Testing Nondeterministic and Probabilistic Processes

Matthew Hennessy

(joint work with Yuxin Deng, Rocco DeNicola, Rob van Glabbeek, Carroll Morgan, Chenyi Zhang)

BASICS, Shanghai October 09

Outline

Background  why bother ?

Testing theory

Testing nondeterministic processes

Testing Probabilistic and nondeterministic processes
Background

Goal: Specification and proof methodologies for probabilistic concurrent systems

Nondeterminism + Probability – why necessary?

- “Nondeterminism” intrinsic to specification development à la CSP
  - underspecified components expressed using “nondeterminism”

```
COMP ⊓ OPTION ⊑ COMP
underspecified ≤ more specified
```

- Analysis of concurrent systems requires “nondeterminism”

∩ - internal choice of CSP

Analysis of concurrent systems

Sys1:

\[ Sys1 \triangleleft (\text{new } s)(A | Sw) \]
\[ A \triangleleft up.U + s?down.D \]
\[ Sw \triangleleft s!stop \]

Sys2:

\[ Sys2 \triangleleft (\text{new } s)(B | Sw) \]
\[ B \triangleleft s?(up.U + down.D) + s?down.D \]
\[ Sw \triangleleft s!stop \]
Analysis of concurrent systems

In CSP theory:

\[
\text{Sys1} \quad \approx \quad \text{Sys2}
\]

semantically equivalent

Both equivalent to the nondeterministic

\[
(up.U + down.D) \sqcap down.D
\]

concurrency = nondeterminism + interleaving

probabilistic concurrency = probability + nondeterminism + interleaving

Testing scenario

- a set of processes \( \mathcal{P}roc \)
- a set of tests \( \mathcal{T} \)
- a set of outcomes \( \mathcal{O} \)
- \( \text{Apply} : \mathcal{T} \times \mathcal{P}roc \rightarrow \mathcal{P}^+(\mathcal{O}) \) – the non-empty set of possible results of applying a test to a process

Comparing sets of outcomes:

- \( O_1 \sqsubseteq_{\text{Ho}} O_2 \) if for every \( o_1 \in O_1 \) there exists some \( o_2 \in O_2 \) such that \( o_1 \leq o_2 \)
- \( O_1 \sqsubseteq_{\text{Sm}} O_2 \) if for every \( o_2 \in O_2 \) there exists some \( o_1 \in O_1 \) such that \( o_1 \leq o_2 \)

\( o_1 \leq o_2 \) : means \( o_2 \) is as least as good as \( o_1 \)
Testing preorders

- \( P \sqsubseteq_{\text{may}} Q \) if \( \text{Apply}(T, P) \sqsubseteq_{\text{Ho}} \text{Apply}(T, Q) \) for every test \( T \)
- \( P \sqsubseteq_{\text{must}} Q \) if \( \text{Apply}(T, P) \sqsubseteq_{\text{Sm}} \text{Apply}(T, Q) \) for every test \( T \)

**Standard testing:**
Use as outcomes \( \mathcal{O} = \{ \top, \bot \} \) with \( \bot \leq \top \)

**Comparisons:**
Possible outcome sets: \( \{ \bot \} \quad \{ \bot, \top \} \quad \{ \top \} \)
- **May:** \( \{ \bot \} <_{\text{Ho}} \{ \bot, \top \} =_{\text{Ho}} \{ \top \} \)
- **Must:** \( \{ \bot \} =_{\text{Sm}} \{ \bot, \top \} <_{\text{Sm}} \{ \top \} \)

**Probabilistic testing:**
Use as \( \mathcal{O} \) the unit interval \([0, 1]\)
Intuition: with \( 0 \leq p \leq q \leq 1 \), passing a test with probability \( q \) better than passing with probability \( p \)

**Comparisons:**
- **May:** \( O_1 \sqsubseteq_{\text{Ho}} O_2 \) is every possibility \( p \in O_1 \) can be improved on by some \( q \in O_2 \)
- **Must:** \( O_1 \sqsubseteq_{\text{Sm}} O_2 \) if every possibility \( q \in O_2 \) is an improvement on some \( p \in O_1 \)
Nondeterministic processes

Intensional semantics:

A process is a state in an LTS

Labelled Transition Systems:
\[ \langle S, \text{Act}_\tau, \rightarrow \rangle \]
- \( S \) - states
- \( \rightarrow \subseteq S \times \text{Act}_\tau \times S \)

\( s_1 \xrightarrow{\mu} s_2 \): process \( s_1 \) can perform action \( \mu \) and continue as \( s_2 \)

