Basic Observables for Probabilistic May Testing

Mieke Massink

C.N.R.-ISTI, Pisa, Italy

—joint work with *Maria Carla Palmeri* and *Rocco De Nicola* (University of Florence) —

Outline

1. Introduction to Basic Observables
2. Probabilistic Basic Observables
3. Observational semantics for PCCS
4. Probabilistic May Testing
5. Results and Conjectures
6. Conclusions and Future Work

Basic Observables

The idea of Basic Observable has its origin in the classical theory of functional programming:

> Two programs M and N are *observationally equivalent* if for every program context C such that both $C[M]$ and $C[N]$ are programs, and for every value v, we have $C[M] \downarrow v$ iff $C[N] \downarrow v$

This paradigm has been the basis for assessing many semantics of *sequential* programming languages.

Subsequently, variants of this paradigm have been studied for *concurrent* systems as well.

Basic Observables for Concurrent Processes

Milner and Sangiorgi (1992): Barbed bisimulation

- Equivalence relation based on a reduction relation and an observation predicate that detects a process' communication capability over a given channel.
- Two processes are barbed equivalent if they have the same communication capabilities and this property is preserved by internal reduction.
- Requires a co-inductive definition.

Basic Observables for Concurrent Processes

Boreale, De Nicola and Pugliese (1999): guaranteed communication

- $P!l$: process P can *only* reach states (via internal actions) from which action l can eventually performed (after internal actions)
- $P \downarrow$: process P converges
- $P \downarrow l$: process P converges also after performing l

The three predicates (used in different combinations) have been shown to induce five contextual pre-orders that coincide with well-known ones such as the fair/should preorder, the *must* pre-order and the, at that time new, *safe-must* pre-order.

The alternative characterisations support simpler methods for proving that two processes are behaviourally related and is similar to that used by Hennessy in the 'Algebraic Theory of Processes (1988)' based on the notion of acceptance sets.

Probabilistic Basic Observables

Given the successful results of the use of Basic Observables to characterise behavioural relations in the non-probabilistic setting it is interesting to study a probabilistic extension of the approach.

We need to introduce a few well-known concepts:

- Probabilistic LTS
- Probabilistic Automaton
- Probabilistic execution
- (Dirac) scheduler
- Probability measure
- Weak transitions

Probabilistic LTS

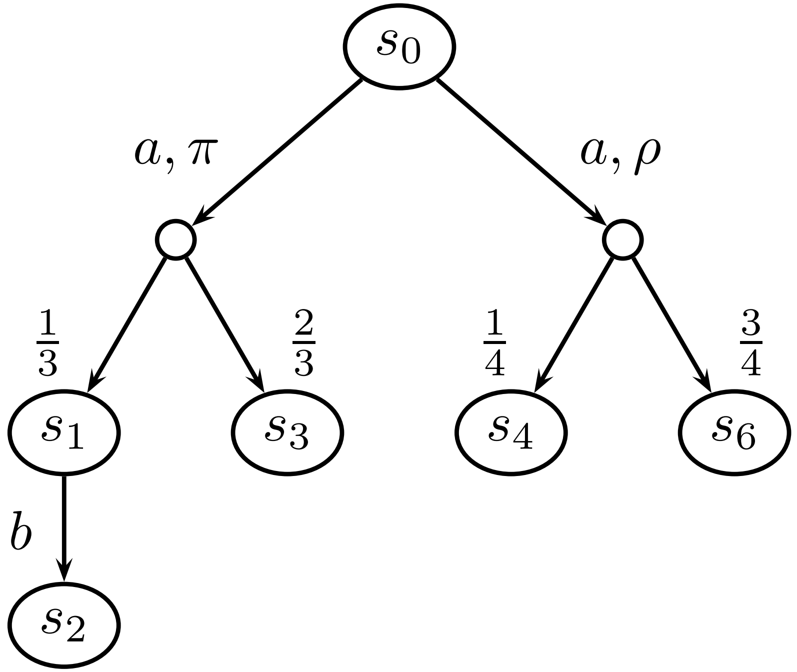
A *probabilistic labelled transition system* (PLTS) is a structure $\mathcal{S} = (S, Act, Steps)$ where

- S is a countable set of *states*,
- Act is a countable set of *actions* containing an *internal action* τ, and
- $Steps \subseteq S \times Act \times Distr(S)$ is a *transition relation*.

