From Branching to Linear Time, Coalgebraically

Corina Cîrstea
University of Southampton

January 8, 2014

The Big Picture

- need to model and verify heterogeneous systems
- requirements concerning correctness, but also resource usage, stochastic behaviour
  - e.g. *irrespective of the environment, the cost of a component achieving a given behaviour is bounded by a given value*
- existing formal verification techniques/tools assume fixed semantic model
  - lack of compositionality at the level of system models !

Summary of Formal Verification Logics

- multitude of temporal logics used in verification:
  - LTL, CTL, CTL*, μ-calculus on non-deterministic transition systems
  - PCTL, probabilistic LTL on probabilistic transition systems
  - ATL on game structures (for reasoning about player strategies)
  - graded CTL (for counting winning strategies in game structures)

  ...

- What are the similarities/differences between these logics?
  - e.g. branching versus linear time
- Are there general recipes for defining temporal logics and asociated verification technologies?
- Can we apply this recipe to new semantic models?
  - e.g. to combinations of the above?

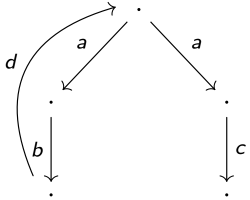
This Talk

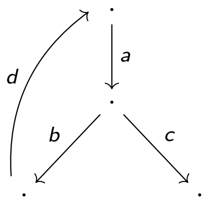
1. What is the linear time behaviour of a state in a system with branching?
    - several different types of branching: non-deterministic, probabilistic, weighted
    - several different types of linear behaviour, e.g. input/output transitions, termination
    - stepping stone to verifying linear time properties
2. What are linear time logics?
    - LTL
    - probabilistic LTL
    - weighted LTL
    - general recipe !

Our Approach in a Nutshell

- coalgebras as semantic models
  - subsumes non-deterministic, probabilistic and weighted models
  - *generic*, *uniform* and *compositional* approach:
    - monads capture branching behaviour
    - polynomial endofunctors capture linear behaviour
- branching monad determines choice of truth values
  - linear time behaviour measures the extent to which a particular trace is exhibited
  - linear time formulas measure the extent to which a linear time property holds

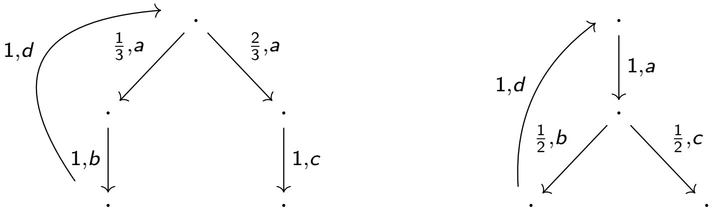
Example: Labelled Transition Systems





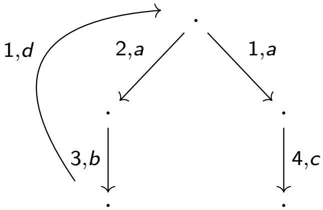
- branching given by non-determinism in choice of transition
- traces given by finite or infinite sequences of labels
- linear time behaviour of a state given by set of maximal traces

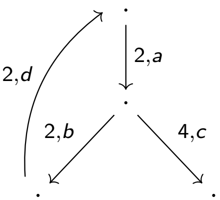
Example: Probabilistic Transition Systems



- branching given by probability distribution over possible transitions
- traces as before !
- linear time behaviour of a state: each maximal trace is a assigned a probability value

Example: Weighted Transition Systems





- branching given by weighted choices over possible transitions
- traces still as before !
- linear time behaviour of a state: if weights measure costs, the minimal cost of exhibiting a maximal trace is of interest !

Coalgebras

For F : Set $\to$ Set, an F-coalgebra is a function $\gamma : S \to F(S)$, where

- S is the state space
- γ defines the one-step behaviour (the transitions)

Examples:

- labelled transition systems (labels as outputs):

$$\gamma : S \to \mathcal{P}(1 + A \times S)$$

$s \mapsto \emptyset$ models deadlock

$s \mapsto * \in 1$ models successful termination

$s \mapsto (a, s')$ models an a-transition

- labelled transition systems (labels as inputs):

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

Coalgebras (Cont'd)

More examples:

- probabilistic transition systems (labels as outputs):

$$\gamma : S \to \mathcal{D}(1 + A \times S)$$

where $\mathcal{D}(X)$ are the subprobability distributions over X

- weighted transition systems (labels as outputs):

$$\gamma : S \to \mathcal{W}(1 + A \times S)$$

where $\mathcal{W}(X) = (\mathbb{N}^\infty)^X$

- systems with input and output:

$$\gamma : S \to \mathcal{P}(1 + B \times S)^A$$

Linear versus Branching Time, Coalgebraically

- branching given by a monad T
    - powerset $\mathcal{P}$
    - subprobability distributions $\mathcal{D}$
    - weights from a semiring S: $\mathcal{W}(X) = S^X$
- transition structure given by a polynomial functor F
    - $1 + A \times \mathsf{Id}$ - deterministic transitions (labels as outputs) with explicit termination
    - Id^A - deterministic transitions (labels as inputs)
    - $(1 + B \times \mathsf{Id})^A$ - deterministic systems with input and output
- goal is to give a uniform, compositional account of linear time semantics in systems with branching
    - systems modelled as coalgebras of type $\mathsf{T} \circ F$ ...

