Generic Trace Semantics and Graded Monads

Stefan Milius\textsuperscript{a}    Dirk Pattinson\textsuperscript{b}    Lutz Schröder\textsuperscript{a}

\textsuperscript{a}Friedrich-Alexander-Universität Erlangen-Nürnberg

\textsuperscript{b}Australian National University, Canberra

CALCO 2015
Introduction

- Coalgebra does bisimilarity
- Traces need algebra: impose additional equational laws
  - See [Kurz/Milius/Pattinson/Schröder 2015] for trace semantics via monads
- Here: refine this to use graded monads
  - control over trace length
- Generalize all previous approaches
- Introduce graded algebras
  - obtain generic trace logic
Trace Semantics and Algebraic Laws

\[ \text{LTS} = \text{Coalgebras for functor } G_X = \mathcal{P}(\Sigma \times X). \]
Algebraically: **Shallow theory**

- Operations \( \sum a_i(\cdot)_i \)
- Shallow laws (e.g. \( a(x) + b(y) = b(y) + a(x) \))

Trace semantics:

- Split operations : \( \sum, a \)
- JSL laws for \( \sum \)
- Distributivity:
  \[ a(\sum x_i) = \sum a(x_i), \]
Trace Semantics, Incrementally

\[
\begin{align*}
C & \xrightarrow{a} C_{10} \xrightarrow{a} C_{11} \\
& \xrightarrow{b} C_{20}
\end{align*}
\]

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

<table>
<thead>
<tr>
<th>Stage</th>
<th>Pretraces</th>
<th>Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 0</td>
<td>{ ((\varepsilon), (c)) }</td>
<td>{ (\varepsilon) }</td>
</tr>
<tr>
<td>Stage 1</td>
<td>{ ((a), (c_{10})), ((a), (c_{11})) }</td>
<td>{ (a) }</td>
</tr>
<tr>
<td>Stage 2</td>
<td>{ ((ab), (c_{20})) }</td>
<td>{ (ab) }</td>
</tr>
<tr>
<td>Stage 3</td>
<td>(\emptyset)</td>
<td>(\emptyset)</td>
</tr>
</tbody>
</table>
Graded Monads

(Smirnov 2008)

- Functors $M_n : \mathbf{C} \to \mathbf{C}$
  - Think: $M_n X = \text{Terms of uniform depth } n \text{ over } X$
- Unit $\eta : id \to M_0$
- Multiplication $\mu^{nk} : M_n M_k \to M_{n+k}$
- Correspond to graded theories
  - operators with assigned depths (e.g. 0, 1)
  - Uniform-depth equations

Here: depth = trace length.
Traces Semantics Via Graded Monads

Trace semantics of $G =$

- Graded monad $(M_n)$
- Natural transformation $\alpha : G \rightarrow M_1$

For coalgebra $\gamma : X \rightarrow GX$ $\alpha$-pretrace maps

$$\gamma^{(0)} = \eta : X \rightarrow M_0X$$

$$\gamma^{(n+1)} = X \xrightarrow{\alpha \gamma} M_1X \xrightarrow{M_1 \gamma^{(n)}} M_1M_nX \xrightarrow{\mu_X^{1n}} M_{n+1}X$$

and $\alpha$-trace maps

$$X \xrightarrow{\gamma^{(n)}} M_nX \xrightarrow{M_n!} M_n1$$
Finite-depth bisimilarity: $M_n = G^n$; trace maps $X \rightarrow G^n 1 = \text{cone into final sequence}$.

Trace semantics of LTS: $M_n X = \mathcal{P}(\Sigma^n \times X)$

Mazurkiewicz traces

Probabilistic traces:

- Generative probabilistic transition systems: $G = \mathcal{D}(\Sigma \times X)$
- $M_n = \mathcal{D}(\Sigma^n \times X)$ from $a(\sum p_i x_i) = \sum p_i a(x_i)$
Example: Trace Semantics, Kleisli Style

(Hasuo/Jacobs/Sokolova 2007)

E.g. LTS with explicit termination (NDA) are $TF$-coalgebras for

$$T = \mathcal{P} \quad F = 1 + \Sigma \times -$$

Kleisli law ($\cong$ lifting $\bar{F}$ of $F$ to Kleisli category)

$$\lambda_X : FT \to TF$$

+ some assumptions $\to$ language semantics via final map

$$X \to T \nu \bar{F} \cong T \mu F = \mathcal{P}(\Sigma^*)$$

(becomes trivial for plain LTS, i.e. $F = \Sigma \times -$)

- $TF^n$ is graded monad
- $\alpha$-trace maps $X \to TF^n 1 \cong \mathcal{P}(\Sigma_{i \leq n} \Sigma^i)$: traces and accepted words
- Language semantics by canonical forgetting
Example: Trace Semantics, Eilenberg-Moore Style

(Bonsangue/Milius/Silva 2013)

