Completeness for coalgebraic fixpoint logic

Fatemeh Seifan
Joint with S. Enqvist and Y. Venema

Institute for Logic, Language and Computation
University of Amsterdam

CSL2016

Standard μ-calculus

Standard μ-calculus

Syntax:

$$\varphi ::= \top \mid \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \neg\varphi \mid \mu.p\varphi' \mid \nu p.\varphi'$$

(provided that all occurrences of p in φ' are positive)

Standard μ-calculus

Syntax:

$$\varphi ::= \top \mid \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \neg\varphi \mid \mu.p\varphi' \mid \nu p.\varphi'$$

(provided that all occurrences of p in φ' are positive)

Semantics:

Standard μ-calculus

Syntax:

$$\varphi ::= \top \mid \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \neg\varphi \mid \mu.p\varphi' \mid \nu p.\varphi'$$

(provided that all occurrences of p in φ' are positive)

Semantics:

$$\begin{aligned} [\![\mu p.\varphi]\!]^{\mathbb{S},V} &:= \bigcap\{U \in \mathcal{P}S \mid [\![\varphi]\!]^{\mathbb{S}[x \mapsto U]} \subseteq U\} \\ [\![\nu p.\varphi]\!]^{\mathbb{S},V} &:= \bigcup\{U \in \mathcal{P}S \mid U \subseteq [\![\varphi]\!]^{\mathbb{S}[x \mapsto U]}\} \end{aligned}$$

Completeness

Kozen Axiomatisation:

- complete calculus for modal logic
- $\varphi(\mu p.\varphi) \vdash_{\mathbf{K}} \mu p.\varphi$
- if $\varphi(\psi) \vdash_{\mathbf{K}} \psi$ then $\mu p.\varphi \vdash_{\mathbf{K}} \psi$

Completeness

Kozen Axiomatisation:

- complete calculus for modal logic
- $\varphi(\mu p.\varphi) \vdash_{\mathbf{K}} \mu p.\varphi$
- if $\varphi(\psi) \vdash_{\mathbf{K}} \psi$ then $\mu p.\varphi \vdash_{\mathbf{K}} \psi$
  ($\varphi \vdash_{\mathbf{K}} \psi$ abbreviates $\vdash_{K} \varphi \to \psi$)

Completeness

Kozen Axiomatisation:

- complete calculus for modal logic
- $\varphi(\mu p.\varphi) \vdash_{\mathbf{K}} \mu p.\varphi$
- if $\varphi(\psi) \vdash_{\mathbf{K}} \psi$ then $\mu p.\varphi \vdash_{\mathbf{K}} \psi$
  ($\varphi \vdash_{\mathbf{K}} \psi$ abbreviates $\vdash_K \varphi \to \psi$)

Theorem (Kozen 1983)
$\vdash_{\mathbf{K}}$ is sound and complete for *aconjunctive* formulas.

Completeness

Kozen Axiomatisation:

- complete calculus for modal logic
- $\varphi(\mu p.\varphi) \vdash_{\mathbf{K}} \mu p.\varphi$
- if $\varphi(\psi) \vdash_{\mathbf{K}} \psi$ then $\mu p.\varphi \vdash_{\mathbf{K}} \psi$
  ($\varphi \vdash_{\mathbf{K}} \psi$ abbreviates $\vdash_K \varphi \to \psi$)

Theorem (Kozen 1983)
$\vdash_{\mathbf{K}}$ is sound and complete for *aconjunctive* formulas.

Theorem (Walukiewicz 1995)
$\vdash_{\mathbf{K}}$ is sound and complete for all formulas.

Completeness

Kozen Axiomatisation:

- complete calculus for modal logic
- $\varphi(\mu p.\varphi) \vdash_{\mathbf{K}} \mu p.\varphi$
- if $\varphi(\psi) \vdash_{\mathbf{K}} \psi$ then $\mu p.\varphi \vdash_{\mathbf{K}} \psi$
  ($\varphi \vdash_{\mathbf{K}} \psi$ abbreviates $\vdash_K \varphi \to \psi$)

Theorem (Kozen 1983)
$\vdash_{\mathbf{K}}$ is sound and complete for *aconjunctive* formulas.

