Of probabilistic *wp* and *CSP*

Carroll Morgan
University of New South Wales

- *Action Systems* (probabilistic)
- Informal translation to *CSP* (probabilistic)
- Probabilistic *relational model* (à la Jifeng He)
- Probabilistic traces via *expected values*
- Expected-value algebra and *duality*
- Probabilistic *healthiness conditions*
- *Compositionality?*

Carroll Morgan.
Of *wp* and *CSP*.
In *Beauty is our Business*, Feijen *et al.*, Springer, 1990.

A probabilistic Action System

demonic choice

probabilistic choice

System $\mathcal{H}$

$$\begin{array}{l} \textbf{var } \mathsf{n} \colon \mathbb{Z} \\ \textbf{initially } \ \mathsf{n} := 0 \ {}_{1/2}\!\oplus \ \pm 1 \end{array}$$

$$\begin{array}{rclcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \neq 0 & \rightarrow & \mathsf{n} := 0 \\ \mathsf{haec} & \widehat{=} & \mathsf{n} = 0 & \rightarrow & \mathsf{n} := -1 \ {}_{1/3}\!\oplus 0 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} < 0 & \rightarrow & \mathsf{n} := \pm 1 \end{array}$$

R.-J.R. Back and R. Kurki-Suonio.
Decentralisation of process nets with centralised control.
2nd ACM SIGACT-SIGOPS Symp. PODC, 131-142, 1983.

Its *pCSP* "equivalent"

$$\textbf{initially} \quad \mathsf{n}{:=}\ 0\ {}_{1/2}{\oplus}\ \ \pm 1$$

$$\begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \neq 0 \ \rightarrow\ \mathsf{n}{:=}\ 0 \\ \mathsf{haec} & \widehat{=} & \mathsf{n} = 0 \ \rightarrow\ \mathsf{n}{:=}\ -1\ {}_{1/3}{\oplus}\ 0 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} < 0 \ \rightarrow\ \mathsf{n}{:=}\ \pm 1 \end{array}$$

Process $\mathcal{H}$

$$\mathcal{H} \quad \widehat{=} \quad \mathcal{H}_0\ {}_{1/2}{\oplus}\ (\mathcal{H}_{-1} \sqcap \mathcal{H}_1)$$

$$\begin{array}{lcl} \mathcal{H}_1 & \widehat{=} & \mathsf{hic} \rightarrow \mathcal{H}_0 \\ \mathcal{H}_0 & \widehat{=} & \mathsf{haec} \rightarrow (\mathcal{H}_{-1}\ {}_{1/3}{\oplus}\ \mathcal{H}_0) \\ \mathcal{H}_{-1} & \widehat{=} & \mathsf{hic} \rightarrow \mathcal{H}_0 \ \ \Box\ \ \mathsf{hoc} \rightarrow (\mathcal{H}_{-1} \sqcap \mathcal{H}_1) \end{array}$$

He; Josephs; Woodcock... and M.J. Butler. A *CSP* Approach to Action Systems. DPhil Thesis, 1992.

Its traces

$$\textbf{initially}\quad \mathsf{n} := 0 \;{}_{1/2}\!\oplus\; \pm 1$$

$$\begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \neq 0 \;\rightarrow\; \mathsf{n} := 0 \\ \mathsf{haec} & \widehat{=} & \mathsf{n} = 0 \;\rightarrow\; \mathsf{n} := -1 \;{}_{1/3}\!\oplus\; 0 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} < 0 \;\rightarrow\; \mathsf{n} := \pm 1 \end{array}$$

$$\begin{array}{llll}
\{ & \langle\rangle, & & \\
 & \langle\mathsf{hic}\rangle, \; \langle\mathsf{haec}\rangle, \; \langle\mathsf{hoc}\rangle, & & \\
 & \langle\mathsf{hic}, & \mathsf{haec}\rangle, & \\
 & \langle\mathsf{haec}, & \mathsf{hic}\rangle, & \\
 & \langle\mathsf{haec}, & \mathsf{haec}\rangle, & \\
 & \langle\mathsf{haec}, & \mathsf{hoc}\rangle, & \\
 & \langle\mathsf{hoc}, & \mathsf{hic}\rangle, & \\
 & \langle\mathsf{hoc}, & \mathsf{hoc}\rangle, & \\
 & \langle\mathsf{hic}, & \mathsf{haec}, & \mathsf{hic}\rangle, \\
 & \langle\mathsf{hic}, & \mathsf{haec}, & \mathsf{haec}\rangle, \\
 & \langle\mathsf{hic}, & \mathsf{haec}, & \mathsf{hoc}\rangle, \\
 & & \vdots & \\
\} & & &
\end{array}$$

The relational model for a *pAS*

- A (*sub-*) *distribution* Δ over a state space S is a function from S to $[0,1]$ such that $(\sum_{s:S} \Delta.s) \leq 1$.
- The *set of all such distributions* is $\overline{S}$.
- A *point distribution* at s is written $\overline{s}$.
- *Non-demonic* probabilistic programs have type $S \rightarrow \overline{S}$.
- *Probabilistic/demonic* programs have type $S \rightarrow \mathbb{P}\overline{S}$.
- The *meaning* of a program *prog* is written $[\![prog]\!]$.

Jifeng He, Annabelle McIver and Karen Seidel.
Probabilistic models for the guarded command language.
Science of Computer Programming 28:171-192, 1997.

Simple examples

- **identity** — $[\![\mathbf{skip}]\!].n \;=\; \{\overline{n}\}$

  The "do-nothing" program **skip** takes any state to itself. Because of our demonic/probabilistic type for programs, however, the result is not just n again, nor even the set $\{n\}$, but rather is the singleton set containing just the point distribution on n.

- **assignment** — $[\![\mathsf{n}{:=}\ \mathsf{n}+1]\!].n \;=\; \{\overline{n+1}\}$

  Non-demonic and non-probabilistic assignments deliver singleton sets of point distributions: singleton sets because there is no demonic choice; point-distributions because there is no (non-trivial) probabilistic choice.

