Coalgebraic CTL: Fixpoint characterization and Poly-time Model Checking

Ryota Kojima[1], Corina Cirstea[2], Koko Muroya[1] and Ichiro Hasuo[3]

RIMS, Kyoto Univ.[1], Univ. of Southampton[2],
and National Institute of Informatics[3]
6 Apr. 2024,CMCS 2024

Contents

1. CTL is efficient, thanks to fixpoint encoding
2. Why is Probabilistic CTL not as good as CTL?
3. We generalize *CTL: Coalgebraic CTL*

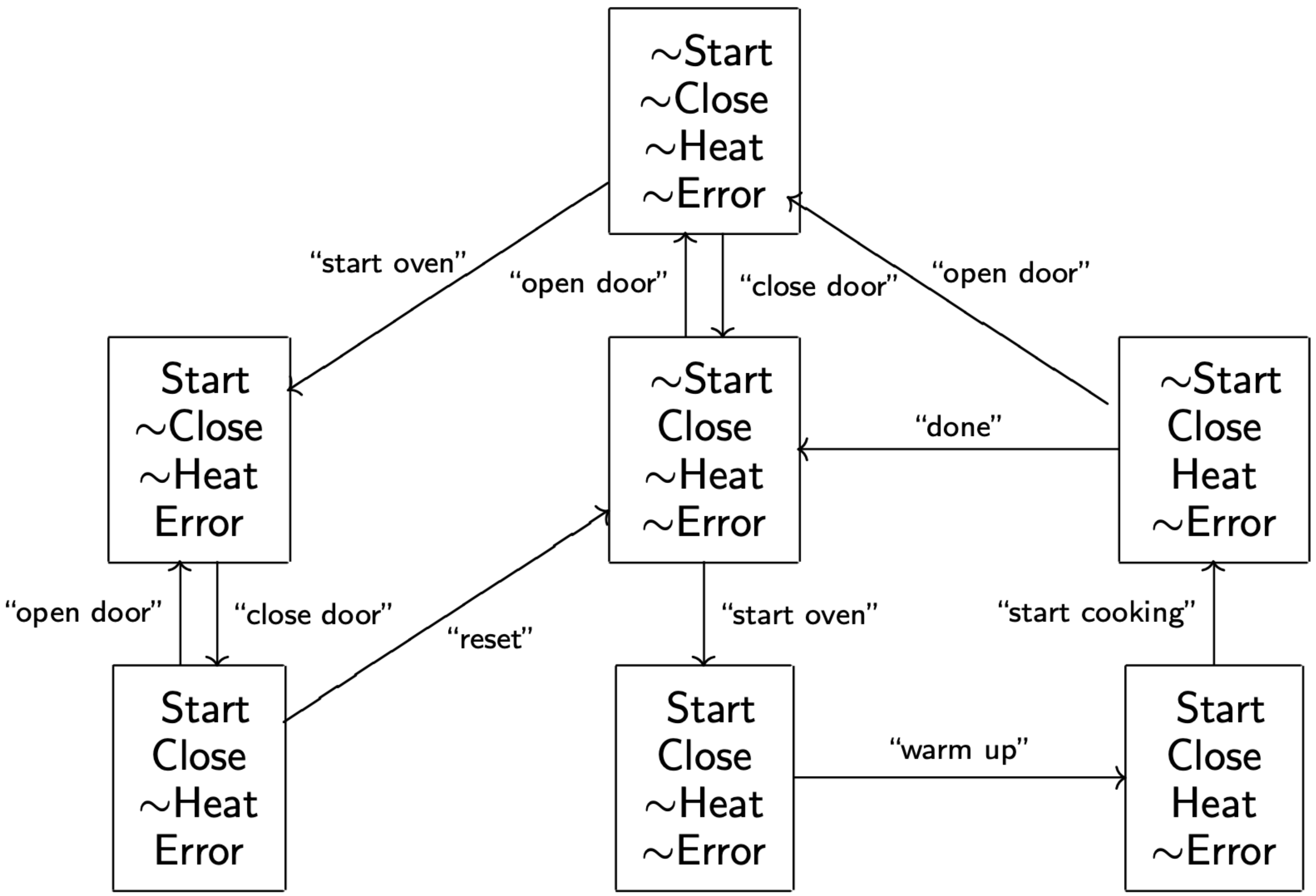
Kojima (RIMS, JP)

Contents

1. CTL is efficient, thanks to fixpoint encoding
2. Why is Probabilistic CTL not as good as CTL?
3. We generalize *CTL: Coalgebraic CTL*

Encode system specifications into modal formulas

	in Math/Logic
(Non-det.) systems	Kripke frames
Specifications	modal formulas



A Kripke frame for a microwave oven [Clarke+'18]

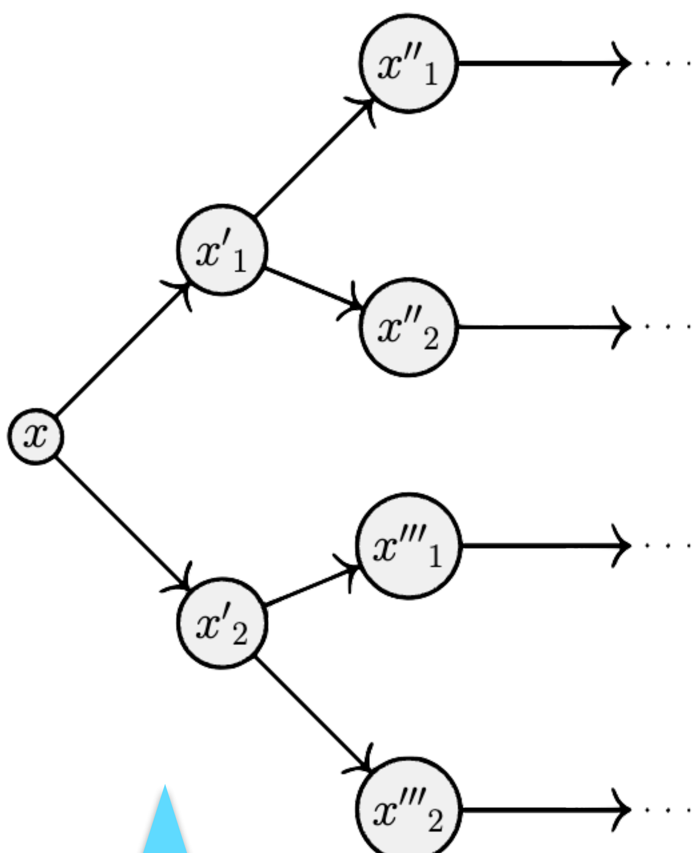
We want to show, for example,

- We can always reach **~Error** ("Liveness property")
- We never reach a critical state **~Close&Heat** ("Safety property")

Specification Language CTL (Computation Tree Logic)

[Emerson&Clarke'82]

CTL is a logic which talks about **computation paths** of a system.



Computation paths of a state x
= "possible futures of x"

Syntax

$$
\begin{aligned}
\theta ::= \; & \top \mid \bot \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2 \\
& \mid \mathrm{EX}\theta \mid \mathrm{AX}\theta \\
& \mid \mathrm{EF}\theta \mid \mathrm{AF}\theta \\
& \mid \mathrm{EG}\theta \mid \mathrm{AG}\theta \\
& \mid \mathrm{E}(\theta_1 \mathrm{U} \theta_2) \mid \mathrm{A}(\theta_1 \mathrm{U} \theta_2) \\
& \mid \mathrm{E}(\theta_1 \mathrm{W} \theta_2) \mid \mathrm{A}(\theta_1 \mathrm{W} \theta_2)
\end{aligned}
$$

CTL has 3 kinds of formulas

$$\theta ::= \top \mid \bot \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2$$

1.**Booleans**

$$\mid \mathsf{EX}\theta \mid \mathsf{AX}\theta$$
$$\mid \mathsf{EF}\theta \mid \mathsf{AF}\theta$$
$$\mid \mathsf{EG}\theta \mid \mathsf{AG}\theta$$
$$\mid \mathsf{E}(\theta_1 \mathsf{U} \theta_2) \mid \mathsf{A}(\theta_1 \mathsf{U} \theta_2)$$
$$\mid \mathsf{E}(\theta_1 \mathsf{W} \theta_2) \mid \mathsf{A}(\theta_1 \mathsf{W} \theta_2)$$

CTL has 3 kinds of formulas

$$
\begin{aligned}
\theta ::= \;& \top \mid \bot \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2 \\
& \mid \mathrm{EX}\theta \mid \mathrm{AX}\theta \\
& \mid \mathrm{EF}\theta \mid \mathrm{AF}\theta \\
& \mid \mathrm{EG}\theta \mid \mathrm{AG}\theta \\
& \mid \mathrm{E}(\theta_1 \mathrm{U} \theta_2) \mid \mathrm{A}(\theta_1 \mathrm{U} \theta_2) \\
& \mid \mathrm{E}(\theta_1 \mathrm{W} \theta_2) \mid \mathrm{A}(\theta_1 \mathrm{W} \theta_2)
\end{aligned}
$$