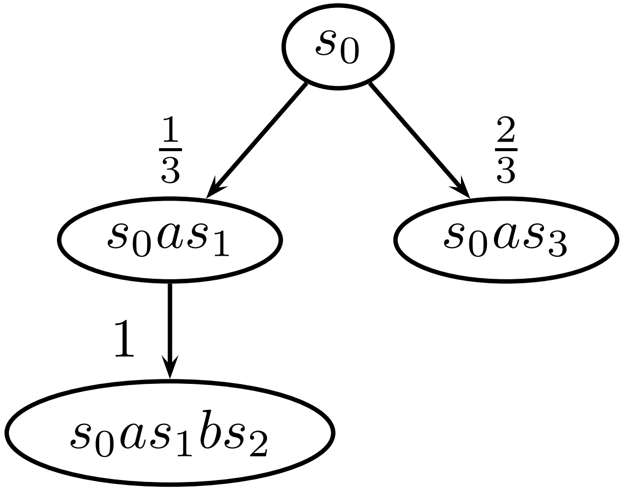
where $Distr(S)$ is the set of (discrete) probability distributions on S.

Probabilistic LTS

Example PLTS:



Example *probabilistic* execution:



Example execution:

$$\alpha = s_0 a s_1 b s_2$$

Resolving *only* non-determism

Resolving non-determinism *and* probabilistic choice

Scheduler

A scheduler is a function that resolves non-determinism in PLTS:
For a PLTS P = (S, Act, Steps)

$$\sigma : execs^*(P) \rightarrow SubDistr(Act \times Distr(S)) \text{ such that}$$

$$\sigma(\alpha) \in SubDistr(Steps(\alpha(\bot))) \text{ for each } \alpha \in execs^*(P)$$

A Dirac scheduler selects exactly *one* branch:

$$\forall \alpha \in execs^*(P) : \sigma(\alpha) = 0 \text{ or } \sigma(\alpha) = \delta_{(a,\mu)} \text{ s.t. } (a, \mu) \in Steps(\alpha(\bot))$$

The probability to reach state $\alpha a s$ from state α is given by

$$\mu_{\sigma(\alpha)}(a, s) = \begin{cases} \mu(s) & \text{if } \sigma(\alpha) = \delta_{(a,\mu)} \\ 0 & \text{if } \sigma(\alpha) = 0 \end{cases}$$

Probability Measure

A probability measure on cones of executions can be defined using a standard Borel space construction:

$$C_\alpha = \{\alpha' \in execs(S) : \alpha \leqslant \alpha'\}$$

Probability measure m_{σ,s_0}:

$$m_{\sigma,\alpha}(C_\alpha) = \begin{cases} 1 & \text{if } \alpha = s_0 \\ \\ \prod_{i=1}^{n} \mu_{\sigma(\alpha_{i-1})}(a_i, s_i) & \text{if } \alpha = s_0 a_1 s_1 a_2 \ldots a_n s_n \text{ and } n \geq 1 \end{cases}$$