- **probabilistic choice** — $[\![\mathsf{n}{:=}\ \mathsf{n}+1 \;{}_{1/3}{\oplus}\; \mathsf{n}{:=}\ \mathsf{n}+2]\!].n$

  $$
  \begin{array}{rll}
  = & \{\Delta'\} & \\
  & \text{where} & \begin{array}{lcll} \Delta'.(n+1) & = & 1/3 & \\ \Delta'.(n+2) & = & 2/3 & \\ \Delta'.n' & = & 0 & \text{for other values } n' \end{array}
  \end{array}
  $$

  Non-demonic but probabilistic assignments deliver singleton sets of non-trivial distributions: again the sets are singleton because there is no demonic choice; but the single element of the set is a proper distribution.

Simple examples

- **demonic choice** — $[\![\mathsf{n}{:=}\ \mathsf{n}+1 \sqcap \mathsf{n}{:=}\ \mathsf{n}+2]\!].n$
  $= \{\overline{n+1}, \overline{n+2}\}$

  A purely demonic (and non-probabilistic) binary choice delivers the distributions contributed by each of its operands.

- **demonic probabilistic choice** — $[\![\mathsf{n}{:=}\ \mathsf{n}+1\ {}_{1/3}\!\oplus_{1/3}\ \mathsf{n}{:=}\ \mathsf{n}+2]\!].n$

$$
\begin{array}{lllll}
= & \{\Delta'_{1/3}, \Delta'_{2/3}\} & & & \\
 & \text{where } \Delta'_p.(n+1) & = & p & \\
 & \phantom{\text{where }} \Delta'_p.(n+2) & = & 1-p & \\
 & \phantom{\text{where }} \Delta'_p.n' & = & 0 & \text{for other values } n'
\end{array}
$$

  The notation ${}_p\!\oplus_q$, for $p+q \leq 1$, abbreviates the demonic choice between the two probabilistic choices ${}_p\!\oplus$ and ${}_{1-q}\!\oplus$: it executes the left branch with probability *at least* p, the right with probability *at least* q and —in any case— it is certain to execute one or the other.

Naked guarded commands

- A "naked" guarded command (*i.e.* not "clothed" by an **if···fi**) is executed only if its guard is true; otherwise it "cannot start".

$$\Delta' \in [\![gd \rightarrow prog]\!].s \quad \widehat{=} \quad \underline{s \in [\![gd]\!]} \;\wedge\; \Delta' \in [\![prog]\!].s$$

- If the guard is false in the current state (here *s*), then the result set of distributions is empty.

Miracles of Morris, Nelson, Morgan, late 80's; used *e.g.* in *Event-B* and elsewhere.

Sequential composition

- First, "lift" functions *f* so that they act over whole incoming initial distributions:

$$f^*.\Delta.s' \quad \widehat{=} \quad \left(\sum_{s:S} \Delta.s * f.s.s'\right)$$

- Then identify the "deterministic refinements" of a general demonic/probabilistic command:

$$r \sqsubseteq f \quad \widehat{=} \quad (\forall s\colon S \cdot r.s \ni f.s)$$

Sequential composition

- $f^*.\Delta.s' \quad \widehat{=} \quad \left(\sum_{s:S} \Delta.s * f.s.s'\right)$

- $r \sqsubseteq f \quad \widehat{=} \quad (\forall s \colon S \cdot r.s \ni f.s)$

r^+ is a form of "down-closure" of r, adding the everywhere-zero sub-distribution where r is empty.

- Then lift the general command by lifting its deterministic components:

$$r^*.\Delta \quad \widehat{=} \quad \{f \colon S \to \overline{S} \mid r^+ \sqsubseteq f \cdot f^*.\Delta\}$$

- And, finally, form the sequential composition by applying the second command "lifted" to all outcomes of the first:

$$[\![prog_1;\, prog_2]\!].s \quad \widehat{=} \quad \{\Delta \colon [\![prog_1]\!].s \cdot [\![prog_2]\!]^*.\Delta\}$$

Traces of a *pAS*

$$
\begin{array}{llll}
\text{System } \mathcal{A}\text{:} & \textbf{initially} \;\; \mathsf{n} := 0 & \mathsf{hic} \;\widehat{=}\; \mathsf{n} \geq 0 \rightarrow \mathsf{n} := +1 \\
 & & \mathsf{hoc} \;\widehat{=}\; \mathsf{n} \leq 0 \rightarrow \mathsf{n} := -1 \\
\\
\text{System } \mathcal{D}\text{:} & \textbf{initially} \;\; \mathsf{n} := \pm 1 & \mathsf{hic} \;\widehat{=}\; \mathsf{n} \geq 0 \rightarrow \mathsf{n} := +1 \\
 & & \mathsf{hoc} \;\widehat{=}\; \mathsf{n} \leq 0 \rightarrow \mathsf{n} := -1 \\
\\
\text{System } \mathcal{P}\text{:} & \textbf{initially} \;\; \mathsf{n} := -1 \;{}_{1/2}{\oplus}\; +1 & \mathsf{hic} \;\widehat{=}\; \mathsf{n} \geq 0 \rightarrow \mathsf{n} := +1 \\
 & & \mathsf{hoc} \;\widehat{=}\; \mathsf{n} \leq 0 \rightarrow \mathsf{n} := -1
\end{array}
$$

All three systems have the same potential traces — but the first is "angelic" (due to external choice), and the second is "demonic" (internal choice).

The third is *probabilistic*, and we look at it more closely...

Traces of a *pAS*

$$\text{Exp}.[\![prog]\!].B.s \quad \widehat{=} \quad (\textstyle\sum_{s':S} \Delta'.s' * B.s') \ ,$$

given that $[\![prog]\!].s = \{\Delta'\}$.

For a *non-demonic* program *prog*, the Exp function gives

the *expected value* of the function *B* over the final distribution produced by *prog* from initial state *s*.

