



Testing Semantics: Processes vs Logics

Michael Mislove

Dusko Pavlovic

James Worrell

AMAST 2006

Partially supported by ONR and NSF



Outline

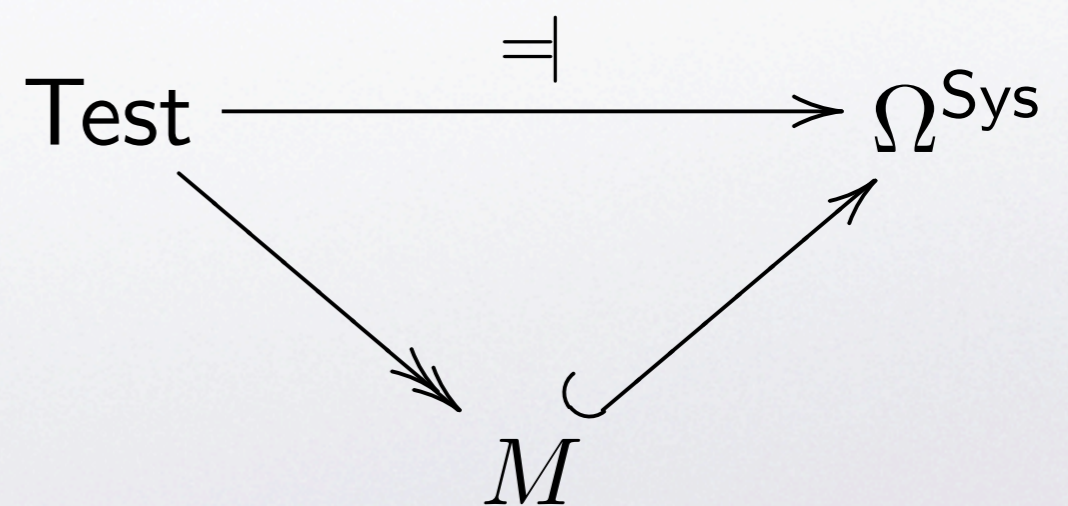
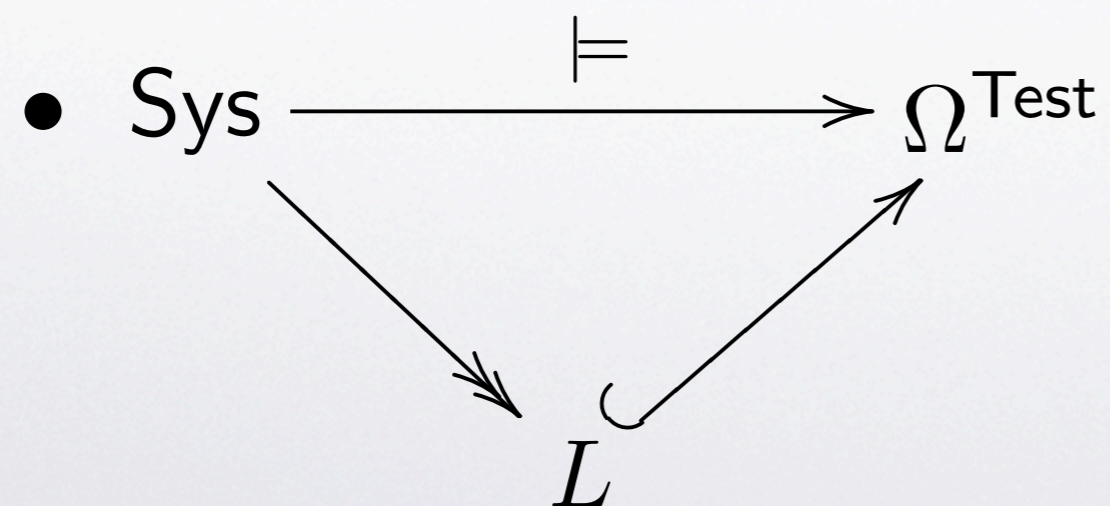
- **Systems and tests**
- **Dualities & Logical Connections**
- ***Examples:* Finite Automata, Pushdown Automata, Turing Machines**
- **Summary and future work**