Weak probabilistic transition

Given PLTS S and state s_0 there exists a *weak transition* from state s_0 to μ

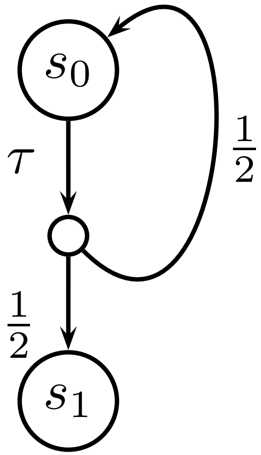
$$s_0 \Longrightarrow \mu$$

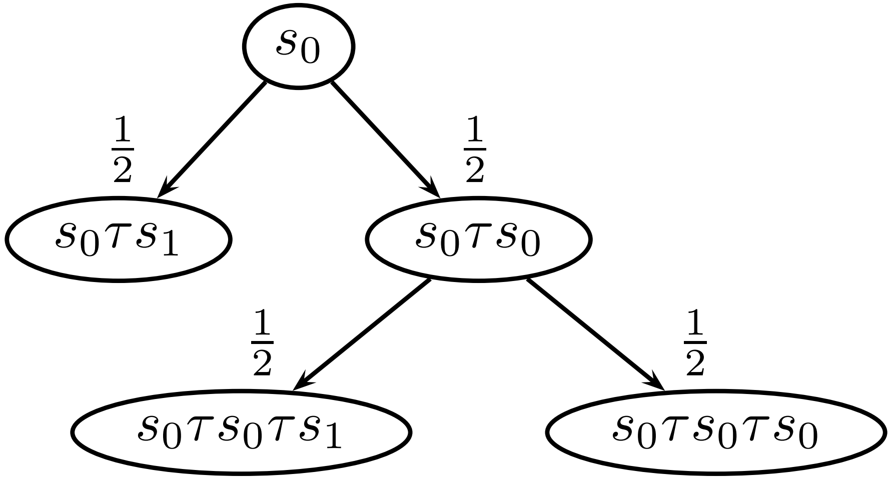
iff there exists a Dirac scheduler such that m_{σ,s_0} satisfies the following conditions

- $m_{\sigma,s_0}(execs^*(S)) = 1$
- $\forall \alpha \in execs^*(S).m_{\sigma,s_0}(\alpha) > 0 \implies trace(\alpha) = \varepsilon$
- $\forall q.\mu(q) = m_{\sigma,s_0}(\{\alpha \in execs^*(S) : \alpha(\perp) = q\})$

M&&T

Example





Probabilistic automaton with cycle

Weak transition generated by σ_2

With σ_2 we obtain $s_0 \Longrightarrow \nu_2$ s.t.:

$$\nu_2(s_0) = \frac{1}{2}.\frac{1}{2} = \frac{1}{4}$$
$$\nu_2(s_1) = \frac{1}{2} + \frac{1}{2}.\frac{1}{2} = \frac{3}{4}$$

Probabilistic CCS

A language to denote probabilistic automata

The syntax of PCCS [Baier, 1998, Ab. Th.]

$$p ::= nil \mid X \mid a.\bigoplus_{i\in I}[\lambda_i]p_i \mid p_1 + p_2 \mid p_1 \mid p_2 \mid p\backslash L \mid p[\ell]$$

Example:

$$p = \tau.([\frac{1}{2}]p \oplus [\frac{1}{2}].nil)$$

Replacing action prefix by probabilistic action prefix

PCCS operational semantics

Most interesting rules:

$$(\textsc{pref}) \qquad a.\bigoplus_{i\in I}[\lambda_i]p_i \xrightarrow{a} \mu \quad \mu(p) = \sum_{i\in I:\ p_i=p} \lambda_i$$

and

$$(\textsc{syn}) \qquad \frac{p_1 \xrightarrow{a} \mu_1 \ p_2 \xrightarrow{\overline{a}} \mu_2}{p_1 \mid p_2 \xrightarrow{\tau} \mu} \ a \neq \tau \ \text{ and } \mu(p) = \begin{cases} \mu_1(p_1')\mu_2(p_2') \text{ if } p = p_1' \mid p_2' \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

PCCS Observational semantics: Basic Observables

Let p be a probabilistic process and a an external action then the *Basic probabilistic observable* associated to p and a is denoted by:

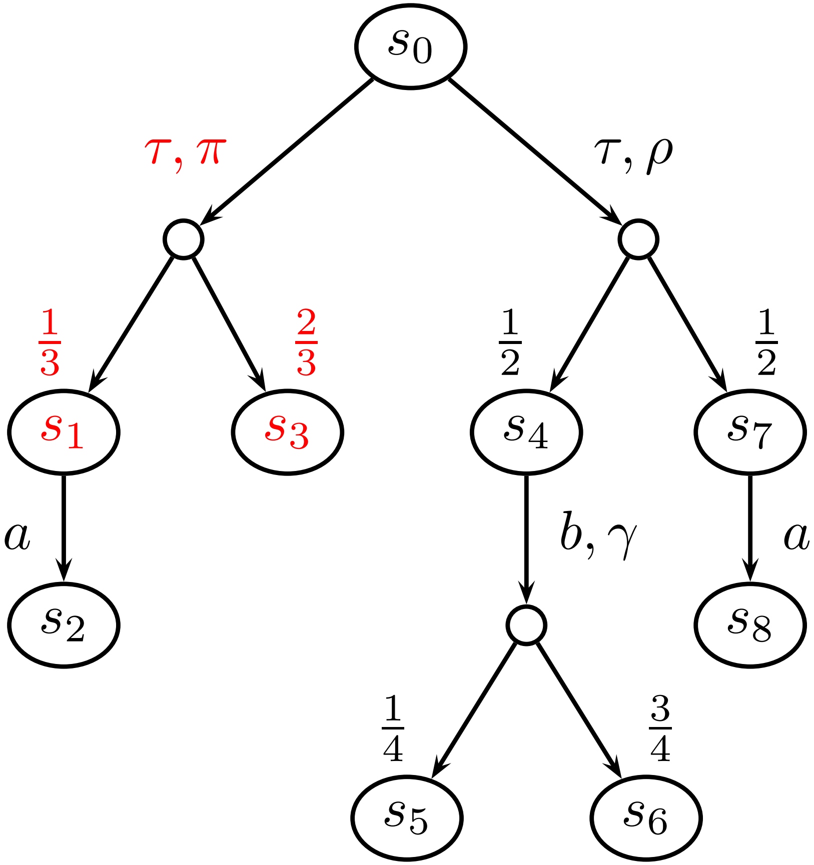
$$\{p \Downarrow_a\}$$

and defined as:

$$\left\{ \sum_{p' : p' \xrightarrow{a}} \mu(p') \text{ such that } p \Longrightarrow \mu \right\}$$

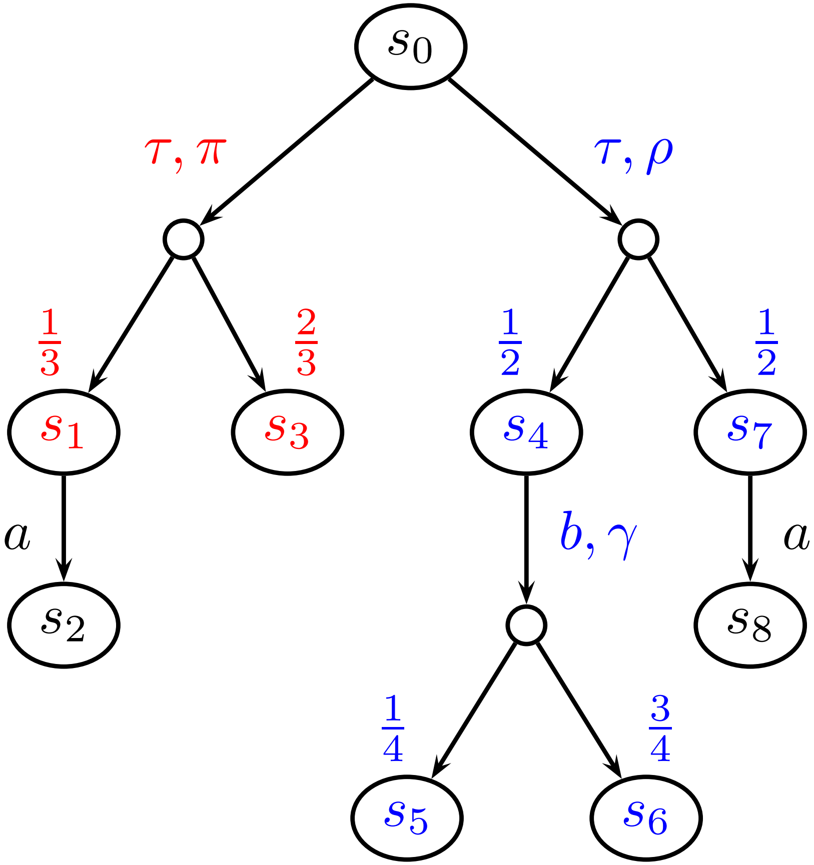
So, $\{p \Downarrow_a\}$ is the set of all probabilities of *initial* communication along channel a for process p.