Traces of a *pAS*

$$\begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \geq 0 \;\rightarrow\; \mathsf{n}{:=}\; +1 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} \leq 0 \;\rightarrow\; \mathsf{n}{:=}\; -1 \end{array}$$

We indicate *characteristic functions* with square brackets and, in that context, an action name stands for its guard; thus

$$[\mathsf{hic}] \qquad\qquad [\mathsf{hic}].n \;=\; \begin{array}{l} 1 \text{ if } n \geq 0 \\ 0 \text{ otherwise} \end{array}$$

is the function of the state space S, that is the integers $\mathbb{Z}$ here, that is one for non-negative arguments and zero elsewhere.

Traces of a *pAS*

$$\textbf{initially}\ \ \mathsf{n}\!:=\ -1\ {}_{1/2}\!\oplus +1 \qquad \begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \geq 0 \ \rightarrow\ \mathsf{n}\!:= +1 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} \leq 0 \ \rightarrow\ \mathsf{n}\!:= -1 \end{array}$$

The expected value of a characteristic function is the probability assigned to its underlying set. Thus

$$\big(\mathrm{Exp}.[\![\, ini \,]\!].[\mathsf{hic}]\big).n$$

is the probability that event hic will be enabled initially.

Traces of a *pAS*

The probability that hic is initially enabled is
$\mathrm{Exp}.[\![ini]\!].[\mathsf{hic}].n$

$$\textbf{initially}\quad \mathsf{n}{:=}\; -1 \;{}_{1/2}\!\oplus +1 \qquad \begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \geq 0 \;\rightarrow\; \mathsf{n}{:=}\; +1 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} \leq 0 \;\rightarrow\; \mathsf{n}{:=}\; -1 \end{array}$$

But [hic] itself is the expected value of the constant, everywhere-one function [true] over the distributions produced by the *action* hic, provided we take that value to be *zero* when the guard is false since —in that case— there is no final distribution. Thus we have

$$[\mathsf{hic}].\cancel{n} \quad = \quad \mathrm{Exp}.[\![\mathsf{hic}]\!].[\mathsf{true}].\cancel{n}$$

for the characteristic function of hic's guard.

If it's enabled, its probability of establishing true is one; if it's not enabled, its probability of establishing true — or indeed anything at all — is zero.

Traces of a *pAS*

The probability that hic is initially enabled is
$\mathrm{Exp}.[\![ini]\!].[\mathsf{hic}].n$

$$\textbf{initially}\quad \mathsf{n}{:=}\ -1\ {}_{1/2}\!\oplus +1 \qquad \begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n}\geq 0 \ \rightarrow\ \mathsf{n}{:=}\ +1 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n}\leq 0 \ \rightarrow\ \mathsf{n}{:=}\ -1 \end{array}$$

But [hic] itself is the expected value of the constant, everywhere-one function [true] over the distributions produced by the *action* hic, provided we take that value to be *zero* when the guard is false since —in that case— there is no final distribution. Thus we have

$$[\mathsf{hic}].n \quad = \quad \mathrm{Exp}.[\![\mathsf{hic}]\!].[\mathsf{true}].n$$

for the characteristic function of hic's guard.

If it's enabled, its probability of establishing true is one; if it's not enabled, its probability of establishing true — or indeed anything at all — is zero.

Traces of a *pAS*

The probability that hic is initially enabled is
$\mathrm{Exp}.[\![ini]\!].[\mathsf{hic}].n$

$$\textbf{initially}\quad \mathsf{n}{:=}\ -1\ {}_{1/2}{\oplus}\ +1 \qquad \begin{array}{rcl} \mathsf{hic} & \widehat{=} & \mathsf{n} \geq 0 \ \rightarrow\ \mathsf{n}{:=}\ +1 \\ \mathsf{hoc} & \widehat{=} & \mathsf{n} \leq 0 \ \rightarrow\ \mathsf{n}{:=}\ -1 \end{array}$$

"The probability of the occurrence of the trace $\langle\mathsf{hic}\rangle$ from initial state n"

$$= \quad \mathrm{Exp}.[\![ini]\!].[\mathsf{hic}].n$$

$[\mathsf{hic}] = \mathrm{Exp}.[\![\mathsf{hic}]\!].[\mathsf{true}]$

$$= \quad \mathrm{Exp}.[\![ini]\!].(\mathrm{Exp}.[\![\mathsf{hic}]\!].[\mathsf{true}]).n$$

$$= \quad \mathrm{Exp}.[\![ini;\ \mathsf{hic}]\!].[\mathsf{true}].n\ .$$

The algebra of the *Exp* functions

- Because of internal choice (demonic nondeterminism), there may be several final distributions produced by a program *prog*.
- Since standard *CSP* includes a trace if it *could* occur, we take the *maximum* probability over all possible distributions, writing

$$\overline{\mathrm{Exp}}.[\![prog]\!].B.s\ .$$

- $\mathrm{Exp}.[\![prog]\!]$ and $\overline{\mathrm{Exp}}.[\![prog]\!]$ are the same if *prog* is non-demonic.
- We have the distribution law

$$\overline{\mathrm{Exp}}.[\![prog_1;prog_2]\!].B.s \quad = \quad \overline{\mathrm{Exp}}.[\![prog_1]\!].(\overline{\mathrm{Exp}}.[\![prog_2]\!].B).s\ .$$

The *Exp* duality

- Since standard *CSP* includes a trace if it *could* occur, we take the *maximum* probability over all possible distributions, writing

$$\overline{\mathrm{Exp}}.[\![prog]\!].B.s\ .$$

- But if we took the "normal", demonic view for sequential programs we would take the *minimum* expected value, writing instead

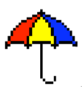
$$\underline{\mathrm{Exp}}.[\![prog]\!].B.s\ .$$

- The two forms are dual, satisfying

$$\overline{\mathrm{Exp}}.[\![prog]\!].B.s \quad = \quad 1 - \underline{\mathrm{Exp}}.[\![prog]\!].(1-B).s\ .$$

- And the latter is exactly the *greatest pre-expectation* for sequential, probabilistic, demonic programs.