Systems & Tests

- Sys – family of *processes*
- Test – *tests* applied to system
- Ω – space of *observations*

- $\text{Sys} \times \text{Test} \xrightarrow{\gamma} \Omega$





Dualities...

$$\bullet \text{ Set}^{\text{op}} \begin{array}{c} \xrightarrow{X \mapsto \text{Set}(X, 2)} \\ \xleftarrow{L \mapsto \text{cBa}(L, 2)} \end{array} \text{cBa}$$

$$\bullet \text{ Sob}^{\text{op}} \begin{array}{c} \xrightarrow{X \mapsto \text{Sob}(X, 2)} \\ \xleftarrow{L \mapsto \text{sHeyt}(L, 2)} \end{array} \text{sHeyt}$$

$$\bullet \text{ Stone}^{\text{op}} \begin{array}{c} \xrightarrow{X \mapsto \text{Top}(X, 2)} \\ \xleftarrow{L \mapsto \text{Ba}(L, 2)} \end{array} \text{Ba}$$

$$\bullet \text{ Top}^{\text{op}} \begin{array}{c} \xrightarrow{X \mapsto \text{Top}(X, 2)} \\ \xleftarrow{L \mapsto \text{Heyt}(L, 2)} \end{array} \text{Heyt}$$

Logical connection: $\mathcal{S}^{\text{op}} \begin{array}{c} \xrightarrow{P} \\ \perp \\ \xleftarrow{M} \end{array} \mathcal{T}$



Finite automata

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

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



Automata are coalgebras

- $\mathcal{A} = (S, \Sigma, s_0, F, \delta) \implies \delta: S \times \Sigma \rightarrow \mathcal{P}(S)$, so

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

- Adding final states: $\chi_F: S \rightarrow 2$

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

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

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



Tests are Algebras

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

Grammar for tests on \mathbf{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 $T: \mathbf{Set} \rightarrow \mathbf{Set}$

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

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



Relating Fsa & Test

$$\text{Set}^{\text{op}} \begin{array}{c} \xrightarrow{\mathcal{P}} \\ \perp \\ \xleftarrow{\mathcal{P}^{\text{op}}} \end{array} \text{Set}$$

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

$$X \rightarrow \mathcal{P}Y \simeq X \rightarrow 2^Y$$

$$\simeq X \times Y \rightarrow 2 \simeq Y \times X \rightarrow 2$$

$$\simeq Y \rightarrow 2^X \simeq Y \rightarrow \mathcal{P}X$$



From Fsa to Test

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

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

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

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



A Distributive Law

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

$$1 + \Sigma \times \mathcal{P}X \hookrightarrow \mathcal{P}(2 \times \mathcal{P}(\Sigma \times X))$$

$$\top \mapsto \{\langle 1, \emptyset \rangle\}$$

$$\langle \sigma, U \rangle \mapsto \{\langle 0, \{\sigma\} \times V \rangle \mid V \in \mathcal{P}U\}$$

$$\lambda: FP \xrightarrow{\cdot} \mathcal{P}G$$



Fsa's map to F -algebras

$$\lambda: F\mathcal{P} \longrightarrow \mathcal{P}G$$

$$A: X_A \xrightarrow{\partial} GX_A \Rightarrow F\mathcal{P}X_A \xrightarrow{\lambda} \mathcal{P}GX_A \xrightarrow{\mathcal{P}\partial} GX_A$$

