Monads and Trace Semantics

Alexander Kurz[a] Stefan Milius[b] Dirk Pattinson[c] Lutz Schröder[b]

[a]University of Leicester

[b]Friedrich-Alexander-Universität Erlangen-Nürnberg

[c]Australian National University, Canberra

IFIP WG 1.3 Meeting, September 2014, Sinaia

Introduction

- Processes are considered modulo equivalences
  - Bisimilarity
  - Mutual similarity
  - 2-nested simulations
  - Completed traces
  - Traces
- Coalgebra on its own is suitable for generic bisimilarity:
  - Behavioural equivalence
  - Aczel-Mendler bisimilarity
  - Λ-bisimulations
  - Aczel-Mendler precongruences
- Trace equivalence needs algebra, i.e. monads

Coalgebra

Generic theory of transition systems:

- Type functor $G : \mathbf{C} \to \mathbf{C}$ (e.g. $G = \mathcal{P} : \mathbf{Set} \to \mathbf{Set}$)
- Systems are G-coalgebras $\gamma : C \to GC$ (e.g. binary relations)
- Morphisms $(C, \gamma) \to (D, \delta)$= maps $C \to D$ with

$$\begin{array}{ccc} C & \xrightarrow{f} & D \\ {\scriptstyle\gamma}\downarrow & & \downarrow{\scriptstyle\delta} \\ GC & \xrightarrow[Gf]{} & GD \end{array}$$

- Morphisms preserve behaviour, so
  - $x \in C$, $y \in D$ behaviourally equivalent if

    $f(x) = g(y)$ for some morphisms f, g

- (Z, ζ) final $\iff \forall C. \exists! f : (C, \gamma) \to (Z, \zeta), \quad f$ morphism.

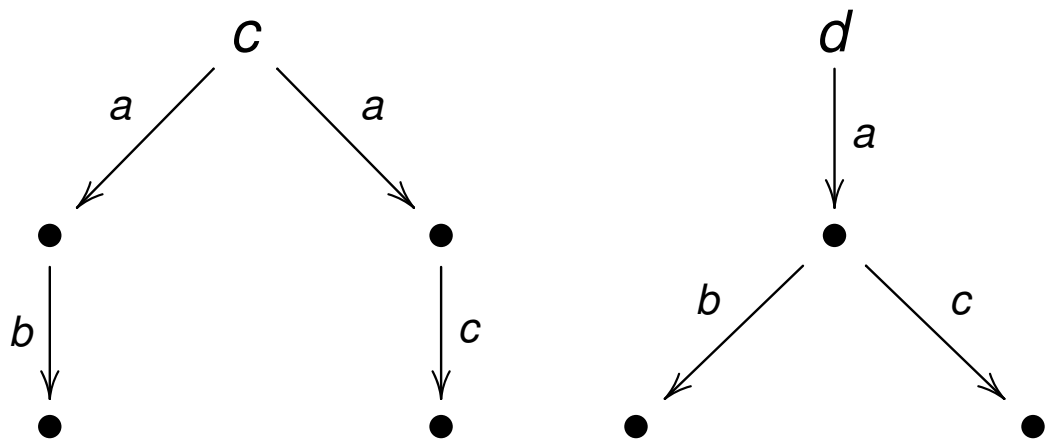
Example: LTS

$GX = \mathcal{P}(\Sigma \times X) \cong (\Sigma \to \mathcal{P}(X))$:

Coalgebras $\gamma : C \to GC$ are labelled transition systems (LTS):

- Fixed set Σ of actions
- Set C of states
- At each state c, sets $\gamma(c)(a)$ of a-successors.

Behavioural equivalence $=$ bisimilarity $\neq$ trace equivalence:



Coalgebraic Trace Semantics: Kleisli

Hasuo/Jacobs/Sokolova 2007:

- $G = TF$, monad T (branching type), functor F (transition type)
- Distributive law
$$\lambda : FT \to TF$$
$\cong$ lifting $\bar{F}$ of F to Kleisli category
- Additional conditions on cppo-enrichment

Then:
$$\nu\bar{F} \cong \mu F$$

$\to$ trace semantics $=$ final map $C \to T\mu F$

Trace Semantics of LTS, Kleisli Style

With explicit termination (NDA):

- $T = \mathcal{P}$
- $F = 1 + \Sigma \times \_$
- $\lambda(a, S) = \{a\} \times S$
- Trace semantics

$$C \to \mathcal{P}(\mu F) = \mathcal{P}(\Sigma^*) \qquad :) \quad (?)$$

(Language semantics)

