Coalgebraic CTL: Fixpoint Characterization and Polynomial-time Model Checking

Ryota Kojima

RIMS, Kyoto University, Japan
kojima@kurims.kyoto-u.ac.jp

Abstract. We introduce a path-based coalgebraic temporal logic, Coalgebraic CTL (CCTL), as a categorical abstraction of standard Computation Tree Logic (CTL). Our logic can be used to formalize properties of systems modeled as coalgebras with branching. We present the syntax and path-based semantics of CCTL, and show how to encode this logic into a coalgebraic fixpoint logic with a step-wise semantics. Our main result shows that this encoding is semantics-preserving. We also present a polynomial-time model-checking algorithm for CCTL, inspired by the standard model-checking algorithm for CTL but described in categorical terms. A key contribution of this work is to identify the categorical essence of the standard encoding of CTL into the modal mu-calculus. This categorical perspective also explains the absence of a similar encoding of PCTL (Probabilistic CTL) into the probabilistic mu-calculus.

1 Introduction

1.1 Path-based Temporal Logics and Categorical Generalization

Temporal logics provide specification-description languages in formal verification on transition systems. Among such logics, CTL* and its fragment CTL (Computation Tree Logic) [13, 14] are well-known and widely used because of their descriptive power. They are *path-based* temporal logics: they refer not just to immediate successors of the current state but also to states reachable along (infinite) computation paths. Such path-based formulas can express eventual and permanent behaviors of transition systems, such as liveness and safety properties [1, 9].

CTL, even though it is a simple fragment, inherits much of the expressive power from CTL*. CTL can express liveness and safety, and its formulas are known to characterize bisimilarity equivalence on transition systems [32].

The restriction to CTL gives us computational efficiency in model checking, an advantage over CTL*. It is well-known that, by implementing a naive fixpoint algorithm, verifying a CTL formula on a state takes at most polynomial time [9], in contrast to the known exponential time complexity bound for CTL*.

The technical core behind this efficiency of CTL is an encoding into a fixpoint modal logic, namely the mu-calculus $\mathbf{L}\boldsymbol{\mu}$ [30]. This encoding can then be used to induce another, *step-wise*, semantics of CTL formulas (Table 1, top-right), in contrast to the path-based semantics (Table 1, top-left). The so-called *fixpoint characterization* [14, Lemma 2.6] states that the encoding is semantics-preserving. The fixpoint characterization enables the verification of path-based specifications expressed in CTL by step-wise, iterative calculation on system states and substantially reduces the complexity of the verification.

The fixpoint characterization seems to define what CTL is, as an optimal solution for the trade-off between descriptive power (inherent to the path-based logics) and efficiency in verification (implemented by step-wise iterations).

Nevertheless, the fixpoint characterization does not come for free among known variants of CTL, instantiated over various systems with different branching types. In quantitative variants of CTL [4, 34], the fixpoint characterization results hold under some restrictions on its parameters. In contrast, the well-known probabilistic variant of CTL, called PCTL [1, 20], does not have a known encoding into a natural probabilistic fixpoint logic, like the probabilistic mu-calculus [8].

We aim to establish a *generic* notion of CTL by which we can uniformly classify known variants of CTL and clarify why the original CTL (with some variants) validates the fixpoint characterization and PCTL does not seem to. To this end, we appeal to *coalgebraic logics* [33, 36], a meta-theory of logics on generic systems modeled as coalgebras.

As a coalgebraic generalization of CTL*, the coalgebraic path-based logic $\mu\mathcal{L}$ is proposed in [5]. The original non-deterministic transition systems, which provide the semantic domain for CTL*, are generalized to TF-coalgebras with their branching

Table 1: Fixpoint characterization in classical CTL on a Kripke frame $c\colon X \to \mathcal{P}^+X$ and in our generalization CCTL on a TF-coalgebra $c\colon X \to TFX$.

	path-based semantics	step-wise semantics
classical	$\text{CTL} \hookrightarrow \text{CTL*} \xrightarrow{[15]} \mathbf{2}^X$	$\text{CTL} \xrightarrow{[14]} \mathbf{L}\boldsymbol{\mu} \xrightarrow{[30]} \mathbf{2}^X$
coalgebraic (ours)	$\mathsf{CCTL} \hookrightarrow \mathsf{SFml} \xrightarrow{(\!\lvert\_\rvert\!)^{\mathsf{SFml}}} \mathbf{2}^X$	$\mathsf{CCTL} \xrightarrow{\iota^{-1}} \mu^{\mathsf{CCTL}} \xrightarrow{[\![\_]\!]} \mathbf{2}^X$

type and transition type specified by a monad T and a functor F, respectively. The notion of computation path in CTL* is replaced by its categorical abstraction, maximal execution map. As shown in [5], this framework encompasses both classical CTL* and an extension of PCTL, by instantiating the branching type by the non-empty powerset monad $\mathcal{P}^+$ and the Giry monad $\mathcal{G}_1$.

1.2 Contributions: Coalgebraic CTL

We follow [5] and introduce our coalgebraic generalization of CTL, dubbed CCTL. As a fragment of $\mu\mathcal{L}$, our CCTL has the genericity of branching and transition type T, F, and sets of liftings Σ, Λ of these type functors. Furthermore, CCTL has novel syntactic parameters of *μ-schemes* and *ν-schemes*, which restrict the allowed form of the least and greatest fixpoints. We describe the path-based semantics $(\!\lvert\_\rvert\!)^{\mathsf{SFml}}$ of CCTL inherited from $\mu\mathcal{L}$ (Table 1, bottom left) on a categorical semantic domain, which we call *BT situation*.

Our theoretical highlight is a coalgebraic version of the fixpoint characterization (Thm. 4.6). We present a bijective and semantics-preserving encoding of CCTL into a restriction μ^{CCTL} of the coalgebraic mu-calculus [8, 21, 41], yielding the step-wise semantics of CCTL (Table 1, bottom right).

Sufficient semantic conditions (Assumption 4.7) for the fixpoint characterization are identified in purely categorical terms. They classify the non-deterministic and probabilistic situations: while classical CTL enjoys all of them, PCTL violates some. The violation explains the absence of the fixpoint characterization for PCTL, in categorical terms.

As significant by-products of our fixpoint characterization, we discovered a coalgebraic abstraction of the *expansion law* [1], which tells how to expand path-based formulas step by step concretely (Prop. 4.9). The coalgebraic expansion law is obtained under weaker assumptions than the fixpoint characterization, and induces a *partial* fixpoint characterization (Prop. 4.10). Remarkably, these results also apply to a qualitative fragment of PCTL.

Our fixpoint characterization (Thm. 4.6) leads to a polynomial-time model-checking algorithm $\mathsf{MC}^{\mathsf{CCTL}}_{\mathcal{S}}$ of CCTL, which is parametrized by a BT situation $\mathcal{S}$. With an additional finiteness condition on $\mathcal{S}$, we obtain termination and correctness of the coalgebraic algorithm $\mathsf{MC}^{\mathsf{CCTL}}_{\mathcal{S}}$. We further conclude the polynomial-time complexity bound of $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$, which recovers the quadratic bound of the known CTL model checking with fixpoints [9] when precisely instantiated.

This work is the first step towards a uniform investigation into efficient and expressive coalgebraic path-based logics. It paves the way to classify known examples, like the quantitative CTL [4, 34], and unknown ones, like a "monotone neighborhood" version of CTL induced from neighborhood frames [17].

This work is organized as follows. §2 recalls necessary categorical notions. §3 defines our semantic domain dubbed *BT situation* and introduces CCTL as a fragment of $\mu\mathcal{L}$ [5]. §4 defines a fragment μ^{CCTL} of the coalgebraic mu-calculus, and provides a bijective encoding of CCTL formulas into μ^{CCTL} formulas. Our main result, the fixpoint characterization (Thm. 4.6), shows this encoding is semantic-preserving. §5 formulates a polynomial-time model-checking algorithm for CCTL.

2 Preliminaries

We use $\mathbb{C}$ for a category with finite products and countable distributive coproducts. As examples, we will use the category **Set** of sets and functions and the category **SB** of standard Borel spaces and measurable functions [12, 37].

Let T be a monad, and F be an endofunctor, on $\mathbb{C}$. We formulate a targeted system as a TF-coalgebra, i.e., a map $c\colon X \to TFX$.

2.1 Functors and Monads

We recall some basic properties of functors and monads. See [26] for details.

In our coalgebraic formulation of transition system with branching, we impose the transition type F (an endofunctor) to be *polynomial* and the branching type T (a monad) to be (affine) *strong* or *commutative*[1].

Definition 2.1 ((simple) polynomial functor, linear functor). A functor on the category $\mathbb{C}$ is a *(simple) polynomial functor* [26, Def. 2.2.1] if it is constructed by the following BNF: $F \ ::= \ \mathrm{Id} \mid C \mid \coprod_{b \in B} F_b \mid F_1 \times F_2$ where C is an arbitrary object and B is a countable set. As a special case, a polynomial functor F of the form $F = C_1 \times \mathrm{Id} + C_2$ for arbitrary objects $C_1, C_2 \in \mathbb{C}$ is called to be *linear*.

A major example is an *arity functor* [26] $F = \coprod_{\alpha \in A} \mathrm{Id}^{|\alpha|}$ for some set A with an arity map $|\_|$. If the category $\mathbb{C}$ has enough copowers [31], like **Set** and **SB**, any polynomial functor can be represented as an arity functor, and vice versa. For simplicity, we assume any polynomial functor hereafter is an arity functor.

Definition 2.2 (strong, commutative monad, [27]). Let $\mathbb{C}$ be a category with finite products.

1. A functor $F\colon \mathbb{C} \to \mathbb{C}$ is called *strong* if it is equipped with a natural transformation $\{\mathsf{st}_{X,Y}\colon X \times FY \to F(X \times Y)\}_{X,Y \in \mathbb{C}}$, called *strength*, which satisfies the following commutative diagrams:

$$\begin{array}{ccc} \mathbf{1} \times FX & \xrightarrow{\mathsf{st}_{\mathbf{1},X}} & F(\mathbf{1} \times X) \\ & {\scriptstyle \pi_2} \searrow & \downarrow {\scriptstyle F\pi_2} \\ & & FX, \end{array} \tag{1}$$

$$\begin{array}{ccccc} X \times (Y \times FZ) & \xrightarrow{\mathrm{id}_X \times \mathsf{st}_{Y,Z}} & X \times F(Y \times Z) & \xrightarrow{\mathsf{st}_{X,Y \times Z}} & F(X \times (Y \times Z)) \\ \cong \downarrow & & & & \downarrow \cong \\ (X \times Y) \times FZ & & \xrightarrow{\mathsf{st}_{X \times Y,Z}} & & F((X \times Y) \times Z). \end{array} \tag{2}$$

[1] In [29], we assumed the monad T to be commutative; this work relaxes this restriction to any strong monad, which especially includes any monad on the category **Set** (see [26]).

2. A monad $T\colon \mathbb{C} \to \mathbb{C}$ is called *strong* if T is strong as a functor and moreover its strength $\mathsf{st}_{X,Y}\colon X \times TY \to T(X \times Y)$ satisfies the following diagrams:

$$\begin{array}{ccc} X \times Y & \xrightarrow{=} & X \times Y \\ {\scriptstyle \mathrm{id}_X \times \eta_Y}\downarrow & & \downarrow{\scriptstyle \eta_{X \times Y}} \\ X \times TY & \xrightarrow{\mathsf{st}_{X,Y}} & T(X \times Y), \end{array} \tag{3}$$

$$\begin{array}{ccccc} X \times T^2Y & \xrightarrow{\mathsf{st}_{X,TY}} & T(X \times TY) & \xrightarrow{T\mathsf{st}_{X,Y}} & T^2(X \times Y) \\ {\scriptstyle \mathrm{id}_X \times \mu_Y}\downarrow & & & & \downarrow{\scriptstyle \mu_{X \times Y}} \\ X \times TY & & \xrightarrow{\mathsf{st}_{X,Y}} & & T(X \times Y). \end{array} \tag{4}$$

3. Let T be a strong monad with its strength st. The monad T is called *commutative* if the following commutative diagram holds:

$$\begin{array}{ccccccc} & & T(TX \times Y) & \xrightarrow{T\mathsf{st}'_{X,Y}} & T^2(X \times Y) & & \\ & \overset{\mathsf{st}_{TX,Y}}{\nearrow} & & & & \overset{\mu_{X \times Y}}{\searrow} & \\ TX \times TY & & & & & & T(X \times Y) \\ & \underset{\mathsf{st}'_{X,TY}}{\searrow} & & & & \underset{\mu_{X \times Y}}{\nearrow} & \\ & & T(X \times TY) & \xrightarrow{T\mathsf{st}_{X,Y}} & T^2(X \times Y) & & \end{array} \tag{5}$$

where the map $\mathsf{st}'_{X,Y}\colon TX \times Y \to T(X \times Y)$ is defined to be the composite

$$TX \times Y \xrightarrow{\cong} Y \times TX \xrightarrow{\mathsf{st}_{Y,X}} T(Y \times X) \xrightarrow{\cong} T(X \times Y). \tag{6}$$

The (unique) map in diagram 5 is called *double strength* and denoted dst.

Example 2.3. 1. (non-determinism) The powerset monad $\mathcal{P}$ on **Set** is commutative and its strength is given by $(x, S) \mapsto \{(x, s) \mid s \in S\}$. Its double strength is given by $(T, S) \mapsto T \times S$, where $\times$ is the set product.

2. (reliability) The sub-Giry monad $\mathcal{G}$ on **SB** is defined as follows. The object part of $\mathcal{G}$ maps a standard Borel space (X, Σ_X) to $(\mathcal{M}_X, \Sigma_{\mathcal{M}_X})$ where $\mathcal{M}_X$ is the set of sub-probability measures on X and $\Sigma_{\mathcal{M}_X}$ is the Borel set generated from $\{\rho \in \mathcal{M}_X \mid \rho(S) \subset [0, 1]$ is measurable w.r.t. $([0, 1], \Sigma_{[0,1]})\}$. The sub-Giry monad $\mathcal{G}$ maps a measurable map $f\colon (X, \Sigma_X) \to (Y, \Sigma_Y)$ to

$$\mathcal{G}f\colon (\mathcal{M}_X, \Sigma_{\mathcal{M}_X}) \to (\mathcal{M}_Y, \Sigma_{\mathcal{M}_Y});\ (\mathcal{G}f)(\rho) = \lambda S.\, \rho(f^{-1}(S)).$$

Furthermore, $\mathcal{G}$ is indeed a commutative monad. The unit $\eta\colon (X, \Sigma_X) \to (\mathcal{M}_X, \Sigma_{\mathcal{M}_X})$ of $\mathcal{G}$ maps each element x to the Dirac distribution δ_x, and the multiplication $\mu\colon (\mathcal{M}_{\mathcal{M}_X}, \Sigma_{\mathcal{M}_{\mathcal{M}_X}}) \to (\mathcal{M}_X, \Sigma_{\mathcal{M}_X})$ maps $\Phi \in \mathcal{M}_{\mathcal{M}_X}$ to the measure defined by the integration $\int_{\rho \in \mathcal{M}_X} \Phi(\rho)\, d\rho$. The strength of $\mathcal{G}$ is

$$X \times \mathcal{G}Y \ni (x, \rho) \longmapsto \delta_x \times \rho \in \mathcal{G}(X \times Y)$$

where $\delta_x \times \rho$ is the product of measures [12] and the double strength is

$$\mathcal{G}X \times \mathcal{G}Y \ni (\rho_1, \rho_2) \longmapsto \rho_1 \times \rho_2 \in \mathcal{G}(X \times Y).$$

In the context of formal verification, we often assume considering systems to be *serial* or *left-total* [9]: the system branching does not cause any deadlock. Coalgebraic rephrase of this restriction on systems is *affine-ness* of the branching-type monad T.

Definition 2.4 (affine monad, [23, Def. 4.1]). Let $T\colon \mathbb{C} \to \mathbb{C}$ be a strong monad. The monad T is called *affine* if the unit $\eta_{\mathbf{1}}\colon \mathbf{1} \to T\mathbf{1}$ is an isomorphism.

Note that the inverse $\eta_{\mathbf{1}}^{-1}$ of the unit $\eta_{\mathbf{1}}$ of an affine monad T is necessarily the terminal map $!_{T\mathbf{1}}\colon T\mathbf{1} \to \mathbf{1}$.

If the ambient category $\mathbb{C}$ has pullbacks, every monad T has the largest affine submonad T^{a}, called the *affine part* [23, Def. 4.5] of T, given by the following pullback:

$$\begin{array}{ccc} T^{\mathrm{a}}X & \longrightarrow & TX \\ \big\downarrow & \lrcorner & \big\downarrow{\scriptstyle T!_X} \\ \mathbf{1} & \xrightarrow{\ \eta_{\mathbf{1}}\ } & T\mathbf{1}. \end{array}$$

The affine part T^{a} is known to inherit the commutativity of the monad T (see [23] for details). [2]

Example 2.5. The affine part of $\mathcal{P}$ is the non-empty powerset monad $\mathcal{P}^+$, and the affine part of the sub-Giry monad $\mathcal{G}\colon \mathbf{SB} \to \mathbf{SB}$ is the Giry monad $\mathcal{G}_1$ [16], which is defined by restricting sub-probability measures in $\mathcal{G}$ to probability measures.

Finally, we show a technical lemma on (affine) strong monads, which will be used in the later sections[3].

Lemma 2.6 (compatibility between strength and projections). *Let T be a strong monad.*

1. *The following diagram commutes:*

$$\begin{array}{ccc} X \times TY & \xrightarrow{\ \mathsf{st}_{X,Y}\ } & T(X \times Y) \\ {\scriptstyle \pi_2}\big\downarrow & & \big\downarrow{\scriptstyle T\pi_2} \\ TX & \xrightarrow{\ =\ } & TX. \end{array} \tag{7}$$

2. *If the monad T is moreover affine, the following diagram also commutes:*

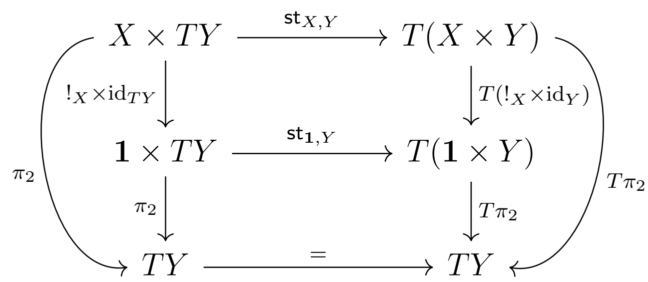
$$\begin{array}{ccc} X \times TY & \xrightarrow{\ \mathsf{st}_{X,Y}\ } & T(X \times Y) \\ {\scriptstyle \pi_1}\big\downarrow & & \big\downarrow{\scriptstyle T\pi_1} \\ X & \xrightarrow{\ \eta_X\ } & TX. \end{array} \tag{8}$$

---

[2] Distributive laws (see Def. 2.7) of any endofunctor over T also restrict to T^{a} [5, Prop. 1].

[3] Lem. 2.6 here is slightly generalized from the previous version [29, Remark A.2, Lemma A.4], where we assumed the monad T to be commutative. The same generalization applies to Prop. 2.8 in §2.2 and Lem. 2.13 in §2.3.

Proof. Item 1 follows from the following commutative diagram:



where the left and right triangles come from the naturality of the projection π_2, the upper square is the naturality of the strength st, and the lower square is the defining diagram of the strong monad T (diagram 1). Note that the outer square of the above diagram is exactly diagram 7.

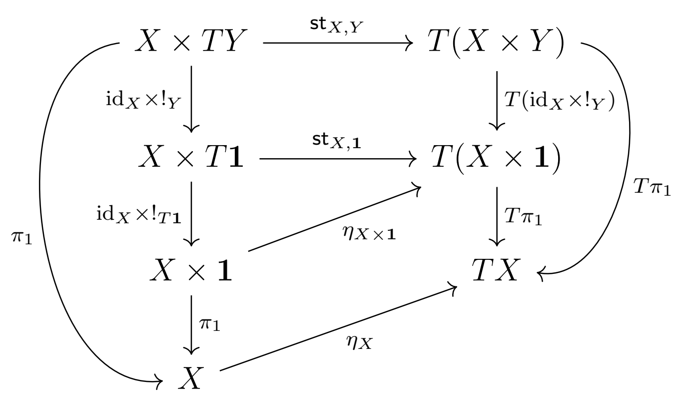
On item 2, since T is strong, we have $\mathsf{st}_{X,\mathbf{1}} \circ \mathrm{id}_X \times \eta_{\mathbf{1}} = \eta_{X\times\mathbf{1}}$. Now that T is moreover affine, we have the inverse map of $\mathrm{id}_X \times \eta_{\mathbf{1}}$, i.e.,

$$(\mathrm{id}_X \times \eta_{\mathbf{1}})^{-1} = \mathrm{id}_X \times \eta_{\mathbf{1}}^{-1} = \mathrm{id}_X \times !_{T\mathbf{1}}.$$

Thus, we have

$$\mathsf{st}_{X,\mathbf{1}} = \mathsf{st}_{X,\mathbf{1}} \circ \mathrm{id}_{X\times T\mathbf{1}} = \mathsf{st}_{X,\mathbf{1}} \circ \mathrm{id}_X \times \eta_{\mathbf{1}} \circ (\mathrm{id}_X \times !_{T\mathbf{1}}) = \eta_{X\times\mathbf{1}} \circ (\mathrm{id}_X \times !_{T\mathbf{1}}). \tag{9}$$

From this equation, the following commutative diagram follows.



where all but the middle triangle follow from naturality of π_1, st and η. The middle triangle is exactly equation 9. The outer square of the above diagram is diagram 8. □

2.2 Distributive Law of Endofunctors over Monads

A *distributive law* of an endofunctor F over a monad T assures compatibility of these two endofunctors: this compatibility is required in the path-based semantics of our coalgebraic temporal logic in §3.3. Conceptually, distributive laws tell us how to cope with iterative composition of system transitions under its branching, by rendering the composite $(TF)^n$ into TF^n with monad multiplication.