Example: Prob. Basic Observable



- $\sigma_1(s_0) = \delta_{(\tau,\pi)}$,
  $\sigma_1(s_0\tau s_1) = 0$,
  $\sigma_1(s_0\tau s_3) = 0$
  $\sum_{p'.p' \xrightarrow{a}} \mu(p') = \frac{1}{3}$

Example: Prob. Basic Observable



- $\sigma_1(s_0) = \delta_{(\tau,\pi)}$,
  $\sigma_1(s_0 \tau s_1) = 0$,
  $\sigma_1(s_0 \tau s_3) = 0$
  $\sum_{p'.p' \xrightarrow{a}} \mu(p') = \frac{1}{3}$
- $\sigma_2(s_0) = \delta_{(\tau,\rho)}$,
  $\sigma_2(s_0 \tau s_4) = \delta_{(b,\gamma)}$,
  $\sigma_2(s_0 \tau s_7) = 0$
  $\sum_{p'.p' \xrightarrow{a}} \mu(p') = \frac{1}{2}$
- $\sigma_3(s_0) = 0$
  $\sum_{p'.p' \xrightarrow{a}} \mu(p') = 0$

$$\{p \Downarrow_a\} = \{0, \tfrac{1}{3}, \tfrac{1}{2}\}$$

Basic Observation Pre-order

Let p and q be two probabilistic processes:

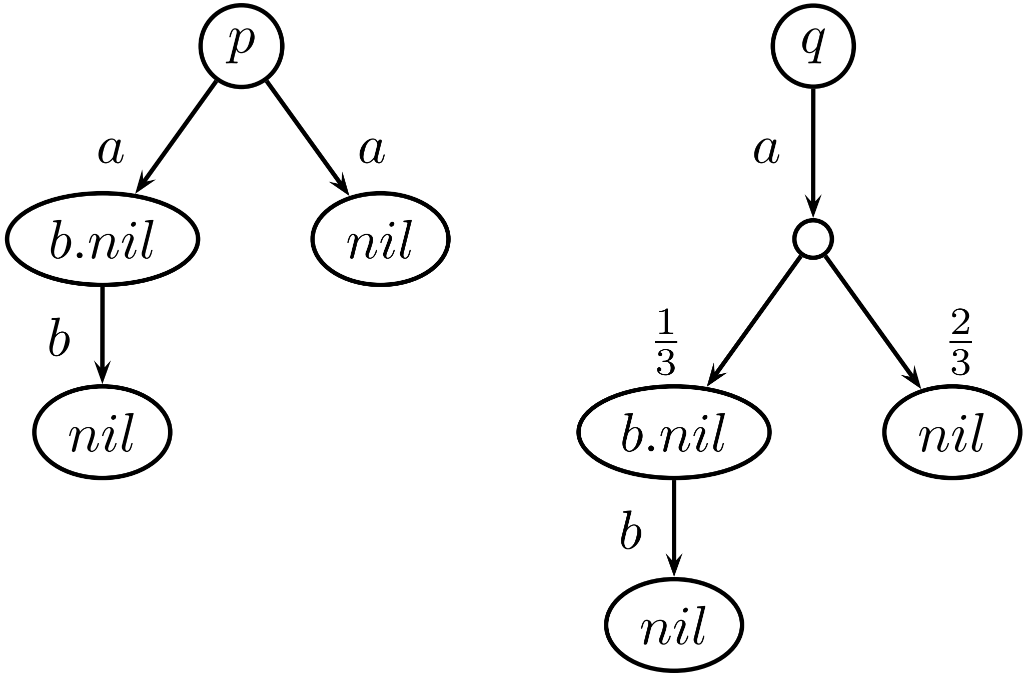
$$p \preccurlyeq_{\mathcal{A}} q \iff \forall a \in \mathcal{A} : \forall y \in \{p \Downarrow_a\}\ \exists y' \in \{q \Downarrow_a\} \text{ s.t. } y \leq y'$$

We can now define a contextual pre-order:

$$p \preccurlyeq^c_{\mathcal{A}} q \iff C[p] \preccurlyeq_{\mathcal{A}} C[q] \quad \forall \text{ context } C$$

The congruence $\preccurlyeq^c_{\mathcal{A}}$ determines an observational semantics for PCCS.