\( s_1 \xrightarrow{\tau} s_2 \) special internal action

Process calculi: Syntax for LTSs

Example process calculus CCS:

- \( 0 \) Do nothing
- \( \mu.P \) Perform \( \mu \) then act as \( P \)
- \( P | Q \) Run \( P \) and \( Q \) in parallel . . . communicating via complementary actions
- \( P + Q \) Nondeterministic choice between \( P \) and \( Q \)
- recursive definitions \( D \Leftarrow P \)

Actions
\( P \xrightarrow{\mu} Q \) defined inductively

lots of other process calculi
Testing nondeterministic processes

Tests:
Any process which may contain new report success action/state $\omega$
$T \leftarrow a.\omega + \overline{b}.T + \overline{c}.0$:
- requests an $a$ action . . .
- after an arbitrary number of $b$ actions . . .
- without doing any $c$ action

Applying test $T$ to process $P$:
- Run the combined process ($T | P$)
- Each execution succeeds or fails
- Each execution contributes $\top$ or $\bot$ to $Apply(T, P)$

Nondeterministic ($T | P$) resolved to a set of deterministic executions

Example

Test: $T \leftarrow \overline{a}.\omega + \overline{b}.T + \overline{c}.0$
Process: $P \leftarrow b.(a.Q + b.P)$

Deterministic executions:

- $T | P \xrightarrow{\tau} b \xrightarrow{\tau} a \omega \ | \ - \\  T | P \xrightarrow{\tau} b \xrightarrow{\tau} b \xrightarrow{\tau} a \omega \ | \ - \\  T | P \ ... \\  T | P \xrightarrow{\tau} b \xrightarrow{\tau} b \xrightarrow{\tau} b \ldots \xrightarrow{\tau} b \ldots \\  \bot$

Result:
$Apply(T, P) = \{ \bot, \top \}$
May-testing nondeterministic processes

Divergence not important:

\[ P + \tau.\tau^\infty \simeq_{\text{may}} P \]

Choice not important:

Must testing nondeterministic processes

Divergence catastrophic:

\[ P + \tau.\tau^\infty \simeq_{\text{must}} \tau^\infty \]

Internal choices not very important:
Characterising nondeterministic processes

Ingredients:

- Traces: \(a_1a_2\ldots a_n\) in \(\text{Traces}(P)\) whenever
  \[
P \xrightarrow{\tau}^* a_1 \xrightarrow{\tau}^* \ldots \xrightarrow{\tau}^* a_n \xrightarrow{\tau}^* P'
  \]

- Divergences/convergences: \(P \Downarrow\) whenever there is no infinite execution
  \[
P \xrightarrow{\tau} \xrightarrow{\tau} \ldots \xrightarrow{\tau} \ldots
  \]

- Failures/Acceptances: \(P \text{ acc } A\) whenever \(P \Downarrow\) and
  \[
P \xrightarrow{\tau}^* P' \text{ implies } P' \xrightarrow{a} \text{ for some } a \text{ in } A
  \]

Probabilistic and nondeterministic processes

Intensional semantics:

A process is a \text{distribution} in an \text{pLTS}

Probabilistic Labelled Transition Systems:
\[
\langle S, \text{Act}_\tau, \xrightarrow{\tau}\rangle
\]

\(S\) - states

\(\xrightarrow{\tau}\subseteq S \times \text{Act}_\tau \times \mathcal{D}(S)\)

\(\mathcal{D}(S)\): Mappings \(\Delta : S \rightarrow [0,1]\) with \(\sum_{s \in S} \Delta(s) = 1\)

\(s_1 \xrightarrow{\mu} \Delta: \text{ process } s_1\)

- can perform action \(\mu\)
- with probability \(\Delta(s_2)\) it continues as process \(s_2\)
Example probabilistic processes

What is the probability of action $a$ happening?

Probabilistic process calculi: Syntax for pLTSs

Example process calculus pCCS:
State terms $S, T$:

- $0$
- $\mu . P$
- $S | T$
- $S + T$

Recursive definitions

Process terms $P, Q$:

- $S$
- $P \oplus Q$ probabilistic choice between $P$ and $Q$

Actions

$s \xrightarrow{\nu} \Delta$ defined inductively

Process terms are distributions over states
Testing probabilistic processes

Tests:
Any (prob. . .) process which may contain report success action/state \( \omega \)
\[ a.\omega \frac{1}{4} \oplus (b + c.\omega) : \]
- 25% of time requests an \( a \) action
- 75% requests a \( c \) action
- 75% requires that \( b \) is not possible in a must test