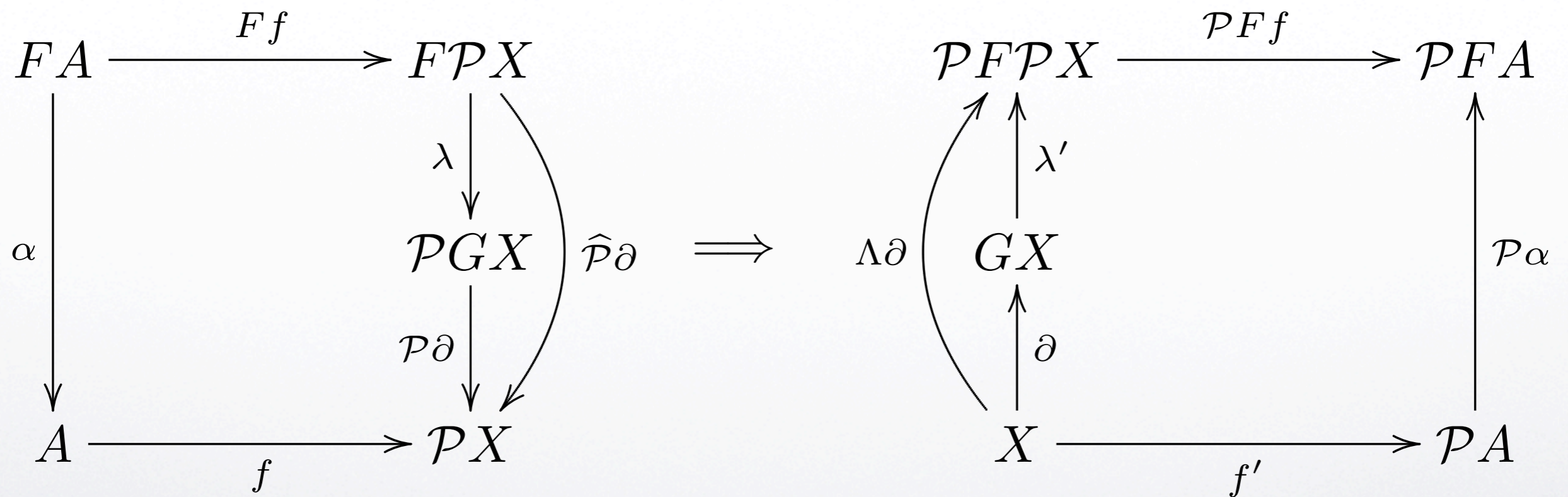
- $\mathcal{P}: \text{Set}^{\text{op}} \longrightarrow \text{Set}$ lifts to $\hat{\mathcal{P}}: ({}_G\text{Set})^{\text{op}} \longrightarrow \text{Set}_F$ by

$$X \xrightarrow{\partial} GX$$

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



Twisted coalgebra morphisms



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



Epi-Mono Factorization

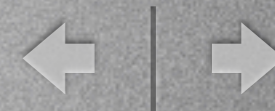
$\text{Aut} \rightarrow G(\text{Aut})$ – final for all *finite* G -coalgebras,
– bisimulation classes of finite-state automata.

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

$$\text{Reg} \rightarrow \mathcal{P}(1 + \Sigma \times \mathcal{P}(\text{Reg}))$$

$$L \mapsto \{\top \mid \langle \rangle \in L\} \cup \{\langle a, U \rangle \mid \partial_a(L) \in U\}$$

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



Aut vs. Reg

$$\begin{array}{ccccc} \mathcal{P}(1 + \Sigma \times \mathcal{P}(\text{Aut})) & \longrightarrow & \mathcal{P}(1 + \Sigma \times \mathcal{P}(\text{Reg})) & \longrightarrow & \mathcal{P}(1 + \Sigma \times \Sigma^*) \\ \uparrow \lambda' & & \uparrow r & & \uparrow \mathcal{P}\alpha \\ 2 \times \mathcal{P}_f(\Sigma \times \text{Aut}) & & & & \\ \uparrow \partial & & & & \\ \text{Aut} & \twoheadrightarrow & \text{Reg}^{\subset} & \rightarrow & \mathcal{P}\Sigma^* \end{array}$$



Psa vs. Cfl

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

$$\begin{array}{ccccc}
 \mathcal{P}(1 + \Sigma \times \mathcal{P}(\text{Psa})) & \longrightarrow & \mathcal{P}(1 + \Sigma \times \mathcal{P}(\text{Cfl})) & \longrightarrow & \mathcal{P}(1 + \Sigma \times \Sigma^*) \\
 \uparrow \lambda' & & \uparrow & & \uparrow \mathcal{P}\alpha \\
 2 \times (\mathcal{P}(\text{Psa} \times \Gamma^*))^{\Sigma+1} & & & & \\
 \uparrow \langle \Phi, \partial \rangle & & & & \\
 \text{Psa} & \longrightarrow & \text{Cfl} & \longrightarrow & \mathcal{P}\Sigma^*
 \end{array}$$