Definition 2.7 (distributive law, [26, Def. 5.2.4]). Let $F\colon \mathbb{C} \to \mathbb{C}$ be a functor and $T\colon \mathbb{C} \to \mathbb{C}$ be a monad. A *distributive law* or *$\mathcal{K}\ell$-law* of F over T is a natural transformation $\xi\colon FT \Rightarrow TF$ with

$$\begin{array}{ccc} FX & \xrightarrow{=} & FX \\ {\scriptstyle F\eta_X}\downarrow & & \downarrow{\scriptstyle \eta_{FX}} \\ FTX & \xrightarrow{\xi_X} & TFX, \end{array} \tag{10}$$

$$\begin{array}{ccccc} FT^2X & \xrightarrow{\xi_{TX}} & TFTX & \xrightarrow{T\xi_X} & T^2FX \\ {\scriptstyle F\mu_X}\downarrow & & & & \downarrow{\scriptstyle \mu_{FX}} \\ FTX & & \xrightarrow{\quad\quad\xi_X\quad\quad} & & TFX. \end{array} \tag{11}$$

For a pair of a polynomial functor and a commutative monad, or that of a linear functor and a strong monad, we always have a canonical distributive law.

Proposition 2.8 ([26, Prop. 5.2.12]). *Let T be a strong monad.*

1. *There is a distributive law of F over T for any linear functor $F = C_1 \times \mathrm{Id} + C_2$.*
2. *If the monad T is moreover commutative, there is a distributive law of F over T for any polynomial functor F.*

We call both of these distributive laws $\xi\colon FT \Rightarrow TF$ the canonical *one.*

We concretely show the canonical distributive law ξ w.r.t. a commutative monad T and an arity functor $F = \coprod_{\alpha \in A} \mathrm{Id}^{|\alpha|}$ for some set A:

$$\begin{array}{ccc} (TX)^{|\alpha|} & \xrightarrow{\mathrm{inj}_\alpha} & \coprod_{\alpha \in A}(TX)^{|\alpha|} \cong FTX \\ {\scriptstyle (\mathsf{dst}_{|\alpha|})_X}\downarrow & & \vdots\,{\scriptstyle \xi_A} \\ T(X^{|\alpha|}) & \xrightarrow{T(\mathrm{inj}_\alpha)} & T(\coprod_{\alpha \in A} X^{|\alpha|}) \cong TFX. \end{array} \tag{12}$$

Note that when T is strong, we obtain the canonical distributive law of $F = C_1 \times \mathrm{Id} + C_2 \cong \coprod_{\alpha \in \mathbb{C}(\mathbf{1}, C_1)} \mathrm{Id} + \coprod_{\alpha \in \mathbb{C}(\mathbf{1}, C_1)} \mathbf{1}$ over T by replacing dst above with st.

2.3 Predicate Liftings

The concept of *predicate lifting* [36] was originally defined on **2**-valued predicates and used in interpreting modalities in coalgebraic modal logics. We slightly generalize the notion so that we can treat Ω-valued predicates for any (complete lattice-like) object Ω in $\mathbb{C}$, dubbed *logical value object*.

Definition 2.9 (logical value object). An object $\Omega \in \mathbb{C}$ is called a *logical value object* if its representation $\Omega^{(\_)} := \mathbb{C}(\_, \Omega)\colon \mathbb{C}^{\mathrm{op}} \to \mathbf{Set}$ restricts to the category of complete lattices and $\{\bot, \top, \vee, \wedge\}$-preserving functions.

The booleans $\mathbf{2} \in \mathbf{Set}$ and the discrete boolean measurable space $(\mathbf{2}, \mathcal{P}\mathbf{2}) \in \mathbf{SB}$ are both logical value objects.

Note that while a logical value object itself is not necessarily a set (as $(\mathbf{2}, \mathcal{P}\mathbf{2})$ is not a set), we can treat the object $\Omega \in \mathbb{C}$ as if it were a complete lattice. Indeed, if Ω is a logical value object, any n-ary boolean operator b induces a monotone natural transformation $\left(\Omega^{(\_)}\right)^n \Rightarrow \Omega^{(\_)}$ since the representation $\Omega^{(\_)}$ restricts to the category of complete lattices by definition. [4] By the Yoneda lemma, this monotone natural transformation in turn (bijectively) corresponds to an n-ary map $\gamma_b \colon \Omega^n \to \Omega$. Especially, we obtain $\gamma_\top, \gamma_\bot \colon \mathbf{1} \to \Omega$ and $\gamma_\vee, \gamma_\wedge \colon \Omega^2 \to \Omega$. We see a logical value object $\Omega \in \mathbb{C}$ as a complete lattice with boolean operators b by identifying a boolean operator b with its induced map γ_b above.

We denote the set of all boolean operators by Γ.

Our definition of predicate lifting of endofunctors is parametrized by a logical value object Ω.

Definition 2.10 (predicate lifting). Let $G \colon \mathbb{C} \to \mathbb{C}$ be an endofunctor, and $\Omega \in \mathbb{C}$ be a logical value object.

1. A *(predicate) lifting* or *modality* of G w.r.t. Ω is a natural transformation $\{\lambda_Y \colon \Omega^Y \to \Omega^{GY}\}_{Y \in \mathbb{C}}$ which is monotone w.r.t. the lattice structures on Ω^Y and Ω^{GY}.
2. We write $\mathrm{ev}_\lambda \colon G\Omega \to \Omega$ for the correspondent of a lifting λ via the Yoneda lemma [5]: this is to say, $\mathrm{ev}_\lambda = \lambda_\Omega(\mathrm{id}_\Omega)$ and $\lambda_Y(p) = \mathrm{ev}_\lambda \circ Gp$ for $p \in \Omega^Y$.

Henceforth, we consistently use the letter σ for a predicate lifting of the branching-type monad T and λ for that of the transition-type functor F. We call σ *"path quantifier"* and λ *"next-time operator."*

Example 2.11. 1. There is a trivial lifting $\mathrm{id}_{\Omega^X} \colon \Omega^X \to \Omega^X$ of the identify functor Id for any logical value object Ω. More generally, there is a canonical predicate lifting $\mathrm{Pred}(F)$ for each polynomial functor F [26, Lemma 6.1.3]. For an arity functor $F = \coprod_{\alpha \in A} (\_)^{|\alpha|}$, the lifting $\mathrm{Pred}(F)$ is induced from the map $[\wedge^{|\alpha|}]_{\alpha \in A} \colon \coprod_{\alpha \in A} \Omega^{|\alpha|} \to \Omega$, where $[\_]$ denotes a cotuple of the coproduct and $\wedge^{|\alpha|} \colon \Omega^{|\alpha|} \to \Omega$ denotes $|\alpha|$-ary conjunction. Thus, $\mathrm{Pred}(F)(Q)$ for a predicate $Q \in \Omega^X$ is given by $\mathrm{Pred}(F)(Q) = [\wedge^{|\alpha|} \circ Q^{|\alpha|}]_{\alpha \in A} \in \Omega^{\coprod_{\alpha \in A} X^{|\alpha|}} = \Omega^{FX}$.

2. (non-determinism) The non-empty powerset monad $\mathcal{P}^+$ has predicate liftings $\mathcal{P}^+_\Diamond, \mathcal{P}^+_\Box$ w.r.t. the booleans $\mathbf{2} \in \mathbf{Set}$ induced by the maps $\Diamond, \Box \colon \mathcal{P}\mathbf{2} \to \mathbf{2}$, respectively (i.e., $\mathcal{P}^+_\Diamond(P) = \Diamond \circ \mathcal{P}^+(P)$ and $\mathcal{P}^+_\Box(P) = \Box \circ \mathcal{P}^+(P)$, recall Def. 2.10). These maps $\Diamond, \Box$ are defined as follows: for $S \in \mathcal{P}^+\mathbf{2}$, $\Diamond(S) = 1$ if and only if $S = \{0,1\}, \{1\}$ and $\Box(S) = 1$ if and only if $S = \{1\}$.
3. (reliability) The Giry monad $\mathcal{G}_1$ has predicate liftings $\mathcal{G}_{1,\geq q}, \mathcal{G}_{1,>q}$ w.r.t. $(\mathbf{2}, \mathcal{P}\mathbf{2}) \in \mathbf{SB}$ induced by respectively the "larger-than-q-or-equal" and "larger-than-q" maps $\geq_q, >_q \colon \mathcal{G}_1(\mathbf{2}, \mathcal{P}\mathbf{2}) \cong ([0,1], \Sigma_{[0,1]}) \to (\mathbf{2}, \mathcal{P}\mathbf{2})$. The map $\geq_q$ is defined by $\geq_q (r) = 1$ if and only if $r \geq q$, and the map $>_q$ is also defined likewise.

[4] Here a *boolean operator* means a map on a complete lattice constructed from operators $\bot, \top, \vee$ and $\wedge$.

[5] Recall the Yoneda lemma: there is a bijective correspondence between natural transformations from $\Omega^{(\_)}$ to $\Omega^{G(\_)}$ and elements of $\Omega^{G\Omega}$.

In the rest of this section, we show a technical lemma (Lem. 2.13) which asserts exchangability of T's lifting (path-quantifier) σ and F's lifting (next-time operator) λ via the canonical distributive law ξ of F over T. To show Lem. 2.13, we recall the notion of *bilinearity* [28, Section 1] of a map w.r.t. Eilenberg-Moore algebras, which plays a crucial role in proving Lem. 2.13 and our main result (Thm. 4.6).

Definition 2.12 (m-linear, bilinear map, Kock [28, Section 1]). Let $T\colon \mathbb{C}\to\mathbb{C}$ be a strong monad. Let $a_m\colon TA_m \to A_m$ for $m = 1,\ldots,n$ and $c\colon TC \to C$ be Eilenberg-Moore T-algebras. An n-ary map $f\colon A_1 \times\cdots\times A_n \to C$ is called *m-linear* $(1 \le m \le n)$ if

$$\begin{array}{ccccccc} A_{\le m-1}\times TA_m\times A_{m+1\le} & \xrightarrow{\mathsf{st}\times\mathrm{id}} & T(A_{\le m-1}\times A_m)\times A_{m+1\le} & \xrightarrow{\mathsf{st}'} & T(\prod_{1\le i\le n}A_i) & \xrightarrow{Tf} & TC \\ \downarrow{\scriptstyle \mathrm{id}\times a_m\times\mathrm{id}} & & & & & & \downarrow{\scriptstyle c}\\ \prod_{1\le i\le n}A_i & & & \xrightarrow{\quad f\quad} & & & C. \end{array} \tag{13}$$

where $A_{\le m-1}$ and $A_{m+1\le}$ denote $\prod_{1\le i\le m-1} A_i$ and $\prod_{m+1\le i\le n} A_i$. We call the map f *bilinear* if f is m-linear for every $m = 1,\ldots,n$.

Remarkably, while the notion of bilinearity apparently seems to depend on the composition order of st and st', it is known to be independent of the order [28]. Thus, the notion is well-defined for any (non-commutative) strong monad.

Nonetheless, if the monad T in question is moreover commutative, we can define bilinearity more concisely using T's double strength dst: a map $f\colon A_1 \times\cdots\times A_n \to C$ is bilinear if the following diagram commutes:

$$\begin{array}{ccccc} TA_1\times\cdots\times TA_n & \xrightarrow{\mathsf{dst}_n} & T(A_1\times\cdots\times A_n) & \xrightarrow{Tf} & TC \\ \downarrow{\scriptstyle a_1\times\cdots\times a_n} & & & & \downarrow{\scriptstyle c}\\ A_1\times\cdots\times A_n & & \xrightarrow{\quad f\quad} & & C \end{array} \tag{14}$$

where dst_n means $(n-1)$-times application of dst.

On top of these categorical machinaries, we can state and prove Lem. 2.13. [6]

Lemma 2.13. *Let T be a commutative (strong, resp.) monad and $F = \coprod_{\alpha\in A}(\_)^{|\alpha|}$ (linear F, resp.), and $\sigma\colon \Omega^{(\_)} \to \Omega^{T(\_)}$ and $\lambda\colon \Omega^{(\_)} \to \Omega^{F(\_)}$ be liftings of T and F, respectively. We suppose ev_σ is an Eilenberg-Moore T-algebra and $\mathrm{ev}_\lambda \circ \mathrm{inj}_\alpha\colon \Omega^{|\alpha|} \to \Omega$ is bilinear w.r.t. ev_σ. Then the following diagram commutes: for each $Y \in \mathbb{C}$,*

$$\begin{array}{ccccc} \Omega^{Y} & \xrightarrow{\sigma_Y} & \Omega^{TY} & \xrightarrow{\lambda_{TY}} & \Omega^{FTY} \\ \downarrow{\scriptstyle \lambda_Y} & & & & \uparrow{\scriptstyle \xi^*_Y}\\ \Omega^{FY} & & \xrightarrow{\quad\sigma_{FY}\quad} & & \Omega^{TFY}, \end{array} \tag{15}$$

where ξ is the canonical distributive law of F over T.

[6] Lem. 2.13 is a mild generalization of [6, Lemma 5.11], where the result is restricted to the case when the branching type is specified by a semiring monad and liftings σ and λ are fixed to specific ones.

Proof. Let $p \in \Omega^Y$ and ξ be the canonical distributive law (Prop. 2.8). Since $\lambda_Y(p) = \mathrm{ev}_\lambda \circ F(p)$ and $\sigma_Y(p) = \mathrm{ev}_\sigma \circ T(p)$ by definition (Def. 2.10), the upper path in diagram 15 reduces to

$$\begin{aligned}
\lambda_{TY} \circ \sigma_Y(p) &= \mathrm{ev}_\lambda \circ F\big(\sigma_Y(p)\big) \\
&= \mathrm{ev}_\lambda \circ F\big(\mathrm{ev}_\sigma \circ T(p)\big) \\
&= \mathrm{ev}_\lambda \circ F\mathrm{ev}_\sigma \circ FTp
\end{aligned}$$

and the lower path reduces to

$$\begin{aligned}
\xi^*_Y \circ \sigma_{FY} \circ \lambda_Y(p) &= \mathrm{ev}_\sigma \circ T\big(\lambda_Y(p)\big) \circ \xi_Y \\
&= \mathrm{ev}_\sigma \circ T\big(\mathrm{ev}_\lambda \circ F(p)\big) \circ \xi_Y \\
&= \mathrm{ev}_\sigma \circ T(\mathrm{ev}_\lambda) \circ TFp \circ \xi_Y \\
&= \mathrm{ev}_\sigma \circ T(\mathrm{ev}_\lambda) \circ \xi_\Omega \circ FTp
\end{aligned}$$

where we used naturality of the distributive law ξ at the last transformation. Thus, it suffices to show the following diagram commutes.

$$\begin{array}{ccccc}
FT\Omega & \xrightarrow{F\mathrm{ev}_\sigma} & F\Omega & \xrightarrow{\mathrm{ev}_\lambda} & \Omega \\
\downarrow{\scriptstyle \xi_\Omega} & & & & \| \\
TF\Omega & \xrightarrow{T\mathrm{ev}_\lambda} & T\Omega & \xrightarrow{\mathrm{ev}_\sigma} & \Omega.
\end{array} \tag{16}$$

We show diagram 16 for a commutative monad T and the arity functor $F = \coprod_{\alpha \in A}(\_)^{|\alpha|}$: the following proof also goes similarly for a strong monad T, by instantiating $F = \coprod_{\alpha \in \mathbb{C}(\mathbf{1}, C_1)} \mathrm{Id} + \coprod_{\alpha \in \mathbb{C}(\mathbf{1}, C_1)} \mathbf{1}$ and replacing dst with st.

Instantiated with $F = \coprod_{\alpha \in A}(\_)^{|\alpha|}$, diagram 16 reduces to the following diagram (using universality of the coproduct $\coprod_{\alpha \in A}(T\Omega)^{|\alpha|}$).

$$\begin{array}{ccccccc}
(T\Omega)^{|\alpha|} & \xrightarrow{\mathrm{inj}_\alpha} & \coprod_{\alpha \in A}(T\Omega)^{|\alpha|} & \xrightarrow{\coprod_{\alpha \in A} \mathrm{ev}_\sigma^{|\alpha|}} & \coprod_{\alpha \in A} \Omega^{|\alpha|} & \xrightarrow{\mathrm{ev}_\lambda} & \Omega \\
 & & \downarrow{\scriptstyle \xi_\Omega} & & & & \| \\
 & & T(\coprod_{\alpha \in A} \Omega^{|\alpha|}) & \xrightarrow{T\mathrm{ev}_\lambda} & T\Omega & \xrightarrow{\mathrm{ev}_\sigma} & \Omega.
\end{array} \tag{17}$$

Commutativity of diagram 17 is assured as follows:

$$
\begin{array}{ccccccc}
 & & \Omega^{|\alpha|} & & & & \\
 & \nearrow^{\mathrm{ev}_\sigma^{|\alpha|}} & & \searrow^{\mathrm{inj}_\alpha} & & \xrightarrow{\mathrm{ev}_\lambda \circ \mathrm{inj}_\alpha} & \\
(T\Omega)^{|\alpha|} & \xrightarrow{\mathrm{inj}_\alpha} & \coprod_{\alpha\in A}(T\Omega)^{|\alpha|} & \xrightarrow[\coprod_{\alpha\in A}\mathrm{ev}_\sigma^{|\alpha|}]{} & \coprod_{\alpha\in A}\Omega^{|\alpha|} & \xrightarrow{\mathrm{ev}_\lambda} & \Omega \\
\downarrow{\scriptstyle \mathsf{dst}_{|\alpha|}} & & \downarrow{\scriptstyle \xi_\Omega} & & & & \| \\
T(\Omega^{|\alpha|}) & \xrightarrow{T\mathrm{inj}_\alpha} & T(\coprod_{\alpha\in A}\Omega^{|\alpha|}) & \xrightarrow{T\mathrm{ev}_\lambda} & T\Omega & \xrightarrow{\mathrm{ev}_\sigma} & \Omega \\
 & & \xrightarrow{T(\mathrm{ev}_\lambda \circ \mathrm{inj}_\alpha)} & & & &
\end{array}
\tag{18}
$$

where the top right and bottom triangles are trivial; the top left triangle is universality of coproduct; and the middle left square comes from the construction of the canonical distributive law (Prop. 2.8). Finally, the commutativity of the outer square of diagram 18 is nothing other than the bilinearity of the map $\mathrm{ev}_\lambda \circ \mathrm{inj}_\alpha$. □

3 Coalgebraic Path-based Temporal Logics: $\mu\mathcal{L}$, CCTL

3.1 Coalgebraic Abstraction of Systems

We first set up a semantic domain of coalgebraic path-based logics $\mu\mathcal{L}$ and CCTL, dubbed *BT situation*. It is categorical data that includes branching and transition types T and F, a coalgebra of these types, path quantifiers, and next-time operators.

Definition 3.1 (BT situation). A *branching-transition situation* (*BT situation*, in short) is given by a tuple $(\mathbb{C}, T, F, c, \Omega, \Sigma, \Lambda)$ where[7]

1. $\mathbb{C}$ is a concrete, finitely complete, and countably cocomplete category, [8]
2. $T\colon \mathbb{C} \to \mathbb{C}$ is a strong monad, whose strength is denoted st,
3. $F\colon \mathbb{C} \to \mathbb{C}$ is a polynomial functor with the canonical distributive law ξ over T,
4. $c\colon X \to TFX$ is a TF-coalgebra,
5. $\Omega \in \mathbb{C}$ is a logical value object (see Def. 2.9) with a labeling function $L\colon \mathrm{AP} \to \Omega^X$ for a fixed set of atomic propositions AP,
6. Σ is a set of path quantifiers (i.e., predicate liftings of T),
7. Λ is a set of next-time operators (i.e., predicate liftings of F).

Note that the polynomial F specified in item 3 is restricted to linear ones when the monad T in item 2 is a non-commutative strong monad since we require the existence of the canonical distributive law ξ of F over T for these F and T in item 3 (see Prop. 2.8).

[7] As two refinements from [29, Definition 3.1] on the date consisting a BT-situation, (a) T is required to be just strong, instead of commutative; (b) we here assume that the logical value object Ω equips a labeling function L, which is required to add atomic propositions to the syntaxes of our logics (see Def. 3.9 and Def. 4.1).