2.**Existetial/Universal “neXt-time” operators**

CTL has 3 kinds of formulas

$$\theta ::= \top \mid \bot \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2$$
$$\mid \mathrm{EX}\theta \mid \mathrm{AX}\theta$$
$$\mid \mathrm{EF}\theta \mid \mathrm{AF}\theta$$
$$\mid \mathrm{EG}\theta \mid \mathrm{AG}\theta$$
$$\mid \mathrm{E}(\theta_1 \mathrm{U} \theta_2) \mid \mathrm{A}(\theta_1 \mathrm{U} \theta_2)$$
$$\mid \mathrm{E}(\theta_1 \mathrm{W} \theta_2) \mid \mathrm{A}(\theta_1 \mathrm{W} \theta_2)$$

3. **Temporal operators**: capturing eventual/ permanent behaviors

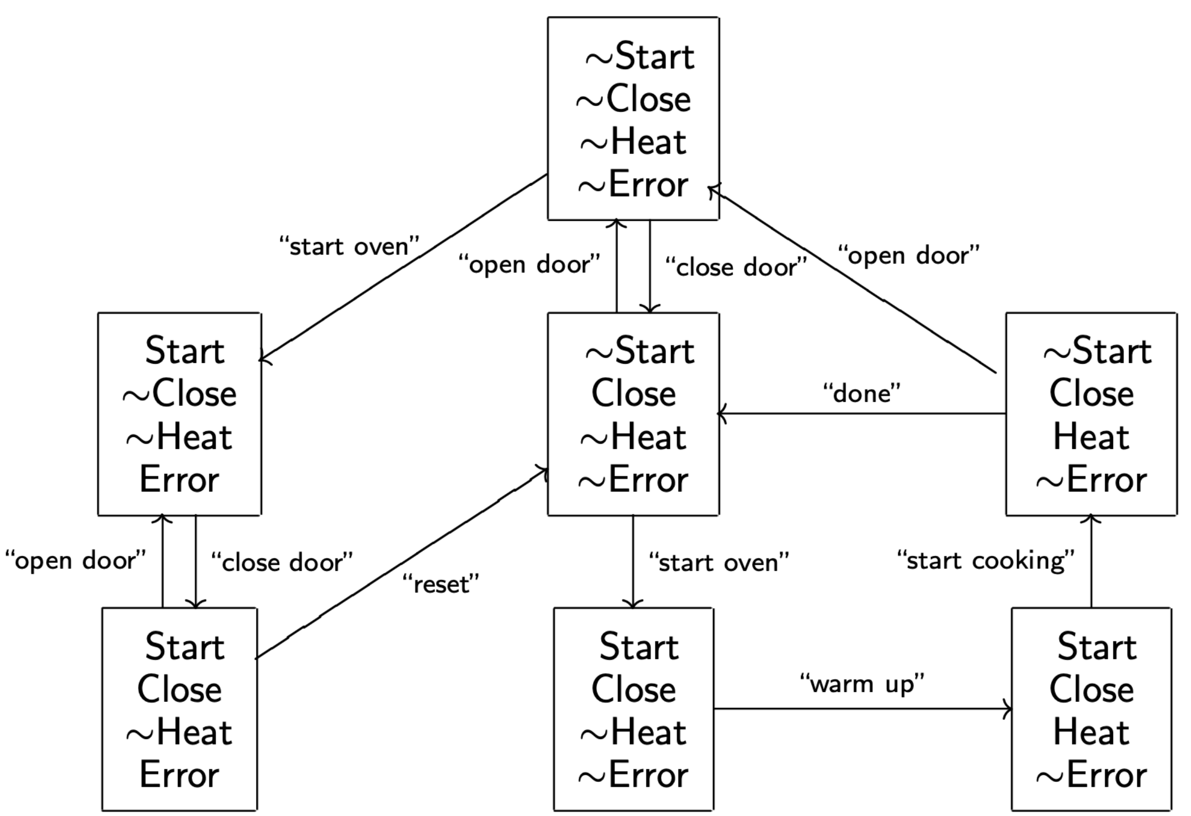
CTL is a logic which talks about **computation paths** of a system.

For example…

- "We can always reach **~Error**"
- "We never reach a critical state **~Close&Heat**"

EF(**~Error**)

AG(**~Close&Heat**)

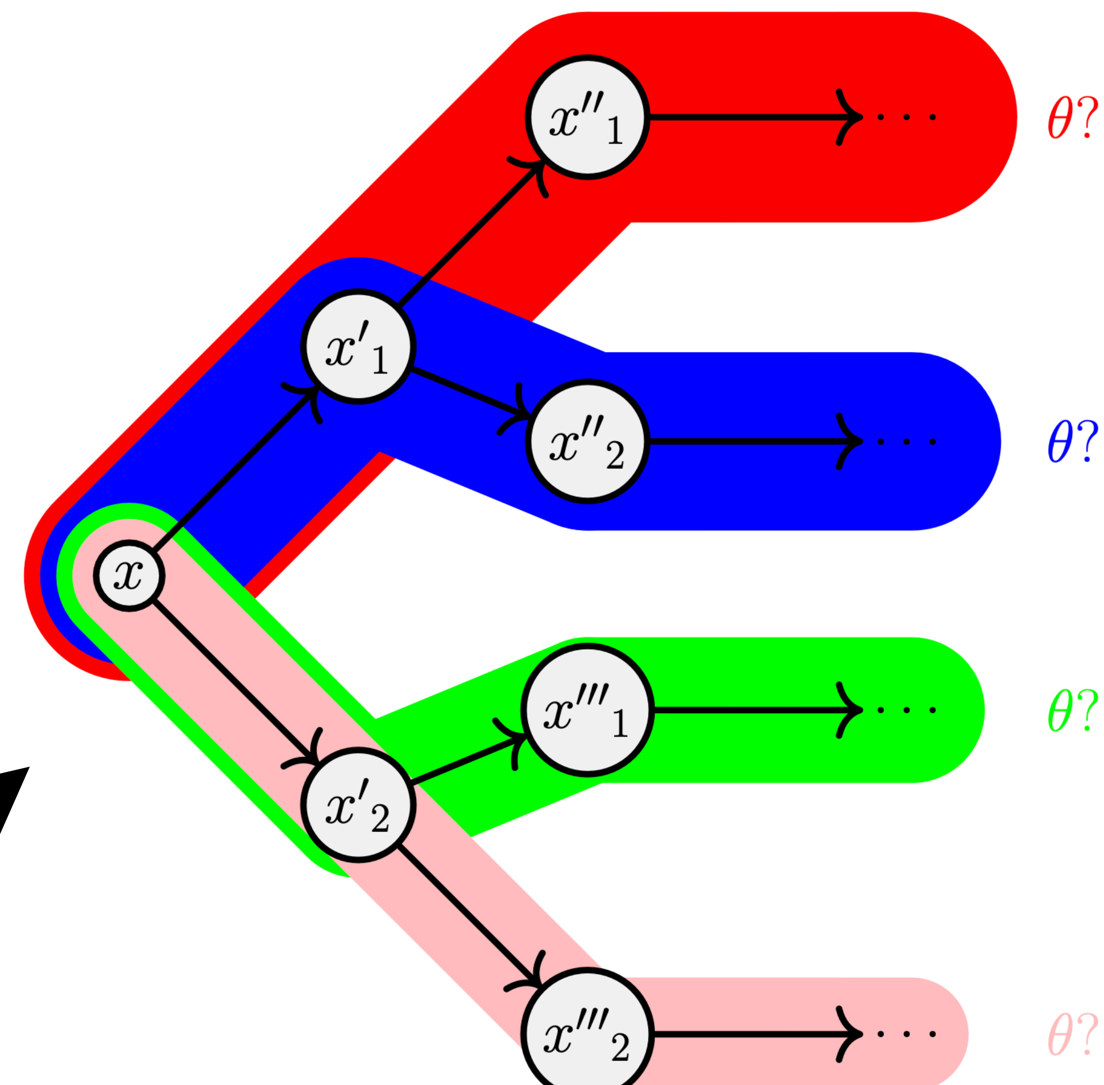


CTL has “**path-based**” semantics

CTL formulas contain path-specifing formulas, like **EF**θ.
So its (default) semantics is exploits computation paths

Concretely

To check **EF**θ, we check along each path whether there is a witness of θ.



CTL is an **optimal** choice!

Among major specification languages…

	CTL	**CTL*** [Emerson&Halpern’85]	**(Alternation-free) Mu-calculus** [Kozen’83]
we saw → Expressive power	High (path-based)	High (path-based)	Low (Step-wise)
Complexity of Model-check	Polynomial/Linear	Exponential	Polynomial

CTL is an **optimal** choice!