E.g. LTS with explicit termination (NDA) are $FT$-coalgebras for

$$T = \mathcal{P} \quad F = 2 \times \Sigma$$

EM-law ($\cong$ lifting $\hat{F}$ of $F$ to EM category)

$$\rho : TF \to FT$$

Language semantics: Final Map

$$X \xrightarrow{\eta} TX \to U\nu\hat{F} \cong \nu F = \mathcal{P}(\Sigma^*)$$

(becomes trivial for plain LTS, i.e. $F = X^{\Sigma}$)

- $F^nT$ is graded monad
- $\alpha$-trace maps $X \to F^nT1 = F^n\mathcal{P}1$: traces and accepted words
- Language semantics by canonical forgetting
Graded Algebras

*M_n*-algebra for \( n < \omega \) (similarly for \( n = \omega \)):

- **Objects** \( A_k, k \leq n \)

- **Maps**

  \[
  a^{mk} : M_mA_k \to A_{m+k} \quad (m+k \leq n)
  \]
(\(M_k X\))_{k \leq n} is the free \(M_n\)-Algebra over \(X\).

\( \rightarrow \) \(M_n\)-Algebras + one truth value \(\tau : 1 \rightarrow A_n\) are trace properties:

\[
\begin{array}{c}
X \xrightarrow{M_n! \gamma(n)} M_n1 \xrightarrow{\tau_n^\#} A_n
\end{array}
\]

Compositional syntax?

That’s what trace logics are about:

\(\langle a\rangle \top \land \langle b\rangle \top\) is trace-invariant, \(\langle a\rangle (\langle a\rangle \top \land \langle b\rangle \top)\) is not.
Depth-1 Graded Monads

- Depth-1 graded theory:
  - all operations and equations have depth \( \leq 1 \)

- Depth-1 graded monad:
  - \( \mu^{nk} \) and \( M_0 \mu^1n \) are pointwise epi
  - \( \mu^1n \) is a (reflexive) coequalizer

\[
M_1 M_0 M_n \xrightarrow{M_1 \mu^0n} M_1 M_n \xrightarrow{\mu^1n} M_{1+n}.
\]

In depth 1, \( M_n \)-algebras are compatible chains of \( M_1 \)-algebras
Generic Trace Logics

Formulas:

- Depth 0: Constants $c$
- Depth $n + 1$: $L(\phi_1, \ldots, \phi_n)$, $L$ modal operator, $\phi_i$ depth $n$.

Semantics:

- $M_0$-algebra $\Omega$ of truth values
- Truth values $\llbracket c \rrbracket : 1 \rightarrow \Omega$
- $M_1$-algebras $\llbracket L \rrbracket : M_1(\Omega^n) \rightarrow \Omega$.
- $\llbracket \phi \rrbracket$ is trace property ($M_n$-algebra, truth value)
Trace logics: Examples

- **LTS** \((G = \mathcal{P}(\Sigma \times X))\):
  - Truth value object: 2 with the usual \(\mathcal{P}\)-algebra structure (complete JSL)
  - Unary operators: e.g.
    \[
    \llbracket \langle a \rangle \rrbracket : \mathcal{P}(\Sigma \times 2) \to 2, \quad S \mapsto \left\{ \begin{array}{ll}
    \top & (a, \top) \in S \\
    \bot & \text{otherwise}
    \end{array} \right.
    \]
  - Generally: \(\llbracket L \rrbracket\) interprets \(a \in \Sigma\) as join-cont. map \(2^n \to 2\)
    \(\to\) add disjunction

- **Finite-depth bisimilarity** \((M_n = G^n)\):
  - Set 2 of truth values
  - \(k\)-ary operators: maps \(\llbracket L \rrbracket : G(2^k) \to 2\) are \(k\)-ary predicate liftings
    - closed under Boolean operators
    \(\to\) coalgebraic modal logic
Example: Probabilistic Trace Logic

For $G = \mathcal{D}(\Sigma \times X)$:

- Set $[0, 1]$ of truth values
- Unary operators: e.g.

$$\llbracket \langle a \rangle \rrbracket : \mathcal{D}(\Sigma \times [0, 1]) \rightarrow [0, 1]$$

$$\mu \mapsto \sum_{p \in [0, 1]} p \mu(a, p)$$

- Generally: $k$-ary operator interprets $a$ as convex morphism $[0, 1]^k \rightarrow [0, 1]$, i.e. affine map
- Thus: add affine combinations, including fuzzy negation $1 - \phi$.
- Precisely:

$$\phi ::= \sum_{a \in \Sigma} \langle a \rangle (c_a + \sum q_{ai} \phi_{ai})$$

where $c_a + \sum q_{ai}(\cdot)_{ai} : [0, 1]^k \rightarrow [0, 1]$. 
Conclusions and Future Work

- Trace semantics via graded monoids
  - subsumes existing approaches to generic finite traces

- **Graded algebras** are formulae in trace logics
  - Cover proper probabilistic systems

- Future work:
  - Expressivity
  - Temporal extensions
  - Model checking
  - Reasoning