The *Exp* duality



$$
\begin{aligned}
wp.\mathbf{abort}.postE &\;:=\; 0 \\
wp.\mathbf{skip}.postE &\;:=\; postE \\
wp.(x := expr).postE &\;:=\; postE\ \langle x \mapsto expr \rangle \\
wp.(prog;\, prog').postE &\;:=\; wp.prog.(wp.prog'.postE) \\
wp.(prog \sqcap prog').postE &\;:=\; wp.prog.postE \ \ \mathsf{min} \ \ wp.prog'.postE \\
wp.(prog\ {}_p\!\oplus prog').postE &\;:=\; p * wp.prog.postE \ + \ \overline{p} * wp.prog'.postE\ .
\end{aligned}
$$

Recall that $\overline{p}$ is the complement of p.

The expression on the right gives the *greatest pre-expectation* of *postE* with respect to each $pGCL$ construct, where *postE* is an expression of type $\mathbb{E}S$ over the variables in state space S. (For historical reasons we continue to write *wp* instead of *gp*.)

The *Exp* duality

$$
\begin{array}{rcl}
wp.\mathbf{abort}.postE & := & 0 \\
wp.\mathbf{skip}.postE & := & postE \\
wp.(x := expr).postE & := & postE\ \langle x \mapsto expr \rangle \\
wp.(prog;\, prog').postE & := & wp.prog.(wp.prog'.postE) \\
wp.(prog \sqcap prog').postE & := & wp.prog.postE \quad \mathsf{min} \quad wp.prog'.postE \\
wp.(prog\ {}_p\!\oplus\ prog').postE & := & p * wp.prog.postE \ + \ \overline{p} * wp.prog'.postE\ .
\end{array}
$$

Recall that $\overline{p}$ is the complement of p.

The expression on the right gives the *greatest pre-expectation* of $postE$ with respect to each $pGCL$ construct, where $postE$ is an expression of type $\mathbb{E}S$ over the variables in state space S. (For historical reasons we continue to write wp instead of gp.)

C.C. Morgan, A.K. McIver and K. Seidel.
Probabilistic predicate transformers.
ACM TOPLAS 18(3):325-353, 1996.

The *Exp* duality

$$\overline{\mathrm{Exp}}.[\![\mathbf{abort}]\!].B \;\;\widehat{=}\;\; 1$$

$$\overline{\mathrm{Exp}}.[\![\mathbf{skip}]\!].B \;\;\widehat{=}\;\; B$$

$$\overline{\mathrm{Exp}}.[\![\mathsf{n}{:=}\;\mathit{expr}]\!].B \;\;\widehat{=}\;\; B_{\mathsf{n}}^{\mathit{expr}}$$

$$\overline{\mathrm{Exp}}.[\![G \rightarrow \mathit{prog}]\!].B \;\;\widehat{=}\;\; [G] * \overline{\mathrm{Exp}}.[\![\mathit{prog}]\!].B$$

$$\overline{\mathrm{Exp}}.[\![\mathit{prog};\;\mathit{prog}']\!].B \;\;\widehat{=}\;\; \overline{\mathrm{Exp}}.[\![\mathit{prog}]\!].(\overline{\mathrm{Exp}}.[\![\mathit{prog}']\!].B)$$

$$\overline{\mathrm{Exp}}.[\![\mathit{prog} \sqcap \mathit{prog}']\!].B \;\;\widehat{=}\;\; \begin{array}{rl} & \overline{\mathrm{Exp}}.[\![\mathit{prog}]\!].B \\ \mathbf{max} & \overline{\mathrm{Exp}}.[\![\mathit{prog}']\!].B \end{array}$$

$$\overline{\mathrm{Exp}}.[\![\mathit{prog}\;{}_{p}\!\oplus\;\mathit{prog}']\!].B \;\;\widehat{=}\;\; \begin{array}{rrcl} & p & * & \overline{\mathrm{Exp}}.[\![\mathit{prog}]\!].B \\ + & (1-p) & * & \overline{\mathrm{Exp}}.[\![\mathit{prog}']\!].B \end{array}$$

26 1. Introduction to $pGCL$

$$\begin{array}{rcl}
wp.\mathbf{abort}.postE & := & 0 \\
wp.\mathbf{skip}.postE & := & postE \\
wp.(x{:=}\;expr).postE & := & postE\;\langle x \mapsto expr\rangle \\
wp.(prog;\,prog').postE & := & wp.prog.(wp.prog'.postE) \\
wp.(prog \sqcap prog').postE & := & wp.prog.postE \;\;\mathsf{min}\;\; wp.prog'.postE \\
wp.(prog\;{}_p\!\oplus\;prog').postE & := & p * wp.prog.postE \;+\; \overline{p} * wp.prog'.postE\,.
\end{array}$$

Recall that $\overline{p}$ is the complement of p.

The expression on the right gives the *greatest pre-expectation* of *postE* with respect to each $pGCL$ construct, where *postE* is an expression of type $\mathbb{E}S$ over the variables in state space S. (For historical reasons we continue to write *wp* instead of *gp*.)

Probabilistic *failures* and *divergences*

- The *failures* of a process can be defined similarly: we have that

$$\overline{\mathrm{Exp}}.[\![ini; \mathsf{es}]\!].[\neg \mathsf{E}].s$$

  is the maximum probability that the process can participate in the trace es and then refuse any event in E, where $[\neg \mathsf{E}]$ is the characteristic function of the set in which no event of E is enabled.

- The *divergences* are defined using

$$\overline{\mathrm{Exp}}.[\![ini; \mathsf{es}]\!].[\mathsf{false}].s \ ,$$

  provided the relational model is (easily) extended to include a bottom "divergent" state $\perp$.

- And the *algebra* of $\overline{\mathrm{Exp}}.[\![\cdot]\!]$ allows us to prove probabilistic "healthiness laws" about the way these observations interact...