[8] By a *concrete* category, we mean a category equipped with a faithful functor to the large category **Set** [31].

Table 2: Examples of BT situation

	parameters	$\mathcal{S}_{\mathrm{ND}}$	$\mathcal{S}_{\mathrm{R}}$
category	$\mathbb{C}$	**Set**	**SB**
branching type	T	$\mathcal{P}^+$	$\mathcal{G}_1$
transition type	F	any polynomial	any polynomial
system	$c\colon X \to TFX$	a Kripke frame	a Markov chain
truth values	$\Omega \in \mathbb{C}$	$\mathbf{2}$	$(\mathbf{2}, \mathcal{P}\mathbf{2})$
path quantifiers	$\{\sigma\}_{\sigma\in\Sigma}$	$\{\mathcal{P}^+_\Diamond, \mathcal{P}^+_\Box\}$	$\{\mathcal{G}_{1,\geq q}, \mathcal{G}_{1,>q}\}_{q\in[0,1]}$
next-time operators	$\{\lambda\}_{\lambda\in\Lambda}$	$\{\mathrm{Pred}(F)\}$	$\{\mathrm{Pred}(F)\}$

Example 3.2. Table 2 defines our examples of BT situation. Note that our instantiations $\mathcal{S}_{\mathrm{ND}}$ and $\mathcal{S}_{\mathrm{R}}$ still have the genericity of F.

1. (non-determinism) In $\mathcal{S}_{\mathrm{ND}}$, $\mathrm{Pred}(F)$ and $\mathcal{P}^+_\Diamond, \mathcal{P}^+_\Box$ are the liftings as in Example 2.11. A $\mathcal{P}^+F$-coalgebra is an "F-generic" left-total Kripke frame: $\mathcal{P}^+F$-coalgebras includes classical Kripke frames (when $F = \mathrm{Id}$), labeled Kripke frames (when $F = \mathcal{P}(\mathrm{AP}) \times \mathrm{Id}$) and Kripke frames with termination (when $F = \mathbf{1} + \mathrm{Id}$).
2. (reliability) In $\mathcal{S}_{\mathrm{R}}$, $\mathcal{G}_1$ is the Giry monad, $\mathrm{Pred}(F)$ and $\mathcal{G}_{1,\geq q}, \mathcal{G}_{1,>q}$ are as in Example 2.11. A $\mathcal{G}_1F$-coalgebra is an F-generic Markov chain, which coincides with a classical one when we restrict the state space to a discrete space $(X, \mathcal{P}X)$ for some countable set X and F is given by Id (or $\mathcal{P}(\mathrm{AP}) \times \mathrm{Id}$).
3. (qualitative reliability) We also define a BT situation $\mathcal{S}_{\mathrm{qR}}$ for *qualitative* reliability by restricting the set of path quantifiers of $\mathcal{S}_{\mathrm{R}}$ to $\{\mathcal{G}_{1,\geq 1}, \mathcal{G}_{1,>0}\}$.

3.2 Maximal Traces as Computation Paths

We recall the concepts of maximal trace map and maximal execution map of TF-coalgebras. The latter is an abstraction of the classical notion of computation paths and will be used in the formulation of our path-based semantics.

We first recall Jacobs' original formulation of maximal trace [24] on the Kleisli category of the monad T [26]. Let $J\colon \mathbb{C} \to \mathcal{K}\ell(T)$ be the canonical left adjoint of the monad T. This J sends an object of $\mathbb{C}$ to itself and a map $f\colon A \to B$ to $\eta_B \circ f$. Given a distributive law $\xi\colon FT \Rightarrow TF$, we have the induced functor $\overline{F}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ that sends a Kleisli arrow $f\colon A \nrightarrow B$ to $\xi_B \circ Ff\colon FA \nrightarrow FB$.

Definition 3.3 (maximal trace, [24, 40]). A tuple (T, F, ξ, c) constitutes *a maximal trace situation* if

1. T is a monad such that each homset of the Kleisli category $\mathcal{K}\ell(T)$ carries an order $\sqsubseteq$,
2. F is an endofunctor with a final coalgebra $\zeta\colon Z \to FZ$,
3. ξ is a distributive law ξ of F over T,

4. c is a TF-coalgebra (equivalently, $\overline{F}$-coalgebra $c\colon X \nrightarrow \overline{F}X$) for which there exists the greatest map $u\colon X \nrightarrow Z$ satisfying $J\zeta \odot u = \overline{F}u \odot c$ w.r.t. the order $\sqsubseteq$, where $\odot$ is the Kleisli composition.

The greatest map u in condition 4 is called the *maximal trace map* w.r.t. the tuple (T, F, ξ, c) and is denoted by $\mathrm{tr}(c)$ (making T, F and ξ implicit).

To fit the above notion (Def. 3.3) into our formulation based on a BT situation, we instantiate it with the data consisting a BT situation.

Definition 3.4 (maximal execution map, [5]). Let $\mathcal{S} = (\mathbb{C}, T, F, c, \Omega, \Sigma, \Lambda)$ be a BT situation.

1. We define the functor $F_X := X \times F$, called *F-path functor* of X, induced from the (polynomial) transition-type functor F and the state space X. We denote the final coalgebra of F_X by [9]

$$\zeta = \langle \zeta_1, \zeta_2 \rangle \colon Z_X \to X \times F Z_X = F_X Z_X.$$

2. We define the *induced $T \circ F_X$-coalgebra* $c'\colon X \to TF_X(X) = T(X \times FX)$ of c by [10]

$$c' = \mathsf{st}_{X,FX} \circ \langle \mathrm{id}_X, c \rangle \colon X \to X \times TFX \to T(X \times FX).$$

3. If the tuple (T, F_X, ξ, c') constitutes a maximal trace situation with ξ the canonical distributive law specified in Def. 3.1, the maximal trace map $\mathrm{tr}(c')$ w.r.t. the tuple (T, F_X, ξ, c') is called the *maximal execution map* of the TF-coalgebra $c\colon X \to TFX$.

Since maximal execution map is a special case of maximal trace map, the following corollary immediately follows from Def. 3.3.

Corollary 3.5. *The maximal execution map* $\mathrm{tr}(c')$ *of the* TF*-coalgebra* c *for a BT situation* $\mathcal{S} = (\mathbb{C}, T, F, c, \Omega, \Sigma, \Lambda)$ *exists if each homset in* $\mathcal{K}\ell(T)$ *carries an order and there exists the greatest map* $u\colon X \to TZ_X$ *satisfying* $J\zeta \odot u = \overline{F_X}u \odot c'$.

We give the intuition behind Def. 3.4.

– We added the auxiliary coefficient X to the transition-type F in defining F-path functor F_X so that we can keep track of the state we have just passed. The coefficient X plays the role of the "storage" of the current state in each transition of the induced $TF_X\big(= T(X \times F)\big)$-coalgebra c'.

[9] Note that every polynomial functor F, and thus F-path functor F_X, is known to have its final coalgebra. See [26] for details.

[10] In [29], we defined the coalgebra c' using the double strength of T (assuming T is a commutative monad). Nonetheless, we can define c' with only the strength as above, and as a result, we can lift the commutativity assumption on T.

– The final F_X-coalgebra Z_X can be seen as the collection of all "X-labeled F-trees". The final coalgebra map $\zeta = \langle \zeta_1, \zeta_2 \rangle \colon Z_X \to X \times FZ_X$ returns, given a labeled tree, the label of the root of the tree (via ζ_1) and its sub-trees (via ζ_2). In the simplest case, $F = \mathrm{Id}$ on the category **Set**, the final $(\mathrm{Id})_X$-coalgebra Z_X coincides with the set X^ω of all infinite paths (or streams) over the state space X, and the maps ζ_1 and ζ_2 correspond to the *head* and *tail operator* on the path space X^ω, respectively.
– The maximal execution $\mathrm{tr}(c') \colon X \to TZ_X$ of the TF-coalgebra c gives "computations", represented in terms of labeled trees, from the current state. When $F = \mathrm{Id}$, the maximal execution $\mathrm{tr}(c')$ returns "(generalized) computation paths" depending on the branching-type T. Concrete instances in Example 3.7 will help to capture this intuition.

Since our focus is path-based semantics of temporal logic and the notion of maximal execution is a coalgebraic generalization of computation paths, a BT situation should naturally equip the maximal execution.

Definition 3.6 (BT situation with maximal execution). A *BT situation with maximal execution* is a BT situation $\mathcal{S}$ with the maximal execution $\mathrm{tr}(c')$ of the TF-coalgebra $c \colon X \to TFX$.

In the remainder of this work, we fix a BT situation $\mathcal{S} = (\mathbb{C}, T, F, c, \Omega, \Sigma, \Lambda)$ with maximal execution, $\mathrm{tr}(c')$.

Example 3.7. All of our leading examples equip their maximal execution. For the sake of simplicity, we fix F to be the identity functor Id on **Set** or **SB** below.

1. (non-determinism) The final coalgebra of $(\mathrm{Id}_{\mathbf{Set}})_X = X \times \mathrm{Id}_{\mathbf{Set}}$ for **Set** is the set X^ω of streams, where ω is the set of finite ordinals. The existence of maximal executions for $\mathcal{P}^+$ is assured by an adaptation of [40, Prop. 4.1]. Concretely, the maximal execution map $\mathrm{tr}(c') \colon X \to \mathcal{P}^+ X^\omega$ maps x to $\{\pi \in X^\omega \mid \pi_0 = x \text{ and } \forall n \in \omega . \pi_{n+1} \in c(\pi_n)\}$.
2. (reliability) The final coalgebra of $(\mathrm{Id}_{\mathbf{SB}})_{(X, \Sigma_X)} = (X, \Sigma_X) \times \mathrm{Id}_{\mathbf{SB}}$ for **SB** is the measurable set $(X^\omega, \Sigma_{X^\omega})$ of streams. Its measurable structure Σ_{X^ω} is generated by the cylinder sets $\mathrm{Cyl}(t) = \{\pi \in X^\omega \mid \pi \text{ has the prefix } t\}$ for every finite path t. The existence of maximal execution for $\mathcal{G}_1$ is assured by [40, Prop. 5.2]. Concretely, the maximal execution map $\mathrm{tr}(c') \colon (X, \Sigma_X) \to \mathcal{G}_1(X^\omega, \Sigma_{X^\omega})$ is determined by
$$\mathrm{tr}(c')(x)\big(\mathrm{Cyl}(t)\big) = c(x)(x_1) \cdot c(x_1)(x_2) \cdot \ldots \cdot c(x_{n-1})(x_n)$$
for each cylinder set $\mathrm{Cyl}(t)$ with $t = xx_1x_2 \ldots x_{n-1}x_n$. For a detailed description, see [39, Def. E.9].

3.3 The logics $\mu\mathcal{L}$ and CCTL

We first recall the coalgebraic logic $\mu\mathcal{L}$ [5]. Its syntax is given by coalgebra-generic *path formulas* and *state formulas*. The following definition is a slight adaptation of the original $\mu\mathcal{L}$.[11]

[11] The notations $\mu\mathcal{L}_F$, $\mu\mathcal{L}$, $[\lambda_F]$ and $[\lambda]$ of the original $\mu\mathcal{L}$ [5] correspond to PFml, SFml, ♡ and ♠, respectively, in our presentation. We also omit variables in SFml.

Definition 3.8 (state formulas and path formulas). Let Σ, Λ be sets, and Γ be a ranked alphabet. Given a set AP of atomic propositions, two sets $\mathsf{PFml}_{\Gamma,\Lambda,\Sigma}$ and $\mathsf{SFml}_{\Gamma,\Lambda,\Sigma}$ (or simply $\mathsf{PFml}, \mathsf{SFml}$) are defined by the following mutual induction:

$$\varphi \in \mathsf{PFml} ::= u \mid \boxdot_\gamma(\varphi_1, \ldots, \varphi_{|\gamma|}) \mid \heartsuit_\lambda \varphi \mid \mu u.\varphi \mid \nu u.\varphi \mid \psi$$
$$\psi \in \mathsf{SFml} ::= p \in \mathrm{AP} \mid \boxdot_\gamma(\psi_1, \ldots, \psi_{|\gamma|}) \mid \spadesuit_\sigma \varphi,$$

where u is a proposition variable, $\gamma \in \Gamma$, $\lambda \in \Lambda$ and $\sigma \in \Sigma$[12]. Furthermore, we assume φ in $\spadesuit_\sigma \varphi \in \mathsf{SFml}$ is closed, i.e., φ has no proposition variables.

The symbols $\boxdot_\gamma$, $\heartsuit_\lambda$ and $\spadesuit_\sigma$ correspond to boolean operators, next-time operators and path quantifiers, respectively.

We can then define the new logic CCTL as a fragment of SFml by restricting the forms of fixpoint formulas.

Definition 3.9 (CCTL). Let Σ, Λ be sets and Γ be a ranked alphabet with subsets $\Gamma_\mu, \Gamma_\nu \subset \Gamma$. Given a set AP of atomic propositions, the set $\mathsf{CCTL}_{\Gamma_\mu,\Gamma_\nu}$ (whose subscripts we will sometimes omit) is the subset of SFml defined by the following grammar:

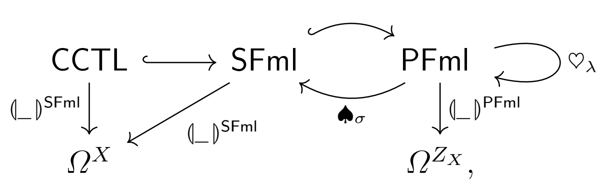
$$\begin{aligned}\psi \in \mathsf{CCTL}_{\Gamma_\mu,\Gamma_\nu} ::= {} & p \in \mathrm{AP} \mid \boxdot_\gamma(\psi_1, \ldots, \psi_{|\gamma|}) \mid \spadesuit_\sigma \heartsuit_\lambda \psi \\ & \mid \spadesuit_\sigma(\mu u.\, \boxdot_{\gamma_\mu}(\psi_1, \ldots, \psi_{|\gamma_\mu|-1}, \heartsuit_\lambda u)) \\ & \mid \spadesuit_\sigma(\nu u.\, \boxdot_{\gamma_\nu}(\psi_1, \ldots, \psi_{|\gamma_\nu|-1}, \heartsuit_\lambda u))\end{aligned}$$

where u is a proposition variable, $\gamma \in \Gamma$, $\lambda \in \Lambda$, $\sigma \in \Sigma$, $\gamma_\mu \in \Gamma_\mu$ and $\gamma_\nu \in \Gamma_\nu$.

The operators γ_μ and γ_ν in the fixpoint formula are called *μ-schemes* and *ν-schemes*, respectively. These are used to recover temporal operators (EF, AF, etc. in classical CTL) and are crucial in characterizing $\mathsf{CCTL}_{\Gamma_\mu,\Gamma_\nu}$ within the mu-calculus.

Example 3.10. In the literature, the modality symbols $\spadesuit_{\mathcal{P}^+_\Diamond}$ and $\spadesuit_{\mathcal{P}^+_\Box}$ in CTL are respectively denoted by E and A. The modality symbols $\spadesuit_{\mathcal{G}_{1,\geq q}}$ and $\spadesuit_{\mathcal{G}_{1,>q}}$ in PCTL are respectively denoted by $\mathbb{P}_{\geq q}$ and $\mathbb{P}_{>q}$ [1]. The modality symbol $\heartsuit_{\mathrm{Pred}(F)}$ in both CTL and PCTL is often denoted by X. In both CTL and PCTL, their sets of μ-schemes and ν-schemes are respectively given by $\{(\_ \vee (\_ \wedge \_))\}$ and $\{(\_ \wedge (\_ \vee \_))\}$. The least/greatest fixpoint formulas made of $(\_ \vee (\_ \wedge \_))/(\_ \wedge (\_ \vee \_))$ is often denoted by U/W. [13]

The relationships between SFml, PFml, CCTL can be summarized as follows:



where the semantics $(\!|\_|\!)^{\mathsf{SFml}}$ and $(\!|\_|\!)^{\mathsf{PFml}}$ is defined below [5].

[12] In addition to the formulas in [29, Definition 3.6], we here added atomic propositions AP to the syntax of SFml. Similar modifications apply to Def. 3.9, Def. 4.1 and all relevant parts throughout this work.

[13] Another (equivalent) choice of Γ_μ and Γ_ν is possible: we can put $\Gamma_\mu = \{\vee, (\_ \vee (\_ \wedge \_))\}$ and $\Gamma_\nu = \{\wedge, (\_ \wedge (\_ \vee \_))\}$, and the least/greatest fixpoint formula made of $\vee/\wedge$ is denoted by F/G.

Definition 3.11 (semantics of PFml and SFml formulas). For each PFml formula φ with free variables $u_1, \dots, u_m$, and each SFml formula ψ, their interpretation $(\!|\varphi|\!)^{\mathsf{PFml}} \colon \left(\Omega^{Z_X}\right)^m \to \Omega^{Z_X}$ and $(\!|\psi|\!)^{\mathsf{SFml}} \colon \Omega^X$ are defined in the following mutually inductive manner[14]: for $\vec{V} = V_1, \dots, V_m$ with $V_i \colon X \to \Omega$,

$$
\begin{aligned}
(\!|u_i|\!)^{\mathsf{PFml}}(\vec{V}) &:= V_i, \\
(\!|\boxdot_\gamma(\varphi_1, \dots, \varphi_{|\gamma|})|\!)^{\mathsf{PFml}}(\vec{V}) &:= \gamma\big(\, (\!|\varphi_1|\!)^{\mathsf{PFml}}(\vec{V}),\ \dots,\ (\!|\varphi_{|\gamma|}|\!)^{\mathsf{PFml}}(\vec{V})\,\big), \\
(\!|\heartsuit_\lambda \varphi|\!)^{\mathsf{PFml}}(\vec{V}) &:= (\!|\heartsuit_\lambda|\!)\big((\!|\varphi|\!)^{\mathsf{PFml}}(\vec{V})\big) \\
(\!|\mu u.\, \varphi|\!)^{\mathsf{PFml}}(\vec{V}) &:= \big(\, \mu\big(\, (\!|\varphi|\!)^{\mathsf{PFml}}(\vec{V}, \_) \colon \Omega^{Z_X} \to \Omega^{Z_X}\,\big)\,\big), \\
(\!|\nu u.\, \varphi|\!)^{\mathsf{PFml}}(\vec{V}) &:= \big(\, \nu\big(\, (\!|\varphi|\!)^{\mathsf{PFml}}(\vec{V}, \_) \colon \Omega^{Z_X} \to \Omega^{Z_X}\,\big)\,\big), \\
(\!|\psi|\!)^{\mathsf{PFml}}(\vec{V}) &:= \zeta_1^*((\!|\psi|\!)^{\mathsf{SFml}})
\end{aligned}
$$

and by

$$
\begin{aligned}
(\!|p|\!)^{\mathsf{SFml}} &:= L(p), \\
(\!|\boxdot_\gamma(\psi_1, \dots, \psi_{|\gamma|})|\!)^{\mathsf{SFml}} &:= \gamma\big(\, (\!|\psi_1|\!)^{\mathsf{SFml}},\ \dots,\ (\!|\psi_{|\gamma|}|\!)^{\mathsf{SFml}}\,\big), \\
(\!|\spadesuit_\sigma \varphi|\!)^{\mathsf{SFml}} &:= (\!|\spadesuit_\sigma|\!)((\!|\varphi|\!)^{\mathsf{PFml}}),
\end{aligned}
$$

where

$$
\begin{aligned}
(\!|\heartsuit_\lambda|\!) &:= \zeta_2^* \circ \lambda_{Z_X} \colon \Omega^{Z_X} \to \Omega^{Z_X}, \\
(\!|\spadesuit_\sigma|\!) &:= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X} \colon \Omega^{Z_X} \to \Omega^X.
\end{aligned}
$$

In this interpretation, f^* denotes the pullback of a map f, and $\mu f, \nu f$ denote the least/greatest fixpoint of a monotone function f on a complete lattice.

Definition 3.12 (path-based semantics of CCTL). The *path-based semantics* of a CCTL formula ψ is given by $(\!|\psi|\!)^{\mathsf{SFml}}$.

The following predicate transformers on Ω^{Z_X} facilitates expressing the interpretations of the restricted fixpoints $\spadesuit_\sigma(\mu u.\, \boxdot_{\gamma_\mu}\ (\psi_1, \dots, \psi_{|\gamma_\mu|-1}, \heartsuit_\lambda u))$ and $\spadesuit_\sigma(\nu u.\, \boxdot_{\gamma_\nu}$ $(\psi_1, \dots, \psi_{|\gamma_\nu|-1}, \heartsuit_\lambda u))$ in CCTL.

Definition 3.13. Let $\gamma \in \Gamma$, $S_1, \dots, S_{|\gamma|-1} \in \Omega^{Z_X}$. We define a monotone operator $\Phi^{\lambda,\gamma}_{(S_1,\dots,S_{|\gamma|-1})}$ by

$$
\Phi^{\lambda,\gamma}_{(S_1,\dots,S_{|\gamma|-1})} := \gamma(S_1, \dots, S_{|\gamma|-1}, (\!|\heartsuit_\lambda|\!)(\_)) \colon \Omega^{Z_X} \to \Omega^{Z_X}
$$

whose subscripts we will sometimes omit.