      ...but also $G \circ \mathsf{T}$ and $F \circ \mathsf{T} \circ G \circ \mathsf{T} \circ \ldots$

Related Work

1. finite traces [Hasuo, Jacobs, Sokolova 2007]
   - applies to coalgebras of type $\mathsf{T} \circ F$
     - non-deterministic systems: $\mathcal{P}(1 + A \times \mathsf{Id})$
     - probabilistic systems: $\mathcal{D}(A \times \mathsf{Id})$

     ...
2. maximal (including infinite) traces [Cîrstea 2011]
   - applies to coalgebras of type $\mathsf{T} \circ F$
3. (finite) traces via determinisation [Jacobs, Silva, Sokolova 2012]
   - applies to coalgebras of type $G \circ \mathsf{T}$
     - non-deterministic automata: $\{0, 1\} \times \mathcal{P}^A$
     - Segala systems: $\mathcal{P}(A \times \mathcal{D})$

     ...

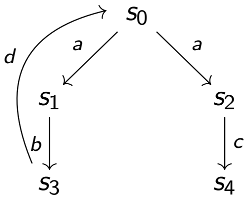
Limitations of Existing Approaches

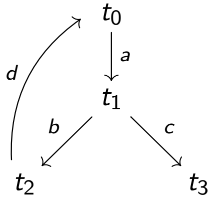
- lack of compositionality in the system type
  - e.g. systems with branching and both input and output not covered:

$$\mathsf{T}(1 + B \times \mathsf{Id})^A$$

- infinite traces only accounted for when models are $\mathsf{T} \circ F$-coalgebras

Bisimulation via Partition Refinement





1. assume $s_i \simeq_0 t_j$ for all i, j
2. for each $s_i \simeq_k t_j$, let

$$s_i \simeq_{k+1} t_j \quad \text{iff} \quad s_i \xrightarrow{l} s' \ \text{ implies } \ t_j \xrightarrow{l} t' \ \text{ and } s' \simeq_k t',$$

and conversely

3. largest bisimulation obtained as greatest fixpoint of monotone operator on lattice of relations

Can this be adapted to check if a state can exhibit a particular trace?

From Branching to Linear Time

Key insight: linear time behaviours of states in $\mathsf{T} \circ F$-coalgebras are states in (final) F-coalgebras!

$\mathcal{P}(1 + A \times \mathsf{Id})$-coalgebra

$$
\begin{tikzcd}
 & s_0 \arrow[dl, "a"'] \arrow[dr, "a"] & \\
s_1 \arrow[d, "b"] & & s_2 \arrow[d, "c"] \\
s_3 \arrow[uur, bend left, "d"] & & s_4
\end{tikzcd}
$$

$1 + A \times \mathsf{Id}$-coalgebra

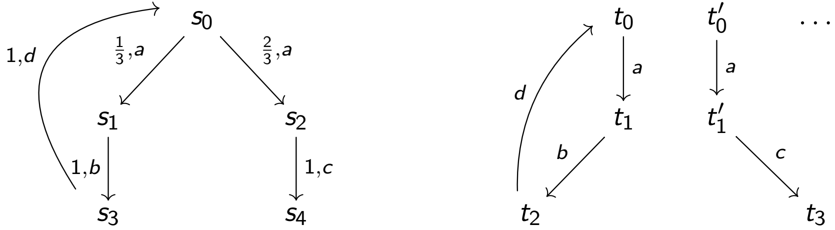
$$
\begin{tikzcd}
 & t_0 \arrow[d, "a"] & t_0' \arrow[d, "a"] & \cdots \\
 & t_1 \arrow[dl, "b"'] & t_1' \arrow[dr, "c"] & \\
t_2 \arrow[uur, bend left, "d"] & & & t_3
\end{tikzcd}
$$

1. assume $s_i \ni_0 t_j$ for all i, j
2. for each $s_i \ni_k t_j$, let

$$s_i \ni_{k+1} t_j \quad \text{iff} \quad t_j \xrightarrow{l} t' \ \text{ implies } \ s_i \xrightarrow{l} s' \ \text{ and } s' \ni_k t'$$

3. relation $\ni$ ("has trace") again obtained as greatest fixpoint !

What Needs To Be Generalised? (I)



1. need to measure the probability of a trace occurring from a state
   $\Longrightarrow$ relations given by maps $S \times T \to [0, 1]$
2. need to measure the ability to exhibit a trace across all branches
   $\Longrightarrow$ (partial) addition operation on $[0, 1]$
3. need to propagate measure along successive transitions
   $\Longrightarrow$ multiplication operation on $[0, 1]$

From Monads to (Ordered) Semirings

Theorem (extends Coumans&Jacobs 2013)

Each commutative, partially additive monad $\mathsf{T} : \mathsf{C} \to \mathsf{C}$ *induces a partial commutative semiring* $(\mathsf{T}(1), +, 0, \times, 1)$ *with an induced preorder* $\sqsubseteq$.