Among major specification languages…

	CTL	**CTL*** [Emerson&Halpern'85]	**(Alternation-free) Mu-calculus** [Kozen'83]
Expressive power	High (path-based)	High (path-based)	Low (Step-wise)
Complexity of Model-check	Polynomial/Linear	Exponential	Polynomial

Why so efficient?

Kojima (RIMS, JP)

CTL is an **optimal** choice!

Among major specification languages…

	CTL	**CTL*** [Emerson&Halpern’85]	**(Alternation-free) Mu-calculus** [Kozen’83]
Expressive power	High (path-base		
Complexity of Model-check	Polynomial/Linear	Exponential	Polynomial

Because CTL has an **encoding into Mu-calculus!**

Why so efficient?

Kojima (RIMS, JP)

CTL is **Efficient** since CTL has a **fixpoint encoding**

For example,…

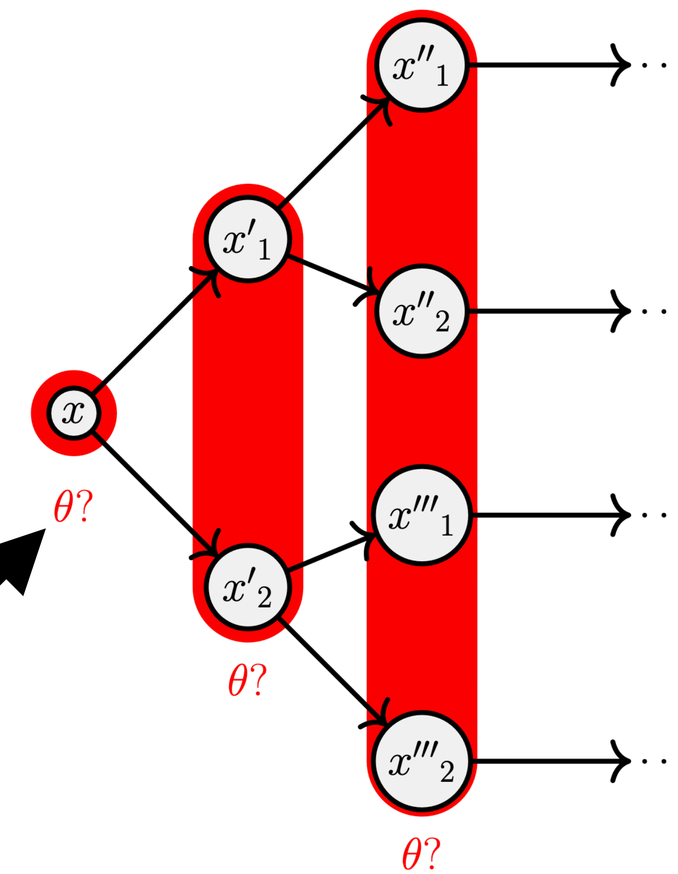
$$\mathrm{EF}\theta \longrightarrow \mu u\,.\,\theta \vee \mathrm{EX}u$$

Mu-calculus [Kozen'83]

$$\theta ::= u \mid \top \mid \bot \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2 \mid \mathrm{EX}\theta \mid \mathrm{AX}\theta \mid \mu u\,.\,\theta \mid \nu u\,.\,\theta$$

A fixpoint formula can be calculated in a **step-wise** manner

To calculate $\mu u \, . \, \theta \vee \mathrm{EX}u$,
we search a witness of θ
step-by-step,
taking succceors each step.

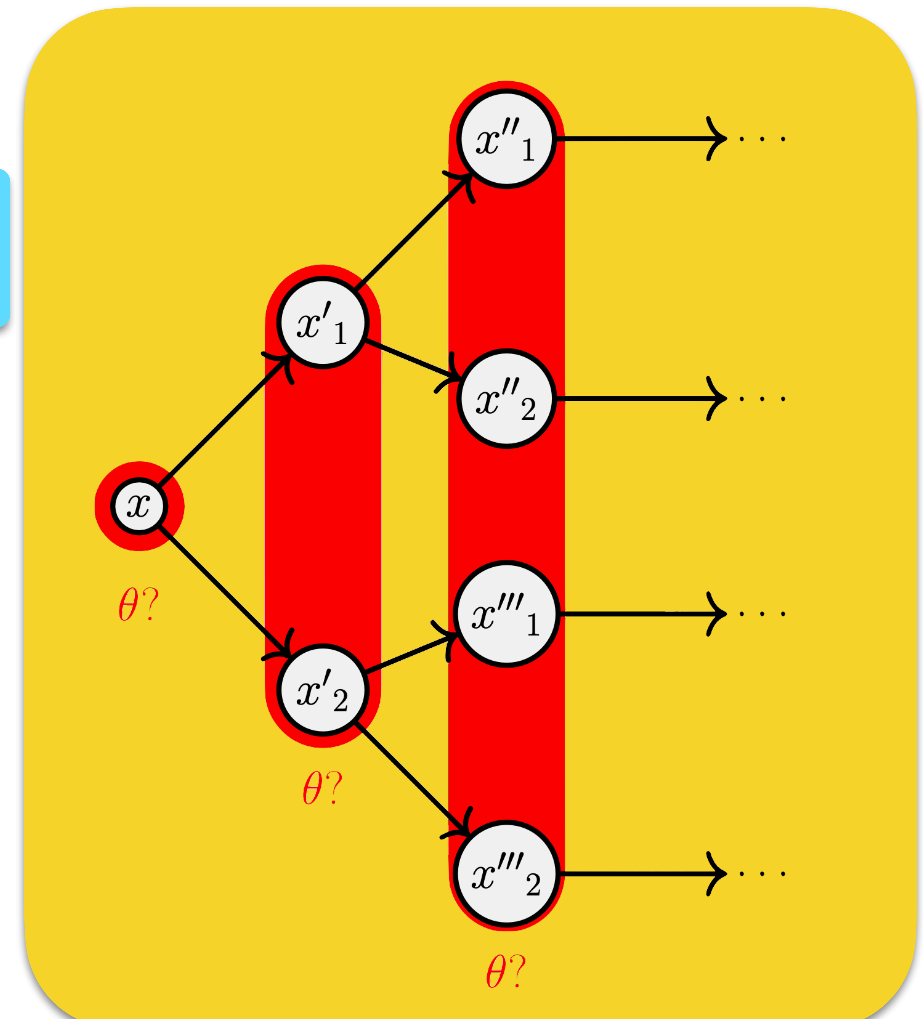


0	$\bot = \emptyset$
1	$\theta \vee \mathrm{EX}\emptyset = \theta$
2	$\theta \vee \mathrm{EX}\theta = \{x \in X \mid \exists x' . \, x \to x \text{ and } x' \vDash \theta\}$
…	

Step-wise semantics of CTL is given by fixpoint encoding

An intermediate fixpoint formula

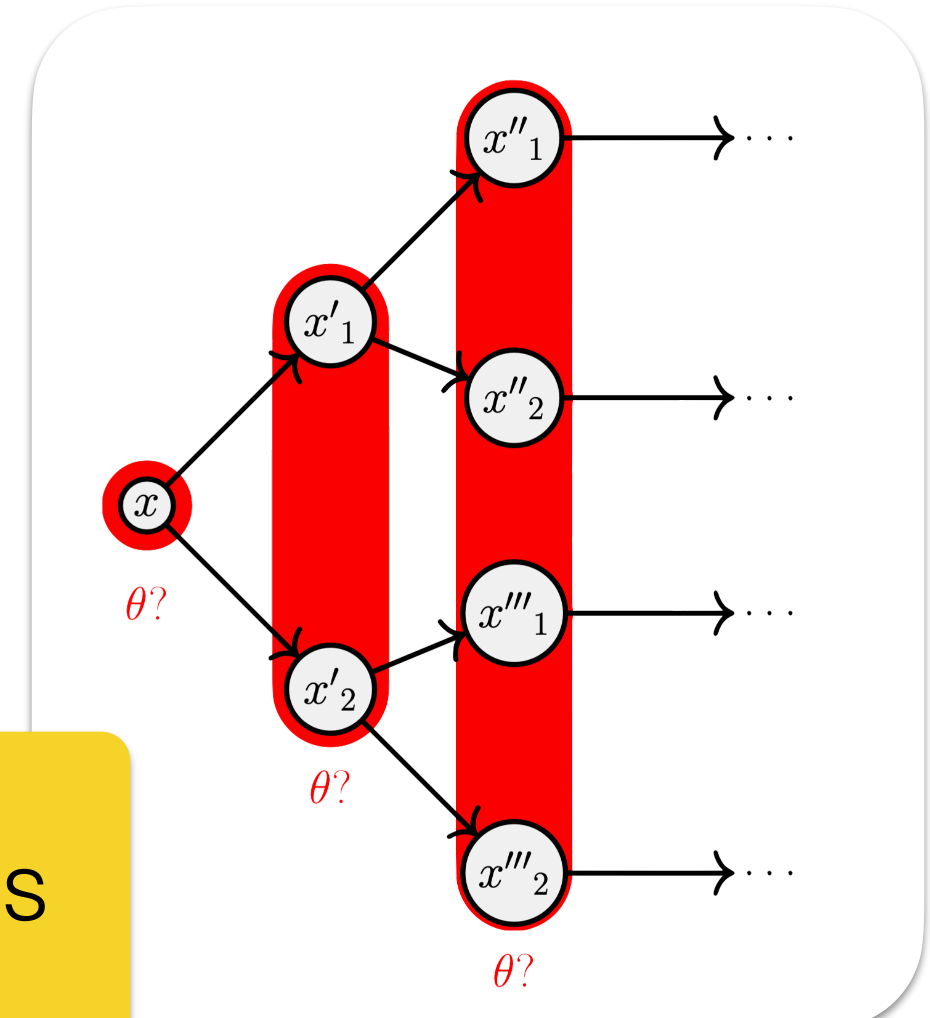
EFθ $\longrightarrow$ $\mu u . \theta \vee \mathsf{EX} u$ $\longrightarrow$



