

Path-Based Coalgebraic Temporal Logics

Corina Cîrstea

School of Electronics and Computer Science
University of Southampton

Overview

- several known **path-based** temporal specification logics:
 - CTL* on transition systems
 - PCTL on probabilistic transition systems
- similarities not sufficiently understood/exploited

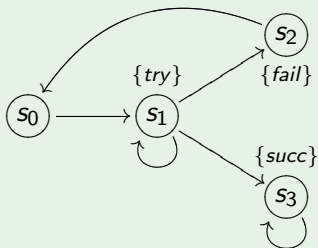
Goal: find a **unifying** pattern, based on a notion of **infinite computation path**

- existing general theory of *finite* traces [Hasuo et. al.]
- existing definition of *infinite* traces for $T = \mathcal{P}$ [Jacobs '04]

Transition Systems

- $\mathcal{P}^+ : \text{Set} \rightarrow \text{Set}$, $\mathcal{P}^+(S) =$ set of *non-empty* subsets of S :
 \mathcal{P}^+ -coalgebras are (restricted) transition systems.

Example



Some **computation paths** from s_0 :

$s_0 \rightarrow s_1 \rightarrow s_1 \dots$

$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_0 \rightarrow s_1 \rightarrow s_2 \dots$

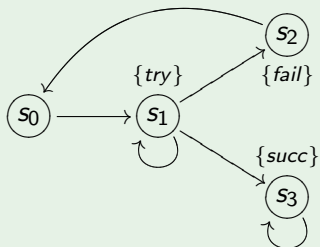
$s_0 \rightarrow s_1 \rightarrow s_3 \rightarrow s_3 \dots$

- to each state, one associates a **set** of computation paths

The Logic CTL*

- **path** formulas: $\varphi ::= \phi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi \mathbf{U}\varphi$
- **state** formulas: $\phi ::= \text{tt} \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \mathbf{E}\varphi \mid \mathbf{A}\varphi$
 - **E** and **A** similar to \diamond and \square modalities ...

Example

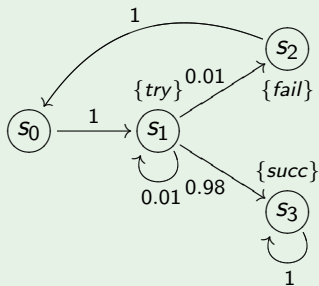


A F (*try***U***succ*)

Probabilistic Transition Systems

- $\mathcal{D} : \text{Set} \rightarrow \text{Set}$, $\mathcal{D}S =$ set of probability distributions over S :
 \mathcal{D} -coalgebras are probabilistic transition systems.

Example



Some computation paths from s_0 :

$s_0 \rightarrow s_1 \rightarrow s_1 \dots$

$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_0 \rightarrow s_1 \rightarrow s_2 \dots$

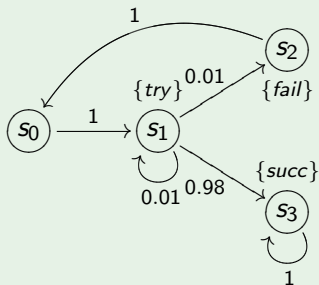
$s_0 \rightarrow s_1 \rightarrow s_3 \rightarrow s_3 \dots$

- to each state, one associates a **probability measure** on the computation paths from that state

The Logic PCTL

- **path** formulas: $\varphi ::= \mathbf{X}\phi \mid \phi \mathbf{U}^{\leq t} \phi$ $t \in \{0, 1, \dots\} \cup \{\infty\}$
- **state** formulas: $\phi ::= \mathbf{tt} \mid p \mid \neg\phi \mid \phi \wedge \phi \mid [\varphi]_{\geq q} \mid [\varphi]_{> q}$

Example



$$[\mathbf{tt} \mathbf{U}^{\leq 3} \mathit{fail}]_{< 0.1}$$

$$[(\mathit{try} \mathbf{U} \mathit{succ})]_{\geq 1}$$

More Examples

- $\mathcal{P}^+ \circ (A \times \text{Id})$ -coalgebras are labelled transition systems (LTSs).
- $\mathcal{D} \circ (A \times \text{Id})$ -coalgebras are generative probabilistic transition systems (GPTSs).

For *both* LTSs and GPTSs, computation paths have the form

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \dots$$

whereas infinite computation traces have the form

$$a_0 a_1 a_2 \dots$$

What LTSs and GPTSs have in common is the *inner* part of the signature functor: $A \times \text{Id}$.

