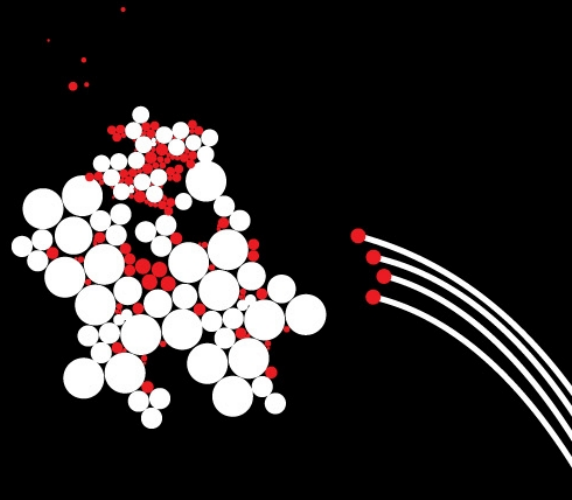
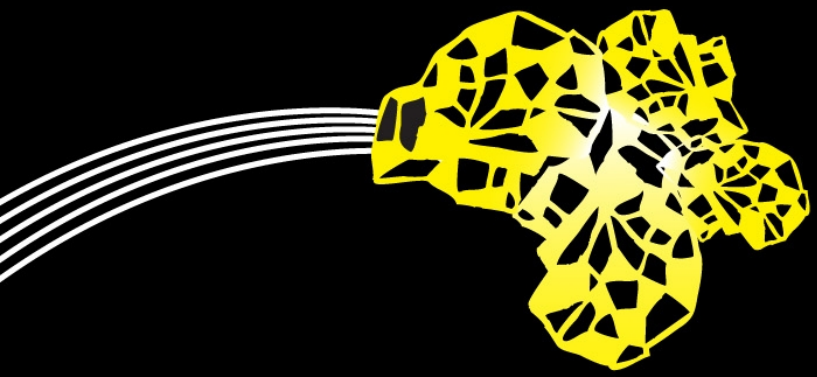
UNIVERSITY OF TWENTE.



ioco Theory for Probabilistic Automata

M. Gerhold, M.I.A. Stoelinga

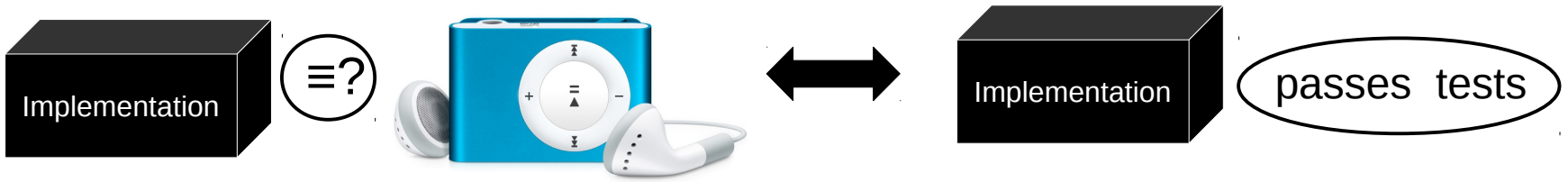
10th MBT Workshop, London 2015



OUTLINE

1. ioco Theory
2. pQTS and pioco
3. Testing with pQTS

Ideally: Impl. conforms to Spec iff Impl. passes test suite



ioco Theory

ioco = Input / Output Conformance

Tretmans (1996) Test Generations with Inputs, Outputs, and Repetitive Quiescence

Tretmans (2008) Model Based Testing with labeled transition Systems

ioco Theory

hioco:

- Osch (2006); *Hybrid input-output conformance and test generation*

mioco:

- Heerink (1998); *Ins and Outs in Refusal Testing*

(r)tioco:

- Brandan Briones, Brinksma (2005); *A test generation framework for quiescent real-time systems*
- Krichen, Tripakis (2004); *Black Box Conformance Testing for real-time systems*
- Hessel, Larsen, Mikucionis, Nielsen, Petersson, Skou (2008); *Online Testing of real-time systems using Uppaal*

sioco:

- Tretmans, Frantzen, Willemse (2004); *A symbolic Framework for MBT*

uioco:

- van der Bijl, Rensink (2003); *Compositional Testing with ioco*

ioco Theory

- Stokkink, Timmer, Stoelinga (2012); *Talking Quiescence a rigorous theory that supports parallel composition, action hiding and determinisation*
- Stokkink, Timmer, Stoelinga (2013); *Divergent Quiescent Transition Systems*
- Brinksma, Timmer, Stoelinga *(2014); An Introduction to Model-Based Testing*

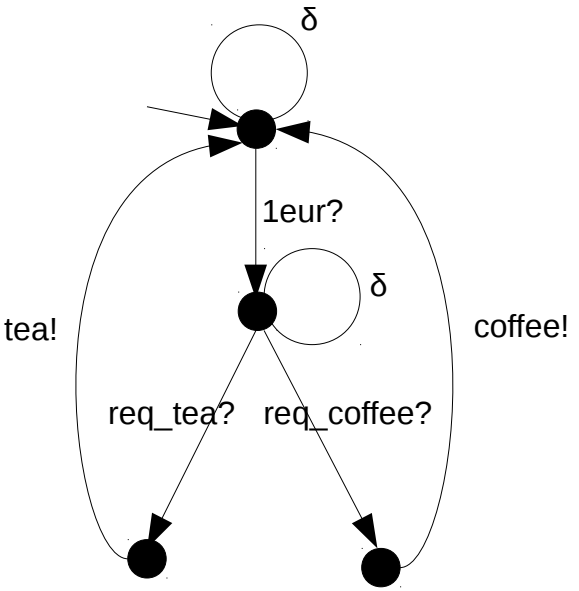
ioco Theory

Definition (QTS): A quiescent transition system is a five tuple

$$A = \langle S, s_0, L_I, L_O^{\delta}, \rightarrow \rangle$$

with

S	a finite set of states
s_0	the starting state
L_I	the input alphabet
L_O^{δ}	the output alphabet + quiescence
$\rightarrow$	the transition relation



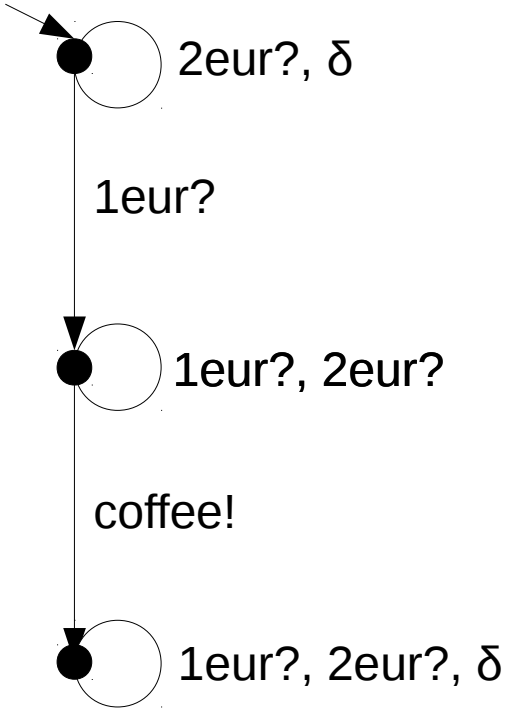
Important: Quiescent output action δ

ioco Theory

Definition (out): For a given QTS *A* and trace σ we write

$$\mathrm{out}_A(\sigma) = \mathrm{after}(\sigma) \cap L_O^{\delta}$$

to denote the set of all output actions as well as quiescence after trace σ.