Kojima (RIMS, JP)

Step-wise semantics of CTL is given by fixpoint encoding

$\mathsf{EF}\theta \longrightarrow \mu u \,.\, \theta \vee \mathsf{EX} u \longrightarrow$

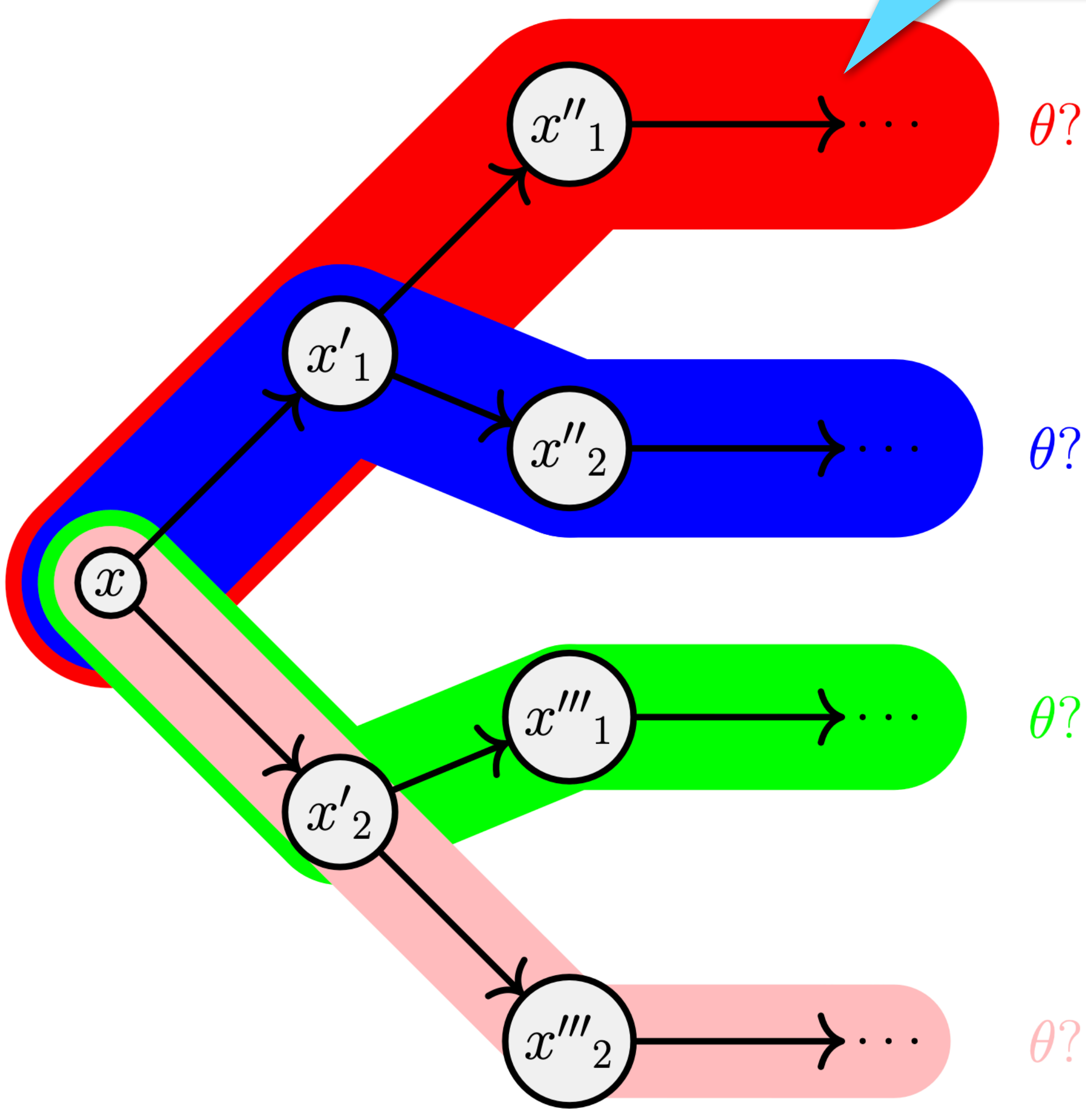


Those formulas which emerge in this encoding are all **alternation-free**, so <u>their model-checking takes only **poly-time**</u>!

Kojima (RIMS, JP)