Theorem (Walukiewicz 1995)
$\vdash_{\mathbf{K}}$ is sound and complete for all formulas.

Question
How to generalise this to similar logics?

Coalgebraic fixpoint logic

- Given a functor T we define T_ω as:

$$T_\omega X = \bigcup \{ TX' \mid X' \subseteq X,\ X' \text{ is finite } \}.$$

Then for every set X we define the function

$$\begin{array}{rcl} Base_X : T_\omega X & \to & \mathcal{P}_\omega X \\ \alpha & \mapsto & \bigcap \{ X' \subseteq X \mid \alpha \in TX' \}. \end{array}$$

Coalgebraic fixpoint logic

- Given a functor T we define T_ω as:

$$\mathsf{T}_\omega X = \bigcup \{\mathsf{T}X' \mid X' \subseteq X,\ X' \text{ is finite } \}.$$

Then for every set X we define the function

$$\begin{array}{rcl} Base_X : \mathsf{T}_\omega X & \to & \mathcal{P}_\omega X \\ \alpha & \mapsto & \bigcap\{X' \subseteq X \mid \alpha \in \mathsf{T}X'\}. \end{array}$$

Syntax of $\mu\mathtt{ML}_\mathsf{T}$:

Coalgebraic fixpoint logic

- Given a functor T we define T_ω as:

$$\mathsf{T}_\omega X = \bigcup\{\mathsf{T}X' \mid X' \subseteq X,\ \ X' \text{ is finite }\}.$$

Then for every set X we define the function

$$\begin{array}{rcl} Base_X : \mathsf{T}_\omega X & \to & \mathcal{P}_\omega X \\ \alpha & \mapsto & \bigcap\{X' \subseteq X \mid \alpha \in \mathsf{T}X'\}. \end{array}$$

Syntax of $\mu\mathtt{ML}_\mathsf{T}$:

$$\varphi ::= \bot \mid \top \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \nabla\alpha \mid \neg\varphi \mid \mu p.\varphi \mid \nu p.\varphi$$

Coalgebraic fixpoint logic

- Given a functor T we define T_ω as:

$$\mathsf{T}_\omega X = \bigcup\{\mathsf{T}X' \mid X' \subseteq X,\ X' \text{ is finite }\}.$$

Then for every set X we define the function

$$\begin{array}{rcl} Base_X : \mathsf{T}_\omega X & \to & \mathcal{P}_\omega X \\ \alpha & \mapsto & \bigcap\{X' \subseteq X \mid \alpha \in \mathsf{T}X'\}. \end{array}$$

Syntax of $\mu\mathtt{ML}_\mathsf{T}$:

$$\varphi ::= \bot \mid \top \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \nabla\alpha \mid \neg\varphi \mid \mu p.\varphi \mid \nu p.\varphi$$

with $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}_\mathsf{T})$. No occurrence of p in φ may be in the scope of an odd number of negations.

Coalgebraic fixpoint logic

Semantics:

Coalgebraic fixpoint logic

Semantics:

- Given a functor T and a binary relation $R : X_1 \nrightarrow X_2$ we define a relation $\overline{\mathsf{T}}R : \mathsf{T}X_1 \nrightarrow \mathsf{T}X_2$ by:

$$\overline{\mathsf{T}}R := \{((\mathsf{T}\pi_1)\rho, (\mathsf{T}\pi_2)\rho) \mid \rho \in \mathsf{T}R\}.$$

Coalgebraic fixpoint logic

Semantics:

- Given a functor T and a binary relation $R : X_1 \nrightarrow X_2$ we define a relation $\overline{\mathsf{T}}R : \mathsf{T}X_1 \nrightarrow \mathsf{T}X_2$ by:

$$\overline{\mathsf{T}}R := \{((\mathsf{T}\pi_1)\rho, (\mathsf{T}\pi_2)\rho) \mid \rho \in \mathsf{T}R\}.$$

$$\mathsf{T}X_1 \xleftarrow{\mathsf{T}\pi_1} \mathsf{T}R \xrightarrow{\mathsf{T}\pi_2} \mathsf{T}X_2$$

