Testing Semantics: Processes vs Logics

Michael Mislove
Dusko Pavlovic
James Worrell

AMAST 2006

Partially supported by ONR and NSF

Outline

- Systems and tests
- Dualities & Logical Connections
- *Examples:* Finite Automata, Pushdown Automata, Turing Machines
- Summary and future work

Systems & Tests

- Sys – family of *processes*
- Test – *tests* applied to system
- Ω – space of *observations*
- $\mathsf{Sys} \times \mathsf{Test} \xrightarrow{\Upsilon} \Omega$
- $$\begin{tikzcd} \mathsf{Sys} \arrow[rr, "\models"] \arrow[dr, two heads] & & \Omega^{\mathsf{Test}} \\ & L \arrow[ur, hook] & \end{tikzcd} \qquad \begin{tikzcd} \mathsf{Test} \arrow[rr, "\dashv"] \arrow[dr, two heads] & & \Omega^{\mathsf{Sys}} \\ & M \arrow[ur, hook] & \end{tikzcd}$$

Dualities...

- $$\begin{tikzcd} \mathsf{Set}^{\mathrm{op}} \arrow[r, bend left, "X \mapsto \mathsf{Set}(X{,}2)"] & \mathsf{cBa} \arrow[l, bend left, "L \mapsto \mathsf{cBa}(L{,}2)"] \end{tikzcd}$$
- $$\begin{tikzcd} \mathsf{Sob}^{\mathrm{op}} \arrow[r, bend left, "X \mapsto \mathsf{Sob}(X{,}2)"] & \mathsf{sHeyt} \arrow[l, bend left, "L \mapsto \mathsf{sHeyt}(L{,}2)"] \end{tikzcd}$$
- $$\begin{tikzcd} \mathsf{Stone}^{\mathrm{op}} \arrow[r, bend left, "X \mapsto \mathsf{Top}(X{,}2)"] & \mathsf{Ba} \arrow[l, bend left, "L \mapsto \mathsf{Ba}(L{,}2)"] \end{tikzcd}$$
- $$\begin{tikzcd} \mathsf{Top}^{\mathrm{op}} \arrow[r, bend left, "X \mapsto \mathsf{Top}(X{,}2)"] & \mathsf{Heyt} \arrow[l, bend left, "L \mapsto \mathsf{Heyt}(L{,}2)"] \end{tikzcd}$$

Logical connection: $$\begin{tikzcd} \mathcal{S}^{\mathrm{op}} \arrow[r, bend left, "P"] \arrow[r, phantom, "\perp"] & \mathcal{T} \arrow[l, bend left, "M"] \end{tikzcd}$$

Finite automata

$\mathcal{A} = (S, \Sigma, s_0, F, \delta)$ where:

- S – *states*
- Σ – *actions*, *inputs*, etc.
- s_0 – initial state
- F – final states
- $\delta\colon S \times \Sigma \to \mathcal{P}(S)$ – transition function

Automata are coalgebras

- $\mathcal{A} = (S, \Sigma, s_0, F, \delta) \quad \Longrightarrow \quad \delta\colon S \times \Sigma \to \mathcal{P}(S)$, so

  $\mathcal{A}$ coalgebra for $\quad X \xrightarrow{\partial} \mathcal{P}(X)^\Sigma$

- Adding final states: $\quad \chi_F\colon S \to 2$

  $\mathcal{A}$ coalgebra for $\quad X \xrightarrow{\chi_F \times \partial} 2 \times \mathcal{P}(X)^\Sigma$

- Adding outputs: $\quad \delta'\colon S \times \Sigma \to \Gamma \times \mathcal{P}(S)$

  $\mathcal{A}$ coalgebra for $\quad X \xrightarrow{\chi_F \times \partial'} 2 \times \mathcal{P}(\Gamma \times X)^\Sigma$

Tests are Algebras

$\mathsf{Fsa} = \ {}_G\mathsf{Set} : \quad X \xrightarrow{\partial} GX = 2 \times \mathcal{P}_f(X)^\Sigma$

Grammar for tests on Fsa: $\ t ::= \top \mid a.t \quad (a \in \Sigma)$

Test – Initial algebra for $\ FA = 1 + \Sigma \times A \ \simeq \ \Sigma^* \top$

obtained via trivial monad $T \colon \mathsf{Set} \to \mathsf{Set}$

$\Theta B = \mu X.\ F(\Sigma \times (B + X))$ – free (weak) F-algebra over B on which Σ acts

$\mathsf{Test} = \ \mathsf{Set}_F : \quad FA = 1 + \Sigma \times A \xrightarrow{\alpha} A$

Relating Fsa & Test

$$\begin{tikzcd} \mathsf{Set}^{\mathrm{op}} \arrow[r, bend left, "\mathcal{P}"] \arrow[r, phantom, "\perp"] & \mathsf{Set} \arrow[l, bend left, "\mathcal{P}^{\mathrm{op}}"] \end{tikzcd}$$

$$\mathcal{P}X \simeq 2^X \quad \Rightarrow$$

$$\begin{aligned} X \to \mathcal{P}Y \quad &\simeq \quad X \to 2^Y \\ &\simeq \quad X \times Y \to 2 \simeq Y \times X \to 2 \\ &\simeq \quad Y \to 2^X \simeq Y \to \mathcal{P}X \end{aligned}$$