CTL is **Expressive**

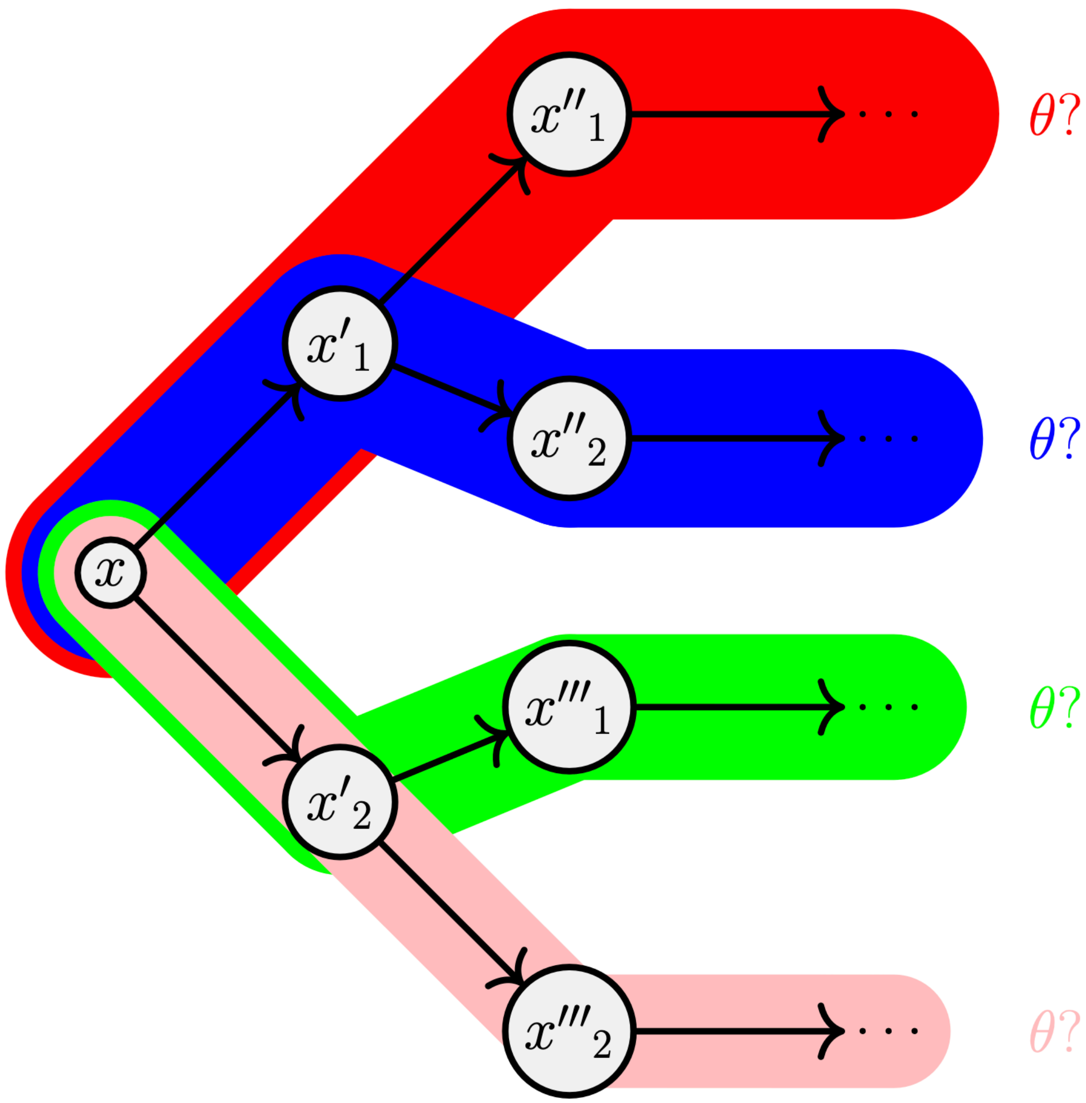
Path-based semantics

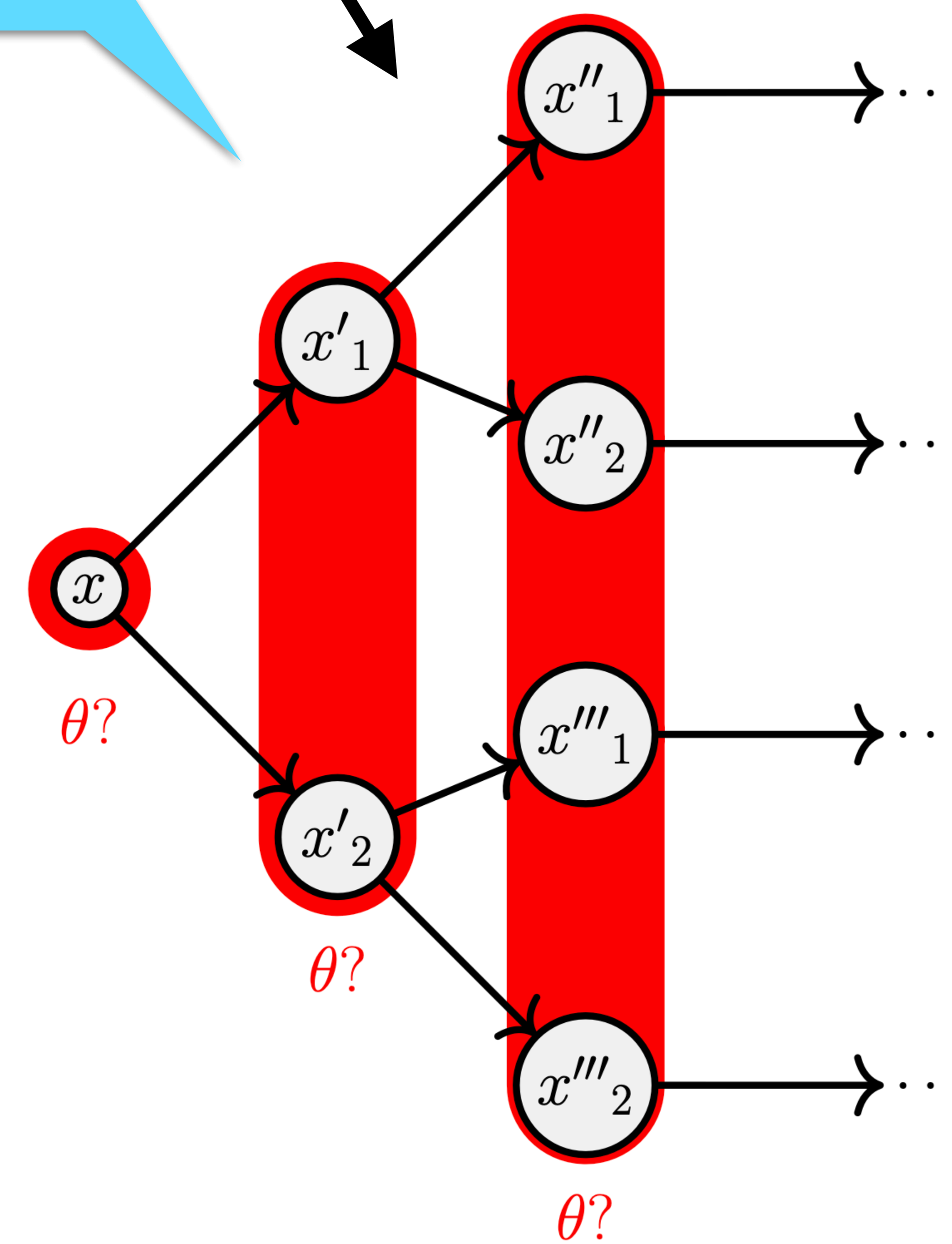


Kojima (RIMS, JP)

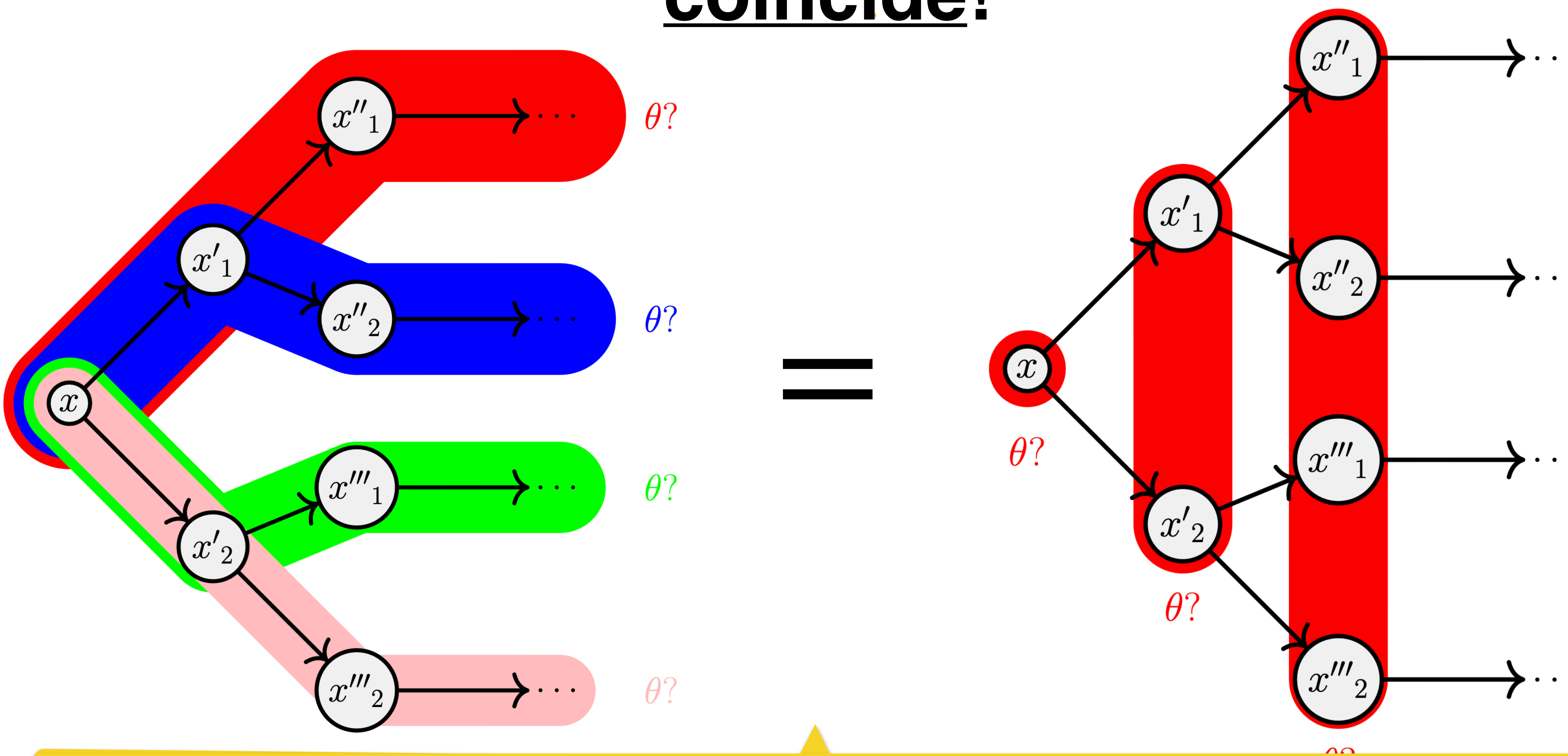
CTL is **Efficient**

Step-wise semantics





CTL is **Optimal** since…
path-based and **step-wise** semantics **coincide!**



"The fixpoint encoding preserves semantics"

= **Fixpoint Characterization** [Emerson&Halpern'85]

Contents

1. CTL is efficient, thanks to fixpoint encoding
2. Why is Probabilistic CTL not as good as CTL?
3. We generalize *CTL: Coalgebraic CTL*

Probabilistic CTL (PCTL)