Coalgebraic fixpoint logic

Semantics:

▶ Given a functor T and a binary relation $R : X_1 \nrightarrow X_2$ we define a relation $\overline{\mathsf{T}}R : \mathsf{T}X_1 \nrightarrow \mathsf{T}X_2$ by:

$$\overline{\mathsf{T}}R := \{((\mathsf{T}\pi_1)\rho, (\mathsf{T}\pi_2)\rho) \mid \rho \in \mathsf{T}R\}.$$

$$\mathsf{T}X_1 \xleftarrow{\mathsf{T}\pi_1} \mathsf{T}R \xrightarrow{\mathsf{T}\pi_2} \mathsf{T}X_2$$

▶ Formulas of $\mu\mathtt{ML}_{\mathsf{T}}$ will be interpreted over T-models $\mathbb{S} = (S, \sigma, m)$ with:

- S is a set
- $\sigma : S \to \mathsf{T}S$ is a coalgebra structure
- $m : S \to \mathcal{P}\mathtt{X}$ is a marking.

We sometimes denote (S, σ, m) by (S, σ_m).

Coalgebraic fixpoint logic

Semantics:

▶ Given a functor T and a binary relation $R : X_1 \nrightarrow X_2$ we define a relation $\overline{\mathsf{T}}R : \mathsf{T}X_1 \nrightarrow \mathsf{T}X_2$ by:

$$\overline{\mathsf{T}}R := \{((\mathsf{T}\pi_1)\rho, (\mathsf{T}\pi_2)\rho) \mid \rho \in \mathsf{T}R\}.$$

$$\mathsf{T}X_1 \xleftarrow{\mathsf{T}\pi_1} \mathsf{T}R \xrightarrow{\mathsf{T}\pi_2} \mathsf{T}X_2$$

▶ Formulas of $\mu\mathtt{ML}_\mathsf{T}$ will be interpreted over T-models $\mathbb{S} = (S, \sigma, m)$ with:

- S is a set
- $\sigma : S \to \mathsf{T}S$ is a coalgebra structure
- $m : S \to \mathcal{P}\mathtt{X}$ is a marking.

We sometimes denote (S, σ, m) by (S, σ_m).
The satisfaction relation $\Vdash_m$ is inductively defined:

$$s \Vdash_m \nabla\alpha \text{ iff } (\sigma_m(s), \alpha) \in \overline{\mathsf{T}}(\Vdash_m).$$

Coalgebraic fixpoint logic

Coalgebraic fixpoint logic

Example: The standard μ-calculus

Coalgebraic fixpoint logic

Example: The standard μ-calculus

Set T to be the powerset functor $\mathcal{P}$:

Coalgebraic fixpoint logic

Example: The standard μ-calculus

Set T to be the powerset functor $\mathcal{P}$:

$$\nabla\Phi \equiv \Box \bigvee \Phi \wedge \bigwedge \Diamond\Phi;$$

$$\Diamond\varphi \equiv \nabla\{\varphi, \top\} \text{ and } \Box\varphi \equiv \nabla\emptyset \vee \{\varphi\}.$$

Coalgebraic fixpoint logic

Coalgebraic fixpoint logic

- Axiomatization of $\mu\mathtt{ML}_{\mathsf{T}}$: complete axiomatixation for $\mathtt{ML}_{\mathsf{T}}$ (Kupke, Kurz & Venema 2012) + axioms and rules for fixpoint operators

Coalgebraic fixpoint logic

- Axiomatization of $\mu\mathtt{ML}_{\mathsf{T}}$: complete axiomatixation for $\mathtt{ML}_{\mathsf{T}}$ (Kupke, Kurz & Venema 2012) $+$ axioms and rules for fixpoint operators
- Equivalence of $\mu\mathtt{ML}_{\mathsf{T}}$-formulas and parity T-automata holds (Venema 2006)