Standard healthiness

$$\begin{array}{lll}
\mathbf{C0} & (\langle\rangle, \emptyset) \in F & \\
\mathbf{C1} & (\mathsf{es} \mathbin{+\!\!+} \mathsf{es}', \mathsf{E}) \in F \quad \Rightarrow \quad (\mathsf{es}, \emptyset) \in F & \\
\mathbf{C2} & (\mathsf{es}, \mathsf{E}) \in F \ \wedge \ \mathsf{E}' \subseteq \mathsf{E} \quad \Rightarrow \quad (\mathsf{es}, \mathsf{E}') \in F & \\
 & & \\
\mathbf{C3} & (\mathsf{es}, \mathsf{E}) \in F \quad \Rightarrow & (\mathsf{es} \mathbin{+\!\!+} \langle \mathsf{e} \rangle, \emptyset) \in F \\
 & & \vee \quad (\mathsf{es}, \mathsf{E} \cup \{\mathsf{e}\}) \in F \\
 & & \\
\mathbf{C4} & \mathsf{es} \in D \quad \Rightarrow \quad \mathsf{es} \mathbin{+\!\!+} \mathsf{es}' \in D & \\
\mathbf{C5} & \mathsf{es} \in D \quad \Rightarrow \quad (\mathsf{es}, \mathsf{E}) \in F &
\end{array}$$

These are the conditions that hold for the failures F and divergences D of a standard *CSP* process.

Probabilistic healthiness

- The probabilistic failure pFail.(es,E) is the maximum probability that the action system can perform its initialisation, then execute es, then enter a state where no actions in E are enabled.

$$\mathrm{pFail}.(\mathsf{es},\mathsf{E}) \quad \widehat{=} \quad \overline{\mathrm{Exp}}.[\![ini; \mathsf{es}]\!].[\neg \mathsf{E}]$$

- The traces are derived from the failures in the usual way.

$$\mathrm{pTr}.\mathsf{es} \quad \widehat{=} \quad \mathrm{pFail}.(\mathsf{es}, \emptyset)$$

- Divergence is the only way to "establish" false.

$$\mathrm{pDiv}.\mathsf{es} \quad \widehat{=} \quad \overline{\mathrm{Exp}}.[\![ini; \mathsf{es}]\!].[\mathsf{false}]$$

Probabilistic healthiness

To compare these with the standard conditions **C0** to **C5**, read "≤" as implication.

pC0 — $$\mathrm{pFail}.(\langle\rangle, \emptyset) = 1$$

It is always possible for a system to start, since its initialisation is unguarded.

pC1 — $$\mathrm{pFail}.(\mathsf{es} \mathbin{+\!\!+} \mathsf{es}', \mathsf{E}) \;\leq\; \mathrm{pFail}.(\mathsf{es}, \emptyset)$$

The probability of continuing a trace is no more than the probability of achieving the trace itself.

pC2 — $$\mathrm{pFail}.(\mathsf{es}, \mathsf{E}) \;\geq\; \mathrm{pFail}.(\mathsf{es}, \mathsf{E} \cup \mathsf{E}')$$

The probability of refusing a set of events is no less than the probability of refusing a superset of it.

pC3 — $$\begin{aligned}\mathrm{pFail}.(\mathsf{es}, \mathsf{E}) \quad\leq\quad & \mathrm{pFail}.(\mathsf{es} \mathbin{+\!\!+} \langle\mathsf{e}\rangle, \emptyset) \\ + \quad & \mathrm{pFail}.(\mathsf{es}, \mathsf{E} \cup \{\mathsf{e}\})\end{aligned}$$

If an event cannot be refused, then it must be accepted.

pC4 — $$\mathrm{pDiv}.\mathsf{es} \;\leq\; \mathrm{pDiv}.(\mathsf{es} \mathbin{+\!\!+} \mathsf{es}')$$

Any event is possible after divergence.

pC5 — $$\mathrm{pDiv}.\mathsf{es} \;\leq\; \mathrm{pFail}.(\mathsf{es}, \mathsf{E})$$

Any refusal is possible after divergence.

Probabilistic healthiness

$$
\begin{array}{lll}
 & \mathrm{pFail}.(\mathsf{es} +\!\!+ \langle \mathsf{e} \rangle, \emptyset) \quad + \quad \mathrm{pFail}.(\mathsf{es}, \mathsf{E} \cup \{\mathsf{e}\}) & \\
\geq & \mathrm{pFail}.(\mathsf{es} +\!\!+ \langle \mathsf{e} \rangle, \emptyset) \quad [\,] \quad \mathrm{pFail}.(\mathsf{es}, \mathsf{E} \cup \{\mathsf{e}\}) & \text{arithmetic} \\
 & & \\
= & \overline{\mathrm{Exp}}.[\![\mathit{ini}; \mathsf{es}; \mathsf{e}]\!].[\mathsf{true}] \quad [\,] \quad \overline{\mathrm{Exp}}.[\![\mathit{ini}; \mathsf{es}]\!].[\neg(\mathsf{E} \cup \{\mathsf{e}\})] & \text{definition pFail} \\
= & \overline{\mathrm{Exp}}.[\![\mathit{ini}; \mathsf{es}]\!].[\mathsf{e}] \quad [\,] \quad \overline{\mathrm{Exp}}.[\![\mathit{ini}; \mathsf{es}]\!].[\neg(\mathsf{E} \cup \{\mathsf{e}\})] & \text{sequential composition} \\
\geq & \overline{\mathrm{Exp}}.[\![\mathit{ini}; \mathsf{es}]\!].([\mathsf{e}] \; [\,] \; [\neg(\mathsf{E} \cup \{\mathsf{e}\})]) & \boxed{\text{super-disjunctivity}} \\
\geq & \overline{\mathrm{Exp}}.[\![\mathit{ini}; \mathsf{es}]\!].[\neg \mathsf{E}]) & \text{set algebra; monotonicity} \\
 & & \\
= & \mathrm{pFail}.(\mathsf{es}, \mathsf{E}) & \text{definition pFail}
\end{array}
$$