Example



It is easy to see that $\{p \Downarrow_a\} = \{q \Downarrow_a\} = \{1\}$ and $\{p \Downarrow_b\} = \{q \Downarrow_b\} = \{0\}$.
For context $C[-] = (- \mid \bar{a}.nil)$ we obtain:

$$\{C[p] \Downarrow_b\} = \{0, 1\} \text{ and } \{C[q] \Downarrow_b\} = \{0, 1/3\}.$$

Furthermore, $C[q] \preccurlyeq_b C[p]$ but $C[p] \not\preccurlyeq_b C[q]$

Probabilistic Testing pre-order

- Probabilistic test:
  - PCCS process over $\mathcal{N} \cup \{w\}$ and $w \notin \mathcal{N}$.
  - w indicates success
  - Test communicates with process-under-test by means of asynchronous parallel composition
- Expected outcomes of tests

$$\begin{array}{ll}\Omega(p,t) = & \Big\{\omega_{p|t}(\sigma) : \ \sigma \text{ Dirac scheduler such that} \\ & m_{\sigma,p|t}(\{\alpha \in \mathit{execs}^* : \mathit{trace}(\alpha) = \varepsilon\}) = 1\Big\} \\ \text{where} & \omega_{p|t}(\sigma) = m_{\sigma,p|t}(\{\alpha \in \mathit{execs}^* : \alpha(\bot) \xrightarrow{w} \})\end{array}$$

For each Dirac scheduler the set contains the probability of success to pass the test.

- Probabilistic weak may pre-order $\sqsubseteq_m$:

$$p \sqsubseteq_m q \iff \forall \text{ test } t : \forall \omega \in \Omega(p,t)\ \exists \omega' \in \Omega(q,t) \text{ s.t. } \omega \leq \omega'$$

Results

- Probabilistic weak may testing $\sqsubseteq_m$ is a congruence over PCCS processes.
- Observation congruence $\preccurlyeq^c_{\mathcal{A}}$ coincides with $\sqsubseteq_m$.
- Conjecture: A strong version of our $\sqsubseteq_m$ coincides with may pre-order of Jonnson and Wang Yi (2000/2002). Moreover, our $\sqsubseteq_m$ coincides with Wang Yi and Larsen (1992).
- The relation to probabilistic testing by Segala (1996) is somewhat more involving. For:
  - finitary processes (finite state and finite branching),
  - considering only one success action,
  - considering only Dirac schedulers,

  our weak testing pre-order coincides with that of Segala.

$\sqsubseteq_m$ is a congruence over PCCS

Let p and q be PCCS processes then

$$\forall C. p \sqsubseteq_m q \implies C[p] \sqsubseteq_m C[q]$$

Proof:

It is shown that $\sqsubseteq_m$ is preserved by each operator of PCCS.

$\preccurlyeq^c_{\mathcal{A}}$ coincides with $\sqsubseteq_m$

Let p and q be PCCS processes then $p \sqsubseteq_m q \equiv p \preccurlyeq^c_{\mathcal{A}} q$
Proof outline:

- We first show that for p, test t and name f not used in p or t

$$\Omega(p,t) = \{(p \mid t[f/w]) \Downarrow_f\}$$

  Renaming does not affect the interaction between p and t.

- Moreover, for p a process and $a \in \mathcal{A}$ we can show that:
  - $\{p \Downarrow_a\} \subseteq \Omega(p, \bar{a}.w.nil)$
  - for each $\omega \in \Omega(p, \bar{a}.w.nil)$ there is $y \in \{p \Downarrow_a\}$ such that $\omega \leq y$.

  Proofs are given by showing that proper schedulers can be constructed to obtain the results.

- Finally, for all p and q we show $p \preccurlyeq^c_{\mathcal{A}} q \implies p \sqsubseteq_m q \implies p \preccurlyeq_{\mathcal{A}} q$

This suffices to show the main theorem.

strong $\preccurlyeq^c_{\mathcal{A}}$ cioncides with may testing by Jonnson and Wang Yi

Jonnson and Wang Yi's probabilistic testing

- processes and tests are *finite probabilistic automata*
- tests are *finite* trees where leaves may be labelled with success
- processes and tests are not able to perform internal actions

$$\mathcal{P} \sqsubseteq^{JW}_m \mathcal{Q} \iff \forall \text{ test } \mathcal{T}: \quad \max \Omega^{JW}(\mathcal{P} \parallel \mathcal{T}) \leq \max \Omega^{JW}(\mathcal{Q} \parallel \mathcal{T})$$

It is easy to see that a strong version of our probabilistic testing corresponds to that of Jonnson and Wang Yi.

