Generic Infinite Traces and 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

Goals:

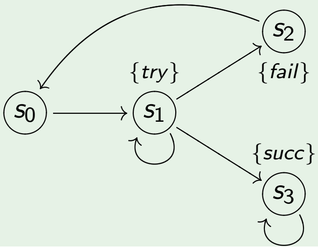
- find a unifying pattern (need infinite computation paths)
  - existing general theory of *finite* traces [Hasuo et. al.]
  - existing definition of *infinite* traces for $T = \mathcal{P}$ [Jacobs '04]
- automatically derive new path-based temporal logics

Restricted Transition Systems

- restricted transition systems are $\mathcal{P}^+$-coalgebras

  ($\mathcal{P}^+(S)$ = set of *non-empty* subsets of S)

Example



Some computation paths from s_0:

$s_0 \longrightarrow s_1 \longrightarrow s_1 \ldots$

$s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_0 \longrightarrow s_1 \longrightarrow s_2 \ldots$

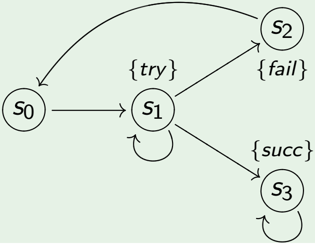
$s_0 \longrightarrow s_1 \longrightarrow s_3 \longrightarrow s_3 \ldots$

- 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: $\phi ::= \mathrm{tt} \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \mathbf{E}\varphi \mid \mathbf{A}\varphi$
  - **E** and **A** similar to $\Diamond$ and $\Box$ modalities ...

Example



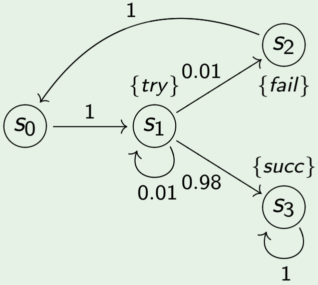
$\mathbf{A}\ \mathbf{F}\ (try\,\mathbf{U}\,succ)$

Probabilistic Transition Systems

- probabilistic transition systems are $\mathcal{D}$-coalgebras

  ($\mathcal{D}(S)$ = set of probability distributions over S)

Example



Some computation paths from s_0:

$s_0 \longrightarrow s_1 \longrightarrow s_1 \ldots$

$s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_0 \longrightarrow s_1 \longrightarrow s_2 \ldots$

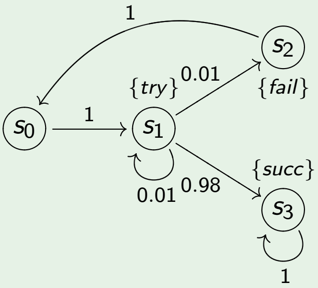
$s_0 \longrightarrow s_1 \longrightarrow s_3 \longrightarrow s_3 \ldots$

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

The Logic PCTL

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

Example



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

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

More Examples

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

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

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

whereas infinite computation traces have the form

$$a_0 \, a_1 \, a_2 \ldots$$

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

The General Setting

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

- *strong monad* $T : \mathsf{C} \to \mathsf{C}$ describes the computation type

  e.g. $\mathcal{P}^+$, $\mathcal{D}$

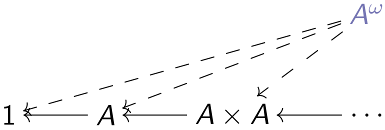
- functor $F : \mathsf{C} \to \mathsf{C}$ describes the transition type
  - require final sequence of F to stabilise at ω

  e.g. Id, $A \times \mathsf{Id}$, $1 + A \times \mathsf{Id}$

- distributive law $\lambda : F \circ T \Rightarrow T \circ F$ (compatible with monad structure) is fixed

Towards Infinite Traces

- the possible infinite traces for both LTSs and GPTSs are elements of A^ω (the *final* $A \times \_$-coalgebra):

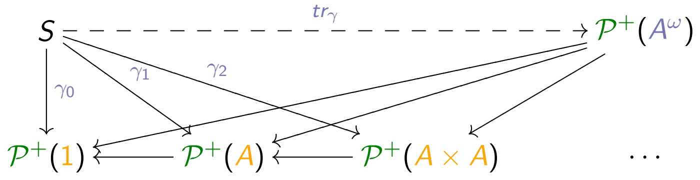


- for an LTS/GPTS (S, γ), the actual infinite traces should be *structured* according to the computation type:

$$tr_\gamma : S \to \mathcal{P}^+(A^\omega) \quad \text{or} \quad tr_\gamma : S \to \mathcal{D}(A^\omega)$$

Defining the Infinite Trace Map (for LTSs)

Fix an LTS $\gamma : S \to \mathcal{P}^+(A \times S)$.