Without explicit termination:

- $T = \mathcal{P}$
- $F = \Sigma \times \_$
- trace semantics

$$C \to \mathcal{P}(\mu F) = \mathcal{P}(\emptyset) = 1 \qquad :/$$

Coalgebraic Trace Semantics: Eilenberg/Moore

(Jacobs/Silva/Sokolova 2012)

- $G = FT$
- Distributive law

$$\rho : TF \to FT$$

  $\cong$ lifting $\hat{F}$ of F to EM category

- makes TC into a $\hat{F}$-coalgebra for $\gamma : C \to TFC$

Then

$$\nu F \cong U \nu \hat{F}$$

$\to$ trace semantics $= C \xrightarrow{\eta} TC \to \nu F$

Trace Semantics of LTS, Kleisli Style

With explicit termination (NDA):

- $T = \mathcal{P}$
- $F = 2 \times \_^{\Sigma}$
- $\rho(S) = ((\exists(\top, \_) \in S), \lambda a. \{f(a) \mid (\_, f) \in S\})$
- Trace semantics

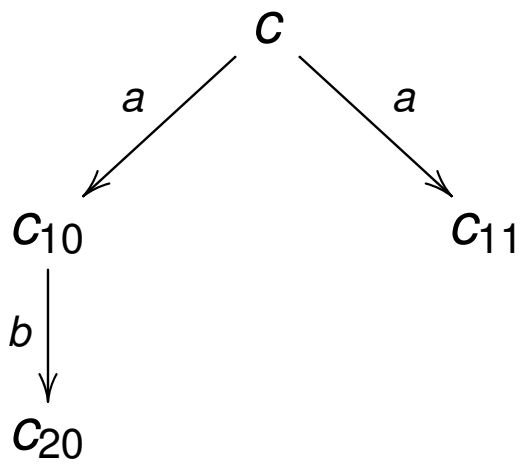
$$C \to \nu F = \mathcal{P}(\Sigma^*) \qquad :) \quad (?)$$

Without explicit termination:

- $T = \mathcal{P}$
- $F = \_^{\Sigma}$
- trace semantics

$$C \to \nu F = 1 \qquad :/$$

Trace Semantics, Incrementally



Pretrace = Pair (u, d), u word, d state.

	Pretraces	Traces
Stage 0 :	$\{(\varepsilon, c)\}$	$\{\varepsilon\}$
Stage 1 :	$\{(a, c_{10}), (a, c_{11})\}$	$\{a\}$
Stage 2 :	$\{(ab, c_{20}))\}$	$\{ab\}$
Stage 3 :	$\emptyset$	$\emptyset$

Monadic Traces

Trace semantics of G = natural transformation $\alpha : G \to M$, M monad

$\gamma^{(n)} : X \to MX$ = n-fold iterate of $\alpha\gamma : X \to MX$ in $\mathsf{Kl}(M)$.

α-trace sequence of $c \in C$:

$$T^{\alpha}_{\gamma}(c) = (\underbrace{M!\gamma^{(n)}(c)}_{\in M1})_{n<\omega}$$

c, d α-trace equivalent $\iff$ $T^{\alpha}_{\gamma}(c) = T^{\alpha}_{\gamma}(d)$

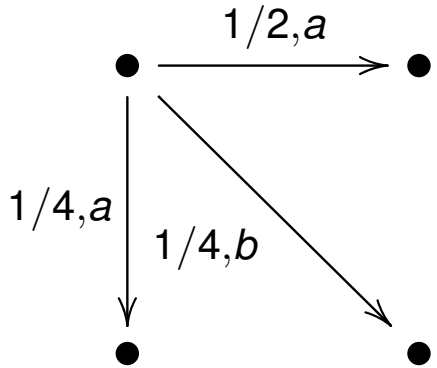
Example: LTS

- $MX = \mathcal{P}(\Sigma^* \times X)$
- $\alpha : \mathcal{P}(\Sigma \times X) \hookrightarrow MX$
- $\gamma^{(n)}(c) = \{(u, d) \in \Sigma^n \times C \mid c \xrightarrow{u} d\}$
- $M1 = \mathcal{P}(\Sigma^* \times 1) \cong \mathcal{P}(\Sigma^*)$
- $M!\gamma^{(n)}(c) = \{u \in \Sigma^n \mid c \xrightarrow{u} \ \}$

so α-trace equivalence is trace equivalence

Generative Probabilistic LTS

= G-coalgebras for $G = \mathcal{D}(\Sigma \times X)$



Traces (for finite Σ):

