The Discriminating Power of Higher-Order Languages: A Process Algebraic Approach

Valeria Vignudelli

University of Bologna & Inria FOCUS

(joint work with Marco Bernardo and Davide Sangiorgi)

Higher-order languages

Variables may be instantiated with terms of the language itself
(e.g. terms can be copied)

- functions (λ-calculus)

  $$(\lambda x.M)N \longrightarrow M\{^N/_x\}$$

- higher-order communication (Higher-Order π-calculus)

  $$a(x).M \mid \overline{a}\langle N\rangle.R \longrightarrow M\{^N/_x\} \mid R$$

- action on locations (kells), as with passivation

  $$[\![M]\!]_l \mid \mathrm{pass}_l(x).N \longrightarrow N\{^M/_x\}$$ M is a running term

  [Schmitt,Stefani GC'04, Lenglet et al. Inf.Comp.'11,
  Piérard,Sumii FOSSACS'12, Koutavas,Hennessy CONCUR'13]

The discriminating power of a language

A testing language $\mathcal{L}$, a set of tested terms $P, Q \ldots$

$$P \simeq_{\mathcal{L}} Q \quad \triangleq \quad C[P] \text{ and } C[Q] \text{ `equally successful'}, \forall \text{ contexts } C \text{ of } \mathcal{L}$$

Otherwise: P, Q are discriminated by $\mathcal{L}$

Two classes of first-order (CCS-like) processes as tested terms P, Q:

- nondeterministic
- reactive probabilistic

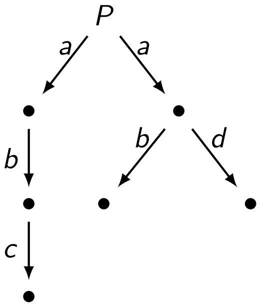
Comparison of testing languages $\mathcal{L}$ with different constructs:

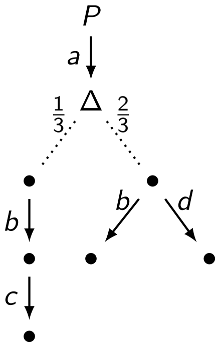
- sequential higher-order (λ-calculus)
- higher-order communication (HOπ)
- ordinary first-order concurrency (CCS)
- locations and passivation
- refusal

no probabilities in $\mathcal{L}$

Tested first-order processes

nondeterministic (LTSs) Vs. reactive probabilistic (RPLTSs)





external & internal nondeterminism

only external nondeterminism

[Larsen,Skou POPL'89, van Breugel et al. Theor.Comp.Sci.'05]

λ-calculus with store (Λ^s)

M is evaluated wrt a store containing P $\qquad \langle P\,;\, M\rangle \longrightarrow \langle P'\,;\, M'\rangle$

- processes are stored values
- read and write values in the store, sequentialization...
- action test $a?$ for stored processes

$$\langle P\,;\, a?\rangle \longrightarrow \langle P'\,;\, \text{true}\rangle \qquad \text{if } P \xrightarrow{a} P'$$

$$\langle P\,;\, a?\rangle \longrightarrow \langle P\,;\, \text{false}\rangle \qquad \text{if } P \stackrel{a}{\not\longrightarrow}$$

Variants: - call by name (${\Lambda^s}_{\mathrm{N}}$) and call by value (${\Lambda^s}_{\mathrm{V}}$)

- refusal free (${\Lambda^s}_{\mathrm{N}(-\mathrm{ref})}, {\Lambda^s}_{\mathrm{V}(-\mathrm{ref})}$)

Concurrent languages

Interactions between context and process:

- action synchronization (CCS)

$$P \mid \overline{a}.M \quad \longrightarrow \quad P' \mid M \qquad\qquad \text{if } P \xrightarrow{a} P'$$

- higher-order communication (HOπ)

$$a(x).M \mid \overline{a}\langle P \rangle.N \longrightarrow M\{{}^{P}\!/\!_{x}\} \mid N$$

- refusal on kells ($\mathrm{CCS}_{\mathrm{ref}}, \mathrm{HO}\pi_{\mathrm{ref}}$)

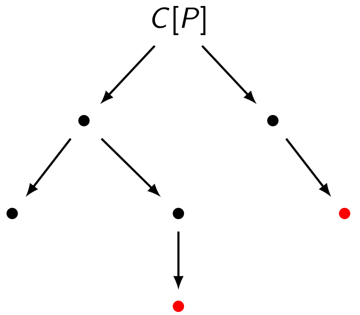
$$[\![P]\!]_l \mid \widetilde{a}_l.M \quad \longrightarrow \quad [\![P]\!]_l \mid M \qquad\qquad \text{if } P \stackrel{a}{\not\longrightarrow}$$

- passivation of kells ($\mathrm{HO}\pi_{\mathtt{pass}}, \mathrm{HO}\pi_{\mathtt{pass},\mathrm{ref}}$)

$$[\![P]\!]_l \mid \mathtt{pass}_l(x).M \longrightarrow M\{{}^{P}\!/\!_{x}\}$$

'Success' on nondeterministic processes

$P \simeq_{\mathcal{L}} Q \quad \triangleq \quad C[P]$ and $C[Q]$ 'equally successful', $\forall$ contexts C of $\mathcal{L}$