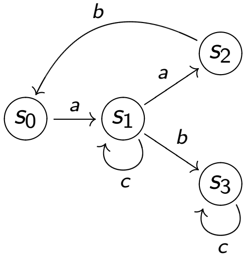


Define $tr_\gamma : S \to \mathcal{P}^+(A^\omega)$ from its finite approximants γ_i.

For existence of tr_γ, we need:

- γ_i's define cone
- $\mathcal{P}^+(A^\omega)$ weakly limiting

Defining the Approximants (for LTSs)



$$\gamma : S \to \mathcal{P}^+(S)$$

$$\begin{aligned} \gamma(s_0) &= \{(a, s_1)\} \\ \gamma(s_1) &= \{(a, s_2), (b, s_3), (c, s_1)\} \\ \gamma(s_2) &= \{(b, s_0)\} \\ \gamma(s_3) &= \{(c, s_3)\} \end{aligned}$$

- one application of γ gives

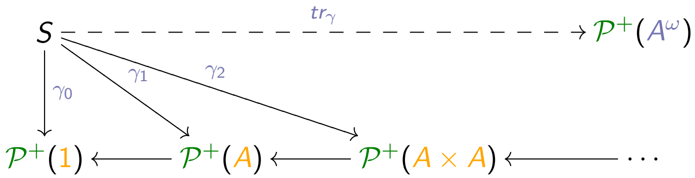
$$\gamma_1(s_1) = \{a, b, c\}$$

- two applications of γ followed by some "flattenning" (use of distributive law) give

$$\gamma_2(s_1) = \{ab, bc, ca, cb, cc\}$$

- ...

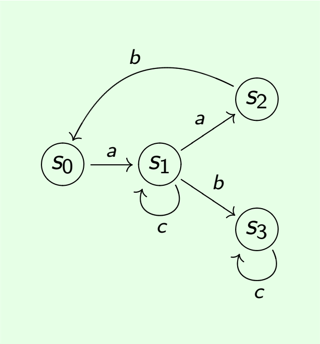
A Problem . . . and its Solution

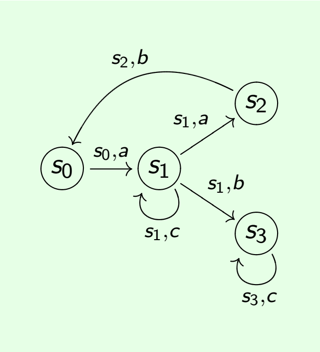


- in general, there are several choices for the infinite trace map . . .
- . . . but there is a canonical (*maximal*) one, assuming:
  - dcpo $\sqsubseteq$ on $S \to \mathcal{P}^+(Z)$
  - mediating maps form directed set
- the trace map can be defined for a general coalgebraic type $T \circ F$ (subject to reasonable constraints)

From Infinite Traces to Infinite Executions

- view $\mathcal{P}^+(A \times \_)$-coalgebra: as $\mathcal{P}^+(S \times A \times \_)$:





- obtain an infinite execution map $exec_\gamma : S \to (S \times A)^\omega$ as the infinite trace map of the new coalgebra !!

“Infinite” Executions: Examples

Take $T = \mathcal{P}^+$.

- $F = \_$ (restricted TSs):

$$s_0\ s_1\ s_2\ \ldots$$

- $F = A \times \_$ (restricted LTSs):

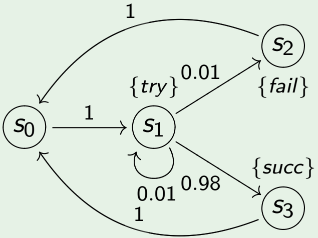
$$s_0\ a_1\ s_1\ a_2\ s_2\ \ldots$$

- $F = 1 + A \times \_$ (LTSs):

$s_0\ a_1\ s_1\ a_2\ s_2\ \ldots$ or $s_0\ a_1\ s_1\ \ldots\ s_n$

The Case of Probabilistic Systems

Example



- working with $T = \mathcal{D}$ over sets does not work:
  - probability measures needed to deal with *uncountably many* traces

$\Rightarrow$ need to work with $T = \mathcal{G}$ (the Giry monad) over measurable spaces

- resulting infinite trace map takes states to probability measures over infinite traces

Coalgebra Structure on Infinite Executions

Fix a $\mathcal{P}^+(A \times \_)$-coalgebra (S, γ).

The *possible* infinite executions have $S \times (A \times \_)$-coalgebra structure.

Hence, one can extract from each infinite execution

- the first state,
- an $A \times \_$-observation.