Applying test \( T \) to process \( P \):
- Execute the combined process \(( T \mid P )\)
- Each execution contributes some probability \( p \) to \( \text{Apply}(T, P) \)
- Each execution is a deterministic resolution of \(( T \mid P )\)

Executing probabilistic nondeterministic processes \(( T \mid P )\)

- Choice points occur during an execution
- choices are made
  - statically
  - or dynamically
- choices are made
  - by schedulers
  - adversaries
  - policies

Executions:
- give deterministic behaviour - but may be probabilistic
- contribute a probability to \( \text{Apply}(T, P) \)
Example of executions

```

Example of executions

Static Policies:

\[ pp_1 : s_1 \mapsto s_0 \]
\[ pp_2 : s_1 \mapsto t_{bd} \]

Possible results:

Using \( pp_1 \):
\[ \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{4} + \left( \frac{3}{4} \right)^2 \cdot \frac{1}{4} + \ldots + \ldots = 1 \]

Using \( pp_2 \):
\[ \frac{1}{4} + \frac{3}{4} \cdot 0 = \frac{1}{4} \]

More executions

Arbitrary policies: combinations of

\[ pp_1 : s_1 \mapsto s_0 \]
\[ pp_2 : s_1 \mapsto t_{bd} \]

Possible results:

Using \( pp_1 \):
\[ 1 \]

Using \( pp_2 \):
\[ \frac{1}{4} \]

In general:
\[ p \cdot 1 + (1 - p) \cdot \frac{1}{4} \] for some \( 0 \leq p \leq 1 \]
Formalising executions I

From pLTSS to LTSs

\[ \Delta \xrightarrow{\mu} \Theta \]

- \( \Delta \) represents a cloud of possible process states
- each possible state must be able to perform \( \mu \)
- all possible residuals combine to \( \Theta \)

Examples:

- \((a.b + a.c)_{1/2} \oplus a.d \xrightarrow{a} b_{1/2} \oplus d\)
- \((a.b + a.c)_{1/2} \oplus a.d \xrightarrow{a} (b_{1/2} \oplus c)_{1/2} \oplus d\)
- \((a.b + a.c)_{1/2} \oplus a.d \xrightarrow{a} (b_{p} \oplus c)_{1/2} \oplus d\)
- \(((\tau.a + \tau.b)_{1/2} \oplus (\tau.a + \tau.c) \xrightarrow{\tau} a_{1/2} \oplus (b_{1/2} \oplus c)\)

From pLTSS to LTSs: formally

\[ \Delta \xrightarrow{\mu} \Theta \]

deep whenever
- \( \Delta = \sum_{i \in I} p_i \cdot s_i \), \( I \) a finite index set
- For each \( i \in I \) there is a distribution \( \Theta_i \) s.t. \( s_i \xrightarrow{\mu} \Theta_i \)
- \( \Theta = \sum_{i \in I} p_i \cdot \Theta_i \)
- \( \sum_{i \in I} p_i = 1 \)

Note: in decomposition \( \sum_{i \in I} p_i \cdot s_i \) states \( s_i \) are not necessarily unique
Formalising executions II

 Executing \((T \mid P)\) to \(\Theta\): \[ (T \mid P) \leadsto \Theta \]

\[
\begin{align*}
(T \mid P) & = \Delta_0 + \Delta_0^{\text{stop}} \\
\Delta_0 & \xrightarrow{\tau} \Delta_1 + \Delta_1^{\text{stop}} \\
\vdots & \xrightarrow{\tau} \Delta_{k+1} + \Delta_{k+1}^{\text{stop}} \\
\Delta_k & \xrightarrow{\tau} \Delta_{(k+1)} \\
\vdots & \xrightarrow{\tau} \Delta_{(k+1)} \\
\end{align*}
\]

Total: \[\Theta = \sum_{k=0}^{\infty} \Delta_k^{\text{stop}}\]

\(\Delta^{\text{stop}}\): all states in \(\Delta\) which
- are successful \(s \xrightarrow{\omega}\)
- or are stuck \(s \not\xrightarrow{}\)

\(\Delta\): all other states, which can proceed \(s \xrightarrow{\tau}\).

note: subdistributions

Applying tests to processes: 
\(\text{Apply}(T, P)\)

- find all executions from \((T \mid P)\):
  \[ (T \mid P) \leadsto \Theta \]