- out(ε) = {δ}
- out(1eur?) = {coffee!}
- out(1eur? 2eur?) = {coffee!}
- out(1eur? coffee!) = {δ}
- out(2eur?) = {δ}
- out(δ) = {δ}
- out(coffee) = ∅

ioco Theory

Definition (ioco): Let *Imp* be an implementation and *Spec* a specification. Furthermore let Imp be input-enabled, then

$$Imp \subseteq_{\text{ioco}} Spec \Leftrightarrow \forall \sigma \in \text{traces}(\text{Spec}) : \text{out}_{\text{Imp}}(\sigma) \subseteq \text{out}_{\text{Spec}}(\sigma)$$

Intuition: *Imp* ioco *Spec* iff for all traces of Spec:

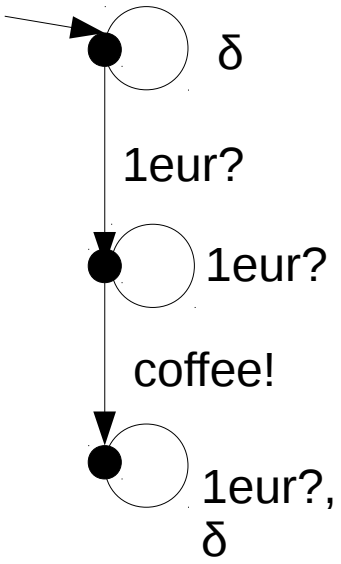
If *Imp* gives output x! After trace σ, then so can *Spec*

Some examples

ioco Theory

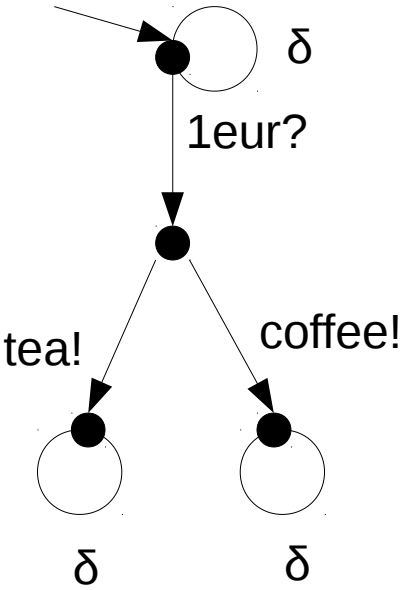
$$i \text{ ioco } s =_{\text{def}} \forall \sigma \in \text{traces}(\text{Spec}) : \text{out}_{\text{Imp}}(\sigma) \subseteq \text{out}_{\text{Spec}}(\sigma)$$

Impl:



ioco

Spec:



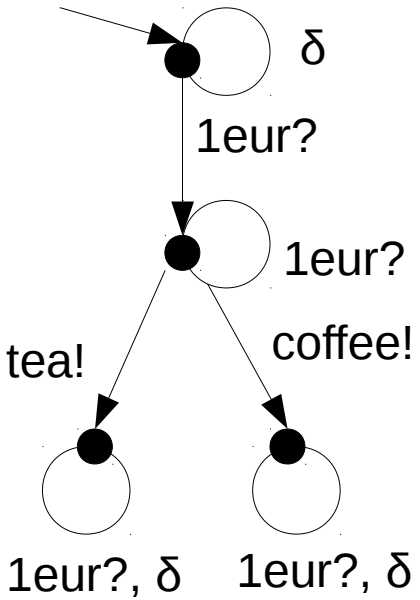
$$\text{out}_{\text{Impl}}(1\text{eur?}) = \{\text{coffee!}\} \subseteq \{\text{tea!}, \text{coffee!}\} = \text{out}_{\text{Spec}}(1\text{eur?})$$

...

ioco Theory

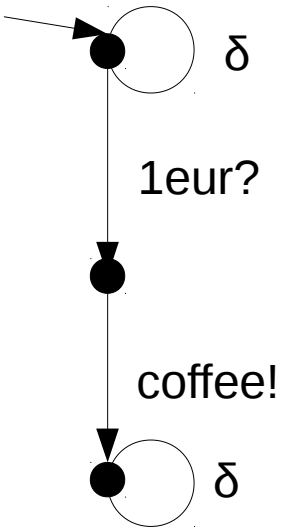
$$i \text{ ioco } s =_{\text{def}} \forall \sigma \in \text{traces}(\text{Spec}) : \text{out}_{\text{Imp}}(\sigma) \subseteq \text{out}_{\text{Spec}}(\sigma)$$

Impl:



not ioco

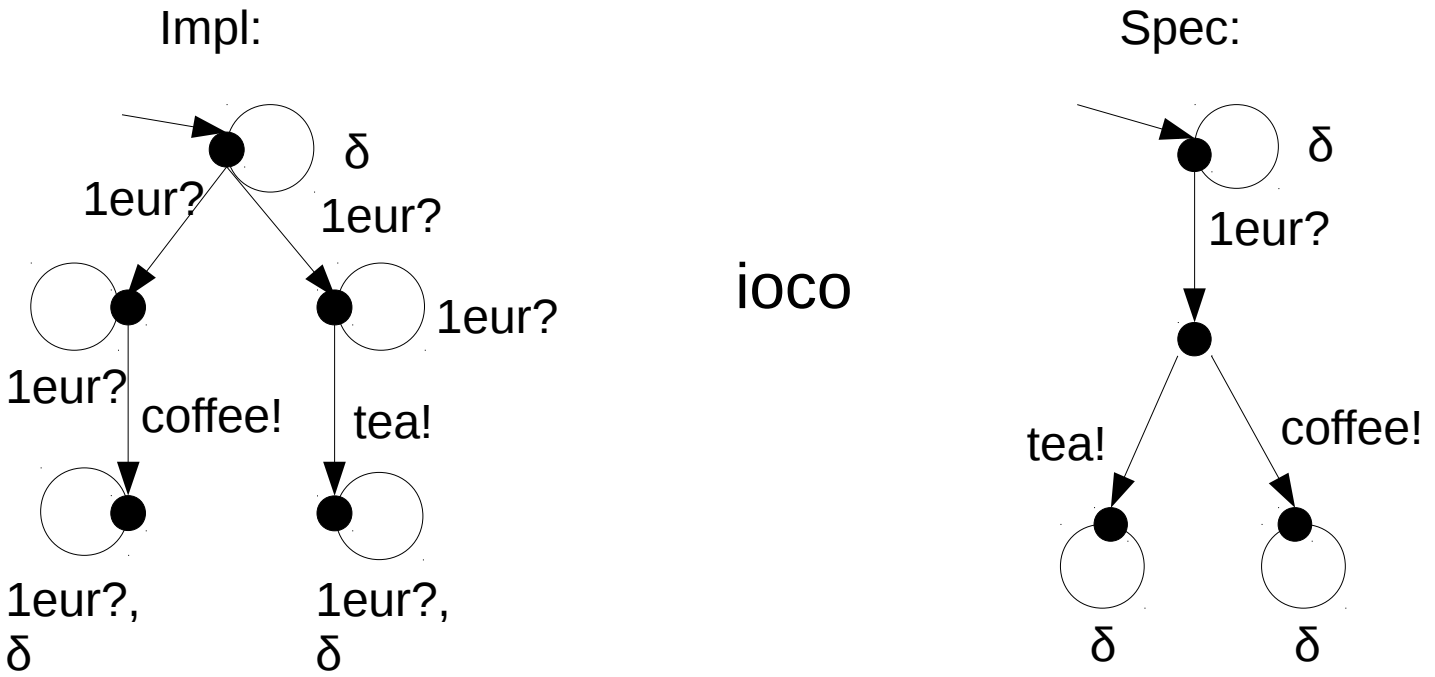
Spec:



$$\text{out}_{\text{Impl}}(1\text{eur?}) = \{\text{tea!}, \text{coffee!}\} \not\subseteq \{\text{coffee!}\} = \text{out}_{\text{Spec}}(1\text{eur?})$$

ioco Theory

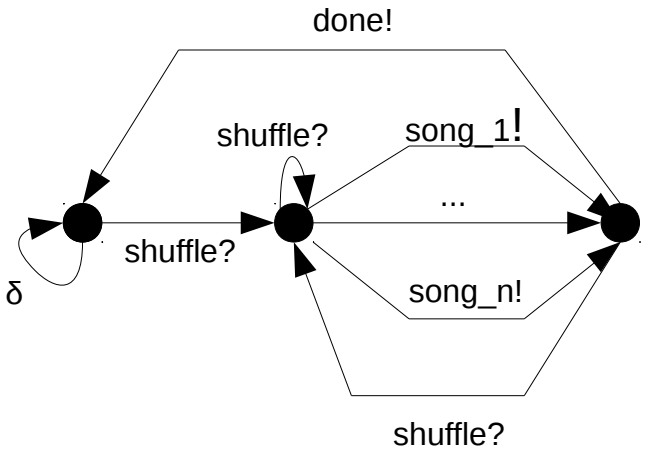
$$i \text{ ioco } s =_{\text{def}} \forall \sigma \in \text{traces}(\text{Spec}) : \text{out}_{\text{Imp}}(\sigma) \subseteq \text{out}_{\text{Spec}}(\sigma)$$



$$\text{out}_{\text{Impl}}(1\text{eur?}) = \{\text{coffee!}\} \subseteq \{\text{tea!}, \text{coffee!}\} = \text{out}_{\text{Spec}}(1\text{eur?})$$

...

Why probabilities?



- Wanted
- Existing MBT testing tools (e.g. JtorX..) already choose next steps probabilistically

pQTS and pioco

Definition (pQTS): A probabilistic quiescent transition system is a five tuple

$$A = \langle S, s_0, L_I, L_O^{\delta}, \Delta \rangle$$

with

S	a finite set of states
s_0	the starting state
L_I	the input alphabet
L_O^{δ}	the output alphabet + quiescence

$$\Delta \subseteq \{(s, \mu) \in S \times \mathrm{Distr}(L \times S)\}$$

Input reactive/output generative probabilistic transition relation

Different to: Segala (1995); *Modeling and Verification of Randomized Distributed Real-Time Systems*

There: $\Delta \subseteq \{(s, a, \mu) \in S \times L \times \mathrm{Distr}(S)\}$

pQTS and pioco

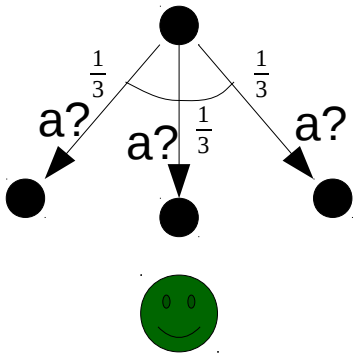
$$\Delta \subseteq \{(s,\mu) \in S \times \mathrm{Distr}(L \times S)\}$$

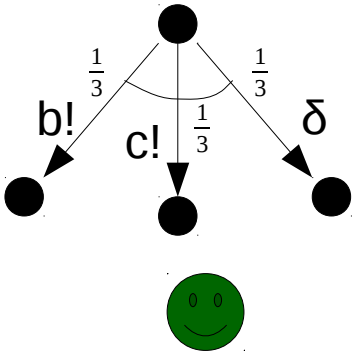
Such that for $(s,\mu) \in \Delta$, we either have a distribution only containing the same Input or a distribution containing only outputs and possibly quiescence

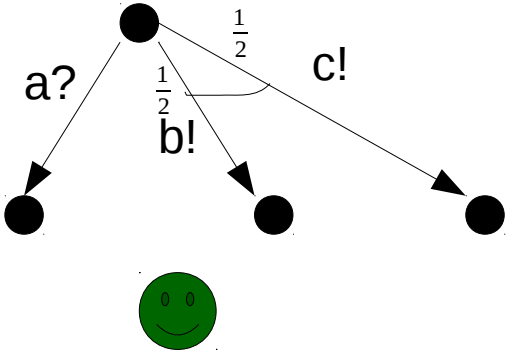
(=input reactive / output generative).

Distribution: $\mu : X \rightarrow [0,1]$ $\quad \sum_{x \in X} \mu(x) = 1$

Examples:







pQTS and pioco

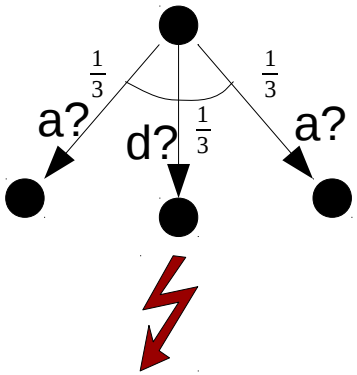
$$\Delta \subseteq \{(s,\mu) \in S \times \mathrm{Distr}(L \times S)\}$$

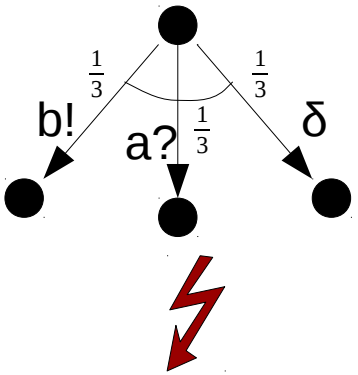
Such that for $(s,\mu) \in \Delta$, we either have a distribution only containing the same Input or a distribution containing only outputs and possibly quiescence

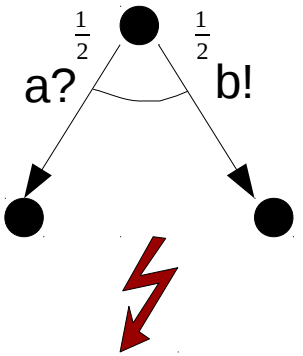
(=input reactive / output generative).