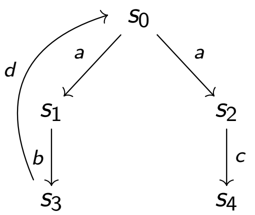
Examples:

- $\mathsf{T} = \mathcal{P}$: $(\{0,1\}, \vee, 0, \wedge, 1, \leq)$, $\top = 1$, $\bot = 0$
- $\mathsf{T} = \mathcal{D}$: $([0,1], +, 0, *, 1, \leq)$, $\top = 1$, $\bot = 0$
- $\mathsf{T} = \mathcal{W}$: $(\mathbb{N}^\infty, \min, \infty, +, 0, \geq)$, $\top = 0$, $\bot = \infty$

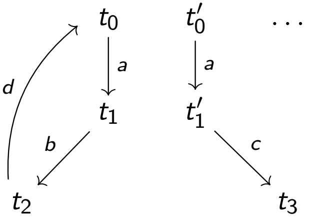
We take relations to be given by functions $R : X \times Y \to \mathsf{T}(1)$

What Needs To Be Generalised? (II)

$\mathcal{P}(1 + A \times \mathsf{Id})$-coalgebra (C, γ)



$1 + A \times \mathsf{Id}$-coalgebra (Z, ζ)



Recall: relation "has trace" obtained as greatest fixpoint of

$$\mathsf{Rel}_{C,Z} \xrightarrow{\mathsf{Rel}(F)} \mathsf{Rel}_{FC,FZ} \xrightarrow{\mathsf{E}_\mathsf{T}} \mathsf{Rel}_{\mathsf{T}(FC),FZ} \xrightarrow{(\gamma \times \zeta)^*} \mathsf{Rel}_{C,Z}$$

where $F = 1 + A \times \mathsf{Id}$ and $T = \mathcal{P}$.

We need generalisations of $\mathsf{Rel}(F)$ and E_T !

Generalised Relation Lifting

- category Rel defined using preorder $\sqsubseteq$ induced by partial semiring S:

$$\begin{tikzcd} X \times Y \arrow[r, "f \times g"] \arrow[d, "R"'] \arrow[dr, phantom, "\sqsubseteq"] & X' \times Y' \arrow[d, "R'"] \\ S \arrow[r, equal] & S \end{tikzcd}$$

- relation lifting $\mathsf{Rel}(F)$ of *polynomial* functor $F : \mathsf{Set} \to \mathsf{Set}$:

$$\begin{tikzcd} \mathsf{Rel} \arrow[r, "\mathsf{Rel}(F)"] \arrow[d, "q"'] & \mathsf{Rel} \arrow[d, "q"] \\ \mathsf{Set} \times \mathsf{Set} \arrow[r, "F \times F"'] & \mathsf{Set} \times \mathsf{Set} \end{tikzcd}$$

defined by structural induction on F.

More on (Strong) Monads

Corollary (Kock 1969)

For $\mathsf{T} : \mathsf{C} \to \mathsf{C}$ *a strong monad, any map*

$$S \times T \longrightarrow \mathsf{T}(1)$$

extends uniquely to a 1-linear map

$$\mathsf{T}(S) \times T \longrightarrow \mathsf{T}(1)$$

This yields an extension lifting E_T:

$$\begin{tikzcd} \mathrm{Rel} \arrow[r, "\mathsf{E}_\mathsf{T}"] \arrow[d, "q"'] & \mathrm{Rel} \arrow[d, "q"] \\ \mathrm{Set} \times \mathrm{Set} \arrow[r, "\mathsf{T} \times \mathrm{Id}"'] & \mathrm{Set} \times \mathrm{Set} \end{tikzcd}$$

More on (Strong) Monads

Corollary (Kock 1969)

For $\mathsf{T} : \mathsf{C} \to \mathsf{C}$ a strong monad, any map

$$S \times T \longrightarrow \mathsf{T}(1)$$

extends uniquely to a 1-linear map

$$\mathsf{T}(S) \times T \longrightarrow \mathsf{T}(1)$$

This yields an extension lifting E_T:

$$\begin{tikzcd}
\mathsf{Rel} \arrow[r, "\mathsf{E}_\mathsf{T}"] \arrow[d, "q"'] & \mathsf{Rel} \arrow[d, "q"] \\
\mathsf{Set} \times \mathsf{Set} \arrow[r, "\mathsf{T} \times \mathsf{Id}"'] & \mathsf{Set} \times \mathsf{Set}
\end{tikzcd}$$

$$\begin{tikzcd}
 & s_0 \arrow[dl, "{\frac{1}{3}, a}"'] \arrow[dr, "{\frac{2}{3}, a}"] & \\
s_1 \arrow[d, "{1, b}"'] & & s_2 \arrow[d, "{1, b}"] \\
s_3 \arrow[uur, bend left=60, "{1, d}"] & & s_4
\end{tikzcd}
\qquad
\begin{tikzcd}
 & t_0 \arrow[d, "a"] & t'_0 \arrow[d, "a"] & \\
 & t_1 \arrow[dl, "b"'] & t'_1 \arrow[dr, "b"] & \\
t_2 \arrow[uur, bend left, "d"] & & & t_3
\end{tikzcd}$$

Two Kinds of Relation Lifting (Example)