[14] As a minor change of notation, this work employs the semantic symbols $(\!|\_|\!)^{\mathsf{PFml}}$ and $(\!|\_|\!)^{\mathsf{SFml}}$ instead of the previous symbols $[\![\_]\!]^{\mathsf{PFml}}$ and $[\![\_]\!]^{\mathsf{SFml}}$ in [29, Definition 3.9], to highlight the difference from another semantic symbol $[\![\_]\!]$ for μ^{CCTL} formulas in Def. 4.1.

Using this operator, we can directly write down interpretations of CCTL formulas:

$$\begin{aligned}
(\!|p|\!)^{\mathsf{SFml}} &= L(p),\\
(\!|\boxdot_\gamma(\psi_1,\dots,\psi_{|\gamma|})|\!)^{\mathsf{SFml}} &= \gamma\big(\,(\!|\psi_1|\!)^{\mathsf{SFml}},\,\dots,\,(\!|\psi_{|\gamma|}|\!)^{\mathsf{SFml}}\,\big),\\
(\!|\spadesuit_\sigma\heartsuit_\lambda\psi|\!)^{\mathsf{SFml}} &= (\!|\spadesuit_\sigma|\!)\circ(\!|\heartsuit_\lambda|\!)\circ\zeta_1^*((\!|\psi|\!)^{\mathsf{SFml}}),\\
(\!|\spadesuit_\sigma(\mu u.\,\boxdot_{\gamma_\mu}(\psi_1,\dots,\psi_{|\gamma_\mu|-1},\heartsuit_\lambda u))|\!)^{\mathsf{SFml}} &= (\!|\spadesuit_\sigma|\!)\big(\mu\,\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*((\!|\psi_1|\!)^{\mathsf{SFml}}),\dots,\zeta_1^*((\!|\psi_{|\gamma_\mu|-1}|\!)^{\mathsf{SFml}}))}\big),\\
(\!|\spadesuit_\sigma(\nu u.\,\boxdot_{\gamma_\nu}(\psi_1,\dots,\psi_{|\gamma_\nu|-1},\heartsuit_\lambda u))|\!)^{\mathsf{SFml}} &= (\!|\spadesuit_\sigma|\!)\big(\mu\,\Phi^{\lambda,\gamma_\nu}_{(\zeta_1^*((\!|\psi_1|\!)^{\mathsf{SFml}}),\dots,\zeta_1^*((\!|\psi_{|\gamma_\nu|-1}|\!)^{\mathsf{SFml}}))}\big).
\end{aligned}$$

Example 3.14. Using the BT situation $\mathcal{S}_{\mathrm{ND}}$ (recall Example 3.2), we recover classical CTL semantics [13] based on computation paths. The instantiated operators $(\!|\mathsf{E}|\!)$ and $(\!|\mathsf{A}|\!)$ respectively map a predicate Q (on computation paths) to the predicates

$$\begin{aligned}
&\{x \in X \mid \text{there is a computation path } \pi \text{ of } x \text{ with } \pi \in Q\},\\
&\{x \in X \mid \text{every computation path } \pi \text{ of } x \text{ belongs to } Q\}.
\end{aligned}$$

The operator $(\!|\mathsf{X}|\!)$ maps a path predicate Q to the path predicate

$$\{\pi \in X^\omega \mid \text{the tail of } \pi \text{ belongs to } Q\}.$$

Using the BT situation $\mathcal{S}_{\mathrm{R}}$, we can also obtain the PCTL semantics [20]. The instantiated operator $(\!|\mathbb{P}_{\geq q}|\!)$ maps a path predicate Q to

$$\left\{ x \in X \,\middle|\, \begin{array}{l} \text{the probability of computation paths of } x \text{ belonging to } Q \\ \text{is greater than or equal } q \end{array} \right\}.$$

4 Fixpoint Characterization of CCTL

The aim of this section is to give an alternative *step-wise* semantics of CCTL, and prove its equivalence to the path-based semantics (Def. 3.12). The equivalence, *fixpoint characterization*, is crucial in obtaining our polynomial time model-checking algorithm of CCTL formulas in §5.

4.1 A Coalgebraic μ-calculus μ^{CCTL}

We first introduce a fragment μ^{CCTL} of the coalgebraic μ-calculus [21, 41]. The fragment instantiates the coalgebraic μ-calculus using composite modalities $\spadesuit_\sigma\heartsuit_\lambda$, and restricts formulas inside fixpoints to be in a particular form.

Definition 4.1 (the μ-calculus μ^{CCTL}). Let Σ, Λ be sets, and Γ be a ranked alphabet. Given a set AP of atomic propositions, we define the μ-calculus $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ by the following grammar:

$$\begin{aligned}\theta \in \mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu} ::= & \, p \in \mathrm{AP} \mid \boxdot_\gamma(\theta_1,\dots,\theta_{|\gamma|}) \mid \spadesuit_\sigma\heartsuit_\lambda\theta \\ & \mid \mu u.\, \boxdot_{\gamma_\mu}(\theta_1,\dots,\theta_{|\gamma_\mu|-1}, \spadesuit_\sigma\heartsuit_\lambda u) \\ & \mid \nu u.\, \boxdot_{\gamma_\nu}(\theta_1,\dots,\theta_{|\gamma_\nu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)\end{aligned}$$

where u is a proposition variable, $\gamma \in \Gamma$, $\lambda \in \Lambda$, $\sigma \in \Sigma$, and $\gamma_\mu \in \Gamma_\mu, \gamma_\nu \in \Gamma_\nu$.

Note here our $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ has no open formula since any occurrence of variables is bound immediately.

Definition 4.2 (semantics of μ^{CCTL} formulas). For each $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ formula θ, its interpretation $[\![\theta]\!] \in \Omega^X$ is defined by:

$$\begin{aligned}[\![p]\!] &:= L(p),\\ [\![\boxdot_\gamma(\theta_1,\dots,\theta_n)]\!] &:= \gamma\big([\![\theta_1]\!],\dots,[\![\theta_n]\!]\big),\\ [\![\spadesuit_\sigma\heartsuit_\lambda\theta]\!] &:= [\![\spadesuit_\sigma\heartsuit_\lambda]\!]\big([\![\theta]\!]\big),\\ [\![\mu u.\, \boxdot_{\gamma_\mu}(\theta_1,\dots,\theta_{|\gamma_\mu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)]\!] &:= \mu\,\Psi^{(\sigma,\lambda),\gamma_\mu}_{([\![\theta_1]\!],\dots,[\![\theta_{|\gamma_\mu|-1}]\!])},\\ [\![\nu u.\, \boxdot_{\gamma_\nu}(\theta_1,\dots,\theta_{|\gamma_\nu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)]\!] &:= \nu\,\Psi^{(\sigma,\lambda),\gamma_\nu}_{([\![\theta_1]\!],\dots,[\![\theta_{|\gamma_\nu|-1}]\!])},\end{aligned}$$

where we denote monotone functions

$$\begin{aligned}[\![\spadesuit_\sigma\heartsuit_\lambda]\!] &:= c^* \circ \sigma_{FX} \circ \lambda_X \colon \Omega^X \to \Omega^X,\\ \Psi^{(\sigma,\lambda),\gamma}_{(S_1,\dots,S_{|\gamma|-1})} &:= \gamma(S_1,\dots,S_{|\gamma|-1}, [\![\spadesuit_\sigma\heartsuit_\lambda]\!](\_))\colon \Omega^X \to \Omega^X,\end{aligned}$$

whose subscripts we will sometimes omit.

Example 4.3. In $\mathcal{S}_{\mathrm{ND}}$, the operator $[\![\mathsf{EX}]\!]$ maps a predicate P on states to the predicate

$$\{x \in X \mid \text{there is a successor } x' \text{ of } x \text{ with } x' \in Q\}.$$

The operator $[\![\mathsf{AX}]\!]$ is also defined similarly. In $\mathcal{S}_{\mathrm{R}}$, the operator $[\![\mathbb{P}_{\geq q}\mathsf{X}]\!]$ maps a predicate P to the predicate

$$\left\{ x \in X \,\middle|\, \begin{array}{l} \text{the probability of successors of } x \text{ belonging to } Q \\ \text{is greater than or equal to } q \end{array} \right\}.$$

4.2 Step-wise Semantics of CCTL and Fixpoint Characterization

To define the step-wise semantics of CCTL, we first define a bijective translation between μ^{CCTL} formulas and CCTL formulas.

Definition 4.4 (translation of μ^{CCTL} into CCTL). We define a translation ι of $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ formulas θ into $\mathsf{CCTL}_{\Gamma_\mu,\Gamma_\nu}$ formulas by

$$\begin{aligned}
\iota(p) &:= p \\
\iota\big(\boxdot_\gamma(\theta_1,\dots,\theta_{|\gamma|})\big) &:= \boxdot_\gamma(\iota\theta_1,\dots,\iota\theta_{|\gamma|}), \\
\iota\big(\spadesuit_\sigma\heartsuit_\lambda\theta\big) &:= \spadesuit_\sigma\heartsuit_\lambda(\iota\theta), \\
\iota\big(\mu u.\boxdot_{\gamma_\mu}(\theta_1,\dots,\theta_{|\gamma_\mu|-1},\spadesuit_\sigma\heartsuit_\lambda u)\big) &:= \spadesuit_\sigma\big(\mu u.\boxdot_{\gamma_\mu}(\iota\theta_1,\dots,\iota\theta_{|\gamma_\mu|-1},\heartsuit_\lambda u)\big), \\
\iota\big(\nu u.\boxdot_{\gamma_\nu}(\theta_1,\dots,\theta_{|\gamma_\nu|-1},\spadesuit_\sigma\heartsuit_\lambda u)\big) &:= \spadesuit_\sigma\big(\nu u.\boxdot_{\gamma_\nu}(\iota\theta_1,\dots,\iota\theta_{|\gamma_\nu|-1},\heartsuit_\lambda u)\big).
\end{aligned}$$

The translation ι is a bijection between μ^{CCTL} formulas and CCTL formulas. We call the inverse map ι^{-1} the *(fixpoint) encoding* of CCTL into μ^{CCTL}. Via this encoding, the semantics of μ^{CCTL} induces another semantics of CCTL, the step-wise semantics of CCTL.

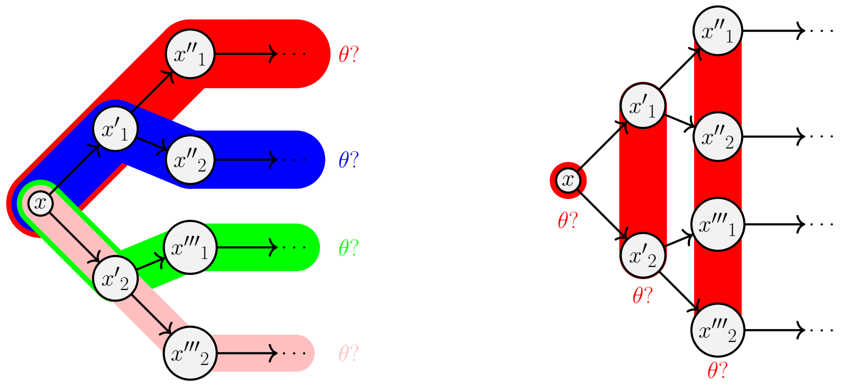
Definition 4.5 (step-wise semantics). The *step-wise semantics* of each CCTL-formula ψ is given by $[\![\iota^{-1}\psi]\!]$.

We will prove the so-called *fixpoint characterization*, which is the equivalence of the path-based semantics (Def. 3.12) and the step-wise semantics (Def. 4.5) of CCTL. The classical fixpoint characterization theorem [14] for CTL asserts, for example, the following equivalence (19). The LHS below is the (path-based) interpretation of the CTL formula $\mathsf{E}(\mu u.\theta \vee \mathsf{X}u)$, and the RHS below is the (step-wise) interpretation of the $\mathbf{L}\boldsymbol{\mu}$ formula that encodes the formula $\mathsf{E}(\mu u.\theta \vee \mathsf{X}u)$.

$$[\![\mathsf{E}(\mu u.\theta \vee \mathsf{X}u)]\!] = [\![\mu u.\theta \vee \mathsf{EX}u]\!]. \tag{19}$$

Fig. 1 illustrates the critical difference between these two interpretations. To verify the CTL formula $\mathsf{E}(\mu u.\theta \vee \mathsf{X}u)$, the path-based semantics (Fig. 1a) searches for a computation path along which the property θ eventually occurs. In contrast, the step-wise semantics (Fig. 1b) searches in a breadth-first manner for a state validating the property θ in the computation tree.

We generalize this classical result to our coalgebraic setting:



(a) Via the path-based semantics (b) Via the step-wise semantics

Fig. 1: Two equivalent interpretations of the CTL formula $\mathsf{E}(\mu u.\theta \vee \mathsf{X}u)$.

Theorem 4.6 (fixpoint characterization). *If the BT situation $\mathcal{S}$ with maximal execution satisfies Assumption 4.7, we have $[\![\theta]\!] = (\!|\iota\theta|\!)^{\mathsf{SFml}}$ for every μ^{CCTL} formula θ, and $[\![\iota^{-1}\psi]\!] = (\!|\psi|\!)^{\mathsf{SFml}}$ for every CCTL formula ψ.*

In this theorem, we identify sufficient conditions on the BT situation in categorical terms so that the fixpoint characterization holds.

Assumption 4.7 (the main assumption).

1. T is an affine monad,
2. the maximal trace $\mathrm{tr}(c')$ satisfies

$$\begin{array}{ccc} X \times TZ_X & \xrightarrow{\mathsf{st}_{X,Z_X}} & T(X \times Z_X) \\ {\scriptstyle \langle \mathrm{id}_X, \mathrm{tr}(c')\rangle}\big\uparrow & & \big\uparrow{\scriptstyle T\langle \zeta_1, \mathrm{id}_{Z_X}\rangle} \\ X & \xrightarrow{\mathrm{tr}(c')} & TZ_X, \end{array} \tag{20}$$

3. for every $\sigma \in \Sigma$, $\mathrm{ev}_\sigma = \sigma_\Omega(\mathrm{id}_\Omega)\colon T\Omega \to \Omega$ is an Eilenberg-Moore T-algebra,
4. for every $\sigma \in \Sigma$, $\lambda \in \Lambda$, and for every μ-scheme $\gamma_\mu \in \Gamma_\mu$ and ν-scheme $\gamma_\nu \in \Gamma_\nu$, we have

$$(\!|\spadesuit_\sigma|\!)\left(\mu\Phi^{\lambda,\gamma_\mu}_{\left(\zeta_1^*((\!|\iota\theta_1|\!)^{\mathsf{SFml}}),\dots,\zeta_1^*((\!|\iota\theta_{|\gamma_\mu|-1}|\!)^{\mathsf{SFml}})\right)}\right) \sqsubseteq \mu\Psi^{(\sigma,\lambda),\gamma_\mu}_{([\![\theta_1]\!],\dots,[\![\theta_{|\gamma_\mu|-1}]\!])} \tag{21}$$

$$(\!|\spadesuit_\sigma|\!)\left(\nu\Phi^{\lambda,\gamma_\nu}_{\left(\zeta_1^*((\!|\iota\theta_1|\!)^{\mathsf{SFml}}),\dots,\zeta_1^*((\!|\iota\theta_{|\gamma_\nu|-1}|\!)^{\mathsf{SFml}})\right)}\right) \sqsupseteq \nu\Psi^{(\sigma,\lambda),\gamma_\nu}_{([\![\theta_1]\!],\dots,[\![\theta_{|\gamma_\nu|-1}]\!])} \tag{22}$$

for every tuple of $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ formulas $\vec{\theta}_{|\gamma|} = (\theta_1,\dots,\theta_{|\gamma|})$, where Ψ, Φ are the operators defined in Def. 3.12 and Def. 4.2,

5. for every $\gamma \in \Gamma_\mu \cup \Gamma_\nu$ and $\sigma \in \Sigma$, $\gamma\colon \Omega^{|\gamma|} \to \Omega$ is $|\gamma|$-linear w.r.t. the T-algebra $\mathrm{ev}_\sigma\colon T\Omega \to \Omega$[15], i.e.,

$$\begin{array}{ccccc} \Omega^n \times T\Omega & \xrightarrow{\mathsf{st}_{\Omega^n,\Omega}} & T(\Omega^n \times \Omega) & \xrightarrow{T\gamma} & T\Omega \\ {\scriptstyle \mathrm{id}_{\Omega^n}\times \mathrm{ev}_\sigma}\downarrow & & & & \downarrow{\scriptstyle \mathrm{ev}_\sigma} \\ \Omega^n \times \Omega & & \xrightarrow{\gamma} & & \Omega \end{array} \tag{23}$$

where $n = |\gamma| - 1$. In the case $|\gamma| = 0$, the above diagram becomes

$$\begin{array}{ccccc} \mathbf{1} \times T\mathbf{1} & \xrightarrow{\mathsf{st}_{\mathbf{1},\mathbf{1}}} & T(\mathbf{1} \times \mathbf{1}) & \xrightarrow{T\gamma} & T\Omega \\ {\scriptstyle \mathrm{id}_{\mathbf{1}}\times !_{T\mathbf{1}}}\downarrow & & & & \downarrow{\scriptstyle \mathrm{ev}_\sigma} \\ \mathbf{1} \times \mathbf{1} & & \xrightarrow{\gamma} & & \Omega, \end{array} \tag{24}$$

6. for every $\sigma \in \Sigma$ and $\lambda \in \Lambda$, the map $\mathrm{ev}_\lambda \circ \mathrm{inj}_\alpha\colon \Omega^{|\alpha|} \to \Omega$ is bilinear w.r.t. ev_σ, where $\mathrm{inj}_\alpha\colon \Omega^{|\alpha|} \to \coprod_{\alpha\in A} \Omega^{|\alpha|}$ is the injection of the index α.

Let us explain each condition in Assumption 4.7.

1. This condition asserts absence of deadlock states. Technically, it is needed here to ensure the compatibility of the strength map of T with the first projection (that is, $T\pi_1 \circ \mathsf{st}_{X,Y} = \eta_X \circ \pi_2$), which, in turn, ensures that the original $T \circ F$-coalgebra structure can be recovered from its execution map $\mathrm{tr}(c')$ (that is, Lem. 4.8).
2. This condition is quite technical but harmless and used to prove one of our key results, Prop. 4.9. A similar condition can be found in [25], as *strong affine-ness*. Indeed, we can show every strongly affine monad satisfies condition 2. Since both $\mathcal{P}^+$ and $\mathcal{G}_1$ are strongly affine, condition 2 is satisfied by both $\mathcal{S}_{\mathrm{ND}}$ and $\mathcal{S}_{\mathrm{R}}$ (see Table 2).
3. This condition, especially the associativity of the Eilenberg-Moore T-algebra ev_σ, enables us to reduce many-fold branching (i.e., several applications of the path quantifier σ) to single branching (i.e., just one application).
4. This condition states that the path quantifier σ preserves the least/greatest fixpoints of the operators Ψ, Φ. In the logical perspective, the inequality (21) means "any path-based witness can be reached in step-wise manner," and the inequality (22) means "step-wise validity guarantees path-based validity."
5. This condition expresses that each application of a path quantifier $\spadesuit_\sigma$ on a formula of the form $\boxdot_\gamma(\vec{\psi}, \heartsuit_\lambda\varphi)$ is calculated by passing $\spadesuit_\sigma$ *inside*, as $\boxdot_\gamma(\vec{\psi}, \spadesuit_\sigma\heartsuit_\lambda\varphi)$.
6. This condition captures the coherence between path quantifiers $\sigma \in \Sigma$ and next-time operators $\lambda \in \Lambda$. If we choose the canonical predicate lifting $\mathrm{Pred}(F)$ as λ, this condition is a consequence of condition 5, because $\mathrm{Pred}(F)$ is constructed from conjunction, i.e., $\mathrm{ev}_\lambda \circ \mathrm{inj}_\alpha = \wedge$; see Example 2.11 (1).

[15] While we assumed γ to be bilinear in [29, Assumption 4.7], this restriction can be harmlessly weakened to $|\gamma|$-linear as above.

Before starting the proof of Thm. 4.6, we introduce two important results, whose proofs are given in appendix A.

The first one (Lem. 4.8) is a consequence of condition 1 of Assumption 4.7, and states that taking the head (ζ_1) of the tail (ζ_2) of paths starting from a state x yields successors of x.

Lemma 4.8. $T(F\zeta_1 \circ \zeta_2) \circ \mathrm{tr}(c') = c$.

The second one (Prop. 4.9) is a coalgebraic generalization of the *expansion law* [1] of CTL. When instantiated to the CTL formula $\mathsf{E}(p_1 \mathsf{U} p_2)$, it means

$$[\![\mathsf{E}(p_1 \mathsf{U} p_2)]\!] = [\![p_2]\!] \cup ([\![p_1]\!] \cap [\![\mathsf{EX}]\!][\![\mathsf{E}(p_1 \mathsf{U} p_2)]\!]).$$

Analogous to the classical one, our coalgebraic expansion law is critically used in the induction in the proof of the fixpoint characterization. It depends on all conditions of Assumption 4.7 but condition 4.