Coalgebraic fixpoint logic

- Axiomatization of $\mu\mathtt{ML}_\mathsf{T}$: complete axiomatixation for $\mathtt{ML}_\mathsf{T}$ (Kupke, Kurz & Venema 2012) + axioms and rules for fixpoint operators
- Equivalence of $\mu\mathtt{ML}_\mathsf{T}$-formulas and parity T-automata holds (Venema 2006)

Question: Can we prove a completeness result for $\mu\mathtt{ML}_\mathsf{T}$?

Coalgebraic fixpoint logic

- Axiomatization of $\mu\mathtt{ML}_\mathsf{T}$: complete axiomatixation for $\mathtt{ML}_\mathsf{T}$ (Kupke, Kurz & Venema 2012) + axioms and rules for fixpoint operators
- Equivalence of $\mu\mathtt{ML}_\mathsf{T}$-formulas and parity T-automata holds (Venema 2006)

Question: Can we prove a completeness result for $\mu\mathtt{ML}_\mathsf{T}$? **Yes!**

Our approach: Principles

Our approach: Principles

- Separate the dynamics from combinatorics

Our approach: Principles

- Separate the dynamics from combinatorics
  Dynamics: Coalgebra
  Combinatorics: Use binary relations to deal with trace combinatorics (trace management)

Our approach: Principles

- Separate the dynamics from combinatorics
  Dynamics: Coalgebra
  Combinatorics: Use binary relations to deal with trace combinatorics (trace management)
- Focus on automata rather than formulas

Our approach: Principles

- Separate the dynamics from combinatorics
  Dynamics: Coalgebra
  Combinatorics: Use binary relations to deal with trace combinatorics (trace management)
- Focus on automata rather than formulas
  Automata:
  - + Uniform, 'clean' presentation of fixpoint formulas
  - + Excellent framework for developing trace theory
  - + Direct formulation of simulation theorem
  - ? Automata and proof theory

Formulas and Automata

Theorem

There are maps $\mathbb{A}_{-} : \mu\mathtt{ML}_{\mathsf{T}} \to Aut(\mathtt{ML}_1)$ *and* $\mathsf{tr} : Aut(\mathtt{ML}_1) \to \mu\mathtt{ML}_{\mathsf{T}}$ *such that:*

(1) *preserve meaning:* $\varphi \equiv \mathbb{A}_{\varphi}$ *and* $\mathbb{A} \equiv \mathsf{tr}(\mathbb{A})$

(2) *satisfy* $\varphi \equiv_{K} \mathsf{tr}(\mathbb{A}_{\varphi})$

(3) *interact nicely with Booleans, modality, fixpoints, and substitution.*

Formulas and Automata

Theorem

There are maps $\mathbb{A}_{-} : \mu\mathtt{ML}_{\mathsf{T}} \to Aut(\mathtt{ML}_1)$ *and* $\mathsf{tr} : Aut(\mathtt{ML}_1) \to \mu\mathtt{ML}_{\mathsf{T}}$ *such that:*

(1) *preserve meaning:* $\varphi \equiv \mathbb{A}_{\varphi}$ *and* $\mathbb{A} \equiv \mathsf{tr}(\mathbb{A})$

(2) *satisfy* $\varphi \equiv_{K} \mathsf{tr}(\mathbb{A}_{\varphi})$

(3) *interact nicely with Booleans, modality, fixpoints, and substitution.*

Item (2) enables us to apply proof-theoretic concepts, such as consistency, to automata.