[Hansson&Jonsson'94]

$$\theta ::= \top \mid \bot \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2$$
$$\mid \mathsf{P}_{\geq r}\mathsf{X}\theta \mid \mathsf{P}_{>r}\mathsf{X}\theta$$
$$\mid \mathsf{P}_{\geq r}\mathsf{F}\theta \mid \mathsf{P}_{>r}\mathsf{F}\theta$$
$$\mid \mathsf{P}_{\geq r}\mathsf{G}\theta \mid \mathsf{P}_{>r}\mathsf{G}\theta$$
$$\mid \mathsf{P}_{\geq r}(\theta_1 \mathsf{U} \theta_2) \mid \mathsf{P}_{>r}(\theta_1 \mathsf{U} \theta_2)$$
$$\mid \mathsf{P}_{\geq r}(\theta_1 \mathsf{W} \theta_2) \mid \mathsf{P}_{>r}(\theta_1 \mathsf{W} \theta_2)$$

PCTL has the "threshold" quantifiers $\mathsf{P}_{\geq r}$, $\mathsf{P}_{>r}$ instead of **E**, **A**

Kojima (RIMS, JP)

Fixpoint characterization **fails** in PCTL…

	CTL	PCTL
Systems	Kripke frames	Markov chains
Path-based sem.	✓	✓
Step-wise sem.	✓	✓
Fix-Pt. Char.	✓	✗
fixpoint MC algo.	Polynomial (Linear)	✗

Fixpoint characterization **fails** in PCTL…

	CTL	PCTL
Systems	Kripke frames	Markov chains
Path-based sem.	✓	✓
Step-wise sem.	✓	✓
Fix-Pt. Char.	✓	✗
fixpoint MC algo.	Polynomial (Linear)	✗

Discontent…

- Not clear what logic deserves the name "CTL" ✗
- No generic notion of "efficient" path-based logic ✗

Our Contributions

Ours!

	CTL	PCTL	CCTL
Systems	Kripke frames	Markov chains	TF -coalgebra
Path-based sem.	✓	✓	✓
Step-wise sem.	✓	✓	✓
Fix-Pt. Char.	✓	✗	✓ Thm. 4.6 & Assum 4.7
fixpoint MC algo.	Polynomial (Linear)	✗	Polynomial (Algo.1)

1. Introduced **Coalgebraic CTL** (CCTL) (Def 3.7)
2. Formulated **Coalgebraic Fix. Ch.** (Thm 4.6)
3. Identified **sufficient condition** for it (Assum 4.7)
4. Introduced a **poly-time MC algo**. for CCTL (Algo.1)

Contents

1. CTL is efficient, thanks to fixpoint encoding
2. Why is Probabilistic CTL not as good as CTL?
3. We generalize *CTL: Coalgebraic CTL*

Our semantic domain has 7 genericities

A *BT-situation* $\mathcal{S} = (\mathbf{C}, T, F, c, \Omega, \Sigma, \Lambda)$ is…

Types	A category $\mathbf{C}$
Branching type	A monad $T\colon \mathbf{C} \to \mathbf{C}$
Transition type	An endofunctor $F\colon \mathbf{C} \to \mathbf{C}$
A system	A coalgebra $c\colon X \to TFX$
Values of predicates	An object $\Omega \in \mathbf{C}$
Path-quantifiers	A set of predicate liftings of T $\Sigma = \{\sigma\colon \Omega^{(\_)} \to \Omega^{T(\_)}\}_{\sigma\in\Sigma}$
Next-time operators	A set of predicate liftings of F $\Lambda = \{\lambda\colon \Omega^{(\_)} \to \Omega^{F(\_)}\}_{\lambda\in\Lambda}$

Kojima (RIMS, JP)

Our semantic domain has 7 genericities

A *BT-situation* $\mathcal{S} = (\mathbf{C}, T, F, c, \Omega, \Sigma, \Lambda)$ is…

The powerset monad in CTL,
The Giry monad in PCTL

Types	A category $\mathbf{C}$
Branching type	A monad $T\colon \mathbf{C} \to \mathbf{C}$
Transition type	An endofunctor $F\colon \mathbf{C} \to \mathbf{C}$
A system	A coalgebra $c\colon X \to TFX$
Values of predicates	An object $\Omega \in \mathbf{C}$
Path-quantifiers	A set of predicate liftings of T $\Sigma = \{\sigma\colon \Omega^{(\_)} \to \Omega^{T(\_)}\}_{\sigma\in\Sigma}$
Next-time operators	A set of predicate liftings of F $\Lambda = \{\lambda\colon \Omega^{(\_)} \to \Omega^{F(\_)}\}_{\lambda\in\Lambda}$

$\{\Diamond, \Box\}$ in CTL,
$\{\geq_r, >_r\}_{r\in[0,1]}$ in PCTL

How to generalize CTL?

<u>There are 2 main ideas:</u>

First, we generalize modalities in CTL to **predicate liftings:**

σ is pred. liftings of T

path quantifiers: E, A $\longrightarrow \spadesuit_\sigma \quad (\sigma \in \Sigma)$

Next-time operators: X $\longrightarrow \heartsuit_\lambda \quad (\lambda \in \Lambda)$

λ is a pred. lifting of F

How to generalize CTL?

There are 2 main ideas:

Second, we use the following identification:

$$\mathsf{F}\theta \equiv \mu u\,.\,\theta \vee \mathsf{X}u$$
$$\mathsf{G}\theta \equiv \nu u\,.\,\theta \wedge \mathsf{X}u$$
$$\ldots$$

Namely, F, G, U, W in CTL are **writen as LFP/GFP of** X!

Here, X is interpreted as an operator on **path-**formulas, an extended class of formulas from CTL.

How to generalize CTL?

<u>There are 2 main ideas:</u>

Second, we use the following identification:

$$\mathsf{F}\theta \equiv \mu u \,.\, \theta \vee \mathsf{X}u$$
$$\mathsf{G}\theta \equiv \nu u \,.\, \theta \wedge \mathsf{X}u$$
$$\ldots$$

Namely, F, G, U, W in CTL are **writen as LFP/GFP of** X!

Thus, we can write, for example, in the F case,

$$\mathsf{EF}\theta \equiv \mathsf{E}(\mu u \,.\, \theta \vee \mathsf{X}u)$$
$$\mathsf{AF}\theta \equiv \mathsf{A}(\mu u \,.\, \theta \vee \mathsf{X}u)$$

Kojima (RIMS, JP)

Coalgebraic CTL (CCTL)

Syntax Σ, Λ : set, Γ : ranked set, $\Gamma_\mu, \Gamma_\nu \subseteq \Gamma$

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

Boolean oper. (made of $\top$, $\bot$, $\wedge$, $\vee$)

Quantified next-time oper. (like EX/AX)

Temporal operators:
Generalization of **EF, EU, AF, AU,** in the LFP (μ) case, and generalization of **EG, EW, AG, AW,** in the GFP (ν) case

Kojima (RIMS, JP)

CCTL's path-based semantics is given by **infinite trace** [Jacobs'04]

Briefly,…

- Notion of computation tree is replaced by infinite trace of T and $F_X = X \times F$.
- The trace map is a Kleisli map

$$\mathbf{tr}\colon X \to TZ_X$$

where Z_X is the final F_X -coalgebra, called **generalized stream object**. Z_X is a coalgebraic version of **path space.**

Fixpoint Encoding of CCTL

Encoding

$$
\begin{aligned}
\epsilon\big(\boxdot_\gamma(\psi_1,\dots,\psi_{|\gamma|})\big) &:= \boxdot_\gamma(\epsilon\psi_1,\dots,\epsilon\psi_{|\gamma|}),\\
\epsilon\big(\spadesuit_\sigma \heartsuit_\lambda \psi\big) &:= \spadesuit_\sigma \heartsuit_\lambda(\epsilon\psi),\\
\epsilon\Big(\spadesuit_\sigma\big(\mu u.\, \boxdot_{\gamma_\mu}\,(\psi_1,\dots,\psi_{|\gamma_\mu|-1}, \heartsuit_\lambda u)\big)\Big) &:= \mu u.\, \boxdot_{\gamma_\mu}\,(\epsilon\psi_1,\dots,\epsilon\psi_{|\gamma_\mu|-1}, \spadesuit_\sigma \heartsuit_\lambda u),\\
\epsilon\Big(\spadesuit_\sigma\big(\nu u.\, \boxdot_{\gamma_\nu}\,(\psi_1,\dots,\psi_{|\gamma_\nu|-1}, \heartsuit_\lambda u)\big)\Big) &:= \nu u.\, \boxdot_{\gamma_\nu}\,(\epsilon\psi_1,\dots,\epsilon\psi_{|\gamma_\nu|-1}, \spadesuit_\sigma \heartsuit_\lambda u).
\end{aligned}
$$