Distribution: $\mu : X \rightarrow [0,1]$ $\quad \sum_{x \in X} \mu(x) = 1$

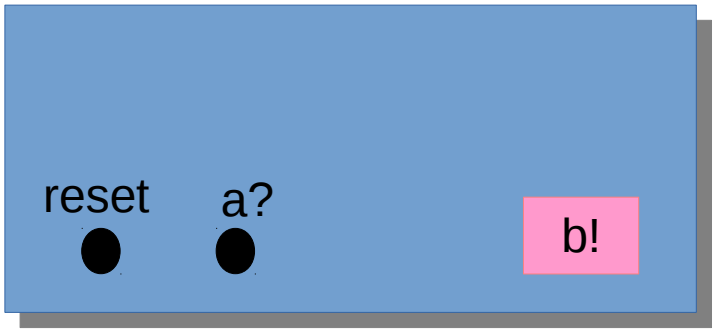
Examples:



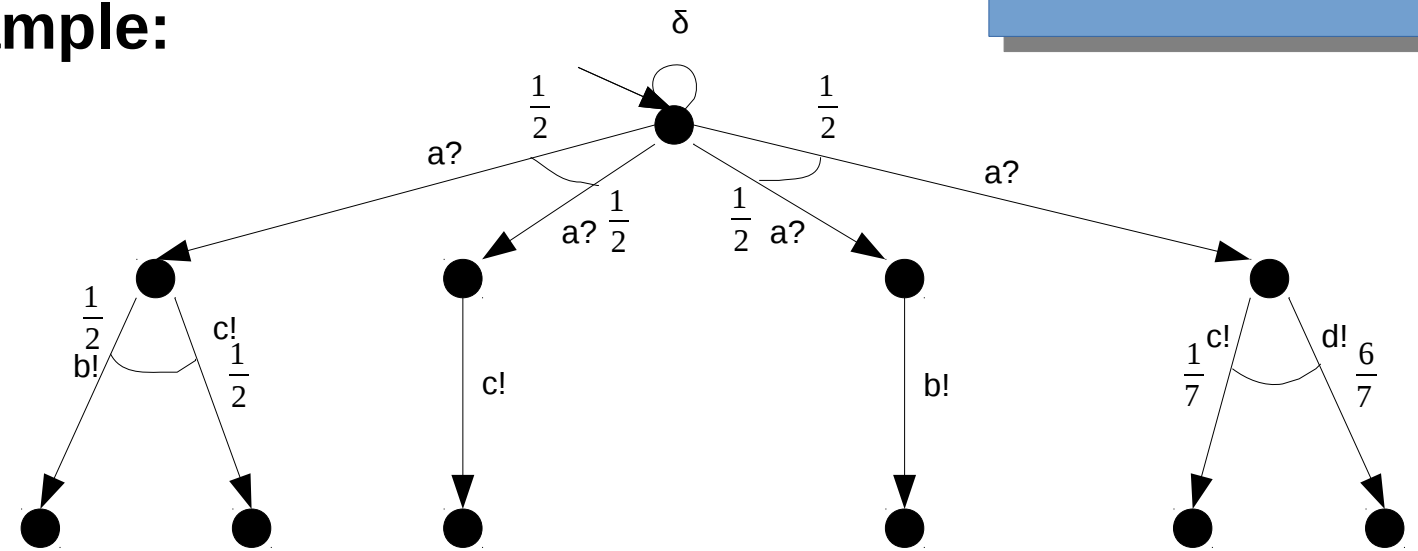




pQTS and pioco



Example:



But what is the probability to get the trace a?c! ?

pQTS and pioco

How do we resolve the remaining non-determinisms?

Definition (Adversary): An adversary *E* is a function

$$E : \mathrm{Path}(A) \rightarrow \mathrm{Distr}(\mathrm{Distr}(L \times S) \cup \mathrm{Halting})$$

such that for each finite path π, if $E(\pi)(\mu) > 0$ then $(\mathrm{last}(\pi), \mu) \in \Delta$.

Definition (Path Probability): We define the function

$$Q^E : \mathrm{Path} \rightarrow [0,1]$$

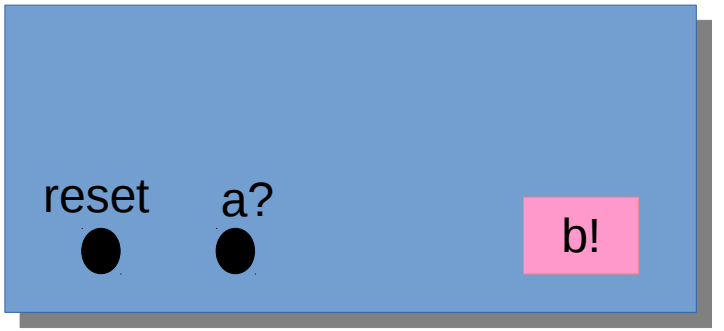
Inductively as follows:

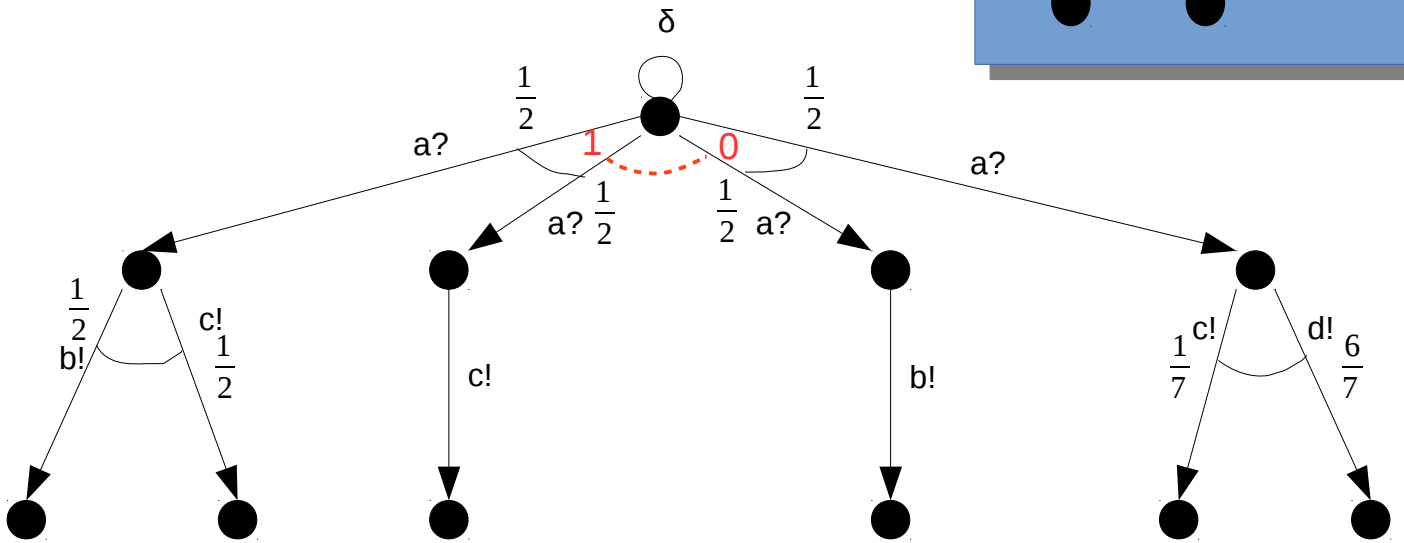
$$Q^E(s_0) = 1 \qquad Q^E(\pi \mu a s) = Q^E(\pi) E(\pi)(\mu) \mu(a, s)$$

Instead of using adversaries (functions), we use probability spaces associated to adversaries to get a probability measure.

pQTS and pioco

Example: Let E be the adversary





$$Q^E(\pi\mu\, a\, s) = Q^E(\pi)\, E(\pi)(\mu)\mu(a,s)$$

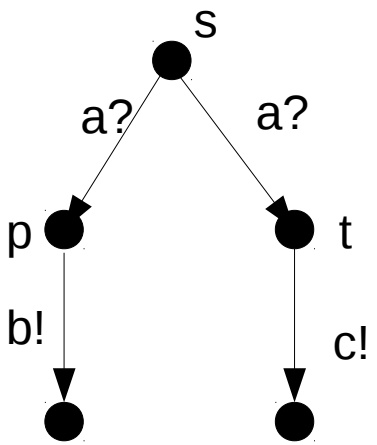
Then the probability for trace a?c! equals $\frac{3}{4}$.