- Satisfiability game $\mathcal{S}(\mathbb{A})$ (Fontain, Leal & Venema 2010)
  - basic positions are binary relations: $R \in \mathcal{P}(A \times A)$
  - R corresponds to $\bigwedge\{\Theta(a) \mid a \in \mathrm{Rng} R\}$
  - $\exists$ wins a match $\rho = R_0 R_1 \dots$ if there is no *bad trace* through ρ

- Satisfiability game $\mathcal{S}(\mathbb{A})$ (Fontain, Leal & Venema 2010)
  - basic positions are binary relations: $R \in \mathcal{P}(A \times A)$
  - R corresponds to $\bigwedge\{\Theta(a) \mid a \in \mathrm{Rng} R\}$
  - $\exists$ wins a match $\rho = R_0 R_1 \ldots$ if there is no *bad trace* through ρ
- $\exists$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ iff $L(\mathbb{A}) \neq \emptyset$

Framework

- Consequence game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$

Framework

- Consequence game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$
  - cf. $\mathcal{S}(\mathbb{A}) \multimap \mathcal{S}(\mathbb{A}')$
  - basic positions are pairs of binary relations: (R, R')
  - winning condition in terms of trace reflection

Framework

- Consequence game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$
  - cf. $\mathcal{S}(\mathbb{A}) \multimap \mathcal{S}(\mathbb{A}')$
  - basic positions are pairs of binary relations: (R, R')
  - winning condition in terms of trace reflection
- $\mathbb{A} \models_C \mathbb{A}'$ implies that $L(\mathbb{A}) \subseteq L(\mathbb{A}')$

Framework

- Consequence game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$
  - cf. $\mathcal{S}(\mathbb{A}) \multimap \mathcal{S}(\mathbb{A}')$
  - basic positions are pairs of binary relations: (R, R')
  - winning condition in terms of trace reflection
- $\mathbb{A} \vDash_C \mathbb{A}'$ implies that $L(\mathbb{A}) \subseteq L(\mathbb{A}')$ but not vice versa

Special automata

- Disjunctive automata, $\Theta : A \to \mathtt{1ML}^{d}_{\mathsf{T}}$

$$\mathtt{LitC}(X)\ \pi ::= \bot \mid \top \mid p \wedge \pi \mid \neg p \wedge \pi$$

$$\mathtt{1ML}^{d}_{\mathsf{T}}(\mathtt{X}, A)\ \alpha ::= \bot \mid \pi \wedge \nabla\beta \mid \alpha \vee \alpha,$$

where $\pi \in \mathtt{LitC}(\mathtt{X})$ and $\beta \in \mathsf{T}A$.

Special automata

- Semi-disjunctive automata, $\Theta : A \to \mathtt{1ML}_{\mathsf{T}}^{s(C)}(\mathtt{X}, A)$

The set of *(zero-step) C-safe conjunctions* $\mathtt{Conj}_0^C(A)$ contains formulas of the form $\bigwedge B$ with $B \subseteq A$, such that:
for all $b_1 \neq b_2 \in B \cap C$, either b_1 or b_2 has the maximal even parity in C.

$$\mathtt{ML}_{\mathsf{T}}^{s(C)}(\mathtt{X}, A)\ \alpha ::= \bot \mid \pi \wedge \nabla\gamma \mid \alpha \vee \alpha,$$

where $\pi \in \mathtt{LitC}(\mathtt{X})$ and $\gamma \in \mathtt{TConj}_0^C(A)$.

Thin relations

- The *(directed) graph* of $\mathbb{A}$ is the structure $(A, E_{\mathbb{A}})$, where $aE_{\mathbb{A}}b$ if a occurs in $\Theta(b)$. The relation $\lhd_{\mathbb{A}}$ denotes the transitive closure of $E_{\mathbb{A}}$.

Thin relations

- The *(directed) graph* of $\mathbb{A}$ is the structure $(A, E_{\mathbb{A}})$, where $aE_{\mathbb{A}}b$ if a occurs in $\Theta(b)$. The relation $\lhd_{\mathbb{A}}$ denotes the transitive closure of $E_{\mathbb{A}}$.

- $R : A \nrightarrow A$ is thin with respect to $\mathbb{A}$ and a if:

(1) for all $b \in A$ with aRb we have $a \lhd_{\mathbb{A}} b$;

(2) for all $b_1, b_2 \in A$ with $b_1, b_2 \in R[a] \cap C_a$, either $b_1 = b_2$ or one of b_1 and b_2 is a maximal even element of C_a.