Idea

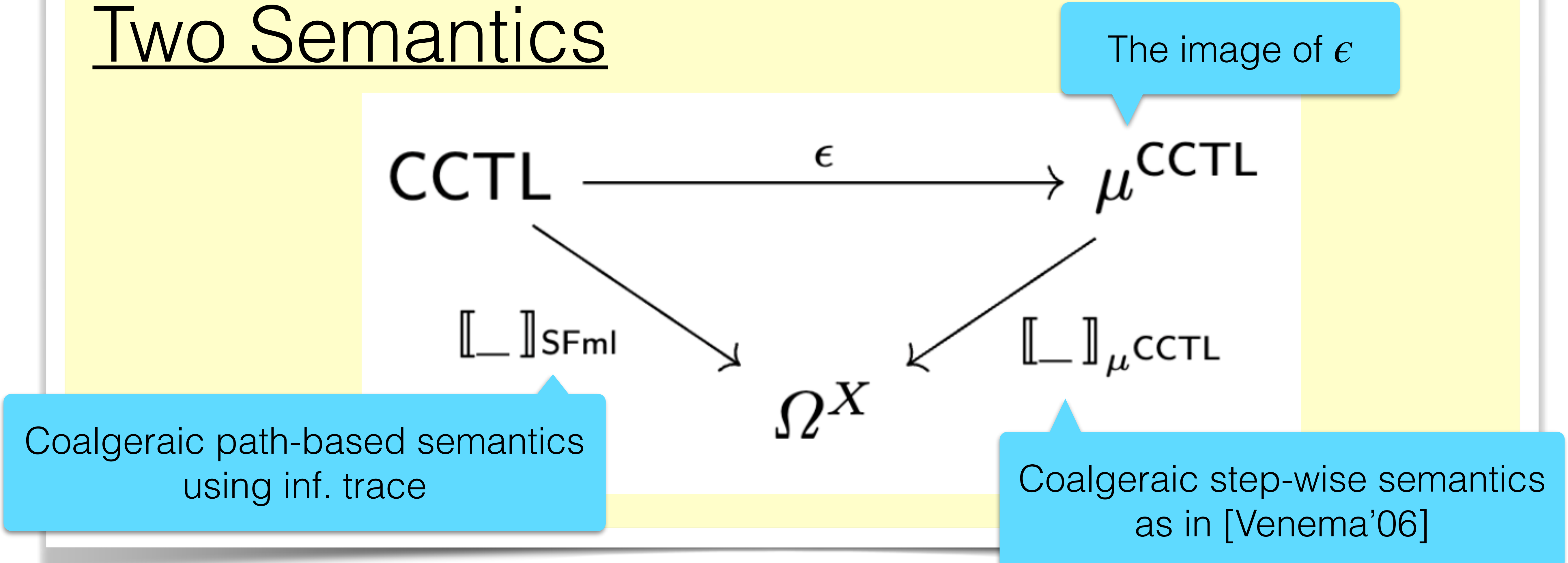
Transform LFP/GFP of **next-time** oper. (X) on paths to LFP/GFP of **quantified next-time** operators (like EX, AX).

Each application of $\spadesuit_\sigma$ is distributed inside μ, ν

MS, JP)

Coalgebraic Fixpoint Characterization

Two Semantics



Thm 4.6

The above triangle commutes.

Sufficient conditions

Assum 4.7

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

$$\begin{array}{ccc} X \times TZ_X & \xrightarrow{\mathsf{st}_{X,Z_X}} & T(X \times Z_X) \\ {\scriptstyle \langle \mathrm{id}_X, \mathrm{tr}(c') \rangle}\uparrow & & \uparrow{\scriptstyle T\langle \zeta_1, \mathrm{id}_{Z_X} \rangle} \\ X & \xrightarrow{\mathrm{tr}(c')} & TZ_X, \end{array} \tag{5}$$

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

$$[\![\spadesuit_\sigma]\!](\mu\Phi_{\lambda,\gamma_\mu,\iota\vec{\theta}_{|\gamma_\mu|}}) \sqsubseteq \mu\Psi_{(\sigma,\lambda),\gamma_\mu,\vec{\theta}_{|\gamma_\mu|}}, \tag{6}$$

$$[\![\spadesuit_\sigma]\!](\nu\Phi_{\lambda\gamma_\nu,\iota\vec{\theta}_{|\gamma_\nu|}}) \sqsupseteq \nu\Psi_{(\sigma,\lambda),\gamma_\nu,\vec{\theta}_{|\gamma_\nu|}}, \tag{7}$$

   for every tuple of $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ formulas $\vec{\theta}_{|\gamma|} = (\theta_1, \ldots, \theta_{|\gamma|})$,
5. for every $\gamma \in \Gamma_\mu \cup \Gamma_\nu$ and $\sigma \in \Sigma$, $\gamma \colon \Omega^{|\gamma|} \to \Omega$ is bilinear [10, Section 1] with respect to the T-algebra $\mathrm{ev}_\sigma \colon T\Omega \to \Omega$,
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.

Sufficient conditions

Assum 4.7

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

$$\begin{array}{ccc} X \times TZ_X & \xrightarrow{\mathrm{st}_{X,Z_X}} & T(X \times Z_X) \\ {\scriptstyle \langle \mathrm{id}_X, \mathrm{tr}(c') \rangle}\uparrow & & \uparrow{\scriptstyle T\langle \zeta_1, \mathrm{id}_{Z_X} \rangle} \\ X & \xrightarrow{\mathrm{tr}(c')} & TZ_X , \end{array} \tag{5}$$

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

$$[\![\spadesuit_\sigma]\!](\mu\Phi_{\lambda,\gamma_\mu,\iota\vec{\theta}_{|\gamma_\mu|}}) \sqsubseteq \mu\Psi_{(\sigma,\lambda),\gamma_\mu,\vec{\theta}_{|\gamma_\mu|}}, \tag{6}$$

$$[\![\spadesuit_\sigma]\!](\nu\Phi_{\lambda\gamma_\nu,\iota\vec{\theta}_{|\gamma_\nu|}}) \sqsupseteq \nu\Psi_{(\sigma,\lambda),\gamma_\nu,\vec{\theta}_{|\gamma_\nu|}}, \tag{7}$$

   for every tuple of $\mu^{\mathsf{CCTL}}_{\Gamma_\mu,\Gamma_\nu}$ formulas $\vec{\theta}_{|\gamma|} = (\theta_1, \ldots, \theta_{|\gamma|})$,
5. for every $\gamma \in \Gamma_\mu \cup \Gamma_\nu$ and $\sigma \in \Sigma$, $\gamma \colon \Omega^{|\gamma|} \to \Omega$ is bilinear [10, Section 1] with respect to the T-algebra $\mathrm{ev}_\sigma \colon T\Omega \to \Omega$,
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.

(4) classifies CTL & PCTL

Cond. (4) in CTL is easy!

Cond. (4) for **EF**, for example, is…

$$x \vDash \mathsf{EF}\theta \Longrightarrow x \vDash \mu u\,.\,\theta \vee \mathsf{EX}u$$

"There is a path π of x, along which we reach θ in future"

"There is a reachable state x' from x with $x' \vDash \theta$"

Cond. (4) in **not** valid in PCTL…

the $\mathsf{P}_{\geq 1}\mathsf{F}$ case (we put here $\theta = p$):

$$x \vDash \mathsf{P}_{\geq 1}\mathsf{F}p \implies x \vDash \mu u\,.\,p \vee \mathsf{P}_{\geq 1}\mathsf{X}u$$

"Almost surely p in future"

LFP of "p or almost surely u in next-step"

Kojima (RIMS, JP)

Cond. (4) in **not** valid in PCTL…

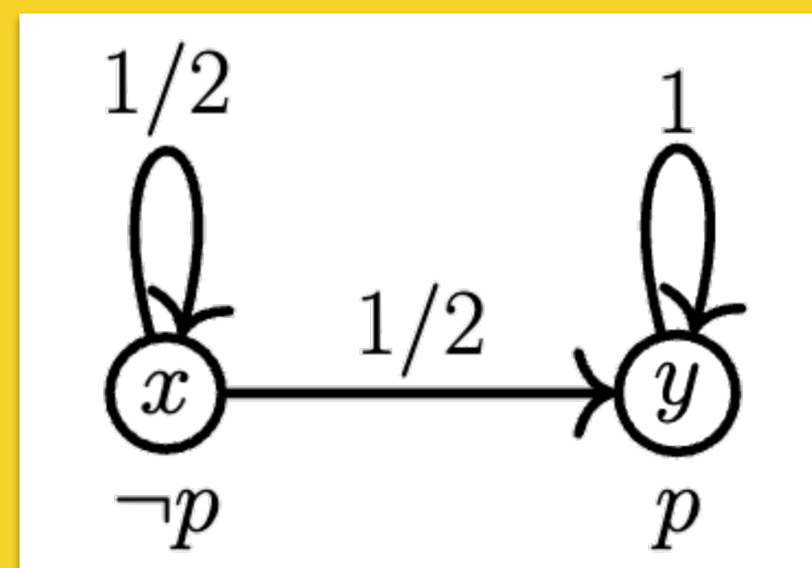
the $\mathsf{P}_{\geq 1}\mathsf{F}$ case (we put here $\theta = p$):

$$x \vDash \mathsf{P}_{\geq 1}\mathsf{F}p \Longrightarrow x \vDash \mu u\,.\,p \vee \mathsf{P}_{\geq 1}\mathsf{X}u$$

"Almost surely p in future"

LFP of "p now or almost surely u in next-step"

- LHS = $\{x, y\} \not\subseteq \{y\}$ = RHS!
- LHS measures "global" behaviour, but RHS only cares "local" behavior.