Instead of using probability spaces associated to adversaries we use probability spaces associated with trace distributions.

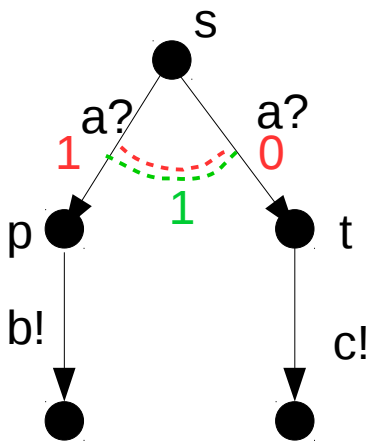
$$P_H(\sigma) = P_E(\text{trace}^{-1}(\sigma))$$

pQTS and pioco

Paths & Traces vs Adversaries & Trace Distributions

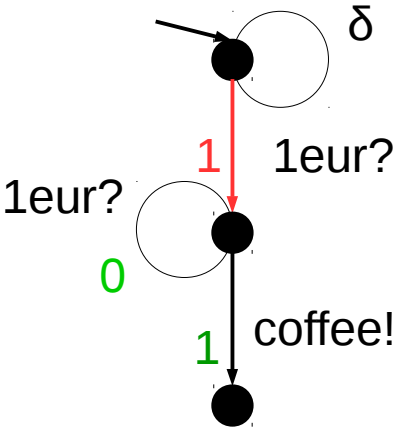


Actions after path s a?p ?
Actions after trace a? ?



Actions with positive prob. after Adversary E
Actions with positive prob. after Trace Distribution H?

pQTS and pioco



loco based on:

$$\mathrm{out}_A(\sigma) = \mathrm{after}(\sigma) \cap L_O^{\delta}$$

Pioco based on: $k \in \mathbb{N} \wedge H \in \mathrm{trd}(A, k)$

$$\mathrm{outcont}(H, A) = \{H' \in \mathrm{trd}(A, k+1) : H \subseteq_k H' \wedge \forall \sigma \in L^k L_I : P_H'(\sigma) = 0\}$$

Intuition: Actions with positive prob. after Trace distribution H?

- It prolongs a trace distribution by length 1
- Only schedules output after length k

pQTS and pioco Theory

Definition (pioco): Let *Imp* be an implementation and *Spec* be a specification. Furthermore let Imp be input enabled, then

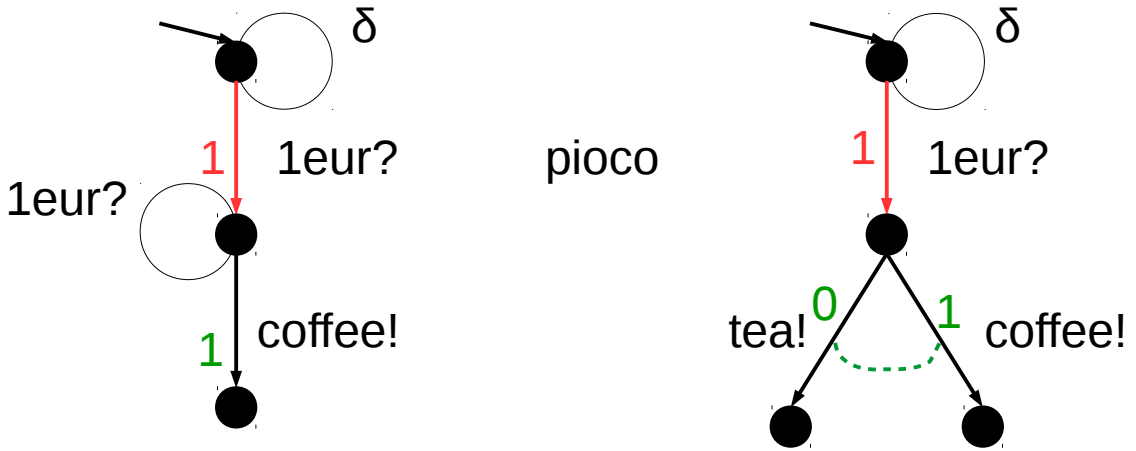
$$Imp \subseteq_{\text{pioco}} Spec \Leftrightarrow \forall\, k \in \mathbb{N}\ \forall\, H \in \text{trd}\left(Spec, k\right):$$
$$\text{outcont}\left(H, Imp\right) \subseteq \text{outcont}\left(H, Spec\right)$$

Intuition: *Imp* pioco to *Spec* iff for all trace distributions H of Spec:
If every behaviour of Impl can be imitated by Spec

Some examples

pQTS and pioco

Example: Take an adversary E of length 1 and a corresponding trace distribution H

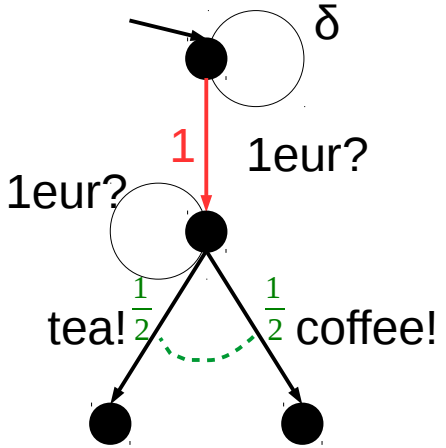


$$\forall\, k \in \mathbb{N}\ \forall\, H \in \mathrm{trd}\left(Spec, k\right):$$

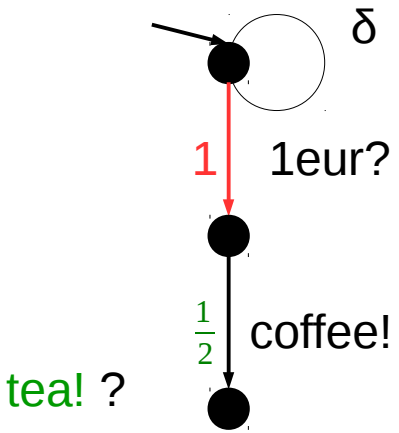
$$\mathrm{outcont}\left(H, Imp\right) \subseteq \mathrm{outcont}\left(H, Spec\right)$$

pQTS and pioco

Example: Take an adversary E of length 1 and a corresponding trace distribution H



Not pioco



Not:

$$\forall k \in \mathbb{N}\ \forall H \in \mathrm{trd}(Spec, k):$$

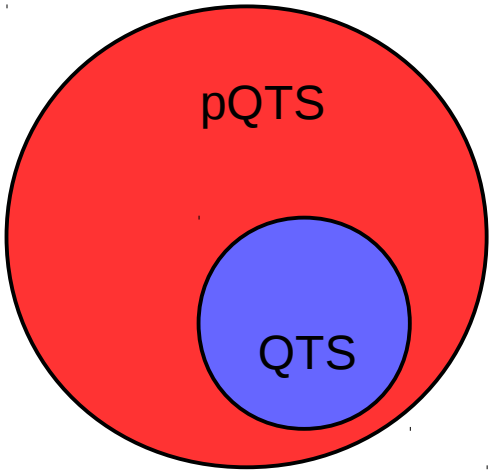
$$\mathrm{outcont}(H, Imp) \subseteq \mathrm{outcont}(H, Spec)$$