The General Setting

Similarly to [Hasuo et. al.], we focus on $T \circ F$ -coalgebras, where:

- $T : \mathcal{C} \rightarrow \mathcal{C}$ is a *strong monad* describing the **computation type**
e.g. \mathcal{P}^+ , \mathcal{D}
- $F : \mathcal{C} \rightarrow \mathcal{C}$ is a functor describing the **transition type**, whose final sequence stabilises at ω
e.g. Id , $A \times \text{Id}$, $1 + A \times \text{Id}$
- **distributive law** $\lambda : F \circ T \Rightarrow T \circ F$ (compatible with monad structure) is fixed

Towards Infinite Traces

- the *possible* infinite traces for $T \circ F$ -coalgebras are given by the *final* F -coalgebra Z , obtained as a limit of:

$$\begin{array}{c}
 \xleftarrow{} \xleftarrow{} \xleftarrow{} \dots \xrightarrow{} \\
 1 \xleftarrow{!} F1 \xleftarrow{F!} F^2 1 \xleftarrow{F^2!} \dots \xrightarrow{} Z
 \end{array}$$

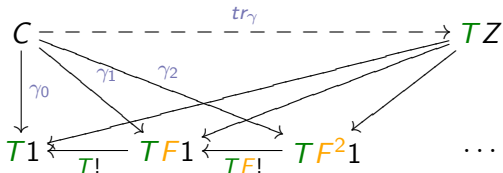
The diagram illustrates a sequence of objects $1, F1, F^2 1, \dots$ connected by solid arrows pointing left, labeled with $!, F!, F^2!, \dots$. Dashed arrows also point left from $F1, F^2 1, \dots$ to 1 . A dashed arrow points from 1 to Z . A series of dashed arrows points from the sequence $F1, F^2 1, \dots$ towards Z , indicating that Z is the limit of this sequence.

- the *actual* infinite traces should be *structured* according to the computation type
 - for $T = \mathcal{P}^+$, we expect *sets* of infinite traces
- in general, we expect an *infinite trace map*

$$tr_\gamma : C \rightarrow TZ$$

for each $T \circ F$ -coalgebra $\gamma : C \rightarrow TFC$.

Defining the γ_i s



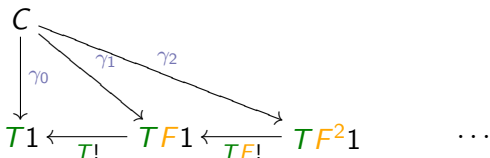
$$\gamma_0: \quad C \xrightarrow{!c} 1 \xrightarrow{\eta_1} T1$$

$$\gamma_{i+1}: \quad C \xrightarrow{\gamma} TFC \xrightarrow{TF\gamma_i} TFFT^i_1 \xrightarrow{T^{\lambda_{Fi_1}}} T^2F^{i+1}_1 \xrightarrow{\mu_{Fi+1}_1} TF^{i+1}_1$$

Under certain conditions on T (satisfied by $T = \mathcal{P}^+$ and $T = \mathcal{D}$, but **not** by $T = \mathcal{P}$), the γ_i s define a cone.

Note: this is also a cone in $Kl(T)$.

A Problem ... and its Solution



- in general, T does not preserve the limit of the final sequence of F
 - not even when $T = \mathcal{P}^+$
- only require T to *weakly* preserve the limit of the final sequence of F
 - need a dcpo \sqsubseteq on $C \rightarrow TZ$ and the assumption that the mediating maps form a directed set to guarantee existence of a *canonical* trace map

The Infinite Trace Map

Definition

For a $T \circ F$ -coalgebra (X, γ) , the infinite trace map

$$tr_\gamma : C \rightarrow TZ$$

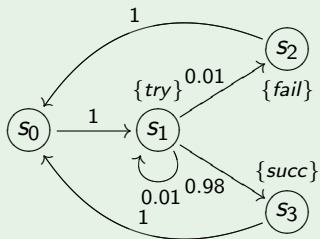
is the **largest** s.t. $C \overset{tr_\gamma}{\dashrightarrow} TZ$ for $i \in \omega$

```
graph TD; C -- tr_gamma [dashed] --> TZ; C -- gamma_i --> TF_i_1[TF^i 1]; TZ -- T pi_i --> TF_i_1;
```

The above applies to $T = \mathcal{P}^+$.