Basic Setup

Logical connection: $\mathcal{S}^{\text{op}} \begin{array}{c} \xrightarrow{P} \\ \perp \\ \xleftarrow{M} \end{array} \mathcal{T}$

Sys: coalgebras for $G: \mathcal{S} \rightarrow \mathcal{S}$

Test: algebras for $F: \mathcal{T} \rightarrow \mathcal{T}$

distributive law $\lambda: FP \dot{\rightarrow} PG$



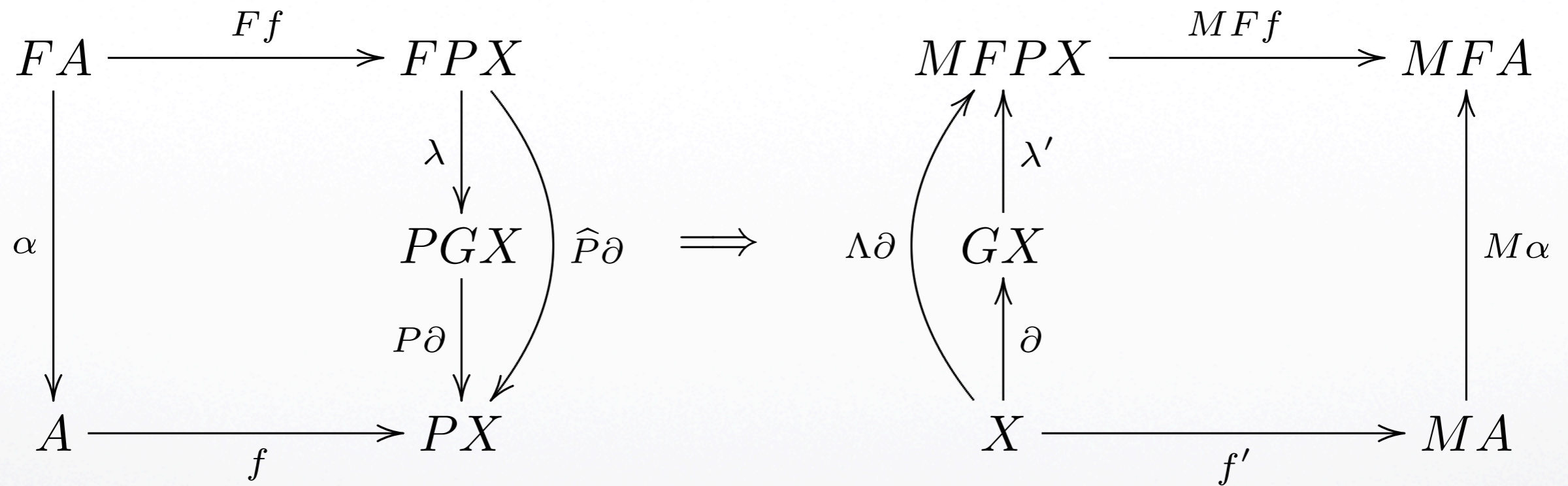
Main Theorem

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

$$\begin{array}{c} X \xrightarrow{\partial} GX \\ \hline \hat{P}\partial : FPX \xrightarrow{\lambda} PGX \xrightarrow{P\partial} PX \end{array}$$



Main Theorem (cont)



$$\Lambda : {}_G\mathcal{S} \rightarrow {}_{MFP}\mathcal{S} : X \xrightarrow{\partial} GX \mapsto X \xrightarrow{\partial} GX \xrightarrow{\lambda'} MFPX$$



Main Theorem (concl)

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

$$\begin{array}{ccccc} MFPX & \xrightarrow{MFP_e} & MFPL & \xrightarrow{MF_{m'}} & MFA \\ \uparrow \lambda' & & \uparrow \ell & & \uparrow M\alpha \\ GX & & & & \\ \uparrow \partial & & & & \\ X & \xrightarrow{e} & L \subset & \xrightarrow{m} & MA \end{array}$$



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