Probabilistic transition system: $\gamma : S \to \mathcal{D}(1 + A \times S)$

Traces: $\delta : Z \to 1 + A \times Z$

1. lift relation $R_i : S \times Z \to [0,1]$ to relation

   $$R_i' : (1 + A \times S) \times (1 + A \times Z) \to [0,1]$$

2. extend relation $R_i' : (1 + A \times S) \times (1 + A \times Z) \to [0,1]$ to relation

   $$R_i'' : \mathcal{D}(1 + A \times S) \times (1 + A \times Z) \to [0,1]$$

3. use $\gamma \times \delta : S \times Z \to \mathcal{D}(1 + A \times S) \times (1 + A \times S)$ to get a relation

   $$R_{i+1} : S \times Z \to [0,1]$$

Linear Time Behaviour as a Fixpoint

Assume: the preorder $\sqsubseteq$ is a ω^{op}-*chain complete* partial order with 1 as $\top$.

Definition

The linear time behaviour of a state in a coalgebra with branching (say of type $G \circ \mathsf{T} \circ F$) is the greatest fixpoint of the operator $\mathcal{O}$ on $\mathsf{Rel}_{S,Z}$ given by

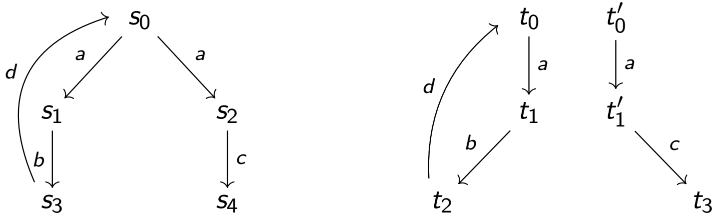
$$\mathsf{Rel}_{S,Z} \xrightarrow{\mathsf{Rel}(F)} \mathsf{Rel}_{FS,FZ} \xrightarrow{\mathsf{E_T}} \mathsf{Rel}_{\mathsf{T}FS,FZ} \xrightarrow{\mathsf{Rel}(G)} \mathsf{Rel}_{G\mathsf{T}FS,GFZ} \xrightarrow{(\gamma\times\delta)^*} \mathsf{Rel}_{S,Z}$$

where:

- $\gamma : S \to G\mathsf{T}FS$ is the system coalgebra
- $\delta : Z \to GFZ$ is the (final) coalgebra of traces

- approach is compositional in the coalgebra type
  - definition of domain of linear time behaviours
  - definition of operator $\mathcal{O}$

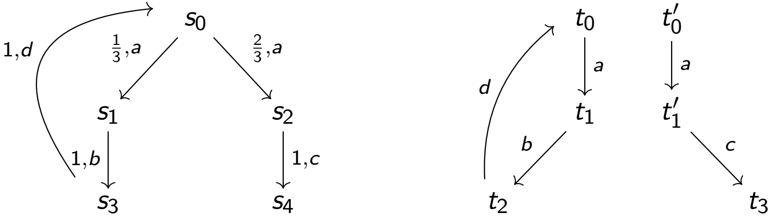
Example: Transition Systems



- $(s_i, t_j) \mapsto 1$
- $(s_3, t_2) \mapsto 1$, $(s_1, t_1) \mapsto 1$, $(s_2, t_1) \mapsto 0$, ...

  ...
- $(s_0, t_0) \mapsto 1$, $\qquad (s_0, t_0') \mapsto 1$

Example: Probabilistic Transition Systems

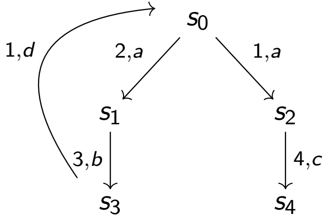


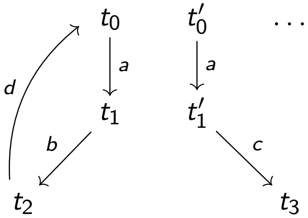
- $(s_i, t_j) \mapsto 1$
- $(s_3, t_2) \mapsto 1$, $(s_1, t_1) \mapsto 1$, $(s_2, t_1) \mapsto 0$, ...
- $(s_3, t_2) \mapsto 1$, $(s_1, t_1) \mapsto 1$, $(s_2, t_1) \mapsto 0$, $(s_0, t_0) \mapsto \frac{1}{3}$
- $(s_3, t_2) \mapsto \frac{1}{3}$, $(s_0, t_0) \mapsto \frac{1}{3}$

  ...
- $(s_0, t_0) \mapsto \frac{1}{9}$

  ...
- $(s_0, t_0) \mapsto 0$, $\quad (s_0, t'_0) \mapsto \frac{2}{3}$

Example: Weighted Transition Systems





- $(s_i, t_j) \mapsto 0$
- $(s_3, t_2) \mapsto 1$, $(s_1, t_1) \mapsto 3$, $(s_2, t_1) \mapsto \infty$, $(s_0, t_0) \mapsto 2$
- $(s_3, t_2) \mapsto 1$, $(s_1, t_1) \mapsto 3$, $(s_2, t_1) \mapsto \infty$, $(s_0, t_0) \mapsto 5$
- $(s_3, t_2) \mapsto 5$, $(s_0, t_0) \mapsto 5$

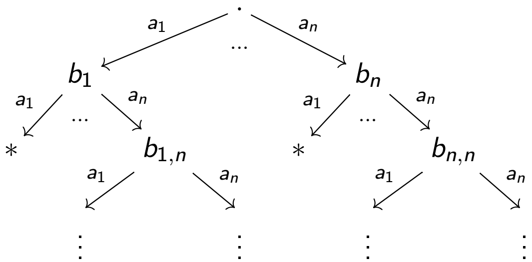
  ...
- $(s_0, t_0) \mapsto 10$

  ...
- $(s_0, t_0) \mapsto \infty$, $(s_0, t_0') \mapsto 5$, ...