The Case of Probabilistic Systems

Example



- the γ_i s define a cone
 - working with $T = \mathcal{D}$ on Set does not work:
 - probability **measures** needed to deal with *uncountably many* traces
- \Rightarrow need to work with $T = \mathcal{G}$ on Meas (the **Giry monad**)

The Case of Probabilistic Systems (Cont'd)

- 1 start with a $\mathcal{D} \circ F$ -coalgebra γ over Set
- 2 lift $F : \text{Set} \rightarrow \text{Set}$ to $\tilde{F} : \text{Meas} \rightarrow \text{Meas}$ (works for *certain polynomial* F s)
- 3 obtain a $\mathcal{G} \circ \tilde{F}$ -coalgebra $\tilde{\gamma}$ over Meas , to which the definition can be applied:
 - we obtain a cone (for any F as above)
 - $\mathcal{G} : \text{Meas} \rightarrow \text{Meas}$ preserves the required limit
 - no dcpo structure needed ...

Relation to the Kleisli Category of T

- $\lambda : F \circ T \Rightarrow T \circ F$ yields lifting $\bar{F} : \text{KI}(T) \rightarrow \text{KI}(T)$ of $F : C \rightarrow C$.

Theorem

The trace map $tr_\gamma : C \rightarrow TZ$ defines an op-lax \bar{F} -coalgebra morphism in $\text{KI}(T)$:

$$\begin{array}{ccc} C & \xrightarrow{tr_\gamma} & Z \\ \gamma \downarrow & \sqsubseteq & \downarrow \eta_{FZ} \circ \zeta \\ \bar{F}C & \xrightarrow{\bar{F}tr_\gamma} & \bar{F}Z \end{array}$$

If T preserves the limit of the final sequence of F , then above is an \bar{F} -coalgebra morphism.

From Infinite Traces to Infinite Paths

Fix a $T \circ F$ -coalgebra $\gamma : C \rightarrow TFC$.

- replace $F : C \rightarrow C$ by $F_C : C \rightarrow C$

$$F_C X = C \times FX$$

\implies *possible* infinite executions: elements of final F_C -coalgebra Z_C

- use distributive law:

$$F_C T \xrightarrow{id_C \times \lambda} C \times TF \xrightarrow{st_{C,F_-}} TF_C$$

- work with the $T \circ F_C$ -coalgebra:

$$C \xrightarrow{\langle id_C, \gamma \rangle} C \times TFC \xrightarrow{st_{C,FC}} TF_C C$$

\implies infinite execution map:

$$C \xrightarrow{exec_\gamma} TZ_C$$

Path-Based Temporal Logics

Goal: define such logics for $T \circ F$ -coalgebras.

- syntax to distinguish between **path** and **state** formulas
- **fixpoint** versus **more restricted** path operators, depending on the *admissible predicates*:
 - complete lattice (e.g. in Set) \implies can consider **fixpoint operators**
 - σ -algebra (e.g. in Meas) \implies can only consider **certain** temporal operators

Fixpoint Logics: the Syntax

Fix

- a base category \mathcal{C} with $U : \mathcal{C} \rightarrow \text{Set}$
- a functor $P : \mathcal{C} \rightarrow \text{Set}^{\text{op}}$ specifying admissible predicates
 - assume $PC \subseteq \mathcal{P}UC$ is a complete lattice

Definition

$$\varphi ::= \text{tt} \mid \text{ff} \mid p^F \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [\lambda_F]\varphi \mid \eta p^F.\varphi$$

$$\phi ::= \text{tt} \mid p \mid \phi \wedge \phi \mid \phi \vee \phi \mid [\lambda]\varphi$$

where:

- $\lambda_C : P \Rightarrow PT$ monotone,
- $\lambda : P \Rightarrow PF$ monotone.

Fixpoint Logics: the Semantics

Fix

- a coalgebra $\gamma : C \rightarrow TFC$ with execution map $\text{exec}_\gamma : C \rightarrow TZ_C$
 - where $\zeta_C : Z_C \rightarrow C \times FZ_C$ is a final $C \times F$ -coalgebra
- a two-sorted valuation $V : (\mathcal{V}_F, \mathcal{V}) \rightarrow (PZ_C, PC)$

Interpret:

- path formulas as sets of paths: $\llbracket \varphi \rrbracket \in PZ_C$
- state formulas as sets of states: $\llbracket \phi \rrbracket \in PC$

Fixpoint Logics: the Semantics (Cont'd)

Some insights:

- to go from $\langle\!\langle\phi\rangle\!\rangle \in PZ_C$ to $\llbracket\langle\lambda\rangle\phi\rrbracket \in PC$ use

$$PZ_C \xrightarrow{(\lambda)_{Z_C}} PTZ_C \xrightarrow{P_{exec_\gamma}} PC$$

- to go from $\llbracket\phi\rrbracket \in PC$ to $\langle\!\langle\phi\rangle\!\rangle \in PZ_C$ use

$$Z_C \xrightarrow{\zeta_C} C \times FZ_C \xrightarrow{\pi_1} C$$

Fixpoint Logics: the Semantics (Cont'd)