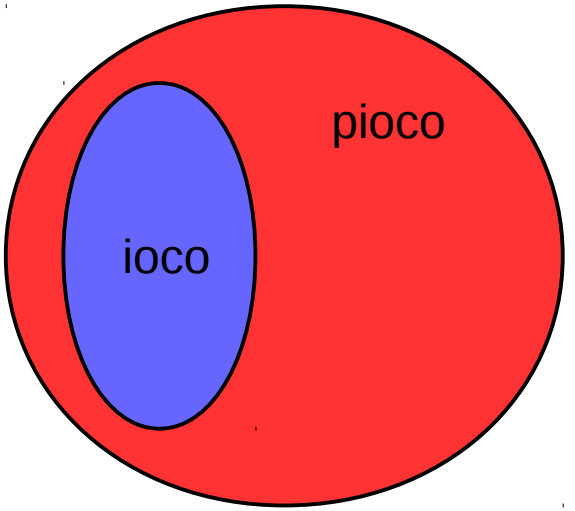
pQTS and pioco

So far we have only considered non-probabilistic QTS

Theorem: Let *Imp and Spec* be two QTS and *Imp* be input-enabled, then

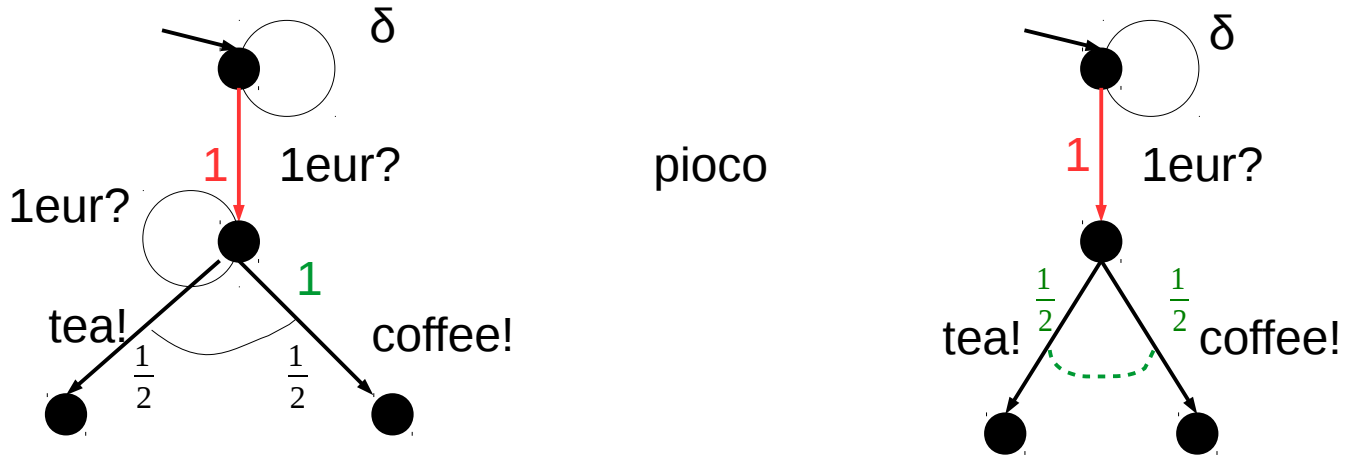
$$Imp \subseteq_{\text{ioco}} Spec \Leftrightarrow Imp \subseteq_{\text{pioco}} Spec$$





pQTS and pioco

Example: Take an adversary E of length 1 and a corresponding trace distribution H

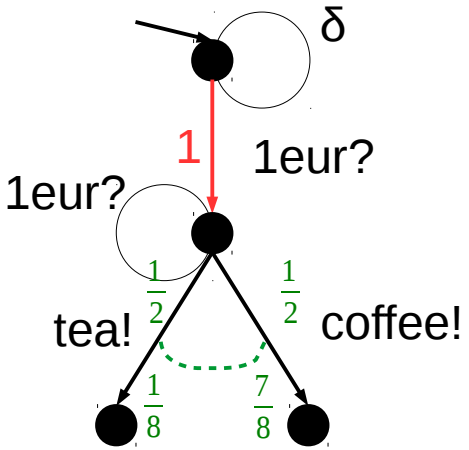


$$\forall k \in \mathbb{N}\, \forall H \in \mathrm{trd}(Spec, k):$$

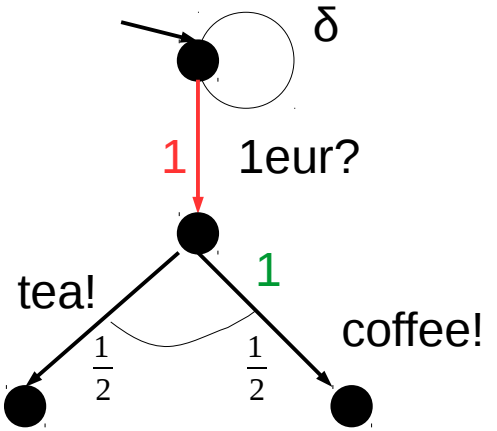
$$\mathrm{outcont}(H, Imp) \subseteq \mathrm{outcont}(H, Spec)$$

pQTS and pioco

Example: Take an adversary E of length 1 and a corresponding trace distribution H



Not pioco



Not:

$$\forall k \in \mathbb{N}\ \forall H \in \mathrm{trd}(Spec, k):$$

$$\mathrm{outcont}(H, Imp) \subseteq \mathrm{outcont}(H, Spec)$$

pQTS and pioco

Imp is always input enabled. What if *Spec* is too?

Theorem (known): Let *Imp and Spec* be two input enabled QTS, then

$$Imp \subseteq_{\text{ioco}} spec \Leftrightarrow Imp \subseteq_{\text{traces}} Spec$$

Theorem: Let *Imp and Spec* be two input enabled pQTS, then

$$Imp \subseteq_{\text{pioco}} spec \Leftrightarrow Imp \subseteq_{\text{TD}} Spec$$

pQTS and pioco

Theorem (Transitivity): Let *A, B* and *C* be three QTS and let *A* and *B* be input-enabled, then

$$A \subseteq_{\text{ioco}} B \text{ and } B \subseteq_{\text{ioco}} C \Rightarrow A \subseteq_{\text{ioco}} C$$

Theorem (Transitivity): Let *A, B* and *C* be three pQTS and let *A* and *B* be input-enabled, then

$$A \subseteq_{\text{pioco}} B \text{ and } B \subseteq_{\text{pioco}} C \Rightarrow A \subseteq_{\text{pioco}} C$$

Testing with pQTS

Verdict 1: System should not give any erroneous output (Classical ioco testing).

Verdict 2: Output frequencies should be matched as specified (Statistical testing).