Example: Systems with Input and Output

Consider coalgebra $\gamma : S \to \mathsf{T}(1 + B \times S)^A$.

Traces are trees (!) :



Linear time behaviour of a state is given by:

1. $\mathsf{T} = \mathcal{P}$: the set of such trees that can be matched
2. $\mathsf{T} = \mathcal{D}$: the probability of matching each such tree - probabilities of different tree branches are multiplied!
3. $\mathsf{T} = \mathcal{W}$: the minimum cost of matching each such tree - costs of different tree branches are added!

Towards Coalgebraic Linear Time Logics

- similar (double) extension lifting can be used to measure the extent to which two states in two coalgebras with branching can exhibit the same behaviour
  - $\mathsf{T} = \mathcal{P}$: existence of a common trace
  - $\mathsf{T} = \mathcal{D}$: probability of a common trace
  - $T = \mathcal{W}$: joint minimal cost of a common trace
- similar approach to temporal logics?
  - instead of individual traces, consider linear temporal logic formulas (sets of acceptable traces)

Generalised Predicate Liftings

- partial commutative semiring $S = (T1, +, 0, \bullet, 1)$ with induced order $\sqsubseteq$ as before
- category Pred defined similarly to Rel:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "P"'] \arrow[dr, phantom, "\sqsubseteq"] & Y \arrow[d, "Q"] \\ S \arrow[r, equal] & S \end{tikzcd}$$

- want to lift predicates over X to predicates over FX ...
- predicate lifting of arity n:

$$\begin{tikzcd} \mathrm{Pred}^n \arrow[r, "\mathsf{L}"] \arrow[d, "p"'] & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Set} \arrow[r, "F"'] & \mathrm{Set} \end{tikzcd}$$

- e.g. $F = 1 + A \times \mathrm{Id}$
  - unary modality $\langle a \rangle$ defined using $\mathsf{L}_a : \mathrm{Pred} \to \mathrm{Pred}$

$$\begin{array}{ccccc} X & 1 + A \times X & * & (a, x) & (a', x) \\ \downarrow P & \downarrow \mathsf{L}_{\langle a \rangle}(P) & \downarrow & \downarrow & \downarrow \\ S & S & 0 & P(x) & 0 \end{array}$$

Generalised Predicate Liftings: More Examples

$F = 1 + A \times \mathsf{Id}$, arbitrary T:

- binary modality $[a]$ defined using $\mathsf{L}_a : \mathsf{Pred} \to \mathsf{Pred}$

$$X \xrightarrow{P_1} S \qquad X \xrightarrow{P_2} S \qquad 1 + A \times X \xrightarrow{\mathsf{L}_{[a]}(P_1,P_2)} S \qquad * \mapsto 0 \qquad (a,x) \mapsto P_1(x) \qquad (a',x) \mapsto P_2(x)$$

- nullary modality $*$ defined using $\mathsf{L}_* : 1 \to \mathsf{Pred}$

$$1 + A \times X \xrightarrow{\mathsf{L}_*} \mathsf{T}1 \qquad * \mapsto 1 \qquad (a,x) \mapsto 0$$

- example formulas: $\langle a \rangle \top$, $[a](\top, *)$

Generalised Predicate Liftings

- *set* of predicate liftings Λ_F for polynomial functor $F = \coprod_{i \in I} \mathrm{Id}^{j_i}$:

$$(L_i)_X(P_1, \ldots, P_{j_i})(f) = \begin{cases} P_1(x_1) \bullet \ldots \bullet P_{j_i}(x_{j_i}) & \text{if } f = (x_1, \ldots, x_{j_i}) \in \iota_i(\mathrm{Id}^{j_i}) \\ 0 & \text{otherwise} \end{cases}$$

Extension Predicate Liftings

- predicates over X canonically induce predicates over subsets, subprobability distributions, or weighted subsets:
  - $\mathsf{T} = \mathcal{P}$: predicate is true on $Y \in \mathcal{P}(X)$ iff if it is true on some $x \in Y$
  - $\mathsf{T} = \mathcal{D}$: $P : X \to [0,1]$ extends to $P' : \mathcal{D}(X) \to [0,1]$

$$\mu : X \to [0,1] \quad \mapsto \quad \sum_{x \in X} \mu(x) * P(x)$$

  - $\mathsf{T} = \mathcal{W}$: $P : X \to \mathbb{N}^\infty$ extends to $P' : \mathcal{W}(X) \to [0,1]$

$$w : X \to \mathbb{N}^\infty \quad \mapsto \quad \min_{x \in X}(w(x) + P(x))$$

- extension lifting P_T:

$$\begin{tikzcd} \mathsf{Pred} \arrow[r, "\mathsf{P}_\mathsf{T}"] \arrow[d, "p"'] & \mathsf{Pred} \arrow[d, "p"] \\ \mathsf{Set} \arrow[r, "\mathsf{T}"'] & \mathsf{Set} \end{tikzcd} \quad \text{given by} \quad \begin{tikzcd} X \arrow[d, "P"'] & \mathsf{T}X \arrow[d, "\mathsf{T}(P)"] & \\ \mathsf{T}1 & \mathsf{T}^2 1 \arrow[r, "\mu_1"] & \mathsf{T}1 \end{tikzcd}$$