- calculate contribution of each \(\Theta\)

Contribution of \(\Theta\):
- all states in \(\Theta\) are successful \(s \xrightarrow{\omega}\) or stuck \(s \not\xrightarrow{}\)
- \(\forall(\Theta) = \sum\{ \Theta(s) \mid s \xrightarrow{\omega} \}\) weight of success

\[\text{Apply}(T, P) = \{ \forall(\Theta) \mid (T \mid P) \leadsto \Theta \}\]

Problem: set of executions \(\{ \Theta \mid (T \mid P) \leadsto \Theta \}\) difficult to calculate
Alternative strategy

Recall:

- $P \sqsubseteq_{p_{\text{may}}} Q$ if $\text{Apply}(T, P) \sqsubseteq_{\text{Ho}} \text{Apply}(T, Q)$ for every test $T$
- $P \sqsubseteq_{p_{\text{must}}} Q$ if $\text{Apply}(T, P) \sqsubseteq_{\text{Sm}} \text{Apply}(T, Q)$ for every test $T$

Maybe:

- $P \sqsubseteq_{p_{\text{may}}} Q$ if $\sup(\text{Apply}(T, P)) \leq \sup(\text{Apply}(T, Q))$ for every test $T$
- $P \sqsubseteq_{p_{\text{must}}} Q$ if $\inf(\text{Apply}(T, P)) \sqsubseteq_{\text{Sm}} \inf(\text{Apply}(T, Q))$ for every test $T$

Strategy:

- calculate $\inf(\text{Apply}(T, -))$ and $\sup(\text{Apply}(T, -))$ directly
- do not calculate the entire set $\text{Apply}(T, -)$

Example: Calculating the sup

Sup-equation:

\[
\begin{align*}
\sup & = \frac{3}{4} \cdot s_1 + \frac{1}{4} \cdot s_2 \\
\end{align*}
\]

\[
\begin{align*}
s_1 & = \max\{r_{\sup}, t_{bd}\} \\
s_2 & = t_{gd} \\
t_{bd} & = 0 \\
t_{gd} & = 1
\end{align*}
\]

$\sup(\text{Apply}(T, P))$ is least solution: $r_{\sup} = 1$
**Example: Calculating the inf**

\[ r_{inf} = \frac{3}{4} \cdot s_1 + \frac{1}{4} \cdot s_2 \]

\[ s_1 = \text{min\{}r_{inf}, t_{bd}\text{}} \]

\[ s_2 = t_{gd} \]

\[ t_{bd} = 0 \]

\[ t_{gd} = 1 \]

\[ \inf(Apply(T, P)) \text{ is least solution: } r_{inf} = \frac{1}{4} \]

---

**Finitary pLTSs**

Whenever

- set of states are finite
- set of actions are finite

In a finitary pLTS:

- execution sets \( \{ \Theta \mid Apply(T, P) \rightharpoonup \Theta \} \) are closed
- \( P \sqsubseteq_{\text{may}} Q \) iff \( \inf(Apply(T, P)) \leq \inf(Apply(T, Q)) \) for every test \( T \)
- \( P \sqsubseteq_{\text{must}} Q \) iff \( \sup(Apply(T, P)) \sqsubseteq_{\text{Sm}} \sup(Apply(T, Q)) \) for every test \( T \)
- \( \inf(Apply(T, -)) \) is least solution of inf-equation
- \( \sup(Apply(T, -)) \) is least solution of sup-equation
Example

\[ r_1 = a.(\tau.b + \tau.c) \quad r_2 = a.b + a.c \quad T = \overline{a}.(b.\omega_{\frac{1}{2}} \oplus c.\omega) \]

\[
\begin{align*}
Apply(T, r_1) & = \begin{cases} 
\inf : 0 \\
\sup : 1 
\end{cases} \\
Apply(T, r_2) & = \begin{cases} 
\inf : \frac{1}{2} \\
\sup : \frac{1}{2} 
\end{cases}
\end{align*}
\]

So choice points do matter: \( r_1 \not\simeq_{\text{pmay}} r_2 \) \( r_1 \not\simeq_{\text{pmust}} r_2 \)

Example

\[
\begin{align*}
Apply(\overline{a}.\omega, P) & = \begin{cases} 
\inf : \frac{1}{2} \\
\sup : 1 
\end{cases} \\
P & \simeq_{\text{pmay}} a.0 \\
P & \sqsubseteq_{\text{pmust}} a.0 \\
a.0 & \not\sqsubseteq_{\text{pmust}} P
\end{align*}
\]
Coming up:

Reasoning techniques for probabilistic processes

Are these distinguishable by any test?

\[
P \quad d \quad \frac{1}{2} \quad \frac{1}{2} \quad b \quad c
\]

\[
Q \quad d \quad \frac{1}{2} \quad \frac{1}{2} \quad a \quad b \quad a \quad c
\]