The algebra of *Exp* (reprise)

- Sub-conjunctivity (from probabilistic *wp*):

$$\underline{\mathrm{Exp}}.[\![prog]\!].B \quad \& \quad \underline{\mathrm{Exp}}.[\![prog]\!].B' \qquad \leq \qquad \underline{\mathrm{Exp}}.[\![prog]\!].(B \,\&\, B')$$

- Super-disjunctivity (from the above, via duality):

$$\overline{\mathrm{Exp}}.[\![prog]\!].B \quad [\!] \quad \overline{\mathrm{Exp}}.[\![prog]\!].B' \qquad \geq \qquad \overline{\mathrm{Exp}}.[\![prog]\!].(B \,[\!]\, B')$$

- Probabilistic conjunction

$$x \,\&\, y \quad \widehat{=} \quad (x + y - 1) \,\mathbf{max}\, 0$$

- and its dual, probabilistic disjunction.

$$x \,[\!]\, y \quad \widehat{=} \quad (x + y) \,\mathbf{min}\, 1$$

Conclusions *re* compositionality

- The probabilistic traces, failures and divergences are *observations* of a probabilistic process, but seem to be too weak to be compositional.

- This is as expected, since the same occurs in sequential programming, but...

- We won't know exactly where the difficulties are until the action-system operations themselves have been defined.

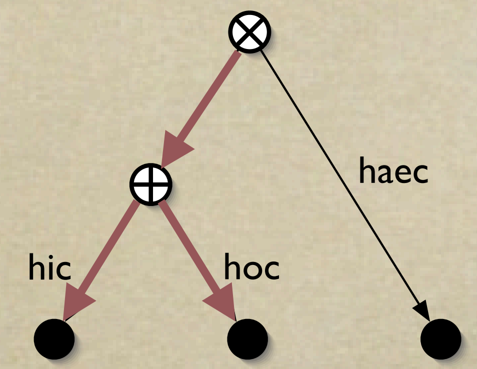
- Of those, parallel composition and hiding seem to be particularly tricky, because of the interaction between probabilistic- and internal choice.

Gavin Lowe.
Representing nondeterministic and probabilistic behaviour in reactive processes.
web.comlab.ox.ac.uk/oucl/work/gavin.lowe/papers/prob.html, 1993.

Conclusions *re* compositionality

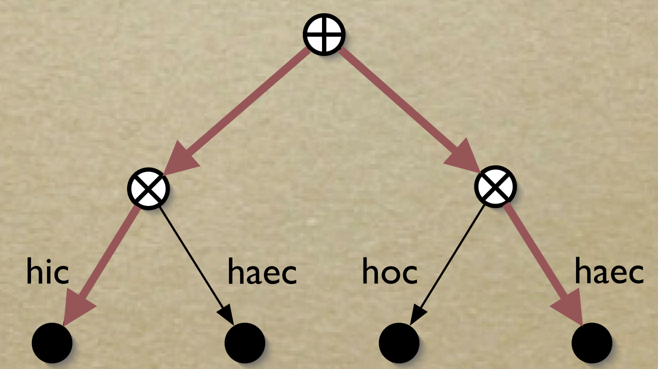
- This process treats **hic** and **hoc** without bias, however much it might favour **haec** over the two together:

$$\begin{array}{ll} & (\mathsf{hic} \rightarrow STOP \;\; {}_{1/2}\oplus \;\; \mathsf{hoc} \rightarrow STOP) \\ \sqcap & \mathsf{haec} \rightarrow STOP \end{array}$$



- But this process might for example execute **hic** half the time and **hoc** not at all:

$$\begin{array}{ll} & (\mathsf{hic} \rightarrow STOP \;\sqcap\; \mathsf{haec} \rightarrow STOP) \\ {}_{1/2}\oplus & (\mathsf{hoc} \rightarrow STOP \sqcap \mathsf{haec} \rightarrow STOP) \end{array}$$



☹ Yet their pFail observations are identical.

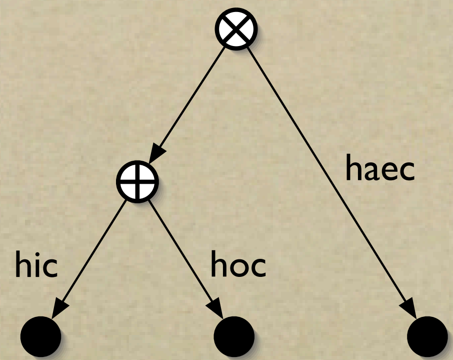
A.K. McIver, C.C. Morgan and J.W. Sanders. Probably Hoare? Hoare Probably! In *Millennial Perspectives in Computer Science*. Davies, Roscoe, Woodcock (eds.), pp 271-282. Palgrave, 2000

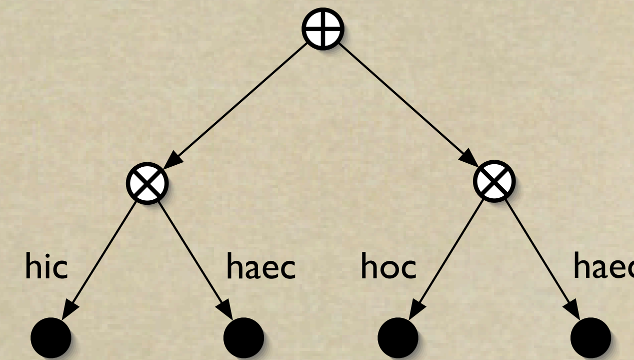
Conclusions *re* compositionality

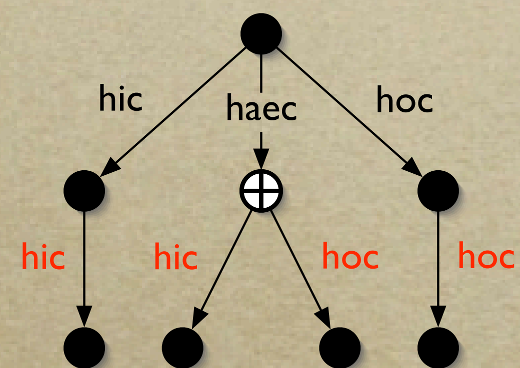
- One could extend the pFail and pDiv operations to act on full *real-valued* expectations rather than only sets of events' guards.