Relation of $\preccurlyeq^{c}_{\mathcal{A}}$ with prob. testing by Segala

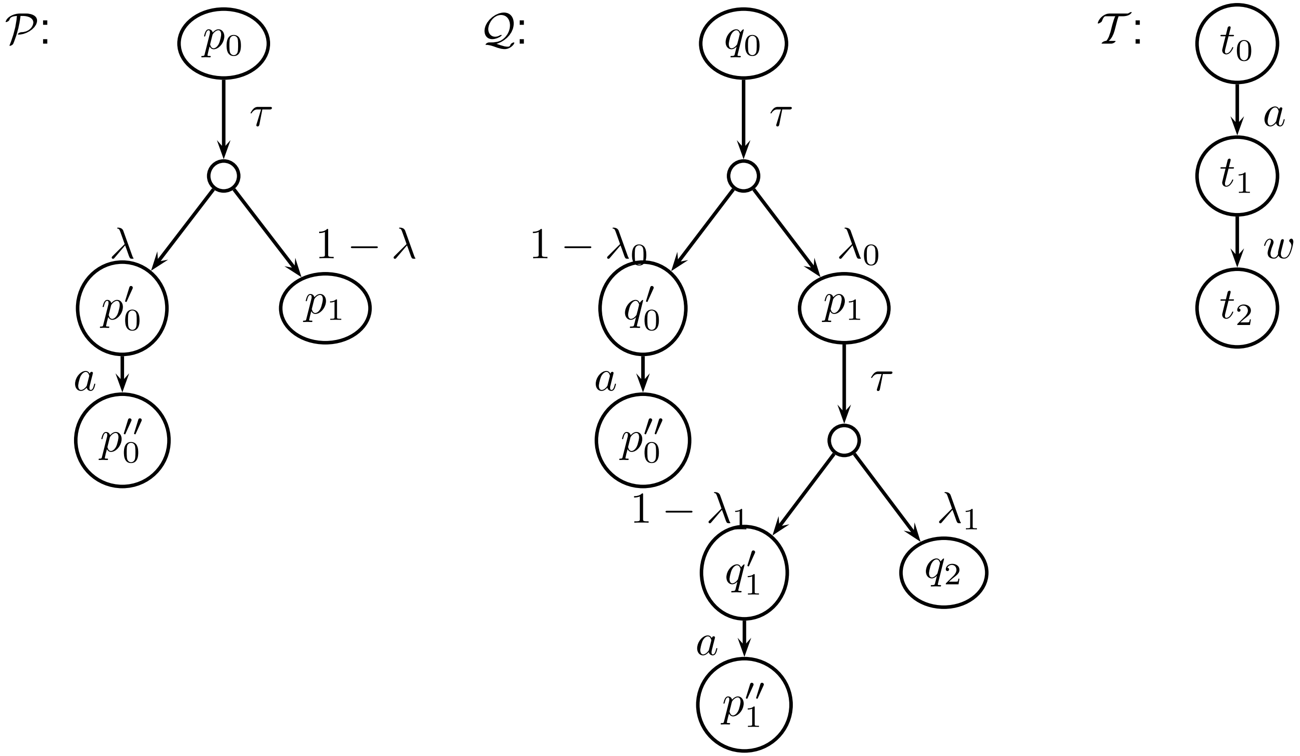
Segala’s probabilistic testing considers

- General schedulers,
- maximal schedulers, but does not require that the total probability of finite executions is 1,
- multiple kinds of success actions, rather than a single kind w

We address each issue:

- Segala shows that in practice general schedulers do not add expressivity w.r.t. Dirac schedulers.
- Next slide
- Next slide