- σ-Algebra on Σ^ω generated by cones $v\uparrow = \{vw \mid w \in \Sigma^\omega\}$
- $\gamma : C \to \mathcal{D}(\Sigma \times C)$ induces distributions μ_c on Σ^ω ($c \in C$) by Hahn-Kolmogorov:

$$\mu_c(\varepsilon\uparrow) = 1$$
$$\mu_c(av\uparrow) = \sum_{c' \in C} \gamma(a, c')\mu_{c'}(v\uparrow)$$

Example: Probabilistic Traces as α-Traces

- $MX = \mathcal{D}(\Sigma^* \times \_)$
- $\alpha(\mu)$ zero outside $\Sigma \times X$
- $M1 \cong \mathcal{D}(\Sigma^*)$
- $M!\gamma^{(n)}(c)$ concentrated at Σ^n.
- $\mu_c(v\uparrow) = (M!\gamma^{(n)}(c))(v)$, so

  α-trace equivalence is probabilistic trace equivalence.

Example: LTS with Explicit Termination

Recall: $G = \mathcal{P}(1 + \Sigma \times X)$.

- $MX = \mathcal{P}(\Sigma * \times (X + 1))$
  - $1 = \{\checkmark\}$ for successful termination
- $\alpha_X(S) = \{(\varepsilon, \checkmark) \mid \checkmark \in S\} \cup \{(a, x) \mid (a, x) \in S\}$
- $M1 \cong \mathcal{P}(\Sigma^*)^2$: sets of accepted and non-blocked words.

$\rightarrow$ α-trace semantics $\neq$ language semantics:

- Similar to CSP traces (distinguish deadlock from successful termination $\checkmark$)
- To get language semantics: w.l.o.g. $G = 2 \times \mathcal{P}_{\neq\emptyset}(X)^\Sigma$ (non-blocking NDAs)

Example: Behavioural Equivalence

- $\alpha : G \to M = G^*$ (algebraically-)free monoid over G
- $G^n 1 \hookrightarrow M1$ (terminal sequence)

α-Trace equivalence $=$ finite-depth behavioural equivalence
($=$ behavioural equivalence if G finitary)

Monads vs. Kleisli Liftings

Recall:

- $G = TF$, distributive law $\lambda : FT \to TF$, lifting $\bar{F}$ to $\mathsf{Kl}(T)$
- Language equivalence via $\nu\bar{F}$
- Equivalently: via final sequence of $\bar{F}$, with entries living over

$$TF^n\emptyset$$

- Assumptions include $\bot : X \to MY$

From λ have monad $M = TF^*$, $\alpha : TF \to TF^*$:

- Have $TF^n\emptyset \hookrightarrow M\emptyset$
- To get from α-trace equivalence to language equivalence, postcompose with

$$\bot^* : M1 \to M\emptyset.$$

Monads vs. Kleisli Liftings: Example

$MX = \mathcal{P}(\Sigma^* \times (X + 1))$:

- $M1 \cong \mathcal{P}(\Sigma^*)^2$
- $M\emptyset \cong \mathcal{P}(\Sigma^*)$

$$\bot^* : M1 \to M\emptyset$$
$$(S, U) \mapsto S$$

i.e. erase non-blocked words

Monads vs. $T1$-Valued Relations

Cirstea 2014:

- Focus on infinite traces
- System types HTF, T monad; for simplicity just TF:
  - Require T to be commutative and partially additive
    $\rightarrow$ $T1$ partial additive semiring; additionally require $T1$ cpo
  - Trace semantics as $T1$-valued relation on $C \times \nu F$
  - E.g. does not handle $\mathcal{D}(\Sigma \times X)$

Monads vs. EM-Liftings

Recall:

- $G = FT$
- Distributive law $\rho : TF \to FT$, lifting $\hat{F}$ to $\mathrm{EM}(T) = \mathbf{Set}^T$
- For F finitary, trace semantics in final $\hat{F}$ sequence with entries over

$$F^n 1$$

For T, F finitary, have $\alpha : FT \to M = (F + T)/\rho$, from adjunction

$$\mathbf{Set} \leftarrow \mathbf{Set}^T \leftarrow \mathrm{Alg}(\hat{F})$$

Have $F^n T \hookrightarrow M$, and $M!\gamma^{(n)} : C \to F^n T 1 \hookrightarrow M1$