From Fsa to Test

Grammar for tests on Fsa: $t ::= \top \mid a.t \quad (a \in \Sigma)$

$\mathcal{A} \in \mathsf{Fsa}, x \in X_{\mathcal{A}}$: $(x \models \top) = \chi_F$ & $(x \models a.t) = \bigvee_{x \xrightarrow{a} y} (y \models t)$

For $\sigma \in \Sigma^*$: $(\mathcal{A} \models \sigma) \leftrightarrow (s_0 \models \sigma.\top)$

$\models$: $\mathsf{Fsa} \to \mathsf{Test}$ by $\mathcal{A} \mapsto \mathcal{L}_{\mathcal{A}} = \{\sigma \in \Sigma^* \mid (\mathcal{A} \models \sigma.\top)\} \subseteq \mathcal{P}\Sigma^*$

A Distributive Law

$$F\mathcal{P}X = 1 + \Sigma \times \mathcal{P}(X) \hookrightarrow \mathcal{P}(2 \times \mathcal{P}(X)^{\Sigma}) = \mathcal{P}GX$$

$$\begin{array}{rcl} 1 + \Sigma \times \mathcal{P}X & \hookrightarrow & \mathcal{P}(2 \times \mathcal{P}(\Sigma \times X)) \\ \top & \mapsto & \{\langle 1, \emptyset \rangle\} \\ \langle \sigma, U \rangle & \mapsto & \{\langle 0, \{\sigma\} \times V \rangle \mid V \in \mathcal{P}U\} \end{array}$$

$$\lambda\colon\ F\mathcal{P} \stackrel{\cdot}{\longrightarrow} \mathcal{P}G$$

Fsa's map to *F*-algebras

$$\lambda\colon\ F\mathcal{P} \stackrel{\cdot}{\longrightarrow} \mathcal{P}G$$

$$\mathcal{A}:\quad X_{\mathcal{A}} \xrightarrow{\ \partial\ } GX_{\mathcal{A}} \ \Rightarrow\ F\mathcal{P}X_{\mathcal{A}} \xrightarrow{\ \lambda\ } \mathcal{P}GX_{\mathcal{A}} \xrightarrow{\ \mathcal{P}\partial\ } GX_{\mathcal{A}}$$

- $\mathcal{P}\colon \mathsf{Set}^{\mathrm{op}} \longrightarrow \mathsf{Set}$ lifts to $\widehat{\mathcal{P}}\colon\ ({}_G\mathsf{Set})^{\mathrm{op}} \longrightarrow \mathsf{Set}_F$ by

$$\frac{X \xrightarrow{\ \partial\ } GX}{\widehat{\mathcal{P}}\partial : F\mathcal{P}X \xrightarrow{\ \lambda\ } \mathcal{P}GX \xrightarrow{\ \mathcal{P}\partial\ } \mathcal{P}X}$$

Twisted coalgebra morphisms

$$
\begin{tikzcd}
FA \arrow[r, "Ff"] \arrow[dd, "\alpha"'] & F\mathcal{P}X \arrow[d, "\lambda"'] \arrow[dd, bend left, "\widehat{\mathcal{P}}\partial"] \\
& \mathcal{P}GX \arrow[d, "\mathcal{P}\partial"'] \\
A \arrow[r, "f"'] & \mathcal{P}X
\end{tikzcd}
\Longrightarrow
\begin{tikzcd}
\mathcal{P}F\mathcal{P}X \arrow[r, "\mathcal{P}Ff"] & \mathcal{P}FA \\
GX \arrow[u, "\lambda'"'] & \\
X \arrow[u, "\partial"'] \arrow[uu, bend left, "\Lambda\partial"] \arrow[r, "f'"'] & \mathcal{P}A \arrow[uu, "\mathcal{P}\alpha"']
\end{tikzcd}
$$

$$\Lambda : {}_G\mathsf{Set} \to {}_{\mathcal{P}F\mathcal{P}}\mathsf{Set} : \ X \xrightarrow{\partial} GX \ \mapsto \ X \xrightarrow{\partial} GX \xrightarrow{\lambda'} \mathcal{P}F\mathcal{P}X$$

Epi-Mono Factorization

$\mathsf{Aut} \to G(\mathsf{Aut})$ – final for all *finite* G-coalgebras,
– bisimulation classes of finite-state automata.