Linear Time Modal Logics: Syntax and Semantics

- modal logic $\mathcal{L}_\Lambda$
  - syntax:

$$\varphi ::= \top \mid [\lambda](\varphi_1, \ldots, \varphi_{\mathrm{ar}(\lambda)})$$

  - semantics w.r.t. coalgebra (C, γ): $[\![\varphi]\!]_\gamma : C \to \mathsf{T}1$

$$[\![\top]\!]_\gamma(c) = \top$$

$$[\![[\lambda](\varphi_1, \ldots, \varphi_{\mathrm{ar}(\lambda)})]\!]_\gamma = \gamma^* \circ \mathsf{P}_\mathsf{T}(\mathsf{P}_\lambda([\![\varphi_1]\!]_\gamma, \ldots, [\![\varphi_n]\!]_\gamma))$$

$$\mathsf{Pred}^n_C \xrightarrow{\mathsf{P}_\lambda} \mathsf{Pred}_{FC} \xrightarrow{\mathsf{P}_\mathsf{T}} \mathsf{Pred}_{\mathsf{T}FC} \xrightarrow{\gamma^*} \mathsf{Pred}_C$$

$$\mathsf{T}FC \xleftarrow{\gamma} C$$

Example: Labelled Transition Systems

$\mathsf{T} = \mathcal{P}$, $F = 1 + A \times \mathsf{Id}$

$$\begin{tikzcd}
 & s_0 \arrow[dl, "a"'] \arrow[dr, "a"] & \\
s_1 \arrow[d, "b"] & & s_2 \arrow[d, "c"] \\
s_3 \arrow[uur, bend left=60, "d"] & & s_4
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & t_0 \arrow[d, "a"] & \\
 & t_1 \arrow[dl, "b"'] \arrow[dr, "c"] & \\
t_2 \arrow[uur, bend left=40, "d"] & & t_3
\end{tikzcd}$$

- $s_1 \models \langle b \rangle \top \qquad t_1 \models \langle b \rangle \top$
- $s_0 \models \langle a \rangle \langle b \rangle \top \qquad t_0 \models \langle a \rangle \langle b \rangle \top$
- $s_0 \not\models \langle a \rangle \langle a \rangle \top \qquad s_0 \not\models *$
- $s_1 \not\models \langle a \rangle \top \qquad s_1 \models [a](\top, \langle d \rangle \top)$

Example: Probabilistic Transition Systems

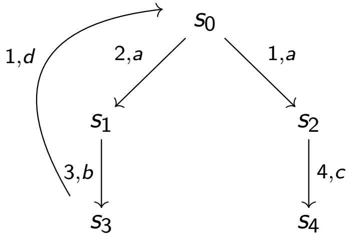
$\mathsf{T} = \mathcal{D}$, $F = 1 + A \times \mathsf{Id}$

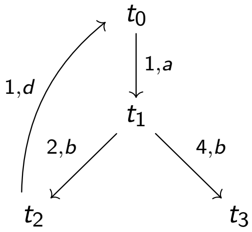
$$\begin{tikzcd}
 & s_0 \arrow[dl, "\frac{1}{3},a"'] \arrow[dr, "\frac{2}{3},a"] & \\
s_1 \arrow[d, "1,b"'] & & s_2 \arrow[d, "1,c"] \\
s_3 \arrow[uur, bend left, "1,d"] & & s_4
\end{tikzcd}
\qquad
\begin{tikzcd}
 & t_0 \arrow[d, "1,a"] & \\
 & t_1 \arrow[dl, "\frac{1}{2},b"'] \arrow[dr, "\frac{1}{2},c"] & \\
t_2 \arrow[uur, bend left, "1,d"] & & t_3
\end{tikzcd}$$

- $(s_1, \langle b \rangle \top) \mapsto 1 \qquad (t_1, \langle b \rangle \top) \mapsto \frac{1}{2}$
- $(s_0, \langle a \rangle \langle b \rangle \top) \mapsto \frac{1}{3} \qquad (t_0, \langle a \rangle \langle b \rangle \top) \mapsto \frac{1}{2}$
- $(s_1, \langle a \rangle \top) \mapsto 0 \qquad (s_0, \langle a \rangle \top) \mapsto 1$
- $(t_1, [b](\langle d \rangle \top, \top)) \mapsto 1$