We call R $\mathbb{A}$-thin or simply thin, if it is thin with respect $\mathbb{A}$ and all $a \in A$.

Thin relations

- The *(directed) graph* of $\mathbb{A}$ is the structure $(A, E_{\mathbb{A}})$, where $aE_{\mathbb{A}}b$ if a occurs in $\Theta(b)$. The relation $\lhd_{\mathbb{A}}$ denotes the transitive closure of $E_{\mathbb{A}}$.

- $R : A \nrightarrow A$ is thin with respect to $\mathbb{A}$ and a if:

(1) for all $b \in A$ with aRb we have $a \lhd_{\mathbb{A}} b$;

(2) for all $b_1, b_2 \in A$ with $b_1, b_2 \in R[a] \cap C_a$, either $b_1 = b_2$ or one of b_1 and b_2 is a maximal even element of C_a.

We call R $\mathbb{A}$-thin or simply thin, if it is thin with respect $\mathbb{A}$ and all $a \in A$.

Fact: For a stream $\rho = R_1 R_2 R_3 ...$ of thin relations there exists a finite collection F of traces on ρ such that any trace t on ρ is bad if and only if there is some $t' \in F$ cofinally equal to t.

Thin satisfiability game

In $\mathcal{S}_{thin}(\mathbb{A})$, $\forall$ has to guarantee that all basic positions are thin relations.

A thin refutation is a winning strategy for $\forall$ in $\mathcal{S}_{thin}(\mathbb{A})$.

Thin satisfiability game

In $\mathcal{S}_{thin}(\mathbb{A})$, $\forall$ has to guarantee that all basic positions are thin relations.

A thin refutation is a winning strategy for $\forall$ in $\mathcal{S}_{thin}(\mathbb{A})$.

Proposition

Given a semi-disjunctive automaton $\mathbb{A}$, then $\exists$ ($\forall$, respectively) has a winning strategy in $\mathcal{S}(\mathbb{A})$ if and only if $\exists$ ($\forall$, respectively) has a winning strategy in $\mathcal{S}_{thin}(\mathbb{A})$.

Lemma (cf. Kozen)

Given an automaton $\mathbb{A}$, if $\mathrm{tr}(\mathbb{A})$ is consistent, then $\exists$ has a winning strategy in the thin satisfiability game for $\mathbb{A}$.

Lemma (cf. Kozen)

Given an automaton $\mathbb{A}$, if $\mathrm{tr}(\mathbb{A})$ is consistent, then $\exists$ has a winning strategy in the thin satisfiability game for $\mathbb{A}$.

Theorem

For every formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$, there is a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\vdash_{\mathbf{K}} \varphi \to \mathrm{tr}(\mathbb{D})$.

Theorem (Completeness)

Every consistent formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$ *is satisfiable.*

Theorem (Completeness)

Every consistent formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$ is satisfiable.

Proof.

There exists a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\mathrm{tr}(\mathbb{D})$ is consistent too.

Theorem (**Completeness**)

Every consistent formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$ *is satisfiable.*

Proof.

There exists a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\mathrm{tr}(\mathbb{D})$ is consistent too.
Now by Lemma $\exists$ has a winning strategy in $\mathcal{S}_{thin}(\mathbb{D})$.

Theorem (**Completeness**)

Every consistent formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$ *is satisfiable.*

Proof.

There exists a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\mathrm{tr}(\mathbb{D})$ is consistent too.
Now by Lemma $\exists$ has a winning strategy in $\mathcal{S}_{thin}(\mathbb{D})$. But $\mathbb{D}$ is disjunctive and hence semi-disjunctive, and so by Proposition $\exists$ also has a winning strategy in $\mathcal{S}(\mathbb{D})$. $\square$

Work in progress

- Obtain completeness for ∇-based μ-calculus for other functors, including the monotone μ-calculus
- Obtain conditions on functor, language and proof system for which one-step completeness + Kozen axiomatization gives completeness