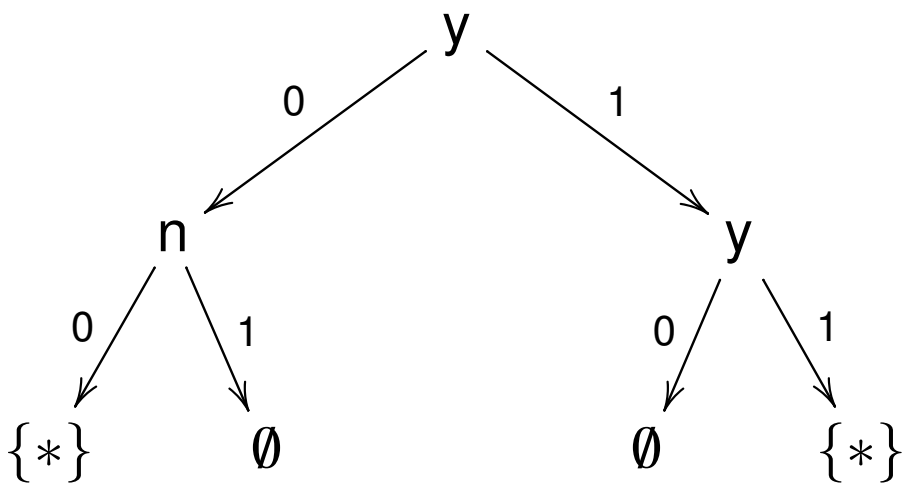
To get from α-trace equivalence to language equivalence, postcompose with

$$F^n !_{T1} : F^n T 1 \to F^n 1.$$

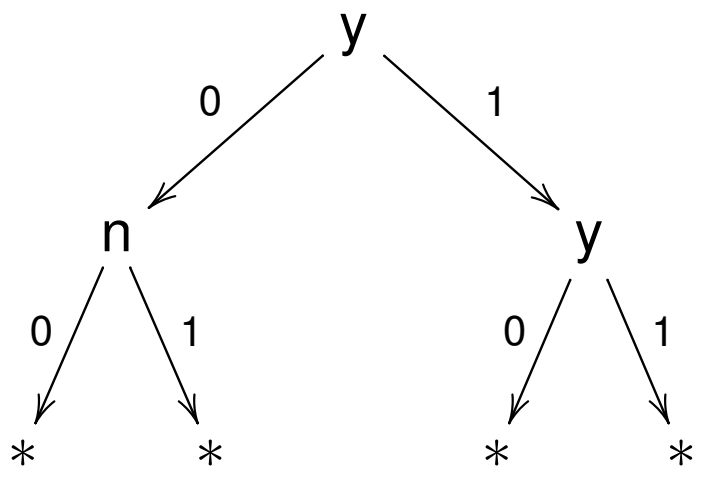
Monads vs. EM-Liftings: Example

$F = 2 \times \_^{\Sigma}$, $\Sigma = \{0,1\}$: two binary operations y, n
$T = \mathcal{P}$

$F^n\mathcal{P}(1)$:



$F^n 1$:



(N.B. $\emptyset = \mathrm{n}(\emptyset, \emptyset)$, so really uniform depth)
Left: 1 accepted, 11 and 0 non-blocked
Right: 1 accepted

Monads vs. Graded Premonads

Lurking behind this:

$((M_n)_{n<\omega}, \eta : \mathrm{id} \to M_0, *)$ graded premonad if

- $f_n^* : M_n X \to M_{k+n} Y$ for $f_X : X \to M_k Y$
- $(f_k^* g)_m^* = f_{m+k}^* g_m^* \qquad \eta_n^* = id \qquad f_0^* \eta = f.$

Captures 'terms of depth n'. Examples:

- M monad, $M_n = M$ for all n
- $\rho : TF \to FT$, $M_n = F^n T$ (also for unranked F, T)
- $\lambda : FT \to TF$, $M_n = TF^n$, e.g.

$$M_n X = \mathcal{P}(\Sigma^n \times X) \qquad M_n X = \mathcal{D}(\Sigma^n \times X)$$

Conclusions and Future Work

- Simple and encompassing framework for trace semantics
- Covers standard trace semantics of vanilla LTS
- Covers probabilistic trace semantics
- Spans full linear-time/branching-time spectrum
- Existing approaches recovered by additional abstraction
- Forthcoming: actual results in the framework