Towards Coalgebraic Path-Based Temporal Logics

- coalgebraic types come equipped with modal languages
- e.g. for $T = \mathcal{P}^+$, the language has modal operators $\Box$ and $\Diamond$:
  - $s \models \Box\phi$ iff $s' \models \phi$ for all s' s.t. $s \rightarrow s'$
  - $s \models \Diamond\phi$ iff $s' \models \phi$ for some s' s.t. $s \rightarrow s'$
- e.g. for $F = A \times \_$, the language has modal operators a and **X**:
  - $s \models a$ iff $s \rightarrow (a, s')$
  - $s \models \mathbf{X}\phi$ iff $s \rightarrow (a, s')$ and $s' \models \phi$
- our coalgebras have type $T \circ F$, so we make use of the above ...

  ... *but with a non-standard interpretation of* $\Box$ *and* $\Diamond$*!*

Path-Based Fixpoint Logics (for TSs)

$T = \mathcal{P}^+$ with monotone $\Box, \Diamond$

$F = \mathsf{Id}$ with monotone $\mathbf{X}$

$$\varphi \ ::= \ \mathsf{tt} \mid \mathsf{ff} \mid p^F \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \mu p^F.\varphi \mid \nu p^F.\varphi$$

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

Given $T \circ F$-coalgebra (S, γ) and suitable valuations (for p^F and p), interpret

- path formulas φ as sets of paths
  - use $S \times F$-coalgebra structure on S^ω to interpret ϕ and $\mathbf{X}\varphi$
- state formulas ϕ as sets of states
  - use infinite execution map $exec_\gamma : S \to \mathcal{P}^+(S^\omega)$ to interpret $\Box\varphi$, $\Diamond\varphi$

General Path-Based Fixpoint Logics

Fix

- base category C with $U : \mathsf{C} \to \mathsf{Set}$
- functor $P : C \to Set^{\mathrm{op}}$ specifying admissible predicates
  - assume $PC \subseteq \mathcal{P}UC$ is a complete lattice
- functors T and F with monotone modal operators Λ and Λ_F, resp.

Definition (Path-Based Fixpoint Language Syntax)

$$\varphi \quad ::= \quad \mathsf{tt} \mid \mathsf{ff} \mid p^F \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [\lambda_F]\varphi \mid \mu p^F.\varphi \mid \nu p^F.\varphi$$

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

- semantics as expected . . .

Recovering (negation-free) CTL*

Define:

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

  ...

- $\mathbf{A}\varphi ::= \Box\varphi$
- $\mathbf{E}\varphi ::= \Diamond\varphi$

How About LTSs?

$T = \mathcal{P}^+$ with modal operators $\Box, \Diamond$

$F = A \times \mathsf{Id}$ with modal operators a ($a \in A$), $\mathbf{X}$

$$\Longrightarrow \quad \varphi \;::=\; \mathsf{tt} \mid \mathsf{ff} \mid p^F \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid a \mid \mathbf{X}\varphi \mid \mu p^F.\varphi \mid \nu p^F.\varphi$$

$$\phi \;::=\; \mathsf{tt} \mid \mathsf{ff} \mid p \mid \phi \wedge \phi \mid \phi \vee \phi \mid \Box\varphi \mid \Diamond\varphi$$

- CTL* operators defined as before !
- can refer to the *next label along a path*:
  - natural encoding of "*a* occurs along every path" as

$$\Box\, F a \;::=\; \Box\, \mu X.(\, a \vee \mathbf{X} X\,)$$

  - compare above to

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

Logics with (Existential) Until Operators

- assume $PC \subseteq \mathcal{P}UC$ is a σ-algebra
- replace fixpoint operators with Until operators $\_\mathbf{U}_L\_$
  - $L \subseteq \Lambda_F$ finite set of (disjunction-preserving) predicate liftings
- semantics defined by

$$(\!|\varphi \mathbf{U}_L \psi|\!) = \bigcup_{i \in \omega} (\!|\varphi \mathbf{U}_L^{\leq i} \psi|\!)$$

where

$$\begin{aligned} \varphi \mathbf{U}_L^{\leq 0} \psi &::= \psi \\ \varphi \mathbf{U}_L^{\leq i+1} \psi &::= \psi \vee (\varphi \wedge \bigvee_{\lambda_F \in L} [\lambda_F](\varphi \mathbf{U}_L^{\leq i} \psi)) \end{aligned}$$

Recovering PCTL as a Fragment

$T = \mathcal{D}$, $F = \mathsf{Id}$

$\Lambda = \{L_q\}$, $\Lambda_F = \{\mathbf{X}\}$

$$\Longrightarrow \quad \varphi \; ::= \; \mathsf{tt} \mid \mathsf{ff} \mid \phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U}_{\mathbf{X}} \varphi$$

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

Define:

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

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)