Testing with pQTS

Verdict 1: Classical ioco testing

1. Given *Imp* and *Spec*
2. Derive test cases from *Spec*
3. Run test case and *Imp* in parallel composition
4. If erroneous output is detected, *fail*

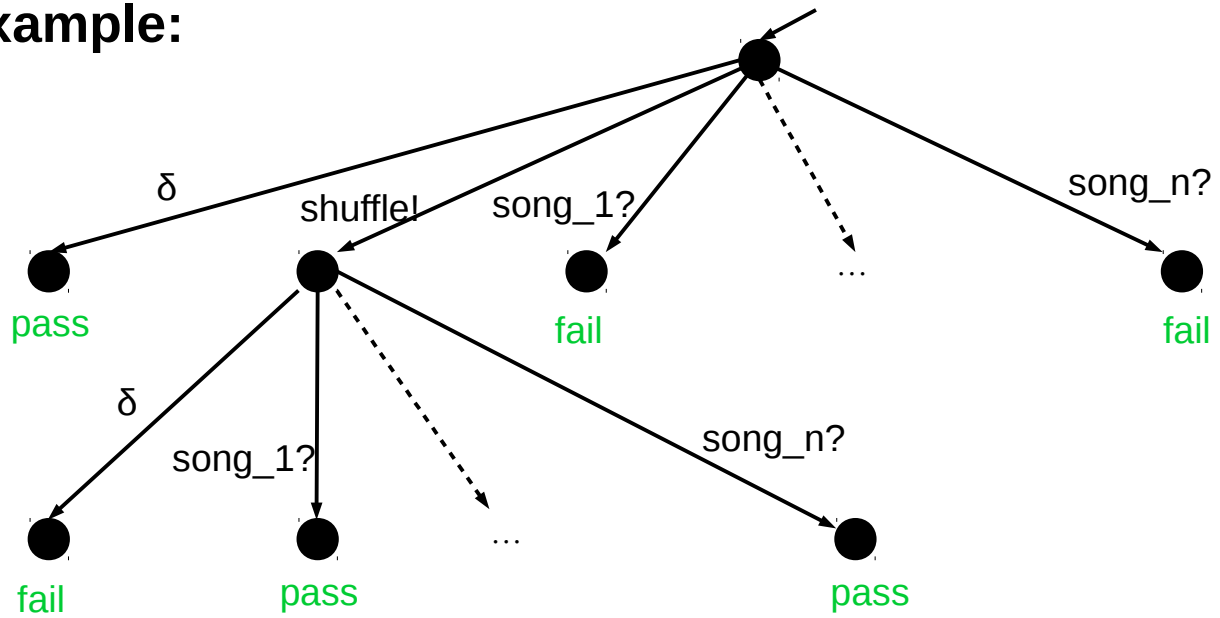
Testing with pQTS

Definition (test): A test is a pQTS of the form

$$t = \langle S, s_0, L_O, L_I \cup \{\delta\}, \Delta \rangle$$

such that t is acyclic, conntected and internally deterministic.

Example:



+ annotation

Summarize in annotated Test suite.

Testing with pQTS

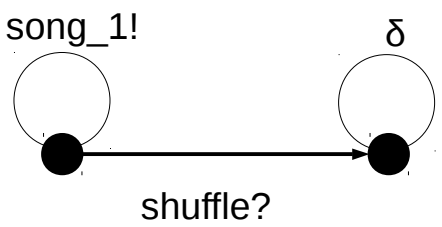
Definition (Verdict Function 1): Given a test *t*, we define the verdict function 1 $v_t : \text{pQTS} \rightarrow \{\text{pass}, \text{fail}\}$ as

$$v_1(A) = \text{pass} \text{ if } \forall \sigma \in \text{traces}(A \,||\, t) \cap \text{ctraces}(t) : a(\sigma) = \text{pass}$$

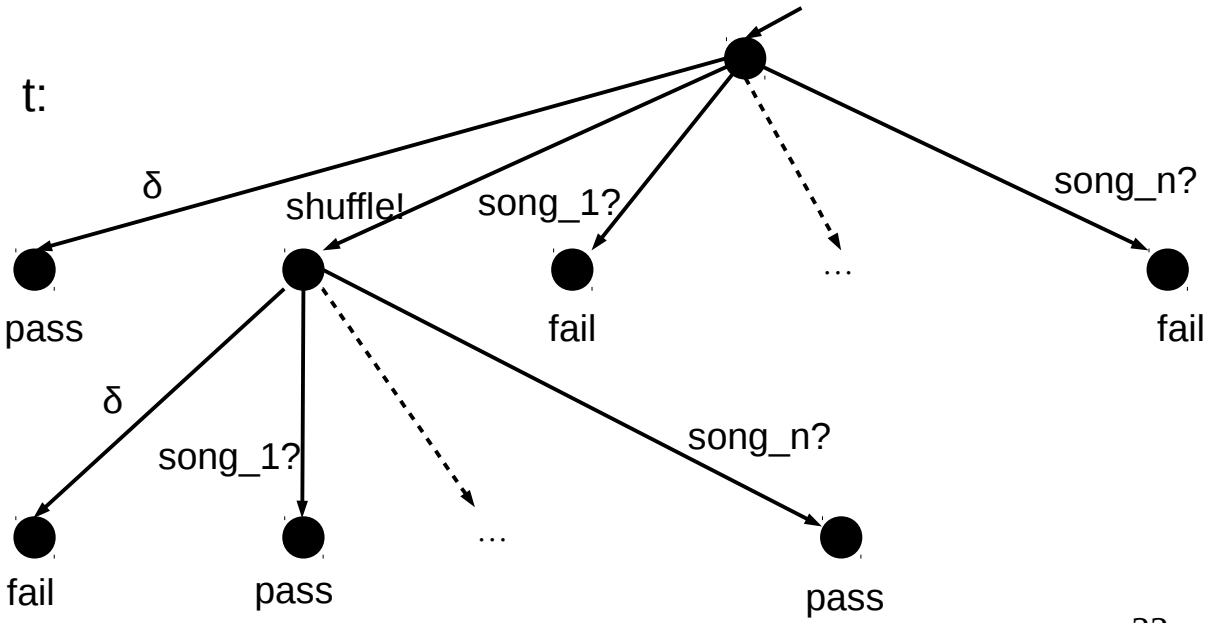
$$v_1(A) = \text{fail} \text{ otherwise}$$

Example:

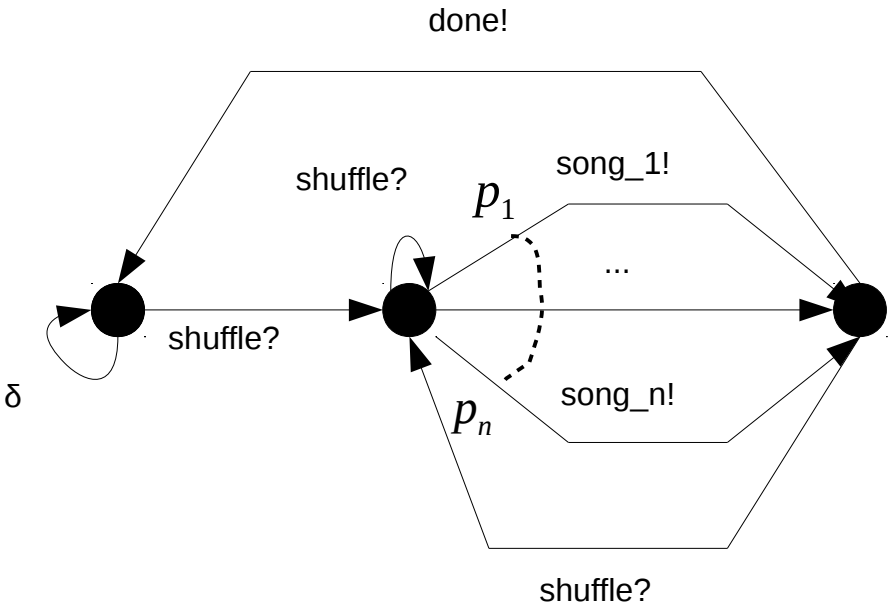
Impl:

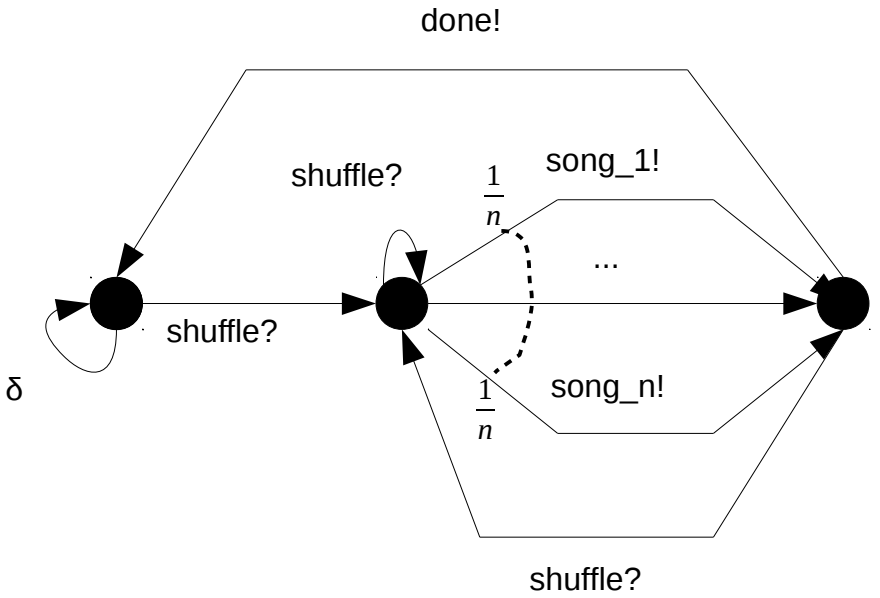


t:



Testing with pQTS





Testing with pQTS

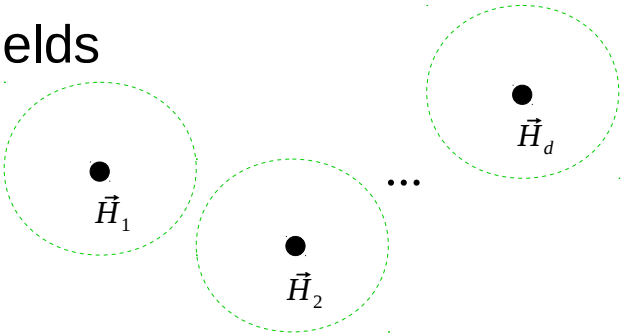
Verdict 2: (Statistical Testing)

1. Given *Imp* and *Spec*
2. Derive (reuse) test cases from *Spec*
3. We run test case and *Imp* in parallel composition
4. Hypothesis Testing and Sampling

Testing with pQTS

Statistical Testing

- We call $O \in \left(L^k\right)^m$ a sample of depth *k* and width *m*
- Therefore assume *m* different trace distributions $\vec{H} = H_1, \ldots, H_m$
- Treat sample as Bernoulli experiment, with success in *i* if $\sigma = \sigma_i$
- Define expected value of σ as $E^{\vec{H}}(\sigma) = \frac{1}{m} \sum_{i=1}^{m} P_{H_i}(\sigma)$
- Then $E^{\vec{H}}$ becomes a distribution
- For given level of significance, are frequencies of *O* in some acceptable distance of expected frequency?
- Unify over all those balls, yields $\mathrm{Obs}(A, \alpha)$



pQTS and pioco

Theorem: Given level of significance α, we know that for all pQTS *A* and *B*

$$\mathrm{Obs}(A,\alpha) \subseteq \mathrm{Obs}(B,\alpha) \Leftrightarrow A \subseteq_{\mathrm{TD}} B$$

M. Stoelinga, F. Vaandrager (2003); A Testing Scenario for probabilistic Automata

Definition (Verdict Function 2): Given a test *t,* a level of significance α ϵ [0,1] and $O \in (\mathrm{Traces}(Impl, k))^m$ we define the verdict function 2 $v_t^{\alpha} : \mathrm{pQTS} \rightarrow \{\mathrm{pass}, \mathrm{fail}\}$ as

$$v_2^{\alpha}(A) = \mathrm{pass} \text{ if } O \in \mathrm{Obs}(Spec)$$

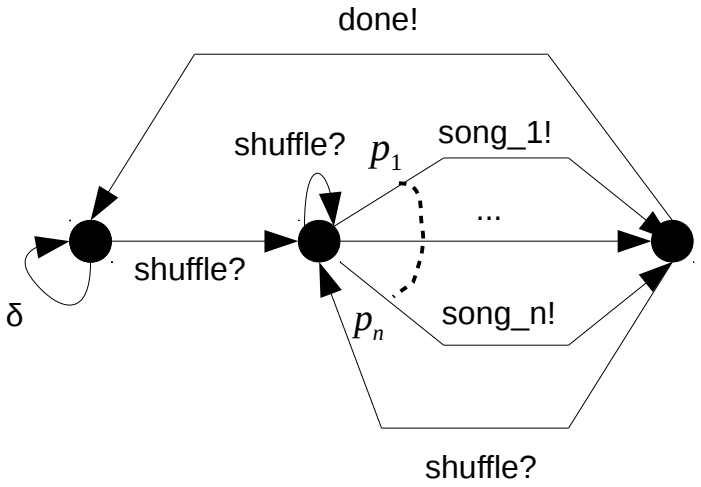
$$v_2^{\alpha}(A) = \mathrm{fail} \text{ otherwise}$$

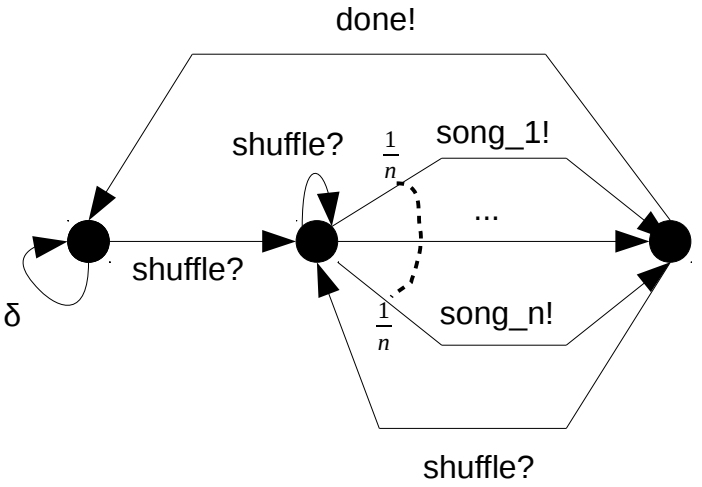
Testing with pQTS

Definition (Overall Verdict): Given a test *t,* a level of significance α ϵ [0,1] and $O \in (\text{Traces}(Impl, k))^m$, we define the verdict function $v : \text{pQTS} \rightarrow \{\text{pass}, \text{fail}\}$ as

$$v(A) = \text{pass} \text{ if } v_1 = \text{pass} \wedge v_2^{\alpha} = \text{pass}$$

$$v(A) = \text{fail} \text{ otherwise}$$





Conclusion

- Definition *pQTS*
- Definition *pioco*

Future Work

- Enable internal actions (pDQTS)
- Prove soundness and (theoretical) completeness
- Use more advanced statistical methods
- Case studies

Thank you for your attention!