Maximal schedulers vs. stopping schedulers

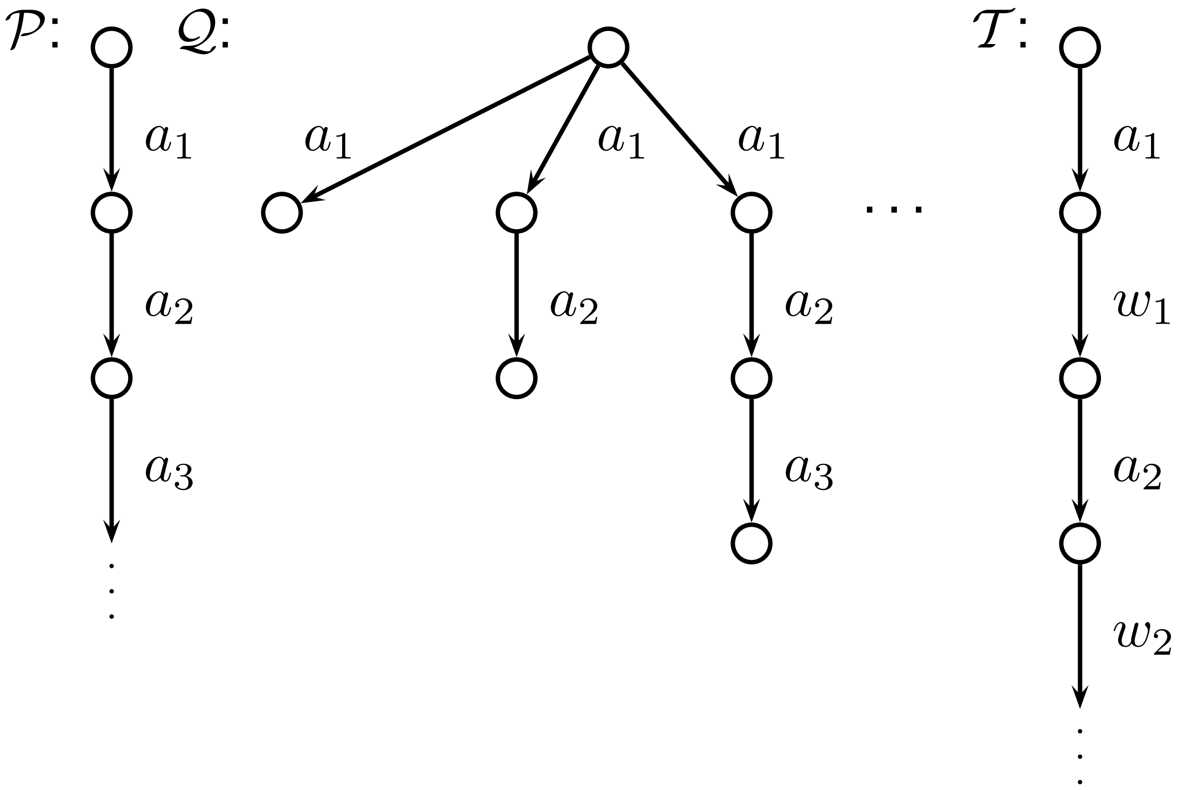


Assume λ_i s.t. $\prod_{i=0}^{\infty} \lambda_i = 1 - \lambda > 0$. We have for stopping schedulers:

$\Omega(\mathcal{P} \parallel \mathcal{T}) = \{0, \lambda\}$ and $\Omega(\mathcal{Q} \parallel \mathcal{T}) = \{0, 1-\lambda_0, 1-\lambda_0\lambda_1, 1-\lambda_0\lambda_1\lambda_2, \ldots\}$

But for maximal sched. $\Omega(\mathcal{P} \parallel \mathcal{T}) = \Omega(\mathcal{Q} \parallel \mathcal{T}) = \{0, \lambda\}$.

Multiple success actions



Test $\mathcal{T}$ can discriminate between $\mathcal{P}$ and $\mathcal{Q}$.

Conclusions and Future work

- Preliminary but promising results for Probabilistic Basic Observables
- Extension of Prob. Basic Obs. to probabilistic (fair) must pre-order.
- Further comparison also with very recent related work by a.o. Deng, Hennessy, Morgan et al.