$\mathsf{Aut} \xrightarrow{\models} \mathcal{P}\Sigma^*$ has image $\mathsf{Reg} \subseteq \mathcal{P}\Sigma^*$

$$\begin{array}{rcl} \mathsf{Reg} & \to & \mathcal{P}\left(1 + \Sigma \times \mathcal{P}(\mathsf{Reg})\right) \\ L & \mapsto & \{\top | \langle\ \rangle \in L\} \cup \{\langle a, U\rangle | \partial_a(L) \in U\rangle\} \end{array}$$

where $\partial_a(L) = \{\sigma \in \Sigma^* |\ a.\sigma \in L\}$.

Aut vs. Reg

$$
\begin{tikzcd}
\mathcal{P}\left(1+\Sigma\times\mathcal{P}(\mathsf{Aut})\right) \arrow[r] & \mathcal{P}\left(1+\Sigma\times\mathcal{P}(\mathsf{Reg})\right) \arrow[r] & \mathcal{P}(1+\Sigma\times\Sigma^*) \\
2\times\mathcal{P}_f(\Sigma\times\mathsf{Aut}) \arrow[u, "\lambda'"] & & \\
\mathsf{Aut} \arrow[u, "\partial"] \arrow[r, two heads] & \mathsf{Reg} \arrow[uu, "r"'] \arrow[r, hook] & \mathcal{P}\Sigma^* \arrow[uu, "\mathcal{P}\alpha"']
\end{tikzcd}
$$

Psa vs. Cfl

$$\mathsf{Psa}\colon\ \text{coalgebras for } GX = 2 \times \mathcal{P}(X \times \Gamma^*)^{\Sigma+1}$$

$$\begin{tikzcd}
\mathcal{P}\left(1+\Sigma\times\mathcal{P}(\mathsf{Psa})\right) \arrow[r] & \mathcal{P}\left(1+\Sigma\times\mathcal{P}(\mathsf{Cfl})\right) \arrow[r] & \mathcal{P}(1+\Sigma\times\Sigma^*) \\
2\times\left(\mathcal{P}(\mathsf{Psa}\times\Gamma^*)\right)^{\Sigma+1} \arrow[u, "\lambda'"] & & \\
\mathsf{Psa} \arrow[u, "<\Phi{,}\partial>"] \arrow[r, two heads] & \mathsf{Cfl} \arrow[uu] \arrow[r, hook] & \mathcal{P}\Sigma^* \arrow[uu, "\mathcal{P}\alpha"']
\end{tikzcd}$$

Basic Setup

$$\textit{Logical connection:}\quad \begin{tikzcd} \mathcal{S}^{\mathrm{op}} \arrow[r, bend left, "P"] \arrow[r, phantom, "\perp"] & \mathcal{T} \arrow[l, bend left, "M"] \end{tikzcd}$$

Sys:	coalgebras for $G\colon \mathcal{S} \to \mathcal{S}$
Test:	algebras for $F\colon \mathcal{T} \to \mathcal{T}$
distributive law	$\lambda\colon FP \stackrel{\cdot}{\to} PG$

Main Theorem

1) $P : \mathcal{S}^{op} \to \mathcal{T}$ lifts to $\widehat{P} : ({}_G\mathcal{S})^{op} \to \mathcal{T}_F$, mapping

$$\frac{X \xrightarrow{\partial} GX}{\widehat{P}\partial : FPX \xrightarrow{\lambda} PGX \xrightarrow{P\partial} PX}$$

Main Theorem (cont)

$$
\begin{tikzcd}
FA \arrow[r, "Ff"] \arrow[dd, "\alpha"'] & FPX \arrow[d, "\lambda"'] \arrow[dd, bend left, "\widehat{P}\partial"] \\
 & PGX \arrow[d, "P\partial"'] \\
A \arrow[r, "f"'] & PX
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
MFPX \arrow[r, "MFf"] & MFA \\
GX \arrow[u, "\lambda'"] & \\
X \arrow[u, "\partial"] \arrow[uu, bend left, "\Lambda\partial"] \arrow[r, "f'"'] & MA \arrow[uu, "M\alpha"']
\end{tikzcd}
$$

$$\Lambda : {}_{G}\mathcal{S} \to {}_{MFP}\mathcal{S} : \; X \xrightarrow{\partial} GX \; \mapsto \; X \xrightarrow{\partial} GX \xrightarrow{\lambda'} MFPX$$

Main Theorem (concl)

$\mathcal{S}$ regular and MFP preserves weak pullbacks imply

$$\begin{tikzcd}
MFPX \arrow[r, "MFPe"] & MFPL \arrow[r, "MFm'"] & MFA \\
GX \arrow[u, "\lambda'"] & & \\
X \arrow[u, "\partial"] \arrow[r, two heads, "e"'] & L \arrow[uu, dashed, "\ell"'] \arrow[r, hook, "m"'] & MA \arrow[uu, "M\alpha"']
\end{tikzcd}$$

Other possibilities

- Turing machines
- Simulation and bisimulation
- Probabilistic models
  - Labeled Markov Processes
- etc....

Related work

- Abramsky (1991), Rutten (2003), Plotkin & Turi (1998?)
- Kupke, Kurz and Pattinson (2005)
- Bonsangue and Kurz (2005)

Why I'm giving this talk