Example: Weighted Transition Systems

$\mathsf{T} = \mathcal{W}$, $F = 1 + A \times \mathsf{Id}$





- $(s_1, \langle b\rangle\top) \mapsto 3$ $\qquad$ $(t_1, \langle b\rangle\top) \mapsto 2$
- $(s_0, \langle a\rangle\top) \mapsto 1$
- $(s_0, \langle a\rangle\langle b\rangle\top) \mapsto 5$ $\qquad$ $(t_0, \langle a\rangle\langle b\rangle\top) \mapsto 3$
- $(t_1, [b](\langle d\rangle\top, \top)) \mapsto 3$

Relational Semantics for Linear Time Modal Logics

- computing $[\![\varphi]\!]_\gamma : C \to \mathsf{T}1$ for all $\varphi \in \mathcal{L}_\Lambda$ same as computing "satisfaction relation"

$$R : C \times \mathcal{L}_\Lambda \to \mathsf{T}1$$

- computing the latter can be done iteratively:
  1. initially $(c, \varphi) \mapsto \top$ for all c and φ
  2. at each step, refine value for $(c, [\lambda]\varphi)$ by unfolding the coalgebra structure on c, and using previous values for (c', φ), with c' "reachable" from c in one step
- as each φ has finite depth, procedure stabilises after finite number of steps for each φ !

Relational Semantics for Linear Time Modal Logics

- $\mathsf{L}_\Lambda = \sum_{\lambda \in \Lambda} \mathsf{Id}^{\mathsf{ar}(\lambda)}$ captures the syntax of $\mathcal{L}_\Lambda$
- $\mathcal{L}_\Lambda$ is carrier of initial $\{\top\} + \mathsf{L}_\Lambda$-algebra ...

  ... and also of a $\{\top\} + \mathsf{L}_\Lambda$-coalgebra $\alpha^{-1} : \mathcal{L}_\Lambda \to \{\top\} + \mathsf{L}_\Lambda(\mathcal{L}_\Lambda)$!
- lifting $\mathsf{D} : \mathsf{Rel} \to \mathsf{Rel}$ of $F \times \mathsf{L}_\Lambda$:

$$\begin{tikzcd} \mathsf{Rel} \arrow[r, "\mathsf{D}"] \arrow[d] & \mathsf{Rel} \arrow[d] \\ \mathsf{Set} \times \mathsf{Set} \arrow[r, "F \times \mathsf{L}_\Lambda"'] & \mathsf{Set} \times \mathsf{Set} \end{tikzcd}$$

  defined using $(P_\lambda)_{\lambda \in \Lambda}$.

Relational Semantics for Linear Time Modal Logics

Theorem

The semantics of $\mathcal{L}_\Lambda$ is the unique fixpoint of the operator on $\mathsf{Rel}_{C,\mathcal{L}_\Lambda}$ given by

$$\mathsf{Rel}_{C,\mathcal{L}_\Lambda} \xrightarrow{\mathsf{D}} \mathsf{Rel}_{FC,\mathsf{L}_\Lambda\mathcal{L}_\Lambda} \xrightarrow{\mathsf{E}_\mathsf{T}} \mathsf{Rel}_{\mathsf{T}FC,\mathsf{L}_\Lambda\mathcal{L}_\Lambda} \xrightarrow{\mathsf{X}} \mathsf{Rel}_{\mathsf{T}FC,\{\top\}+\mathsf{L}_\Lambda\mathcal{L}_\Lambda} \xrightarrow{(\gamma\times\alpha^{-1})^*} \mathsf{Rel}_{C,\mathcal{L}_\Lambda}$$

Intuition:

- D - one linear step
- E_T - amalgamate across different branches
- X - incorporate $\top$
- $(\gamma \times \alpha^{-1})^*$ - unfold the coalgebra structures of states and formulas

Linear Time Fixpoint Logics

- modal logic $\mu\mathcal{L}_\Lambda$
  - syntax:
    $$\varphi ::= x \mid \top \mid [\lambda](\varphi_1, \ldots, \varphi_{\mathrm{ar}(\lambda)}) \mid \mu x.\varphi \mid \nu x.\varphi$$
  - semantics $[\![\varphi]\!]_\gamma^V$ w.r.t. coalgebra (C, γ) and valuation $V : \mathcal{V} \to \mathrm{Pred}_C$:
    - $[\![x]\!]_\gamma^V = V(x)$

      ...
    - $[\![\mu x.\varphi]\!]$ and $[\![\nu x.\varphi]\!]$ defined using least/greatest fixpoints of operator on Pred_C:
      $$P \longmapsto [\![\varphi]\!]_\gamma^{V[P/x]}$$

Example: Labelled Transition Systems

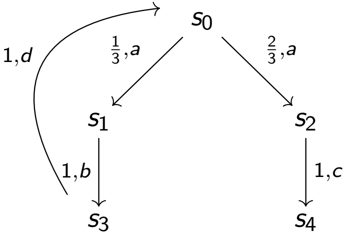
$\mathsf{T} = \mathcal{P}$, $F = 1 + A \times \mathsf{Id}$

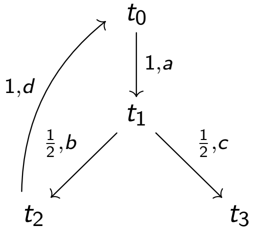
$$\begin{tikzcd}
 & s_0 \arrow[dl, "a"] \arrow[dr, "a"] & \\
 s_1 \arrow[d, "b"] & & s_2 \arrow[d, "c"] \\
 s_3 \arrow[uur, bend left, "d"] & & s_4
\end{tikzcd}$$