Results obtained without cond. (4):

- <u>Coalgebraic expansion law</u>

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*

$$[\![\spadesuit_\sigma]\!](\mu \Phi_{\lambda,\gamma_\mu,\iota\vec{\theta}_{|\gamma_\mu|-1}}) \sqsupseteq \Psi_{(\sigma,\lambda),\gamma_\mu,\vec{\theta}_{|\gamma_\mu|-1}}([\![\spadesuit_\sigma]\!](\mu \Phi_{\lambda,\gamma_\mu,\iota\vec{\theta}_{|\gamma_\mu|-1}})) \tag{7}$$

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

$$[\![\spadesuit_\sigma]\!](\nu \Phi_{\lambda,\gamma_\nu,\iota\vec{\theta}_{|\gamma_\nu|-1}}) \sqsubseteq \Psi_{(\sigma,\lambda),\gamma_\nu,\vec{\theta}_{|\gamma_\nu|-1}}([\![\spadesuit_\sigma]\!](\nu \Phi_{\lambda,\gamma_\nu,\iota\vec{\theta}_{|\gamma_\nu|-1}})) \tag{8}$$

for $\theta_1, \ldots, \theta_{|\gamma_\nu|-1}$ with $[\![\iota\theta_i]\!]_{\mathsf{SFml}} \sqsubseteq [\![\theta_i]\!]_{\mu\mathsf{CCTL}}$ for $i = 1, \ldots, |\gamma_\nu| - 1$. Furthermore, if $[\![\iota\theta_i]\!]_{\mathsf{SFml}} = [\![\theta_i]\!]_{\mu\mathsf{CCTL}}$ for every subformula θ_i, the inequalities 7 and 8 are both equalities.

- <u>Partial Fixpoint Characterization</u>

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

1. *$[\![\theta]\!]_{\mu\mathsf{CCTL}} = [\![\iota\theta]\!]_{\mathsf{SFml}}$ for a formula θ without any μ or ν,*
2. *$[\![\theta]\!]_{\mu\mathsf{CCTL}} \sqsubseteq [\![\iota\theta]\!]_{\mathsf{SFml}}$ for a formula θ with only μs, and*
3. *$[\![\theta]\!]_{\mu\mathsf{CCTL}} \sqsupseteq [\![\iota\theta]\!]_{\mathsf{SFml}}$ for a formula θ with only νs.*

Results valid without (4):

- Coalgebraic expansion law

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*

$$[\![\spadesuit_\sigma]\!](\mu\Phi_{\lambda,\gamma_\mu,\iota\vec{\theta}_{|\gamma_\mu|-1}}) \sqsupseteq \Psi_{(\sigma,\lambda \ldots} \ldots \quad (7)$$

for $\theta_1, \ldots, \theta_{|\gamma_\mu|-1}$ with $[\![\iota\theta_i]\!]_{\mathsf{SFml}} \sqsupseteq [\![\theta_i \ldots$

$$[\![\spadesuit_\sigma]\!](\nu\Phi_{\lambda,\gamma_\nu,\iota\vec{\theta}_{|\gamma_\nu|-1}}) \sqsubseteq \Psi_{(\sigma,\lambda \ldots}$$

for $\theta_1, \ldots, \theta_{|\gamma_\nu|-1}$ with $[\![\iota\theta_i]\!]_{\mathsf{SFml}} \sqsubseteq [\![\theta_i \ldots$ if $[\![\iota\theta_i]\!]_{\mathsf{SFml}} = [\![\theta_i]\!]_{\mu^{\mathsf{CCTL}}}$ for every subf… equalities.

Qualitative variant of PCTL satisfies all but(4), so it enjoys partial Fix. Ch.

- Partial fixpoint characterization

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

1. $[\![\theta]\!]_{\mu^{\mathsf{CCTL}}} = [\![\iota\theta]\!]_{\mathsf{SFml}}$ *for a formula θ without any μ or ν,*
2. $[\![\theta]\!]_{\mu^{\mathsf{CCTL}}} \sqsubseteq [\![\iota\theta]\!]_{\mathsf{SFml}}$ *for a formula θ with only μs, and*
3. $[\![\theta]\!]_{\mu^{\mathsf{CCTL}}} \sqsupseteq [\![\iota\theta]\!]_{\mathsf{SFml}}$ *for a formula θ with only νs.*

Poly-time MC for CCTL

Idea behind our algo.

1. Encode CCTL into a (coalgebaic) fixpoint logic
2. Calculate fixpoint formulas, step-wisely

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

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

procedure CHECK(θ)
  switch θ **do**

    case $\boxdot_\gamma(\theta_1, \ldots, \theta_{|\gamma|})$
      return $\gamma(\mathrm{CHECK}(\theta_1), \ldots, \mathrm{CHECK}(\theta_{|\gamma|}))$
    end case

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

    case $\mu u.\, \boxdot_\gamma\, (\theta_1, \ldots, \theta_{|\gamma_\mu|-1}, \spadesuit_\sigma \heartsuit_\lambda u)$
      $U := \bot;\ V := \gamma_\mu(\mathrm{CHECK}(\theta_1), \ldots, \mathrm{CHECK}(\theta_{|\gamma_\mu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](\bot))$
      while $U \neq V$ **do**
        $U := V$
        $V := \gamma_\mu(\mathrm{CHECK}(\theta_1), \ldots, \mathrm{CHECK}(\theta_{|\gamma_\mu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](U))$
      end while
      return U
    end case

    case $\nu u.\, \boxdot_{\gamma_\nu}\, (\theta_1, \ldots, \theta_{|\gamma_\nu|-1}, \spadesuit_\sigma \heartsuit_\lambda u)$
      $U := \top;\ V := \gamma_\nu(\mathrm{CHECK}(\theta_1), \ldots, \mathrm{CHECK}(\theta_{|\gamma_\nu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](\top))$
      while $U \neq V$ **do**
        $U := V$
        $V := \gamma_\nu(\mathrm{CHECK}(\theta_1), \ldots, \mathrm{CHECK}(\theta_{|\gamma_\nu|-1}), [\![\spadesuit_\sigma \heartsuit_\lambda]\!](U))$
      end while
      return U
    end case

end procedure
return CHECK($\iota^{-1}\psi$)

Our model checking algorithm $\mathsf{MC}^{\mathsf{CCTL}}_{\mathcal{S}}$

Here suppose **finite** coalgebra in a **concrete** category

Kojima (RIMS, JP)

Poly-time MC for CCTL

Correctness (prop. 5.2)

$\mathrm{MC}^{\mathrm{CCTL}}_{\mathcal{S}}$ terminates and returns $[\![\psi]\!]_{\mathsf{SFml}}$ for $\psi \in \mathrm{CCTL}$.

A key is semantics-preservation of our encoding!

Complexity (prop. 5.4)

$$\mathcal{O}(|\psi| \cdot |X| \cdot N \cdot t(\sigma, \lambda))$$

- $|\psi|$: the number of subformulas
- N : the maximal time to execute boolean opr.
- $t(\sigma, \lambda)$: the maximal time to solve $x \in [\![\spadesuit_\sigma \heartsuit_\lambda]\!](U)$ for $x \in X$ and $U \in \Omega^X$

Our encoding is linear-time, and encoded formula is alternation-free!

	CTL	PCTL	CCTL
Systems	Kripke frames	Markov chains	TF-coalgebra
Path-based sm.	✓	✓	✓
Step-wise sm.	✓	✓	✓
Fix-Pt. Ch.	✓	✗	✓ Thm. 4.6 & Assum 4.7
fixpoint MC algo.	Polynomial (Linear)	✗	Polynomial (Algo.1)

Ours!

Future Work

- Find a nice **probablilistic path-based logic** in which Fix-Pt. Ch. holds
- Formalize a **path-based version of Parikh's game logic** [Parikh'85], analyzing the neighbouhood monad
- Generalize a fixpoint encoding of **CTL*** [Cirstea'11]

Kojima (RIMS, JP)

	CTL	PCTL	CCTL
Systems	Kripke frames	Markov chains	TF-coalgebra
Path-based sm.	✓	✓	✓
Step-wise sm.	✓	✓	✓
Fix-Pt. Ch.	✓	✗	✓ Thm. 4.6 & Assum 4.7
fixpoint MC algo.	Polynomial (Linear)	✗	Polynomial (Algo.1)

Ours!

Future Work

- Find a nice **probablilistic path-based logic** in which Fix-Pt. Ch. holds.
- Formalize a **path-based version of Parikh's game logic** [Parikh'85], analyzing the neighbouhood monad
- Generalize a fixpoint encoding of **CTL*** [Cirstea'11]

Thanks!