Proposition 4.9 (coalgebraic expansion law). *Let $\sigma \in \Sigma$, $\lambda \in \Lambda$, and μ-schemes $\gamma_\mu \in \Gamma_\mu$ and ν-schemes $\gamma_\nu \in \Gamma_\nu$. We have*

$$(\!|\spadesuit_\sigma|\!)\big(\mu\Phi^{\lambda,\gamma_\mu}_{\left(\zeta_1^*((\!|\iota\theta_1|\!)^{\mathsf{SFml}}),\ldots,\zeta_1^*((\!|\iota\theta_{|\gamma_\mu|-1}|\!)^{\mathsf{SFml}})\right)}\big) \sqsupseteq \mu\Psi^{(\sigma,\lambda),\gamma_\mu}_{([\![\theta_1]\!],\ldots,[\![\theta_{|\gamma_\mu|-1}]\!])} \tag{25}$$

for $\theta_1, \ldots, \theta_{|\gamma_\mu|-1}$ with $(\!|\iota\theta_i|\!)^{\mathsf{SFml}} \sqsupseteq [\![\theta_i]\!]$ for $i = 1, \ldots, |\gamma_\mu| - 1$, and

$$(\!|\spadesuit_\sigma|\!)\big(\nu\Phi^{\lambda,\gamma_\nu}_{\left(\zeta_1^*((\!|\iota\theta_1|\!)^{\mathsf{SFml}}),\ldots,\zeta_1^*((\!|\iota\theta_{|\gamma_\nu|-1}|\!)^{\mathsf{SFml}})\right)}\big) \sqsubseteq \nu\Psi^{(\sigma,\lambda),\gamma_\nu}_{([\![\theta_1]\!],\ldots,[\![\theta_{|\gamma_\nu|-1}]\!])} \tag{26}$$

for $\theta_1, \ldots, \theta_{|\gamma_\nu|-1}$ with $(\!|\iota\theta_i|\!)^{\mathsf{SFml}} \sqsubseteq [\![\theta_i]\!]$ for $i = 1, \ldots, |\gamma_\nu| - 1$. Furthermore, if $(\!|\iota\theta_i|\!)^{\mathsf{SFml}} = [\![\theta_i]\!]$ for every subformula θ_i, the inequalities 25 and 26 are both equalities.

Proof (Thm. 4.6). Since ι is a bijection between μ^{CCTL} and CCTL, it suffices to show

$$[\![\theta]\!] = (\!|\iota\theta|\!)^{\mathsf{SFml}} \tag{27}$$

for every $\theta \in \mu^{\mathsf{CCTL}}$. We prove equation 27 by induction on the construction of θ.

For $\theta = p \in \mathrm{AP}$ or $\theta = \boxdot_\gamma(\theta_1, \ldots, \theta_{|\gamma|})$, equation 27 is straightforward.

For $\theta = \spadesuit_\sigma \heartsuit_\lambda \theta'$, by the induction hypothesis, we have $(\!|\iota\theta'|\!)^{\mathsf{SFml}} = [\![\theta']\!]$. Thus, we obtain, by Def. 3.11,

$$\begin{aligned}
(\!|\iota(\spadesuit_\sigma \heartsuit_\lambda \theta')|\!)^{\mathsf{SFml}} &= (\!|\spadesuit_\sigma \heartsuit_\lambda \iota\theta'|\!)^{\mathsf{SFml}} \\
&= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X}((\!|\heartsuit_\lambda \iota\theta'|\!)^{\mathsf{PFml}}) \\
&= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X}(\zeta_2^* \circ \lambda_{Z_X}(\!|\iota\theta'|\!)^{\mathsf{PFml}}) \\
&= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X}\big(\zeta_2^* \circ \lambda_{Z_X}(\zeta_1^*(\!|\iota\theta'|\!)^{\mathsf{SFml}})\big) \\
&= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X} \circ \zeta_2^* \circ \lambda_{Z_X} \circ \zeta_1^*([\![\theta']\!]).
\end{aligned}$$

Using naturality of λ and σ, the above equation is

$$\begin{aligned}(\!|\iota(\spadesuit_\sigma\heartsuit_\lambda\theta')|\!)^{\mathsf{SFml}} &= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X} \circ \zeta_2^* \circ F\zeta_1^* \circ \lambda_X([\![\theta']\!]) \\ &= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X} \circ (F\zeta_1 \circ \zeta_2)^* \circ \lambda_X([\![\theta']\!]) \\ &= \big(\mathrm{tr}(c')\big)^* \circ \big(T(F\zeta_1 \circ \zeta_2)\big)^* \circ \sigma_{FX} \circ \lambda_X([\![\theta']\!]) \\ &= \big(T(F\zeta_1 \circ \zeta_2) \circ \mathrm{tr}(c')\big)^* \circ \sigma_{FX} \circ \lambda_X([\![\theta']\!]).\end{aligned}$$

Since $T(F\zeta_1 \circ \zeta_2) \circ \mathrm{tr}(c') = c$ by Lem. 4.8, we finally have

$$\begin{aligned}(\!|\iota(\spadesuit_\sigma\heartsuit_\lambda\theta')|\!)^{\mathsf{SFml}} &= c^* \circ \sigma_{FX} \circ \lambda_X([\![\theta']\!]) \\ &= [\![\spadesuit_\sigma\heartsuit_\lambda\theta']\!].\end{aligned}$$

Next, we move on to the case $\theta = \mu u.\, \boxdot_{\gamma_\mu} (\theta_1, \dots, \theta_{|\gamma_\mu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)$. Firstly, we hypothesize $\theta_1, \dots, \theta_{|\gamma_\mu|-1}$ with $(\!|\iota\theta_i|\!)^{\mathsf{SFml}} = [\![\theta_i]\!]$ for $i = 1, \dots, |\gamma_\mu|-1$. Under the notation introduced in Def. 3.12 and Def. 4.2, we have

$$\begin{aligned}[\![\mu u.\, \boxdot_{\gamma_\mu} (\theta_1, \dots, \theta_{|\gamma_\mu|}, \spadesuit_\sigma\heartsuit_\lambda u)]\!] &= \mu\Psi^{(\sigma,\lambda),\gamma_\mu}_{([\![\vec{\theta}]\!])}, \\ (\!|\iota\big(\mu u.\, \boxdot_{\gamma_\mu} (\theta_1, \dots, \theta_{|\gamma_\mu|}, \spadesuit_\sigma\heartsuit_\lambda u)\big)|\!)^{\mathsf{SFml}} &= (\!|\spadesuit_\sigma|\!)(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^* (\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})}).\end{aligned}$$

Thus, the last task is to prove $(\!|\spadesuit_\sigma|\!)(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^* (\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})}) = \mu\Psi^{(\sigma,\lambda),\gamma_\mu}_{([\![\vec{\theta}]\!])}$. The direction LHS $\sqsubseteq$ RHS is already assumed in condition 4 of Assumption 4.7.

We show the other direction, LHS $\sqsupseteq$ RHS. To prove this, we recall the Knaster-Tarski fixpoint theorem [38]: the least fixpoint of a monotone function on a complete lattice is exactly the minimal of all pre-fixpoints of the function. Since LHS is a pre-fixpoint of the operator Ψ by Prop. 4.9, we conclude LHS $\sqsupseteq$ RHS by the Knaster-Tarski fixpoint theorem.

The proof for the case $\theta = \nu u.\, \boxdot_{\gamma_\nu} (\theta_1, \dots, \theta_{|\gamma_\nu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)$ is similar to the least fixpoint case since condition 4 of Assumption 4.7 is symmetric to μ and ν. □

Examining the above proof, we can also obtain a *partial* fixpoint characterization.

Proposition 4.10 (partial fixpoint characterization). *Under the same assumption of Thm. 4.6 (Assumption 4.7) but without condition 4, we have*

1. $[\![\theta]\!] = (\!|\iota\theta|\!)^{\mathsf{SFml}}$ *for a formula* θ *without any* μ *or* ν*,*
2. $[\![\theta]\!] \sqsubseteq (\!|\iota\theta|\!)^{\mathsf{SFml}}$ *for a formula* θ *with only* μ*'s,*
3. $[\![\theta]\!] \sqsupseteq (\!|\iota\theta|\!)^{\mathsf{SFml}}$ *for a formula* θ *with only* ν*'s.*

The implication of Prop. 4.10 is that: although item 3 says we can only over-approximate simple safety properties in CCTL (CCTL formulas with only the μ operators), item 2 says the liveness properties in CCTL can be *under-approximated* by the fixpoint modal logic μ^{CCTL} via the fixpoint encoding $\iota^{-1}\colon \mathsf{CCTL} \to \mu^{\mathsf{CCTL}}$ and Thm. 4.6. Since techniques on under-approximation of temporal properties are of interest in formal verification, Prop. 4.10 has (at least limited) applicability for checking practical liveness properties over systems where the (full) fixpoint characterizasion does not hold. As an example of such a system, we will see the BT situation $\mathcal{S}_{\mathrm{qR}}$ of qualitative reliability validates the partial fixpoint characterization (Prop. 4.10) in Prop. 4.13 while it breaks Thm. 4.6.

4.3 Examples and Non-examples of Assumption 4.7

The non-deterministic BT situation $\mathcal{S}_{\mathrm{ND}}$ satisfies Assumption 4.7, as expected.

Proposition 4.11. *$\mathcal{S}_{ND}$ satisfies Assumption 4.7 with $\Gamma_\mu = \{(\_ \vee (\_ \wedge \_))\}$ and $\Gamma_\nu = \{(\_ \wedge (\_ \vee \_))\}$. Thus, $\mathcal{S}_{ND}$ enjoys the fixpoint characterization (Thm. 4.6).*

Proof. We prove conditions of Assumption 4.7.

On condition 1, we already saw the non-empty powerset monad $\mathcal{P}^+$ is affine in Example 2.5.

On condition 2, we have

$$\begin{aligned}\mathsf{st}_{X,Z_X} \circ \langle \mathrm{id}_X, \mathrm{tr}(c')\rangle(x) &= \{(x,z) \mid z \in \mathrm{tr}(c')(x)\},\\ \mathcal{P}^+\langle \zeta_1, \mathrm{id}_{Z_X}\rangle \circ \langle \mathrm{id}_X, \mathrm{tr}(c')\rangle(x) &= \{(\zeta_1(z), z) \mid z \in \mathrm{tr}(c')(x)\}.\end{aligned}$$

Thus, condition 2 comes from $\zeta_1(z) = x$ for $z \in \mathrm{tr}(c')(x)$. This equality can be obtained by Lem. A.5 in appendix A. [16] Indeed, we have

$$\begin{aligned}\{\zeta_1(z) \mid z \in \mathrm{tr}(c')(x)\} &= \mathcal{P}^+(\zeta_1) \circ \mathrm{tr}(c')(x)\\ &= \eta_X(x)\\ &= \{x\}.\end{aligned}$$

Condition 4 is instantiated in $\mathcal{S}_{\mathrm{ND}}$ as

$$\begin{aligned}[\![\mathsf{E}(\theta_1 \mathsf{U} \theta_2)]\!] &\subseteq \mu u.\, \theta_2 \cup (\theta_1 \cap [\![\mathsf{EX}]\!] u)\\ [\![\mathsf{E}(\theta_1 \mathsf{W} \theta_2)]\!] &\supseteq \nu u.\, \theta_1 \cap (\theta_2 \cup [\![\mathsf{EX}]\!] u)\end{aligned}$$

for E (the A case is given likewise). Proof of these inequalities is presented in [1, Thm. 6.23]. Note that although there the state set is assumed to be finite, this assumption can be lifted: the proof uses the expansion law, but the law can be obtained by checking the conditions of Assumption 4.7 other than 4 (recall our proof of the coalgebraic expansion law does not depend on condition 4). The rest of the proof in [1] can be done without the finiteness assumption.

On condition 3, we have to show the diamond $\Diamond$ and box modalities $\Box$ are Eilenberg-Moore $\mathcal{P}^+$-algebra, which is shown in [6]. Note that, whereas the diamond modality is also an Eilenberg-Moore $\mathcal{P}$-algebra,the box modality is not.

[16] We can use Lem. A.5 here because it depends only on condition 1 and we proved affine-ness of $\mathcal{P}^+$.

On condition 5, we have four connectives $\bot$, $\top$, $\vee$ and $\wedge$. We can easily check the 0-ary operators $\bot$ and $\top$ satisfy diagram 24 in condition 5. We here prove diagram 23 for the conjunction $\wedge$ case with the diamond modality $\Diamond$: other cases, $\wedge$ with $\Box$ and $\vee$ with $\Diamond$ and $\Box$, are calculated quite similarly. Let $t \in \mathbf{2}$ and $S \in \mathcal{P}^{+}\mathbf{2}$. By concrete calculation, we have

$$\Diamond \circ \mathcal{P}^{+}(\wedge) \circ \mathsf{st}_{\mathbf{2},\mathbf{2}}(t, S) = \Diamond(\{t \wedge s \mid s \in S\}),$$
$$\wedge \circ (\mathrm{id}_{\mathbf{2}} \times \Diamond)(t, S) = t \wedge \Diamond(S).$$

For every (non-empty) subset S, these two expressions coincide.

On condition 6, as we mentioned after Assumption 4.7, the canonical predicate lifting $\mathrm{Pred}(F)$ for the polynomial F is bilinear w.r.t. ev_σ if boolean operators are so. Thus, condition 6 follows from validity of condition 5 above. $\Box$

On the other hand, the probabilistic BT situations ($\mathcal{S}_{\mathrm{R}}$ and $\mathcal{S}_{\mathrm{qR}}$) fail to satisfy some conditions of Assumption 4.7, and hence to have the fixpoint characterization.

Fact 4.12. *$\mathcal{S}_R$ and $\mathcal{S}_{qR}$ do not satisfy Assumption 4.7.*

Firstly, $\mathcal{S}_{\mathrm{R}}$ does not satisfy condition 3 of Assumption 4.7, i.e., the requirement for the $\mathcal{G}_1$-modality $\sigma = \geq_q : \mathcal{G}_1(\mathbf{2}, \mathcal{P}\mathbf{2}) \to (\mathbf{2}, \mathcal{P}\mathbf{2})$ to be an Eilenberg-Moore $\mathcal{G}_1$-algebra in **SB**. Indeed, the modality breaks the associativity condition of Eilenberg-Moore $\mathcal{G}_1$-algebras. The associativity means the following diagram commutes for every $\rho \in \mathcal{G}_1(\mathcal{G}_1(\mathbf{2}, \mathcal{P}\mathbf{2})) \cong \mathcal{G}_1([0,1], \Sigma_{[0,1]})$, where $\Sigma_{[0,1]}$ is the Borel set generated from the usual topology of $[0, 1]$:

$$\begin{array}{ccc}
\mathcal{G}_1(\mathcal{G}_1(\mathbf{2}, \mathcal{P}\mathbf{2})) \cong \mathcal{G}_1([0,1], \Sigma_{[0,1]}) & \xrightarrow{\mathcal{G}_1(\geq_q)} & \mathcal{G}_1(\mathbf{2}, \mathcal{P}\mathbf{2}) \cong ([0,1], \Sigma_{[0,1]}) \\
\downarrow{\scriptstyle \mu_{(\mathbf{2},\mathcal{P}\mathbf{2})}} & & \downarrow{\scriptstyle \geq_q} \\
\mathcal{G}_1(\mathbf{2}, \mathcal{P}\mathbf{2}) \cong ([0,1], \Sigma_{[0,1]}) & \xrightarrow[\geq_q]{} & (\mathbf{2}, \mathcal{P}\mathbf{2}).
\end{array}$$

The commutativity of this diagram can be further rephrased as follows: the condition $\rho([q, 1]) \geq q$ is equivalent to $\int_{r \in [0,1]} \rho(r)\, dr \geq q$ for every measure ρ. However, by taking a real number q other than 0 or 1, this equivalence fails. Thus, the associativity condition of Eilenberg-Moore $\mathcal{G}_1$-algebras also fails for q other than 0 or 1.

This suggests that by restricting the modality parameter q to 0 or 1, we can make condition 3 hold. This restriction is realized by the BT situation $\mathcal{S}_{\mathrm{qR}}$ (see Example 3.2).

Nevertheless, for $\mathcal{S}_{\mathrm{qR}}$, condition 4 of Assumption 4.7 is violated. The violation can be seen in a simple counterexample shown in Fig. 2 (found in [3]). While the PCTL formula $\mathbb{P}_{\geq 1}(\mu u.p \vee \mathsf{X}u)$ is interpreted as $\{x, y\}$ in this example, the encoded probabilistic mu-formula $\mu u.p \vee \mathbb{P}_{\geq 1}\mathsf{X}u$ is interpreted as $\{y\}$. Thus, we have $\mathbb{P}_{\geq 1}(\mu u.p \vee \mathsf{X}u) \sqsupset \mu u.p \vee \mathbb{P}_{\geq 1}\mathsf{X}u$, which breaks condition 4 of Assumption 4.7.

Nonetheless, we also have the following positive result.

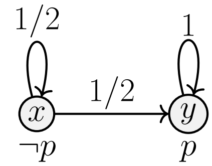


Fig. 2: A counterexample Markov chain.

Proposition 4.13. *$\mathcal{S}_{qR}$ with its state space $(X, \mathcal{P}X)$ for a countable set X satisfies the other conditions of Assumption 4.7 than condition 4 with $\Gamma_\mu = \{(\_ \vee (\_ \wedge \_))\}$ and $\Gamma_\nu = \{(\_ \wedge (\_ \vee \_))\}$ Thus, $\mathcal{S}_{qR}$ with countable $(X, \mathcal{P}X)$ enjoys the partial fixpoint characterization (Prop. 4.10).*

Proof. On condition 1, we already saw the Giry monad $\mathcal{G}_1$ is affine in Example 2.5.

On condition 2, by Example 3.7, we have, for $x \in X$,

$$\begin{aligned}
\mathsf{st}_{X,Z_X} \circ \langle \mathrm{id}_X, \mathrm{tr}(c') \rangle (x) &= \delta_x \times \big(\mathrm{tr}(c')(x)\big) \\
&= \lambda S \in \Sigma_{X \times Z_X}.\ \mathrm{tr}(c')(x)(S_x), \\
\mathcal{G}_1 \langle \zeta_1, \mathrm{id}_{Z_X} \rangle \circ \langle \mathrm{id}_X, \mathrm{tr}(c') \rangle (x) &= \lambda S \in \Sigma_{X \times Z_X}.\ \mathrm{tr}(c')(x)\big(\langle \zeta_1, \mathrm{id}_{Z_X} \rangle^{-1}(S)\big) \\
&= \lambda S \in \Sigma_{X \times Z_X}.\ \mathrm{tr}(c')(x)\big(\{z \in Z_X \mid (\zeta_1(z), z) \in S\}\big),
\end{aligned}$$

where λ is the lambda function notation, $\Sigma_{X \times Z_X}$ is the canonical measurable structure on the product $X \times Z_X$, and S_x is the x-section $\{z \in Z_X \mid (x, z) \in S\}$ of S. We want

$$\mathrm{tr}(c')(x)(S_x) = \mathrm{tr}(c')(x)\big(\{z \in Z_X \mid (\zeta_1(z), z) \in S\}\big)$$

for every measurable set $S \in \Sigma_{X \times Z_X}$. Since X is supposed to be countable, we have a countable sum

$$\{z \in Z_X \mid (\zeta_1(z), z) \in S\} = \bigcup_{y \in X} \{z \in Z_X \mid y = \zeta_1(z) \text{ and } (y, z) \in S\}.$$

Note that this countable sum is indeed a disjoint sum. By Lem. A.5 in appendix A, we have

$$\begin{aligned}
\delta_x &= \eta_X(x) \\
&= \mathcal{G}_1(\zeta_1) \circ \mathrm{tr}(c')(x) \\
&= \lambda A \in \Sigma_X.\ \mathrm{tr}(c')(x)\big(\zeta_1^{-1}(A)\big).
\end{aligned}$$

Since $\{x\} \in \Sigma_X = \mathcal{P}X$, we have, for $B \in \Sigma_{Z_X}$,

$$\begin{aligned}
B \subseteq \zeta_1^{-1}(\{x\}) &\Longrightarrow \mathrm{tr}(c')(x)(B) = 1, \\
B \not\subseteq \zeta_1^{-1}(\{x\}) &\Longrightarrow \mathrm{tr}(c')(x)(B) = 0.
\end{aligned}$$

Thus, by sigma-additivity of probability measures, we have

$$\begin{aligned}
&\mathrm{tr}(c')(x)\big(\bigcup_{y\in X}\{z\in Z_X \mid y=\zeta_1(z) \text{ and } (y,z)\in S\}\big)\\
&=\sum_{y\in X}\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid y=\zeta_1(z) \text{ and } (y,z)\in S\}\big)\\
&=\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid x=\zeta_1(z) \text{ and } (x,z)\in S\}\big).
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
&\mathrm{tr}(c')(x)\big(S_x\big)\\
&=\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid x=\zeta_1(z) \text{ and } (x,z)\in S\}\cup\{z\in Z_X \mid x\neq\zeta_1(z) \text{ and } (x,z)\in S\}\big)\\
&=\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid x=\zeta_1(z) \text{ and } (x,z)\in S\}\big)+\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid x\neq\zeta_1(z) \text{ and } (x,z)\in S\}\big)\\
&=\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid x=\zeta_1(z) \text{ and } (x,z)\in S\}\big).
\end{aligned}$$

Finally, we obtain

$$\begin{aligned}
&\mathrm{tr}(c')(x)\big(S_x\big)\\
&=\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid x=\zeta_1(z) \text{ and } (x,z)\in S\}\big)\\
&=\mathrm{tr}(c')(x)\big(\bigcup_{y\in X}\{z\in Z_X \mid y=\zeta_1(z) \text{ and } (y,z)\in S\}\big)\\
&=\mathrm{tr}(c')(x)\big(\{z\in Z_X \mid (\zeta_1(z),z)\in S\}\big).
\end{aligned}$$

On condition 3, we saw this in §4.3.