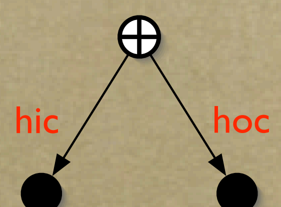
- ☺ This recovers compositionality for sequential programming but, for concurrency, ...

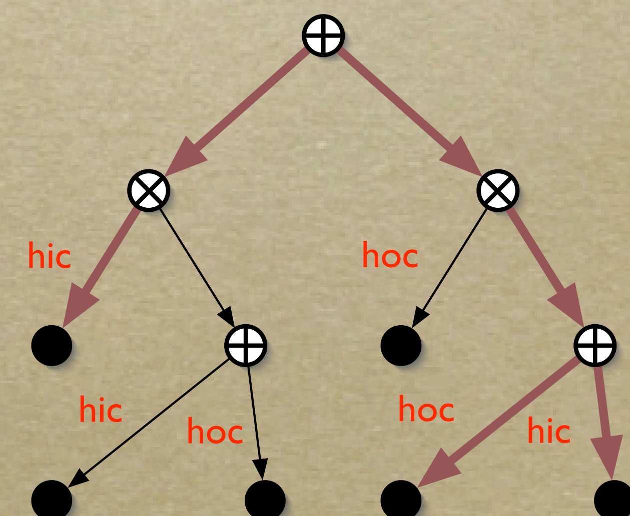
- ? It might be necessary to extend further, to allow more elaborate "tree-like" *and quantitative* testing observations, especially for hiding.