- $s_0 \not\models \nu x.\langle a \rangle x$
- $s_0 \models \mu x.[a](\top, x)$
- $s_0 \models \nu x.\mu y.[a](x, y)$

Example: Probabilistic Transition Systems

$\mathsf{T} = \mathcal{D}$, $F = 1 + A \times \mathsf{Id}$



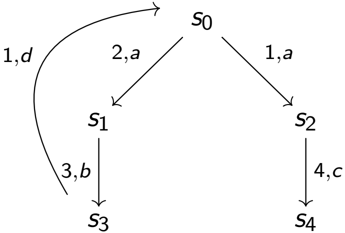


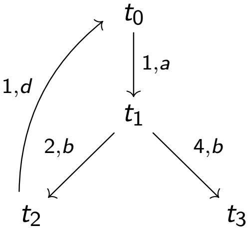
- $(s_0, \nu x.\langle a \rangle x) \mapsto 0$
- $(s_0, \mu x.[a](\top, x)) \mapsto 1$
- $(s_0, \nu x.\mu y.[a](x, y)) \mapsto 0$

$(t_0, \mu x.[b](\top, x)) \mapsto \frac{1}{2}$

Example: Weighted Transition Systems

$\mathsf{T} = \mathcal{W}$, $F = 1 + A \times \mathsf{Id}$





- $(s_0, \nu x.\langle a \rangle x) \mapsto \infty$ (because $(s_1, \langle a \rangle \nu x.\langle a \rangle x) \mapsto \infty$)
- $(s_0, \mu x.[a](\top, x)) \mapsto 1$ $(t_0, \mu x.[b](\top, x)) \mapsto 3$
- $(s_0, \nu x.\mu y.[a](x, y)) \mapsto \infty$

Relational Semantics for Linear Time Fixpoint Logics

- same iterative approach works for fixpoint formulas (assuming only one type of fixpoints)
- initially $(c, \varphi) \mapsto \bot$ ($(c, \varphi) \mapsto \top$) for lfp formulas (resp. gfp formulas)
- at each step, unfold the formula structure, and if needed also the coalgebra structure
  - to compute new approximation for $(s, [\lambda]\varphi)$, unfold γ on s and use previous values for (s', φ)
  - to compute new approximation for $(s, \mu x.\varphi)$, use value for $(s, \varphi[\mu x.\varphi/x])$
- sufficient to work with formulas in the closure of the formula of interest, as opposed to the entire fixpoint language !

Relational Semantics for Fixpoint Logics

Theorem

Let $\varphi \in \mathcal{L}_\Lambda$ be clean, guarded, containing no free variables and only least (greatest) fixpoint operators. Then $[\![\varphi]\!]_\gamma$ can be obtained from the least (greatest) fixpoint of the operator on $\mathsf{Rel}_{C,\mathcal{F}}$ given by

$$\mathsf{Rel}_{C,\mathcal{F}} \xrightarrow{\mathsf{F}} \mathsf{Rel}_{\mathsf{T}F C\times C, \mathsf{L}_\Lambda\mathcal{F}+\mathcal{F}} \xrightarrow{\mathsf{X}} \mathsf{Rel}_{\mathsf{T}F C\times C, \{\top\}+\mathsf{L}_\Lambda\mathcal{F}+\mathcal{F}} \xrightarrow{(\langle\gamma,\mathrm{id}_C\rangle\times\alpha)^*} \mathsf{Rel}_{C,\mathcal{F}}$$

where:

- $\mathcal{F} = \mathsf{Cl}(\varphi)$
- $\alpha : \mathcal{F} \to \{\top\} + \mathsf{L}_\Lambda\mathcal{F} + \mathcal{F}$ *is the formula coalgebra*

Intuition:

- F - one linear step and one branching step
- X - incorporate $\top$
- $(\langle\gamma, \mathrm{id}_C\rangle \times \alpha)^*$ - unfold the coalgebra structure of states and formulas

So How Can We Use This ?

- can check whether $\mu x.\varphi$ holds with probability at least p using iterative approach
  - stop (with Yes) as soon as value p reached
- can also check whether $\nu x.\varphi$ holds with probability at least p
  - stop (with No) as soon as value below p reached
- similarly for weighted systems (cost at most C)
  - for $\mu x.\varphi$, stop (with Yes) as soon as C reached (from above!)
  - for $\nu x.\varphi$, stop (with No) as soon as C reached (from below!)

Conclusions and Future Work

Summary

- general account of linear time behaviour in systems with branching
- general notion of linear time fixpoint logic for systems with branching
- both are parametric in the choice of branching and linear behaviours
- relational semantics supports approximation-based approach to model-checking

Ongoing/future work

- study expressiveness of linear time logics
- generalise linear time logics to coalgebras of type $F \circ \mathsf{T} \circ G$, ...
- coalgebraic model-checking of linear time properties
    - localised model checking
    - large state spaces still a challenge !
    - new algorithms for known semantic models ?
    - compositional approach: combine different types of branching !