On condition 5, it suffices to check it for four connectives $\perp$, $\top$, $\vee$ and $\wedge$. We can easily check the 0-ary operators $\perp$ and $\top$ satisfy diagram 24 in condition 5. We here prove diagram 23 for conjunction $\wedge$ and the modality $\geq_1$ since the $\vee$ and $>_0$ case can be seen in the same manner. We want the diagram

$$\begin{array}{ccccc}
(\mathbf{2},\mathcal{P}\mathbf{2})\times\mathcal{G}_1(\mathbf{2},\mathcal{P}\mathbf{2}) & \xrightarrow{\mathsf{st}_{(\mathbf{2},\mathcal{P}\mathbf{2}),(\mathbf{2},\mathcal{P}\mathbf{2})}} & \mathcal{G}_1((\mathbf{2},\mathcal{P}\mathbf{2})\times(\mathbf{2},\mathcal{P}\mathbf{2})) & \xrightarrow{\mathcal{G}_1(\wedge)} & \mathcal{G}_1(\mathbf{2},\mathcal{P}\mathbf{2})\\
\Big\downarrow{\scriptstyle \mathrm{id}_{(\mathbf{2},\mathcal{P}\mathbf{2})}\times\geq_1} & & & & \Big\downarrow{\scriptstyle\geq_1}\\
(\mathbf{2},\mathcal{P}\mathbf{2})\times(\mathbf{2},\mathcal{P}\mathbf{2}) & & \xrightarrow{\qquad\qquad\wedge\qquad\qquad} & & (\mathbf{2},\mathcal{P}\mathbf{2})
\end{array}$$

to commute. Each path of this diagram can be calculated as

$$\geq_q \circ\mathcal{G}_1(\wedge)\circ\mathsf{st}_{(\mathbf{2},\mathcal{P}\mathbf{2}),(\mathbf{2},\mathcal{P}\mathbf{2})}(t,r)=\begin{cases}\geq_1(r) & t=1\\ 0 & t=0\end{cases},$$

$$\wedge\circ(\mathrm{id}_{(\mathbf{2},\mathcal{P}\mathbf{2})}\times\geq_q)(t,r)=t\wedge\geq_1(r)$$

for $t\in\mathbf{2}$ and $r\in\mathcal{M}_{(\mathbf{2},\mathcal{P}\mathbf{2})}\cong[0,1]$. These coincide since $\geq_1(r)$ means $r\geq 1$ by the definition of the modality $\geq_1$ (Example 2.11).

On condition 6, by the same reason as we mentioned in the proof of Prop. 4.11, condition 6 follows from condition 5, which we proved now. □

Remark 4.14. We saw we can not construct a step-wise semantics of PCTL equivalent to its path-based one via our encoding (Def. 4.4). In fact, we can make a stronger statement: *any* fixpoint encoding of PCTL into the probabilistic mu-calculus does not preserve semantics. Indeed, PCTL does not have the finite model property [3], whereas the probabilistic mu-calculus does [8]. One example of PCTL formula with no finite model is $\mathbb{P}_{>0}\mathsf{G}(\neg p \wedge \mathbb{P}_{>0}\mathsf{F}p)$ for any atomic predicate p.

5 A Polynomial-time Model-Checking Algorithm for CCTL

Thanks to the fixpoint characterization, we can obtain a polynomial-time model-checking algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ for CCTL. It is based on the standard model-checking algorithm for CTL [9]. Nevertheless, the algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ is described in categorical terms, with the following additional conditions on the BT situation $\mathcal{S}$.

Assumption 5.1.

1. The ambient category $\mathbb{C}$ is concrete [31].
2. The underlying set of X is finite, with its size denoted by $|X|$.
3. The underlying set of Ω is $\mathbf{2}$.

By Assumption 5.1, we can identify Ω-predicates with subsets of the underlying set of X and the maps γ and $[\![\spadesuit_\sigma\heartsuit_\lambda]\!]$ with corresponding predicate transformers.

Given the BT situation $\mathcal{S}$ and a specification $\psi \in \mathsf{CCTL}$, the algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ calculates $(\!|\psi|\!)^{\mathsf{SFml}}$, which is the interpretation of ψ. The calculation steps are shown in Algorithm 1. Firstly, the CCTL formula ψ is encoded into a μ^{CCTL} formula $\iota^{-1}\psi$ (cf. Def. 4.4). Next, the μ^{CCTL} formula $\iota^{-1}\psi$ is passed to the procedure CHECK($\iota^{-1}\psi$), which is the core of $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$. The procedure call calculates $[\![\iota^{-1}\psi]\!]$ in a step-wise manner. The calculation result coincides with $(\!|\psi|\!)^{\mathsf{SFml}}$ by the fixpoint characterization (Thm. 4.6).

The procedure CHECK(θ) is a simplification of an existing model-checking algorithm for the coalgebraic μ-calculus $\mathbf{C}\boldsymbol{\mu}$ [21]. In the body of CHECK(θ), one out of four cases is chosen according to the structure of θ. The first two cases, one for boolean operators and one for modalities, are straightforward. In the least fixpoint case, we exploit the Cousot-Cousot fixpoint theorem [10], which approximates the least fixpoint by an ascending chain in Ω^X starting from the least element $\perp$. The greatest fixpoint case is similar to the least fixpoint case.

Termination of CHECK(θ), and hence $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ as a whole, is a direct consequence of our finiteness assumption in Assumption 5.1. The encoding ι^{-1} is also terminating. Correctness, particularly that of the two while loops (at Line 14 and Line 22), follows from the Cousot-Cousot fixpoint theorem.

Proposition 5.2 (termination and correctness of $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$). *For a given* CCTL *formula ψ, the algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ terminates and returns $(\!|\psi|\!)^{\mathsf{SFml}}$.*

Proof. We check termination and correctness of $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ simultaneously.

Firstly, the encoding ι^{-1} is terminating by its definition. Correctness of ι^{-1} is assured in Thm. 4.6.

Next, we check termination and correctness of the procedure CHECK(θ), i.e., whether CHECK(θ) calculates $[\![\theta]\!]$ for a given μ^{CCTL} formula θ in finite steps. Among the five cases inside CHECK(θ), the $\boxdot_\gamma$ case and the $\spadesuit_\sigma\heartsuit_\lambda$ case are clear, see Def. 4.2.

We move on to the μ case. Hypothesizing CHECK(θ_i) $= [\![\theta_i]\!]$ for every subformula θ_i ($i = 1, \ldots, |\gamma_\mu|-1$), the procedure CHECK($\mu u. \boxdot_{\gamma_\mu}(\theta_1, \ldots, \theta_{|\gamma_\mu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)$) calculates

Algorithm 1 A CCTL model-checking algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$.

Input: A CCTL formula ψ.
Output: An Ω-predicate $U \in \Omega^X$. ▷ where $\mathcal{S} = (\mathbb{C}, T, F, c, \Omega, \Sigma, \Lambda)$.

1: **procedure** CHECK(θ)
2: **switch** θ **do**

3: **case** p
4: **return** $L(p)$
5: **end case**

6: **case** $\boxdot_\gamma(\theta_1, \dots, \theta_{|\gamma|})$
7: **return** $\gamma(\mathrm{CHECK}(\theta_1), \dots, \mathrm{CHECK}(\theta_{|\gamma|}))$
8: **end case**

9: **case** $\spadesuit_\sigma \heartsuit_\lambda \theta'$
10: **return** $[\![\spadesuit_\sigma \heartsuit_\lambda]\!](\mathrm{CHECK}(\theta'))$
11: **end case**

12: **case** $\mu u. \boxdot_\gamma\ (\theta_1, \dots, \theta_{|\gamma_\mu|-1}, \spadesuit_\sigma \heartsuit_\lambda u)$
13: $U := \bot$; $V := \gamma_\mu\big(\mathrm{CHECK}(\theta_1), \dots, \mathrm{CHECK}(\theta_{|\gamma_\mu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](\bot)\big)$
14: **while** $U \neq V$ **do**
15: $U := V$
16: $V := \gamma_\mu\big(\mathrm{CHECK}(\theta_1), \dots, \mathrm{CHECK}(\theta_{|\gamma_\mu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](U)\big)$
17: **end while**
18: **return** U
19: **end case**

20: **case** $\nu u. \boxdot_{\gamma_\nu}\ (\theta_1, \dots, \theta_{|\gamma_\nu|-1}, \spadesuit_\sigma \heartsuit_\lambda u)$
21: $U := \top$; $V := \gamma_\nu\big(\mathrm{CHECK}(\theta_1), \dots, \mathrm{CHECK}(\theta_{|\gamma_\nu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](\top)\big)$
22: **while** $U \neq V$ **do**
23: $U := V$
24: $V := \gamma_\nu\big(\mathrm{CHECK}(\theta_1), \dots, \mathrm{CHECK}(\theta_{|\gamma_\nu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](U)\big)$
25: **end while**
26: **return** U
27: **end case**

28: **end procedure**
29: **return** CHECK($\iota^{-1}\psi$)

the chain

$$\bot \sqsubseteq \Psi(\bot) \sqsubseteq \cdots \sqsubseteq \Psi^n(\bot) \sqsubseteq \dots \tag{28}$$

in Ω, where Ψ represents $\gamma_\mu\big([\![\theta_1]\!], \dots, [\![\theta_{|\gamma_\mu|-1}]\!], [\![\spadesuit_\sigma \heartsuit_\lambda]\!](\_)\big)$ (in accordance with the notation of Def. 4.2). Indeed, Line 13 initializes U as $\bot$ and V as $\Psi(\bot) = \Psi(U) = \gamma([\![\theta_1]\!], \dots, [\![\theta_{|\gamma_\mu|-1}]\!], [\![\spadesuit_\sigma \heartsuit_\lambda]\!]\bot)$. Each while loop (Line 14) first sets $U := V$ (Line 15). Then Line 16 updates V to $\Psi(V)$. Thus, at the end of each iteration of the while loop, the equality $U = \Psi(V)$ is an invariant, which means U at the end of n-th iteration is exactly $\Psi^n(\bot)$.

The Cousot-Cousot theorem [10] assures chain 28 approximates the least fixpoint of the monotone function Ψ. Since this least fixpoint $\mu\Psi$ coincides with $[\![\theta]\!]$ by Def. 4.2, the while loop (Line 14) returns $[\![\theta]\!]$ if it terminates (correctness). By the finiteness of X, as imposed in Assumption 5.1, the while loop (Line 14) indeed terminates (termination): the number of its iteration steps is bounded by $|X|$.

The ν case is treated in the same way as the μ case. □

To estimate the complexity bound of our algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$, we focus on the time to compute each composite modality $\spadesuit_\sigma\heartsuit_\lambda$ for a given state and predicate. This computation problem is called *one-step satisfaction problem* in [22].

Definition 5.3 (one-step satisfaction problem, [22, Def. 2]). The *one-step satisfaction problem* w.r.t. σ and λ for a state $x \in X$ and an Ω-predicate U is to decide whether $x \in [\![\spadesuit_\sigma\heartsuit_\lambda]\!](U)$ or not. We denote the time to solve this problem by $t\big((\sigma,\lambda),x,U\big)$ and define[17]

$$t(\sigma,\lambda) := \sum_{x\in X} \max_{U\in\Omega^X} t\big((\sigma,\lambda),x,U\big).$$

The time $t(\sigma,\lambda)$ is often polynomial (or even linear) to the size of transition systems in many familiar cases (see Example 5.5 below and [22] for more details). We show $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ is at most quadratic to $t(\sigma,\lambda)$ under our assumption (Assumption 5.1).

Proposition 5.4 (complexity bound of $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$). *Let $|\psi|$ be the number of subformulas in ψ, and N be a constant that bounds the time to execute the boolean operations used in ψ. The complexity of $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ is given by*

$$O\Big(|\psi|\cdot\big(|X|\cdot N + t(\sigma,\lambda) + 2\cdot|X|\cdot t(\sigma,\lambda)\cdot N\big) + |\psi|\Big).$$

Especially, it is quadratic to $t(\sigma,\lambda)$ (since $|X| \le t(\sigma,\lambda)$) and linear to $|\psi|$[18].

Proof. Since the translation ι^{-1} takes only linear-time to the size of the CCTL formula ψ, the total complexity of Algorithm 1 is $O(|\psi| + C)$ where C is the complexity of $\mathrm{CHECK}(\iota^{-1}\psi)$ in Algorithm 1. We decide this C.

- The $p \in \mathrm{AP}$ case: the time for this procedure is irrespective of $|X|$.
- The $\boxdot_\gamma(\theta_1,\ldots,\theta_{|\gamma|})$ case: we check whether $x \in \gamma([\![\theta_1]\!],\ldots,[\![\theta_{|\gamma|}]\!])$ for every $x \in X$. By the definition of the constant N, the time to solve this problem is bounded by $|X|\cdot N$.
- The $\spadesuit_\sigma\heartsuit_\lambda\theta'$ case: we check whether $x \in [\![\spadesuit_\sigma\heartsuit_\lambda]\!]([\![\theta']\!])$ for every $x \in X$. Since solving this problem costs at most $\max_{U\in\Omega^X} t\big((\sigma,\lambda),x,U\big)$ for each $x \in X$, the time for calculating $[\![\spadesuit_\sigma\heartsuit_\lambda]\!]([\![\theta']\!])$ is bounded by $t(\sigma,\lambda)$ by Def. 5.3.
- The $\mu u.\boxdot_{\gamma_\mu}(\theta_1,\ldots,\theta_{|\gamma_\mu|-1},\spadesuit_\sigma\heartsuit_\lambda u)$ case: we check whether

$$x \in \mu u.\gamma([\![\theta_1]\!],\ldots,[\![\theta_{|\gamma_\mu|-1}]\!],[\![\spadesuit_\sigma\heartsuit_\lambda]\!](u)) \tag{29}$$

[17] The definition of $t(\sigma,\lambda)$ is modified from the previous version [29, Definition 5.3]: the previous definition

$$t(\sigma,\lambda) := \max_{x\in X, U\in\Omega^X} t\big((\sigma,\lambda),x,U\big)$$

is in general larger than the one employed here.

[18] As an improvement on complexity analysis here from the previous version [29, Proposition 5.4], we focus on the complexity bound relative to the time $t(\sigma,\lambda)$. This is justified because the amount $t(\sigma,\lambda)$ can be seen as a generalization notion of "size of a transition system", which could even be applicable to such complex systems where we can hardly count the number of edges/transitions or other ingredients of the systems.

for every $x \in X$. Each iteration of the while loop (Line 14) checks whether

$$x \in \gamma(\llbracket\theta_1\rrbracket, \dots, \llbracket\theta_{|\gamma_\mu|-1}\rrbracket, \llbracket\spadesuit_\sigma\heartsuit_\lambda\rrbracket(Q))$$

for every $x \in X$, where Q is some Ω-predicate. Thus, the time to compute each iteration of the while loop is bounded by $t(\sigma, \lambda) \cdot N$. Furthermore, the number of the while loop iterations (Line 14) is bounded by $|X|$ since the Cousot-Cousot theorem and our finiteness assumption assure we obtain the least fixpoint with at most $|X|$ steps, as we saw in the proof of Prop. 5.2. Hence, the time to decide whether condition 29 for every $x \in X$ is bounded by $|X| \cdot t(\sigma, \lambda) \cdot N$.

– The $\nu u. \boxdot_{\gamma_\nu} (\theta_1, \dots, \theta_{|\gamma_\nu|-1}, \spadesuit_\sigma\heartsuit_\lambda u)$ case is the same as the μ case.

Therefore, each call of the switch has the complexity

$$\begin{aligned} &O\big(|X| \cdot N + t(\sigma, \lambda) + |X| \cdot t(\sigma, \lambda) \cdot N + |X| \cdot t(\sigma, \lambda) \cdot N\big) \\ &= O\big(|X| \cdot N + t(\sigma, \lambda) + 2 \cdot |X| \cdot t(\sigma, \lambda) \cdot N\big). \end{aligned}$$

Thus, the complexity C of CHECK$(\iota^{-1}\psi)$ is

$$O\Big(|\psi| \cdot \big(|X| \cdot N + t(\sigma, \lambda) + 2 \cdot |X| \cdot t(\sigma, \lambda) \cdot N\big)\Big).$$

Finally, we conclude the total complexity of $\mathsf{MC}^{\mathsf{CCTL}}_{\mathcal{S}}$ as

$$O(C + |\psi|) = O\Big(|\psi| \cdot \big(|X| \cdot N + t(\sigma, \lambda) + 2 \cdot |X| \cdot t(\sigma, \lambda) \cdot N\big) + |\psi|\Big).$$

□

Example 5.5 (fixpoint model checking for CTL). The instance $\mathsf{MC}^{\mathsf{CCTL}}_{\mathcal{S}_{\mathrm{ND}}}$ corresponds to the well-known model-checking algorithm for CTL via fixpoints [9]. Since the time $t(\sigma, \lambda)$ is bounded by $|X| + |E|$, where $|E|$ is the number of edges of the Kripke frame, as in [22, Example 3], Our complexity in Prop. 5.4 recovers the known quadratic complexity bound of the classical algorithm.

6 Conclusion and Future Work

We formulated a new path-based coalgebraic logic CCTL (Def. 3.9), as an abstraction of classical CTL. We introduced an encoding of CCTL formulas into step-wise μ^{CCTL} formulas, which captures the categorical essence of the standard encoding of CTL into $\mathbf{L}\boldsymbol{\mu}$. This encoding is proven to preserve the semantics (Thm. 4.6) under some semantic conditions (Assumption 4.7) formulated in purely categorical terms. We saw these conditions distinguish classical CTL, which enjoys the fixpoint characterization (Prop. 4.11), and PCTL, which violates some conditions and enjoys only limited results (Prop. 4.13). Our coalgebraic fixpoint characterization yielded a naive model-checking algorithm $\mathsf{MC}_{\mathcal{S}}^{\mathsf{CCTL}}$ of CCTL, whose complexity is analyzed to be polynomial (Prop. 5.4).

The genericity of our framework of CCTL will allow several interesting extensions: n-ary next-time operators and non-boolean logical connectives could be smoothly incorporated. By changing the branching type T, our framework is expected to not only encompass other known examples like quantitative variants of CTL [4, 34] but also yield novel efficient path-based logics. Especially, we are investigating monotone neighborhood frames [17] and aim to establish "Monotone Neighborhood CTL" which would provide a CTL-like specification-description language as an efficient fragment of Parikh's game logic [19, 35]. [19] We will also explore $[0, 1]$-valued probabilistic path-based logics and corresponding probabilistic mu-calculus validating the fixpoint characterization. Such path-based logics would resemble, but computationally much superior to, existing Quantitative LTL [7].

We will also extend our encoding ι^{-1} to the coalgebraic path-based logic $\mu\mathcal{L}$, as an abstraction of classical exponential encodings of CTL* into the mu-calculus [2, 11].

Acknowledgement. This work is an extended version of a joint work [29] with Corina Cîrstea, Koko Muroya and Ichiro Hasuo, and supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603) and JST the establishment of university fellowships towards the creation of science technology innovation (No. JPMJFS2123). The author is grateful to all coauthors and his supervisor Masahito Hasegawa for their beneficial advices and commitment.

[19] Monotone neighborhood frames are coalgebras of the monotone neighborhood monad $\mathcal{M}$ [18]. While the monad $\mathcal{M}$ is not commutative unlike $\mathcal{P}^+$ and $\mathcal{G}_1$, our framework also applies to $\mathcal{M}$ by restricting allowable transition-type functors to linear polynomial ones since $\mathcal{M}$ is a strong monad.