P is an LTS, $C[P] \Downarrow$

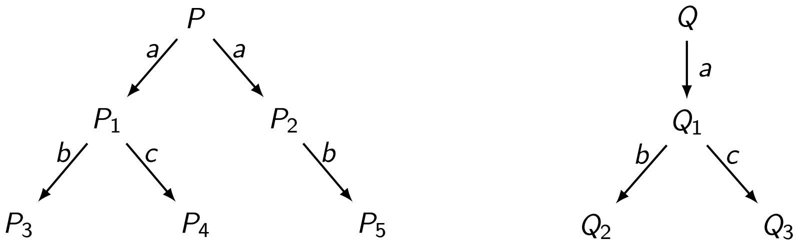
Success states $\bullet$ are:

$M \xrightarrow{\omega}$ in CCS, HOπ

$\langle s\,;\,\text{true}\rangle$ in Λ^s

Success is $\Downarrow$ ($=$ a success state $\bullet$ is reachable) [may success]

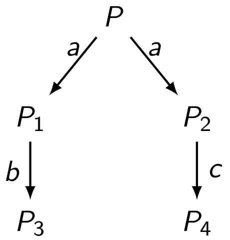
Example: discriminating LTSs

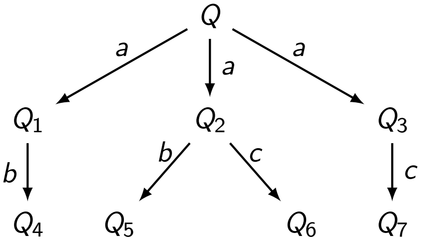


- test actions
- test refusal of actions
- perform tests in sequence

completed trace/simulation equivalent but not failure equivalent processes
[see van Glabbeek '90 spectrum]

Example: discriminating LTSs





- conjunction of tests

  $\Longrightarrow$ copying processes (during the execution)

failure trace equivalent but not ready trace equivalent processes

Copying

- higher-order languages can make copies of P before interaction
- only Λ^s call by value and HOπ with passivation can make copies of (and run tests on) P after interaction

$$\langle P\,;\ \texttt{if}\ a?\ \texttt{then}\ (\lambda x.M)\mathrm{read}\ \texttt{else}\ \mathsf{false}\rangle$$

The Spectrum on LTSs

ready simulation equivalence

$\Lambda^s{}_{\mathrm{V}} = \mathsf{HO}\pi_{\mathrm{ref},\mathbf{pass}}$

simulation equivalence

$\Lambda^s{}_{\mathrm{V}(-\mathrm{ref})} = \mathsf{HO}\pi_{\mathbf{pass}}$

failure trace equivalence

$\Lambda^s{}_{\mathrm{N}} = \mathsf{HO}\pi_{\mathrm{ref}} = \mathsf{CCS}_{\mathrm{ref}}$

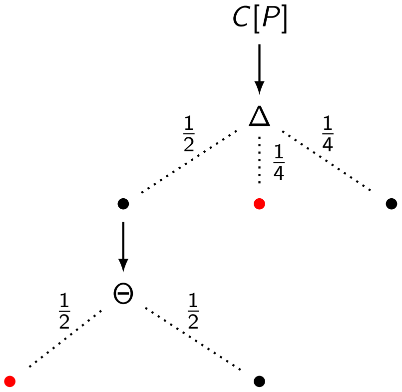
$\Lambda^s{}_{\mathrm{N}(-\mathrm{ref})} = \mathsf{HO}\pi = \mathsf{CCS}$

trace equivalence

- CBV $=$ passivation in HOπ
- sequential $=$ concurrent
- first-order communication $=$ higher-order communication

‘Success’ on reactive probabilistic processes

$P \simeq_{\mathcal{L}} Q \quad \triangleq \quad C[P]$ and $C[Q]$ ‘equally successful’, $\forall$ contexts C of $\mathcal{L}$



P is an RPLTS, $C[P] \Downarrow_{\frac{1}{2}}$

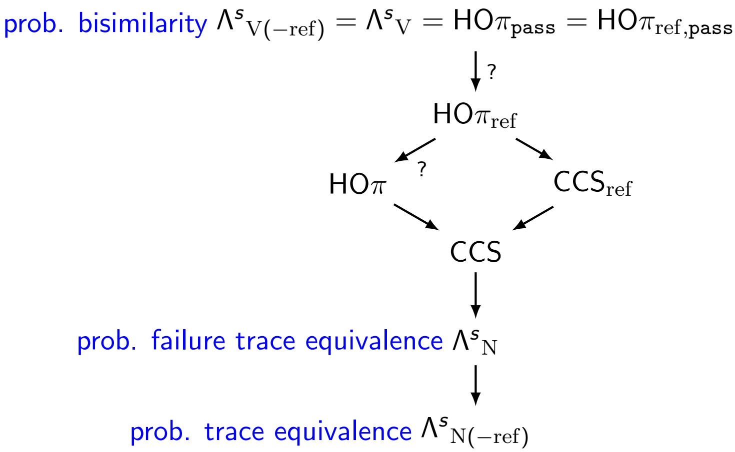
Success states ● are:

$M \xrightarrow{\omega}$ in CCS, HOπ

$\langle N\,;\,\mathsf{true}\rangle$ in Λ^s

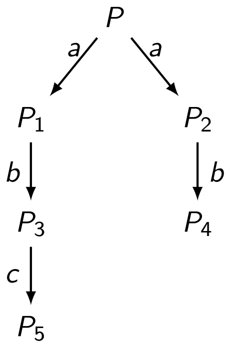
Success is $\Downarrow_p$, where $p =$ sum of probabilities of all success paths

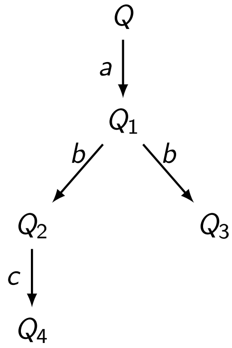
The Spectrum on RPLTSs



- CBV = passivation in $\mathsf{HO}\pi \pm \mathrm{ref}$
- sequential $\neq$ concurrent
- first-order communication $\neq$ higher-order communication

Testing LTSs: refusal and conjunction

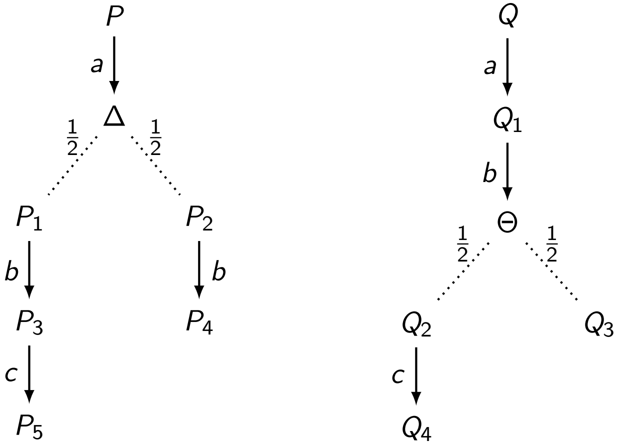




$C[P] \not\Downarrow$ $\qquad$ $C[Q] \Downarrow$

$$C = \langle \cdot \; ; \; \underline{\mathrm{Seq}} \; \mathrm{T}_a \, (\, \underline{\mathrm{And}} \, (\, \underline{\mathrm{Seq}} \; \mathrm{T}_b \; \mathrm{T}_c \,)(\, \underline{\mathrm{Seq}} \; \mathrm{T}_b \; \mathrm{T}_{\neg c} \,)) \rangle$$

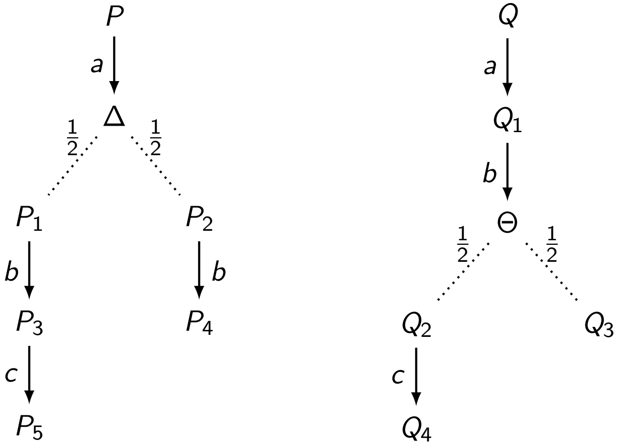
Testing RPLTSs: refusal and conjunction



$C[P] \Downarrow_0$ $\qquad$ $C[Q] \Downarrow_{\frac{1}{4}}$

$$C = \langle \cdot \; ; \; \underline{\mathrm{Seq}}\ \mathrm{T}_a\,(\,\underline{\mathrm{And}}\,(\,\underline{\mathrm{Seq}}\ \mathrm{T}_b\ \mathrm{T}_c\,)(\,\underline{\mathrm{Seq}}\ \mathrm{T}_b\ \mathrm{T}_{\neg c}\,))\rangle$$

Testing RPLTSs: refusal and conjunction



$C[P] \Downarrow_{\frac{1}{2}}$ $\quad$ $C[Q] \Downarrow_{\frac{1}{4}}$

$$C = \langle \cdot \ ; \ \underline{\mathrm{Seq}} \ \mathrm{T}_a \, (\, \underline{\mathrm{And}} \, (\, \underline{\mathrm{Seq}} \ \mathrm{T}_b \ \mathrm{T}_c \,)(\, \underline{\mathrm{Seq}} \ \mathrm{T}_b \ \mathrm{T}_c \,))\rangle$$

Conclusion

- characterization of the testing equivalences in terms of known behavioural equivalences on processes
- spectrum of the discriminating powers of different languages on both nondeterministic and reactive probabilistic processes

Conclusion

- characterization of the testing equivalences in terms of known behavioural equivalences on processes
- spectrum of the discriminating powers of different languages on both nondeterministic and reactive probabilistic processes

Thank you!