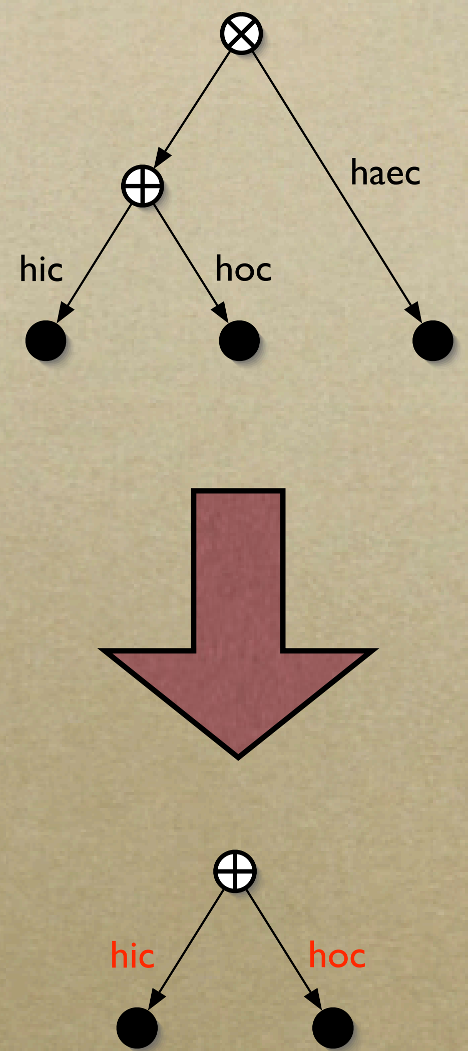

haec
hic
hoc


hic
haec
hoc
haec

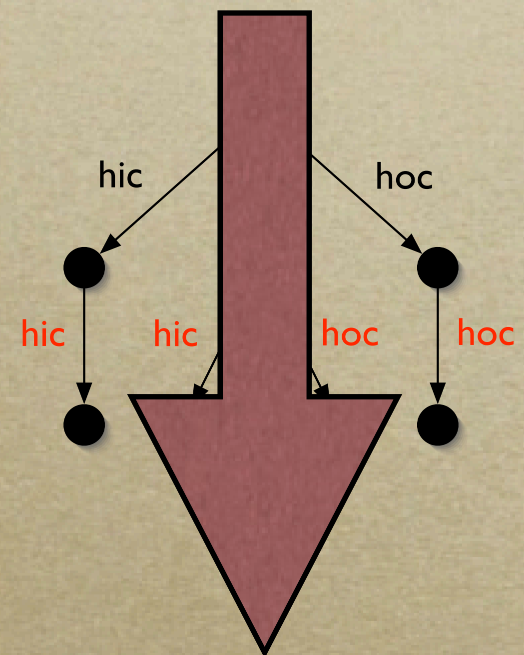

hic
haec
hoc
hic
hic
hoc
hoc


hic
hoc

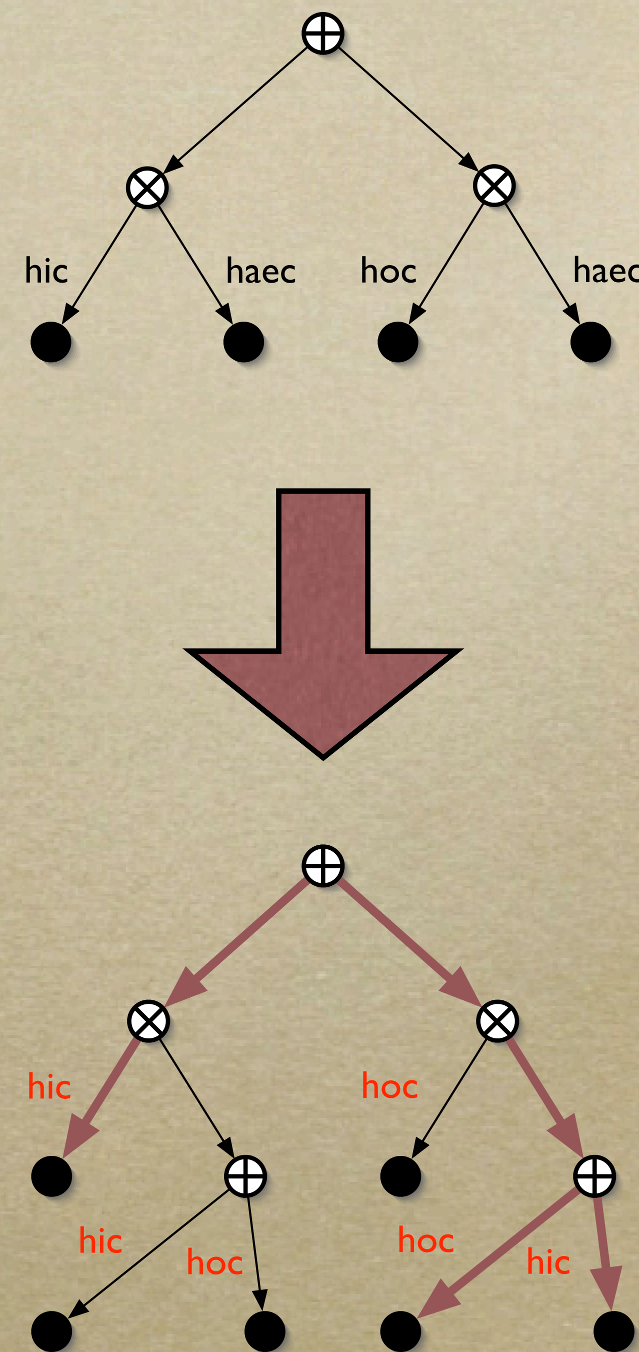

hic
hoc
hic
hoc
hoc
hic

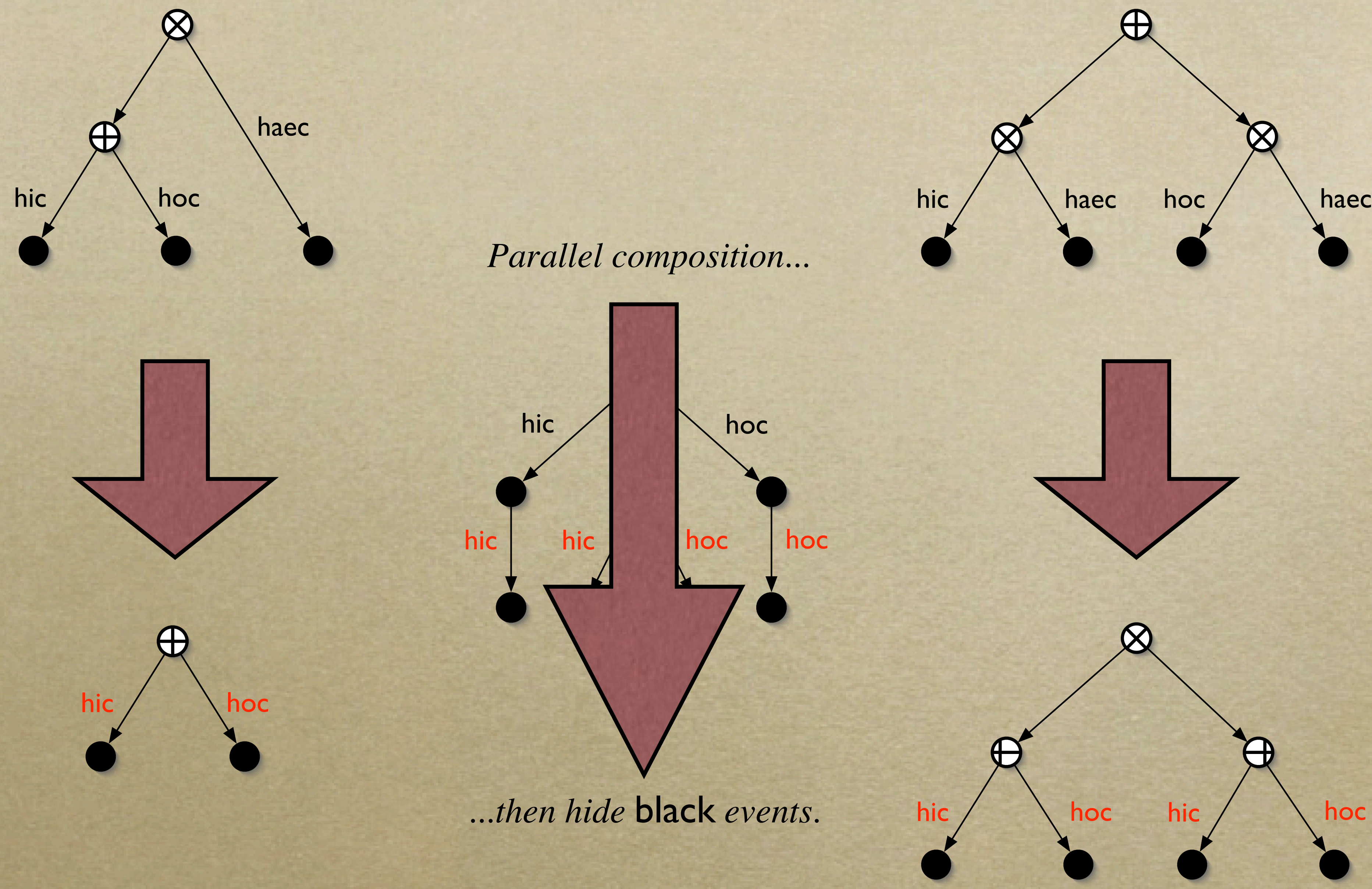


Parallel composition...



...then hide black *events.*




haec
hic
hoc
hic
haec
hoc
haec
Parallel composition...
hic
hoc
hic
hic
hoc
hoc
hic
hoc
...then hide black *events.*
hic
hoc
hic
hoc