References

[1] Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)

[2] Bhat, G., Cleaveland, R.: Efficient model checking via the equational μ-calculus. In: Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996. pp. 304–312. IEEE Computer Society (1996). https://doi.org/10.1109/LICS.1996.561358, https://doi.org/10.1109/LICS.1996.561358

[3] Brázdil, T., Forejt, V., Kretínský, J., Kucera, A.: The satisfiability problem for probabilistic CTL. In: Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24-27 June 2008, Pittsburgh, PA, USA. pp. 391–402. IEEE Computer Society (2008). https://doi.org/10.1109/LICS.2008.21, https://doi.org/10.1109/LICS.2008.21

[4] Chechik, M., Devereux, B., Easterbrook, S.M., Gurfinkel, A.: Multi-valued symbolic model-checking. ACM Trans. Softw. Eng. Methodol. **12**(4), 371–408 (2003). https://doi.org/10.1145/990010.990011, https://doi.org/10.1145/990010.990011

[5] Cîrstea, C.: Maximal traces and path-based coalgebraic temporal logics. Theor. Comput. Sci. **412**(38), 5025–5042 (2011). https://doi.org/10.1016/j.tcs.2011.04.025, https://doi.org/10.1016/j.tcs.2011.04.025

[6] Cirstea, C.: Canonical Coalgebraic Linear Time Logics. In: Moss, L.S., Sobocinski, P. (eds.) 6th Conference on Algebra and Coalgebra in Computer Science (CALCO 2015). Leibniz International Proceedings in Informatics (LIPIcs), vol. 35, pp. 66–85. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2015). https://doi.org/10.4230/LIPIcs.CALCO.2015.66, http://drops.dagstuhl.de/opus/volltexte/2015/5527

[7] Cirstea, C.: Linear-time logics – a coalgebraic perspective (2023)

[8] Cîrstea, C., Kupke, C., Pattinson, D.: EXPTIME tableaux for the coalgebraic μ-calculus. In: Grädel, E., Kahle, R. (eds.) Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5771, pp. 179–193. Springer (2009). https://doi.org/10.1007/978-3-642-04027-6_15, https://doi.org/10.1007/978-3-642-04027-6_15

[9] Clarke, E.M., Grumberg, O., Kroening, D., Peled, D.A., Veith, H.: Model checking, 2nd Edition. MIT Press (2018), https://mitpress.mit.edu/books/model-checking-second-edition

[10] Cousot, P., Cousot, R.: Constructive versions of Tarski's fixed point theorems. Pacific Journal of Mathematics **82**(1), 43 – 57 (1979)

[11] Dam, M.: Ctl* and ectl* as fragments of the modal mu-calculus. Theor. Comput. Sci. **126**(1), 77–96 (1994). https://doi.org/10.1016/0304-3975(94)90269-0, https://doi.org/10.1016/0304-3975(94)90269-0

[12] Doob, J.L.: Measure Theory. Springer (1994). https://doi.org/https://doi.org/10.1007/978-1-4612-0877-8

[13] Emerson, E.A., Clarke, E.M.: Using branching time temporal logic to synthesize synchronization skeletons. Sci. Comput. Program. **2**(3), 241–266 (1982). https:

//doi.org/10.1016/0167-6423(83)90017-5, https://doi.org/10.1016/0167-6423(83)90017-5

[14] Emerson, E.A., Halpern, J.Y.: Decision procedures and expressiveness in the temporal logic of branching time. J. Comput. Syst. Sci. **30**(1), 1–24 (1985). https://doi.org/10.1016/0022-0000(85)90001-7, https://doi.org/10.1016/0022-0000(85)90001-7

[15] Emerson, E.A., Halpern, J.Y.: "sometimes" and "not never" revisited: on branching versus linear time temporal logic. J. ACM **33**(1), 151–178 (1986). https://doi.org/10.1145/4904.4999, https://doi.org/10.1145/4904.4999

[16] Giry, M.: A categorical approach to probability theory. In: Banaschewski, B. (ed.) Categorical Aspects of Topology and Analysis. pp. 68–85. Springer Berlin Heidelberg, Berlin, Heidelberg (1982)

[17] Hansen, H.H., Kupke, C.: A coalgebraic perspective on monotone modal logic. In: Adámek, J., Milius, S. (eds.) Proceedings of the Workshop on Coalgebraic Methods in Computer Science, CMCS 2004, Barcelona, Spain, March 27-29, 2004. Electronic Notes in Theoretical Computer Science, vol. 106, pp. 121–143. Elsevier (2004). https://doi.org/10.1016/j.entcs.2004.02.028, https://doi.org/10.1016/j.entcs.2004.02.028

[18] Hansen, H.H., Kupke, C., Leal, R.A.: Strong completeness for iteration-free coalgebraic dynamic logics. In: Díaz, J., Lanese, I., Sangiorgi, D. (eds.) Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8705, pp. 281–295. Springer (2014). https://doi.org/10.1007/978-3-662-44602-7_22, https://doi.org/10.1007/978-3-662-44602-7_22

[19] Hansen, H.H., Kupke, C., Marti, J., Venema, Y.: Parity games and automata for game logic (extended version). CoRR **abs/1709.00777** (2017), http://arxiv.org/abs/1709.00777

[20] Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects Comput. **6**(5), 512–535 (1994). https://doi.org/10.1007/BF01211866, https://doi.org/10.1007/BF01211866

[21] Hasuo, I., Shimizu, S., Cîrstea, C.: Lattice-theoretic progress measures and coalgebraic model checking. In: Bodík, R., Majumdar, R. (eds.) Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016. pp. 718–732. ACM (2016). https://doi.org/10.1145/2837614.2837673, https://doi.org/10.1145/2837614.2837673

[22] Hausmann, D., Schröder, L.: Game-based local model checking for the coalgebraic mu-calculus. In: Fokkink, W.J., van Glabbeek, R. (eds.) 30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands. LIPIcs, vol. 140, pp. 35:1–35:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.CONCUR.2019.35, https://doi.org/10.4230/LIPIcs.CONCUR.2019.35

[23] Jacobs, B.: Semantics of weakening and contraction. Ann. Pure Appl. Log. **69**(1), 73–106 (1994). https://doi.org/10.1016/0168-0072(94)90020-5, https://doi.org/10.1016/0168-0072(94)90020-5

[24] Jacobs, B.: Trace semantics for coalgebras. In: Adámek, J., Milius, S. (eds.) Proceedings of the Workshop on Coalgebraic Methods in Computer Science, CMCS 2004, Barcelona, Spain, March 27-29, 2004. Electronic Notes in Theoretical Computer Science, vol. 106, pp. 167–184. Elsevier (2004). https://doi.org/10.1016/j.entcs.2004.02.031, https://doi.org/10.1016/j.entcs.2004.02.031

[25] Jacobs, B.: Affine monads and side-effect-freeness. In: Hasuo, I. (ed.) Coalgebraic Methods in Computer Science - 13th IFIP WG 1.3 International Workshop, CMCS 2016, Colocated with ETAPS 2016, Eindhoven, The Netherlands, April 2-3, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9608, pp. 53–72. Springer (2016). https://doi.org/10.1007/978-3-319-40370-0_5, https://doi.org/10.1007/978-3-319-40370-0_5

[26] Jacobs, B.: Introduction to Coalgebra: Towards Mathematics of States and Observation. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2016). https://doi.org/10.1017/CBO9781316823187

[27] Kock, A.: Monads on symmetric monoidal closed categories. Archiv der Mathematik **21**(1), 1–10 (1970). https://doi.org/10.1007/BF01220868, https://doi.org/10.1007/BF01220868

[28] Kock, A.: Bilinearity and cartesian closed monads. Mathematica Scandinavica **29**, 161–174 (1971)

[29] Kojima, R., Cîrstea, C., Muroya, K., Hasuo, I.: Coalgebraic CTL: fixpoint characterization and polynomial-time model checking. In: König, B., Urbat, H. (eds.) Coalgebraic Methods in Computer Science - 17th IFIP WG 1.3 International Workshop, CMCS 2024, Colocated with ETAPS 2024, Luxembourg City, Luxembourg, April 6-7, 2024, Proceedings. Lecture Notes in Computer Science, vol. 14617, pp. 1–22. Springer (2024). https://doi.org/10.1007/978-3-031-66438-0_1, https://doi.org/10.1007/978-3-031-66438-0_1

[30] Kozen, D.: Results on the propositional mu-calculus. Theor. Comput. Sci. **27**, 333–354 (1983). https://doi.org/10.1016/0304-3975(82)90125-6, https://doi.org/10.1016/0304-3975(82)90125-6

[31] MacLane, S.: Categories for the Working Mathematician. Springer-Verlag, New York (1971), graduate Texts in Mathematics, Vol. 5

[32] Milner, R.: Communication and concurrency. PHI Series in computer science, Prentice Hall (1989)

[33] Moss, L.S.: Coalgebraic logic. Ann. Pure Appl. Log. **96**(1-3), 277–317 (1999). https://doi.org/10.1016/S0168-0072(98)00042-6, https://doi.org/10.1016/S0168-0072(98)00042-6

[34] Pan, H., Li, Y., Cao, Y., Ma, Z.: Model checking computation tree logic over finite lattices. Theor. Comput. Sci. **612**, 45–62 (2016). https://doi.org/10.1016/J.TCS.2015.10.014, https://doi.org/10.1016/j.tcs.2015.10.014

[35] Parikh, R.: The logic of games and its applications. In: Selected Papers of the International Conference on "Foundations of Computation Theory" on Topics in the Theory of Computation. p. 111–139. Elsevier North-Holland, Inc., USA (1985)

[36] Pattinson, D.: Coalgebraic modal logic: soundness, completeness and decidability of local consequence. Theor. Comput. Sci. **309**(1-3), 177–193 (2003). https://doi.org/10.1016/S0304-3975(03)00201-9, https://doi.org/10.1016/S0304-3975(03)00201-9

[37] Schubert, C.: Terminal coalgebras for measure-polynomial functors. In: Chen, J., Cooper, S.B. (eds.) Theory and Applications of Models of Computation, 6th Annual Conference, TAMC 2009, Changsha, China, May 18-22, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5532, pp. 325–334. Springer (2009). https://doi.org/10.1007/978-3-642-02017-9_35, https://doi.org/10.1007/978-3-642-02017-9_35

[38] Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics **5**, 285–309 (1955), https://api.semanticscholar.org/CorpusID:13651629

[39] Urabe, N., Hasuo, I.: Categorical büchi and parity conditions via alternating fixed points of functors. In: Cîrstea, C. (ed.) Coalgebraic Methods in Computer Science - 14th IFIP WG 1.3 International Workshop, CMCS 2018, Colocated with ETAPS 2018, Thessaloniki, Greece, April 14-15, 2018, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11202, pp. 214–234. Springer (2018). https://doi.org/10.1007/978-3-030-00389-0_12, https://doi.org/10.1007/978-3-030-00389-0_12

[40] Urabe, N., Hasuo, I.: Coalgebraic infinite traces and kleisli simulations. Log. Methods Comput. Sci. **14**(3) (2018). https://doi.org/10.23638/LMCS-14(3:15)2018, https://doi.org/10.23638/LMCS-14(3:15)2018

[41] Venema, Y.: Automata and fixed point logic: A coalgebraic perspective. Inf. Comput. **204**(4), 637–678 (2006). https://doi.org/10.1016/J.IC.2005.06.003, https://doi.org/10.1016/j.ic.2005.06.003

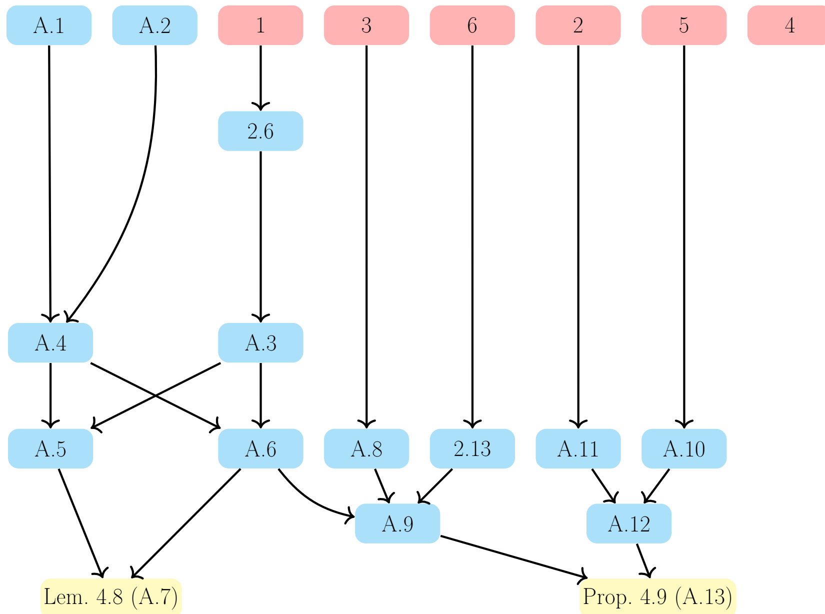


Fig. 3: Dependencies between lemmas (in blue) and assumtions (in red) for proving Lem. 4.8 (A.7) and Prop. 4.9 (A.13) (in yellow).

A Omitted Proofs

We fix a BT situation $\mathcal{S} = (\mathbb{C}, T, F, c, \Omega, \Sigma, \Lambda)$ with maximal execution $\mathrm{tr}(c')$, and suppose $\mathcal{S}$ satisfies Assumption 4.7.

We prove Lem. 4.8 (Lem. A.7) in appendix A.1 and Prop. 4.9 (Prop. A.13) in appendix A.2 (see Fig. 3).

A.1 Proof of Lem. 4.8

Here we fix a distributive law ξ of F over T, assured by Prop. 2.8.

Lemma A.1.

$$\overline{F_X}\,\mathrm{tr}(c') = \mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big). \tag{30}$$

Proof. Firstly, when we have a distributive law $\xi\colon FT \Rightarrow TF$, we also have a distributive law $\xi'\colon F_X T \Rightarrow TF_X$ by

$$\xi'_A = \mathsf{st}_{X,FA} \circ (\mathrm{id}_X \times \xi_A) \colon X \times FTA \to X \times TFA \to T(X \times FA).$$

By the definition of the Kleisli lifting (see §3.2), the Kleisli lifting $\overline{F_X}$ of F_X maps $\mathrm{tr}(c')$ to

$$\overline{F_X}\,\mathrm{tr}(c') = \xi'_{Z_X} \circ (\mathrm{id}_X \times F\,\mathrm{tr}(c')).$$

This equation can be reduced further by the definition of ξ':

$$\begin{aligned}\overline{F_X}\,\mathrm{tr}(c') &= \xi'_{Z_X} \circ (\mathrm{id}_X \times F\,\mathrm{tr}(c')) \\ &= \mathsf{st}_{X,FZ_X} \circ (\mathrm{id}_X \times \xi_{Z_X}) \circ (\mathrm{id}_X \times F\,\mathrm{tr}(c')) \\ &= \mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big).\end{aligned}$$

This concludes the proof. □

Lemma A.2.

$$(\eta_{X\times FZ_X} \circ \zeta) \odot \mathrm{tr}(c') = \overline{F_X}\,\mathrm{tr}(c') \odot c'. \tag{31}$$

Proof. The statement comes from the equation $J\zeta \odot u = \overline{F}u \odot c\,\mathrm{tr}(c')$ defining maximal trace (Def. 3.3.) Note that $J\zeta = \eta_{X\times FZ_X} \circ \zeta$ by the definition of the Kleisli embedding J. □

Lemma A.3. *1.* $T\pi_1 \circ c' = \eta_X$*,*
2. $T\pi_2 \circ c' = c$*.*

Proof. On item 1, from condition 1 of Assumption 4.7 and item 2 of Lem. 2.6, we have $T\pi_1 \circ \mathsf{st}_{X,FX} = \eta_X \circ \pi_1$. Thus, we obtain

$$T\pi_1 \circ c' = T\pi_1 \circ \mathsf{st}_{X,FX} \circ \langle \mathrm{id}_X, c\rangle = \eta_X \circ \pi_1 \circ \langle \mathrm{id}_X, c\rangle = \eta_X$$

On item 2, by item 1 of Lem. 2.6, we have $T\pi_2 \circ \mathsf{st}_{X,FX} = \pi_2$. Thus, we obtain

$$T\pi_2 \circ c' = T\pi_2 \circ \mathsf{st}_{X,FX} \circ \langle \mathrm{id}_X, c\rangle = \pi_2 \circ \langle \mathrm{id}_X, c\rangle = c.$$

□

Lemma A.4. $T(\zeta) \circ \mathrm{tr}(c') = \mu_{X\times FZ_X} \circ T\Big(\mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c'$.

Proof. By the unitality of the monad T and the definition of the Kleisli composition $\odot$, we have

$$\begin{aligned}T(\zeta) \circ \mathrm{tr}(c') &= \mathrm{id}_{T(X\times FZ_X)} \circ T(\zeta) \circ \mathrm{tr}(c') \\ &= \big(\mu_{X\times FZ_X} \circ T(\eta_{X\times FZ_X})\big) \circ T(\zeta) \circ \mathrm{tr}(c') \\ &= \mu_{X\times FZ_X} \circ T(\eta_{X\times FZ_X} \circ \zeta) \circ \mathrm{tr}(c') \\ &= (\eta_{X\times FZ_X} \circ \zeta) \odot \mathrm{tr}(c').\end{aligned}$$

By Lem. A.2, we have $(\eta_{X\times FZ_X} \circ \zeta) \odot \mathrm{tr}(c') = \overline{F_X}\,\mathrm{tr}(c') \odot c'$. The RHS of this equation can be reduced to $\overline{F_X}\,\mathrm{tr}(c') \odot c' = \mu_{X\times FZ_X} \circ T\big(\overline{F_X}\,\mathrm{tr}(c')\big) \circ c'$ by the definition of $\odot$. Finally, since $\overline{F_X}\,\mathrm{tr}(c') = \mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)$ by Lem. A.1, we obtain

$$\begin{aligned}T(\zeta) \circ \mathrm{tr}(c') &= \mu_{X\times FZ_X} \circ T\big(\overline{F_X}\,\mathrm{tr}(c')\big) \circ c' \\ &= \mu_{X\times FZ_X} \circ T\Big(\mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c'.\end{aligned}$$

□

Lemma A.5. $T\zeta_1 \circ \mathrm{tr}(c') = \eta_X$.

Proof. Since $\zeta_1 = \pi_1 \circ \zeta$, we have $T(\zeta_1) \circ \mathrm{tr}(c') = T(\pi_1) \circ T(\zeta) \circ \mathrm{tr}(c')$. Thus, by Lem. A.4 and the naturality of μ, we have

$$\begin{aligned}
T(\zeta_1) \circ \mathrm{tr}(c') &= T(\pi_1) \circ T(\zeta) \circ \mathrm{tr}(c') \\
&= T(\pi_1) \circ \mu_{X \times FZ_X} \circ T\Big(\mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_X \circ T^2(\pi_1) \circ T\Big(\mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_X \circ T\Big(T(\pi_1) \circ \mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c'.
\end{aligned}$$

By $T\pi_1 \circ \mathsf{st}_{X,FZ_X} = \eta_X \circ \pi_1$ by item 2 of Lem. 2.6, the above equation is

$$\begin{aligned}
T(\zeta_1) \circ \mathrm{tr}(c') &= \mu_X \circ T\Big(T\pi_1 \circ \mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_X \circ T\Big(\eta_X \circ \pi_1 \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_X \circ T(\eta_X \circ \pi_1) \circ c' \\
&= \mu_X \circ T\eta_X \circ T\pi_1 \circ c'.
\end{aligned}$$

Finally, by the monad unitality $\mu_X \circ T\eta_X = \mathrm{id}_{TX}$ and item 1 of Lem. A.3, we have

$$\begin{aligned}
T(\zeta_1) \circ \mathrm{tr}(c') &= \mu_X \circ T\eta_X \circ T\pi_1 \circ c' \\
&= T\pi_1 \circ c' \\
&= \eta_X.
\end{aligned}$$

□

Lemma A.6.

$$T(\zeta_2) \circ \mathrm{tr}(c') = \mu_{FZ_X} \circ T\big(\xi \circ F\,\mathrm{tr}(c')\big) \circ c.$$

Proof. Since $T(\zeta_2) \circ \mathrm{tr}(c') = T\pi_2 \circ T\zeta$, by Lem. A.4, we have

$$\begin{aligned}
T(\zeta_2) \circ \mathrm{tr}(c') &= T\pi_2 \circ T\zeta \circ \mathrm{tr}(c') \\
&= T\pi_2 \circ \mu_{X \times FZ_X} \circ T\Big(\mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c'.
\end{aligned}$$

Now we have $T\pi_2 \circ \mu_{X \times FZ_X} = \mu_{FZ_X} \circ T^2\pi_2 \colon T^2(X \times FZ_X) \to TFZ_X$ by naturality of the multiplication μ. Combining this and $T\pi_2 \circ \mathsf{st}_{X,FZ_X} = \pi_2$ in item 1 of Lem. 2.6 yields

$$\begin{aligned}
T(\zeta_2) \circ \mathrm{tr}(c') &= \mu_{FZ_X} \circ T^2\pi_2 \circ T\Big(\mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_{FZ_X} \circ T\Big(T\pi_2 \circ \mathsf{st}_{X,FZ_X} \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_{FZ_X} \circ T\Big(\pi_2 \circ \big(\mathrm{id} \times (\xi_{Z_X} \circ F\,\mathrm{tr}(c'))\big)\Big) \circ c' \\
&= \mu_{FZ_X} \circ T\big(\xi \circ F\,\mathrm{tr}(c') \circ \pi_2\big) \circ c' \\
&= \mu_{FZ_X} \circ T\big(\xi \circ F\,\mathrm{tr}(c')\big) \circ T\pi_2 \circ c'.
\end{aligned}$$

Finally, by $T\pi_2 \circ c' = c$ in Lem. A.3, we have

$$\begin{aligned} T(\zeta_2) \circ \mathrm{tr}(c') &= \mu_{FZ_X} \circ T\big(\xi \circ F\,\mathrm{tr}(c')\big) \circ T\pi_2 \circ c' \\ &= \mu_{FZ_X} \circ T\big(\xi \circ F\,\mathrm{tr}(c')\big) \circ c. \end{aligned}$$

□

Lemma A.7 (Lem. 4.8). *$T(F\zeta_1 \circ \zeta_2) \circ \mathrm{tr}(c') = c$.*

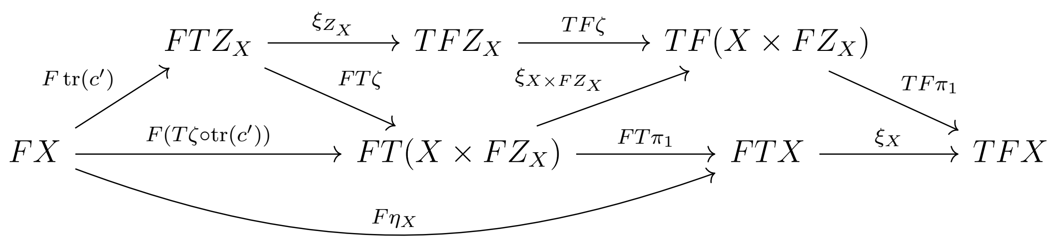
Proof. By Lem. A.6, we have

$$\begin{aligned} T(F\zeta_1 \circ \zeta_2) \circ \mathrm{tr}(c') &= T(F\zeta_1) \circ T(\zeta_2) \circ \mathrm{tr}(c') \\ &= TF\zeta_1 \circ \mu_{FZ_X} \circ T\big(\xi \circ F\,\mathrm{tr}(c')\big) \circ c. \end{aligned}$$

Since $TF\zeta_1 \circ \mu_{FZ_X} = \mu_{FX} \circ T^2F\zeta_1$ by the naturality of the multiplication μ, we have

$$T(F\zeta_1 \circ \zeta_2) \circ \mathrm{tr}(c') = \mu_{FX} \circ T\big(TF\zeta_1 \circ \xi \circ F\,\mathrm{tr}(c')\big) \circ c.$$

Thus, it suffices to show $TF\zeta_1 \circ \xi \circ F\,\mathrm{tr}(c') = \eta_{FX}$ since $\mu_{FX} \circ T\eta_{FX} \circ c = c$. This is obtained by the following diagram:



Here

- the upper-left triangle is trivial,
- the upper-middle and upper-right triangles come from the naturality of the distributive law ξ, and
- the semicircle below is by Lem. A.5.

