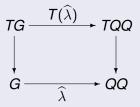
Towards Planes; more complex operations

Other Operations

How do we obtain axioms like

$$\lambda^{lpha\cupeta}\varphi\iff\lambda^{lpha}arpha\wedge\lambda^{eta}arphi?$$

Answer 2: $\widehat{\lambda}$ is a homomorphism, i.e. a diagram like



commutes.

◆□▶ ◆□▶ ◆□▶ ◆□▶ ● ● ● ●

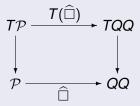
Towards Planes; more complex operations

Other Operations

How do we obtain axioms like

$$\lambda^{\alpha\cup\beta}\varphi\iff\lambda^{lpha}\varphi\wedge\lambda^{eta}\varphi?$$

Answer 2: in PDL... $\widehat{\Box}$ is a homomorphism.



commutes.

◆□▶ ◆□▶ ◆□▶ ◆□▶ ● ● ● ●

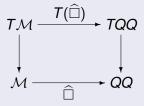
Towards Planes; more complex operations

Other Operations

How do we obtain axioms like

$$\lambda^{\alpha\cup\beta}\varphi\iff\lambda^{lpha}\varphi\wedge\lambda^{eta}\varphi$$
?

Answer 2: in Game Logic... $\widehat{\Box}$ is NOT a homomorphism.



commutes.

◆□▶ ◆□▶ ◆□▶ ◆□▶ ● ● ● ●

Other issues

Input/output

We do not understand how to deal with input/output (functors that are not monads)

(Java)
$$F(S) := (1 + S \times B + S \times E)^A$$

Idea: Use

 $J(B) := (1 + S \times B + S \times E)^S$

which is a monad.

Problem: Actions are subject to typing conditions.

◆□▶ ◆□▶ ▲□▶ ▲□▶ ■ ののの

Other issues

Input/output

We do not understand how to deal with input/output (functors that are not monads)

(Java)
$$F(S) := (1 + S \times B + S \times E)^A$$

Idea: Use

$$J(B) := (1 + S \times B + S \times E)^S$$

which is a monad.

Problem: Actions are subject to typing conditions.

◆□▶ ◆□▶ ▲□▶ ▲□▶ ■ ののの

Other issues

Input/output

We do not understand how to deal with input/output (functors that are not monads)

(Java)
$$F(S) := (1 + S \times B + S \times E)^A$$

Idea: Use

$$J(B) := (1 + S \times B + S \times E)^S$$

which is a monad.

Problem: Actions are subject to typing conditions.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

Other issues

Definability

We can now define operations on the label even if they make "no sense" for the coalgebra; e.g.

$$\lambda^{\alpha \cup \beta} = \lambda^{\alpha} \cup \lambda^{\beta}.$$

When are those definable and what do they express is unclear to us.

◆□▶ ◆□▶ ▲□▶ ▲□▶ ■ ののの

The End!!!

- We understand how to label modalities.
- We can explain the axiom of sequential composition.
- We can explain axioms for algebraic operations.
- We can not see any bialgebra.
- The general picture is still unclear.
- The Test modality is still evasive.
- Input/output should be worked out.

◆□▶ ◆□▶ ▲□▶ ▲□▶ ■ ののの

The End!!!

- We understand how to label modalities.
- We can explain the axiom of sequential composition.
- We can explain axioms for algebraic operations.
- We can not see any bialgebra.
- The general picture is still unclear.
- The Test modality is still evasive.
- Input/output should be worked out.