Definition

$$\llbracket p^F \rrbracket = V(p^F)$$

$$\llbracket \phi \rrbracket = P(\pi_1 \circ \zeta_C)(\llbracket \phi \rrbracket)$$

$$\llbracket [\lambda_F] \varphi \rrbracket = P(\pi_2 \circ \zeta_C) \circ (\lambda_F)_{Z_C}(\llbracket \varphi \rrbracket)$$

$$\llbracket \mu p^F . \varphi \rrbracket = \text{Ifp}(M_\varphi)$$

$$\llbracket p \rrbracket = V(p)$$

$$\llbracket [\lambda] \varphi \rrbracket = (P_{\text{exec}_\gamma} \circ \lambda_{Z_C})(\llbracket \varphi \rrbracket)$$

where $M_\varphi : PZ_C \rightarrow PZ_C$ is the monotone map given by

$$M_\varphi(Y) = \llbracket \varphi \rrbracket_{p^F \mapsto Y}$$

Recovering (Negation-Free) CTL*

$$T = \mathcal{P}^+, \quad F = \text{Id}$$

$$\Lambda = \{\Box, \Diamond\}, \quad \Lambda_F = \{\circ\}$$

$$\Rightarrow \varphi ::= \text{tt} \mid \text{ff} \mid p^F \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \circ\varphi \mid \eta p^F.\varphi$$

$$\phi ::= \text{tt} \mid p \mid \phi \wedge \phi \mid \phi \vee \phi \mid \Box\varphi \mid \Diamond\varphi$$

Define:

- $\varphi \mathbf{U} \psi ::= \mu X.(\psi \vee (\varphi \wedge \circ X))$
- $\mathbf{G}\varphi ::= \nu X.(\varphi \wedge \circ X)$
- $\mathbf{A}\varphi ::= \Box\varphi$

...

How About LTSs?

$$T = \mathcal{P}^+, \quad F = A \times \text{Id}$$

$$\Lambda = \{\Box, \Diamond\}, \quad \Lambda_F = \{a \mid a \in A\} \cup \{\circ\}$$

$$\Rightarrow \varphi ::= \text{tt} \mid \text{ff} \mid p^F \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid a \mid \circ\varphi \mid \eta p^F.\varphi$$

$$\phi ::= \text{tt} \mid p \mid \phi \wedge \phi \mid \phi \vee \phi \mid \Box\varphi \mid \Diamond\varphi$$

- more natural encoding of “a occurs along every path” as

$$\Box \mu X. (a \vee \circ X)$$

- compare above to

$$\mu X. (\langle - \rangle \text{tt} \wedge [-a]X)$$

Logics with (Existential) Until Operators

- assume $PC \subseteq \mathcal{PUC}$ is a σ -algebra
- can drop requirement on monotonicity of λ , λ_F and add negation to both path and state formulas
- replace fixpoint operators with Until operators \mathbf{U}_L
 - $L \subseteq \Lambda_F$ finite set of (disjunction-preserving) predicate liftings
- semantics defined by

$$\llbracket \varphi \mathbf{U}_L \psi \rrbracket = \bigcup_{i \in \omega} \llbracket \varphi \mathbf{U}_L^{\leq i} \psi \rrbracket$$

where

$$\varphi \mathbf{U}_L^{\leq 0} \psi ::= \psi$$

$$\varphi \mathbf{U}_L^{\leq i+1} \psi ::= \psi \vee (\varphi \wedge \bigvee_{\lambda_F \in L} [\lambda_F](\varphi \mathbf{U}_L^{\leq i} \psi))$$

Recovering PCTL as a Fragment

$$T = \mathcal{D}, \quad F = \text{Id}$$

$$\Lambda = \{L_q\}, \quad \Lambda_F = \{\circ\}$$

$$\implies \varphi ::= \text{tt} \mid \text{ff} \mid \phi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \circ\varphi \mid \varphi \mathbf{U} \circ\varphi$$

$$\phi ::= \text{tt} \mid p \mid \neg\phi \mid \phi \wedge \phi \mid L_q\varphi$$

Define:

- $\mathbf{X}\varphi ::= \circ\varphi$
- $\varphi \mathbf{U} \psi ::= \varphi \mathbf{U} \circ\psi$
- $[\varphi]_{\geq q} ::= L_q\varphi$

Future Work

- other computational monads
 - e.g. the *finite multiset* monad and *graded temporal logics*?
- investigate *linear fragments* of path-based temporal logics
 - automata-based model-checking techniques (parameterised by computation type)