In conclusion, the bottom path of the above diagram $\xi_X \circ F\eta_X$ is reduced as $\xi_X \circ F\eta_X = \eta_{FX}$ by the definition of distributive laws (diagram 10 in Def. 2.7). □

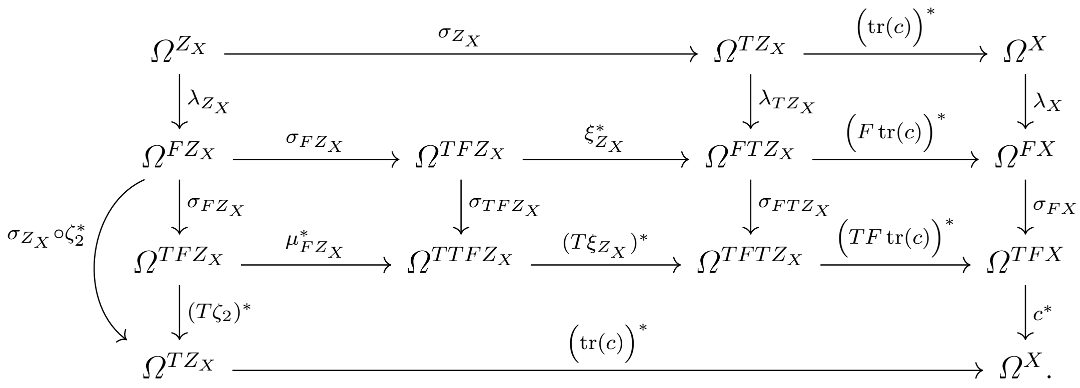
A.2 Proof of Prop. 4.9

Lemma A.8 (Cîrstea [5, Lemma 3.1]). *For $\sigma \in \Sigma$, we have $\sigma_{TY} \circ \sigma_Y = \mu^*_Y \circ \sigma_Y$ for each $Y \in \mathbb{C}$.*

Proof. Straightforward from the assumption that $(\Omega, \mathrm{ev}_\sigma)$ is a T-algebra (condition 3 in Assumption 4.7). See [5, Lemma 3.1] for details. □

Lemma A.9. *For $\sigma \in \Sigma$ and $\lambda \in \Lambda$, we have* $[\![\spadesuit_\sigma \heartsuit_\lambda]\!] \circ (\!|\spadesuit_\sigma|\!) = (\!|\spadesuit_\sigma|\!) \circ (\!|\heartsuit_\lambda|\!)$.

Proof. See the diagram below.



(32)

Here,

- the top-left rectangle is by Lem. 2.13 (which is assured by 6 in Assumption 4.7),
- the top-right rectangle is by the naturality of λ,
- the middle-center and middle-right rectangles and the left hemisphere are by the naturality of σ,
- the middle-left rectangles come from Lem. A.8, and
- the bottom rectangle is by Lem. A.6.

$\square$

Lemma A.10. *For $\gamma \in \Gamma_\mu \cup \Gamma_\nu$ and $\sigma \in \Sigma$, we have the following: for every $f_1, \ldots, f_n \colon X \to \Omega$ and $g \colon Z_X \to \Omega$,*

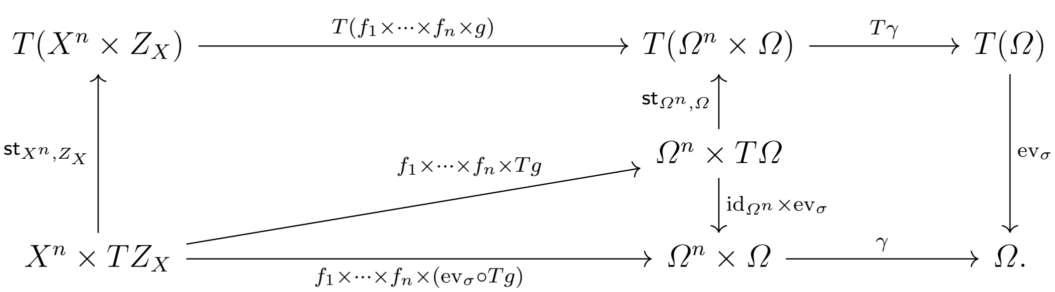
$$\big(\mathsf{st}_{X^n, Z_X}\big)^* \circ \sigma_{X^n \times Z_X}\big(\gamma \circ (f_1 \times \cdots \times f_n \times g)\big) = \gamma\big(f_1 \times \cdots \times f_n \times \sigma_{Z_X}(g)\big) \qquad (33)$$

or, equivalently,

$$\mathrm{ev}_\sigma \circ T\gamma \circ T(f_1 \times \cdots \times f_n \times g) \circ \mathsf{st}_{X^n, Z_X} = \gamma\big(f_1 \times \cdots \times f_n \times (\mathrm{ev}_\sigma \circ Tg)\big), \qquad (34)$$

where $n = |\gamma| - 1$.

Proof. The LHS and RHS of equation 34 is respectively depicted as the top and bottom paths in the following diagram:



Here,

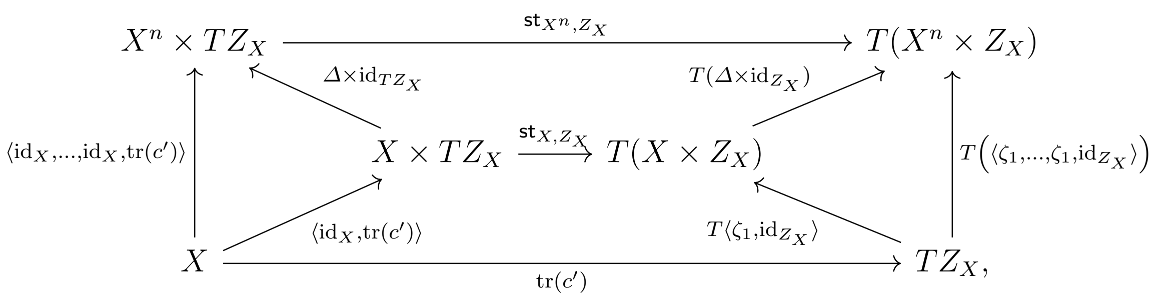
- the top-left trapezoid comes from the naturality of the strength st,
- the triangle below is straightforward, and
- the right rectangle is assumed in 5 in Assumption 4.7.

□

Lemma A.11.

$$T\big(\langle \zeta_1, \ldots, \zeta_1, \mathrm{id}_{Z_X}\rangle\big) \circ \mathrm{tr}(c') = \mathsf{st}_{X^n, Z_X} \circ \langle \mathrm{id}_X, \ldots, \mathrm{id}_X, \mathrm{tr}(c')\rangle \tag{35}$$

Proof. The LHS and RHS of equation 35 is respectively depicted as the bottom and top paths of the following commutative diagram:



where $\Delta\colon X \to X^n$ is the diagonal map. Here,

- the left and right triangles comes from the definition of the diagonal Δ,
- the top trapezoid is the naturality of the strength (Def. 2.2), and
- the top trapezoid is condition 2 of Assumption 4.7.

□

Lemma A.12. *For $\gamma \in \Gamma_\mu \cup \Gamma_\nu$, $\psi_1, \ldots, \psi_{|\gamma|-1} \in \mathsf{SFml}$ and $\varphi \in \mathsf{PFml}$, we have*

$$(\!|\spadesuit_\sigma\big(\boxdot_\gamma(\psi_1, \ldots, \psi_{|\gamma|-1}, \varphi)\big)|\!)^{\mathsf{SFml}} = (\!|\boxdot_\gamma(\psi_1, \ldots, \psi_{|\gamma|-1}, \spadesuit_\sigma\varphi)|\!)^{\mathsf{SFml}} \tag{36}$$

in other words,

$$(\!|\spadesuit_\sigma|\!)\big(\gamma((\!|\psi_1|\!)^{\mathsf{PFml}}, \ldots, (\!|\psi_{|\gamma|-1}|\!)^{\mathsf{PFml}}, (\!|\varphi|\!)^{\mathsf{PFml}})\big) = \gamma\big((\!|\psi_1|\!)^{\mathsf{SFml}}, \ldots, (\!|\psi_{|\gamma|-1}|\!)^{\mathsf{SFml}}, (\!|\spadesuit_\sigma|\!)((\!|\varphi|\!)^{\mathsf{PFml}})\big). \tag{37}$$

Proof. Since $\sigma(f) = \mathrm{ev}_\sigma \circ T(f)$, the LHS and RHS of equation 36 (or equation 37) are expressed as follows.

$$\begin{aligned}
\mathrm{LHS} &= \big(\mathrm{tr}(c')\big)^* \circ \sigma_{Z_X}\big(\gamma((\!|\psi_1|\!)^{\mathsf{PFml}}, \ldots, (\!|\psi_{|\gamma|-1}|\!)^{\mathsf{PFml}}, (\!|\varphi|\!)^{\mathsf{PFml}})\big)\\
&= \mathrm{ev}_\sigma \circ T\Big(\gamma\big(\zeta_1^*((\!|\psi_1|\!)^{\mathsf{SFml}}), \ldots, \zeta_1^*((\!|\psi_{|\gamma|-1}|\!)^{\mathsf{SFml}}), (\!|\varphi|\!)^{\mathsf{PFml}}\big)\Big) \circ \mathrm{tr}(c')\\
&= \mathrm{ev}_\sigma \circ T\gamma \circ T\big((\!|\psi_1|\!)^{\mathsf{SFml}} \times \cdots \times (\!|\psi_{|\gamma|-1}|\!)^{\mathsf{SFml}} \times (\!|\varphi|\!)^{\mathsf{PFml}}\big) \circ T\big(\langle \zeta_1, \ldots, \zeta_1, \mathrm{id}_{Z_X}\rangle\big) \circ \mathrm{tr}(c')\\
&= \mathrm{ev}_\sigma \circ T\gamma \circ T\big((\!|\psi_1|\!)^{\mathsf{SFml}} \times \cdots \times (\!|\psi_{|\gamma|-1}|\!)^{\mathsf{SFml}} \times (\!|\varphi|\!)^{\mathsf{PFml}}\big) \circ \mathsf{st}_{X^n, Z_X} \circ \langle \mathrm{id}_X, \ldots, \mathrm{id}_X, \mathrm{tr}(c')\rangle,
\end{aligned}$$

where the last transformation uses Lem. A.11. On the other hand, the RHS can be written as

$$\mathrm{RHS} = \gamma\big(\llparenthesis\psi_1\rrparenthesis^{\mathsf{SFml}}, \dots, \llparenthesis\psi_{|\gamma|-1}\rrparenthesis^{\mathsf{SFml}}, \mathrm{ev}_\sigma \circ T(\llparenthesis\varphi\rrparenthesis^{\mathsf{PFml}})\big).$$

They are respectively the top and bottom paths from X to Ω in the diagram below.

$$\begin{array}{ccccccc}
 & & T(X^n \times Z_X) & \xrightarrow{T(\llparenthesis\psi_1\rrparenthesis^{\mathsf{SFml}}\times\cdots\times\llparenthesis\psi_{|\gamma|-1}\rrparenthesis^{\mathsf{SFml}}\times\llparenthesis\varphi\rrparenthesis^{\mathsf{PFml}})} & T(\Omega^n \times \Omega) & \xrightarrow{T\gamma} & T(\Omega) \\
 & & {\scriptstyle \mathsf{st}_{X^n,Z_X}}\big\uparrow & & & & \big\downarrow{\scriptstyle \mathrm{ev}_\sigma} \\
X & \xrightarrow{\langle \mathrm{id}_X,\dots,\mathrm{id}_X,\mathrm{tr}(c')\rangle} & X^n \times TZ_X & \xrightarrow{\llparenthesis\psi_1\rrparenthesis^{\mathsf{SFml}}\times\cdots\times\llparenthesis\psi_{|\gamma|-1}\rrparenthesis^{\mathsf{SFml}}\times(\mathrm{ev}_\sigma\circ T\llparenthesis\varphi\rrparenthesis^{\mathsf{PFml}})} & \Omega^n \times \Omega & \xrightarrow{\gamma} & \Omega.
\end{array} \tag{38}$$

Commutativity of the rectangle in this diagram is guaranteed by Lem. A.10, by letting $f_i = \llparenthesis\psi_i\rrparenthesis^{\mathsf{SFml}}$ and $g = \llparenthesis\varphi\rrparenthesis^{\mathsf{PFml}}$. □

Proposition A.13 (coalgebraic expansion law, Prop. 4.9). *Let $\sigma \in \Sigma$, $\lambda \in \Lambda$, and μ-schemes $\gamma_\mu \in \Gamma_\mu$ and ν-schemes $\gamma_\nu \in \Gamma_\nu$. We have*

$$\llparenthesis\spadesuit_\sigma\rrparenthesis\big(\mu\Phi^{\lambda,\gamma_\mu}_{\left(\zeta_1^*(\llparenthesis\iota\theta_1\rrparenthesis^{\mathsf{SFml}}),\dots,\zeta_1^*(\llparenthesis\iota\theta_{|\gamma_\mu|-1}\rrparenthesis^{\mathsf{SFml}})\right)}\big) \sqsupseteq \mu\Psi^{(\sigma,\lambda),\gamma_\mu}_{(\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\mu|-1}\rrbracket)} \tag{39}$$

for $\theta_1,\dots,\theta_{|\gamma_\mu|-1}$ with $\llparenthesis\iota\theta_i\rrparenthesis^{\mathsf{SFml}} \sqsupseteq \llbracket\theta_i\rrbracket$ for $i = 1,\dots,|\gamma_\mu|-1$, and

$$\llparenthesis\spadesuit_\sigma\rrparenthesis\big(\nu\Phi^{\lambda,\gamma_\nu}_{\left(\zeta_1^*(\llparenthesis\iota\theta_1\rrparenthesis^{\mathsf{SFml}}),\dots,\zeta_1^*(\llparenthesis\iota\theta_{|\gamma_\nu|-1}\rrparenthesis^{\mathsf{SFml}})\right)}\big) \sqsubseteq \nu\Psi^{(\sigma,\lambda),\gamma_\nu}_{(\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\nu|-1}\rrbracket)} \tag{40}$$

for $\theta_1,\dots,\theta_{|\gamma_\nu|-1}$ with $\llparenthesis\iota\theta_i\rrparenthesis^{\mathsf{SFml}} \sqsubseteq \llbracket\theta_i\rrbracket$ for $i = 1,\dots,|\gamma_\nu|-1$. Furthermore, if $\llparenthesis\iota\theta_i\rrparenthesis^{\mathsf{SFml}} = \llbracket\theta_i\rrbracket$ for every subformula θ_i, the inequalities 39 and 40 are both equalities.

Proof. We prove the μ case. The ν case is proven in the same way. Let

$$\begin{aligned}
(\llbracket\vec{\theta}\rrbracket) &= (\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\mu|-1}\rrbracket) \\
(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}}) &= (\zeta_1^*\llparenthesis\iota\theta_1\rrparenthesis^{\mathsf{SFml}},\dots,\zeta_1^*\llparenthesis\iota\theta_{|\gamma_\mu|-1}\rrparenthesis^{\mathsf{SFml}}).
\end{aligned}$$

By Lem. A.9, we have

$$\begin{aligned}
\Psi^{(\sigma,\lambda),\gamma_\mu}_{(\llbracket\vec{\theta}\rrbracket)}\big(\llparenthesis\spadesuit_\sigma\rrparenthesis(\mu\Phi_{\lambda,\gamma_\mu,\iota\vec{\theta}_{|\gamma_\mu|-1}})\big) &= \gamma_\mu\big(\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\mu|-1}\rrbracket, \llbracket\spadesuit_\sigma\heartsuit_\lambda\rrbracket\llparenthesis\spadesuit_\sigma\rrparenthesis(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}})})\big) \\
&= \gamma_\mu\big(\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\mu|-1}\rrbracket, \llparenthesis\spadesuit_\sigma\rrparenthesis\llparenthesis\heartsuit_\lambda\rrparenthesis(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}})})\big).
\end{aligned}$$

Furthermore, by Lem. A.12, we obtain

$$\begin{aligned}
\Psi^{(\sigma,\lambda),\gamma_\mu}_{(\llbracket\vec{\theta}\rrbracket)}\big(\llparenthesis\spadesuit_\sigma\rrparenthesis(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}})})\big) &= \gamma_\mu\big(\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\mu|-1}\rrbracket, \llparenthesis\spadesuit_\sigma\rrparenthesis\llparenthesis\heartsuit_\lambda\rrparenthesis(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}})})\big) \\
&= \llparenthesis\spadesuit_\sigma\rrparenthesis\Big(\gamma_\mu\big(\llbracket\theta_1\rrbracket,\dots,\llbracket\theta_{|\gamma_\mu|-1}\rrbracket, \llparenthesis\heartsuit_\lambda\rrparenthesis(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}})})\big)\Big) \\
&\sqsubseteq \llparenthesis\spadesuit_\sigma\rrparenthesis\Big(\gamma_\mu\big(\llparenthesis\iota\theta_1\rrparenthesis^{\mathsf{SFml}},\dots,\llparenthesis\iota\theta_{|\gamma_\mu|-1}\rrparenthesis^{\mathsf{SFml}}, \llparenthesis\heartsuit_\lambda\rrparenthesis(\mu\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*\llparenthesis\iota\vec{\theta}\rrparenthesis^{\mathsf{SFml}})})\big)\Big).
\end{aligned}$$

Here the last transformation comes from $(\!|\iota\theta_i|\!)^{\mathsf{SFml}} \sqsupseteq [\![\theta_i]\!]$ for $i = 1, \ldots, |\gamma_\mu| - 1$ and monotonicity of $(\!|\spadesuit_\sigma|\!)$ and γ_μ (following from the definition of predicate liftings (Def. 2.10)). Since $\mu\,\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})}$ is a fixpoint of $\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})}$, we conclude

$$\begin{aligned}\Psi^{(\sigma,\lambda),\gamma_\mu}_{([\![\vec{\theta}]\!])}\big((\!|\spadesuit_\sigma|\!)(\mu\,\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})})\big) &\sqsubseteq (\!|\spadesuit_\sigma|\!)\Big(\gamma_\mu\big((\!|\iota\theta_1|\!)^{\mathsf{SFml}}, \ldots, (\!|\iota\theta_{|\gamma_\mu|-1}|\!)^{\mathsf{SFml}}, [\![\heartsuit_\lambda]\!](\mu\,\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})})\big)\Big) \\ &= (\!|\spadesuit_\sigma|\!)\big(\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})}(\mu\,\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})})\big) \\ &= (\!|\spadesuit_\sigma|\!)(\mu\,\Phi^{\lambda,\gamma_\mu}_{(\zeta_1^*(\!|\iota\vec{\theta}|\!)^{\mathsf{SFml}})}).\end{aligned}$$

□