

"Performance Probe Specification"

Allan Clark and Stephen Gilmore

LFCS, University of Edinburgh

Allan Clark, Edinburgh, July 2007

"Introduction"

"Performance Measurement Probes"

In this talk I hope you will learn

- ▶ What a performance specification probe is

"Introduction"

"Performance Measurement Probes"

In this talk I hope you will learn

- ▶ What a performance specification probe is
- ▶ The regular-expression-like language for defining probes used in ipc

”Introduction”

”Performance Measurement Probes”

In this talk I hope you will learn

- ▶ What a performance specification probe is
- ▶ The regular-expression-like language for defining probes used in ipc
- ▶ A new method of compilation from probe specification to probe components

"Performance Measure Probes"

"What is a performance measurement probe"

- ▶ A performance measurement probe is a separate component which may be attached to the model to be measured.

"Performance Measure Probes"

"What is a performance measurement probe"

- ▶ A performance measurement probe is a separate component which may be attached to the model to be measured.
- ▶ The probe component observes the actions of the measured model and changes its state accordingly

"Performance Measure Probes"

"What is a performance measurement probe"

- ▶ A performance measurement probe is a separate component which may be attached to the model to be measured.
- ▶ The probe component observes the actions of the measured model and changes its state accordingly
- ▶ To measure the model the probe can then be interrogated for its state.

"Pepa Probes"

"What is required?"

- ▶ We require multi-way synchronisation: $(P \underset{a}{\bowtie} Q) \underset{a}{\bowtie} R$

"Pepa Probes"

"What is required?"

- ▶ We require multi-way synchronisation: $(P \bowtie_a Q) \bowtie_a R$
- ▶ The probe component must be able to passively observe an action from the model and change its state accordingly:

$$R \stackrel{def}{=} (a, \top).R$$

"Pepa Probes"

"What is required?"

- ▶ We require multi-way synchronisation: $(P \bowtie_a Q) \bowtie_a R$
- ▶ The probe component must be able to passively observe an action from the model and change its state accordingly:

$$R \stackrel{def}{=} (a, \top).R$$

- ▶ We require choice such that a component is capable of performing as acting as one of a choice of components:

$$R \stackrel{def}{=} S + T$$

"Very Simple Example"

"Example"



$$\begin{aligned} \textit{Probe} &\stackrel{\textit{def}}{=} (\textit{break}, \top). \textit{Probe}_{\textit{broken}} \\ \textit{Probe}_{\textit{broken}} &\stackrel{\textit{def}}{=} (\textit{repair}, \top). \textit{Probe} \end{aligned}$$

"Very Simple Example"

"Example"



$$\begin{aligned} \textit{Probe} &\stackrel{\textit{def}}{=} (\textit{break}, \top). \textit{Probe}_{\textit{broken}} \\ \textit{Probe}_{\textit{broken}} &\stackrel{\textit{def}}{=} (\textit{repair}, \top). \textit{Probe} \end{aligned}$$



$$\textit{Probe} \underset{\textit{break}, \textit{repair}}{\boxtimes} \textit{Model}$$

"Very Simple Example"

"Example"



$$\begin{aligned} \textit{Probe} &\stackrel{\textit{def}}{=} (\textit{break}, \top). \textit{Probe}_{\textit{broken}} \\ \textit{Probe}_{\textit{broken}} &\stackrel{\textit{def}}{=} (\textit{repair}, \top). \textit{Probe} \end{aligned}$$

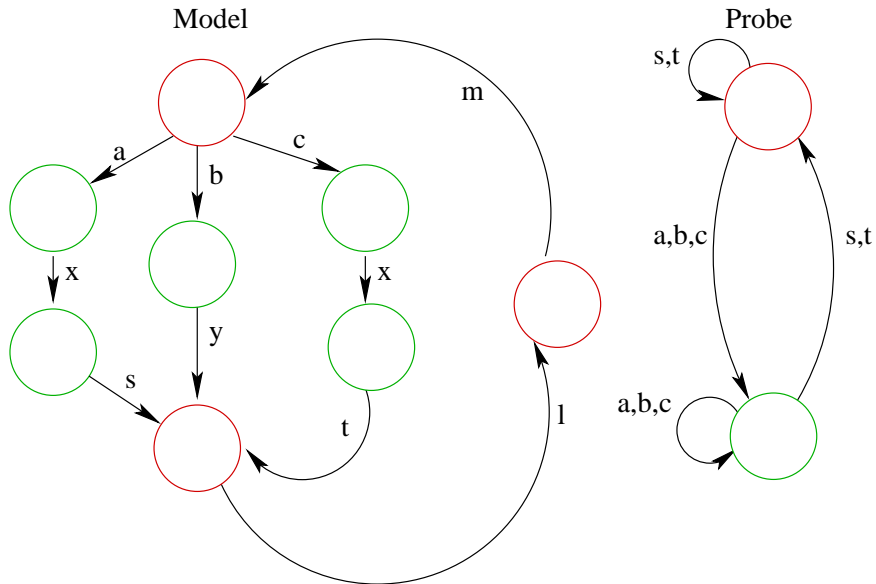


$$\textit{Probe} \underset{\textit{break}, \textit{repair}}{\boxtimes} \textit{Model}$$

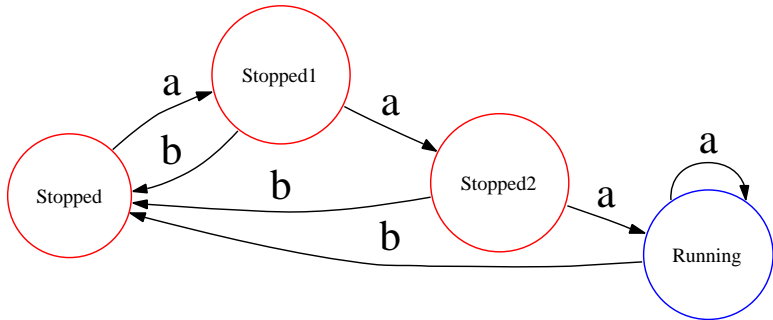


$$\begin{aligned} \textit{Probe} &\stackrel{\textit{def}}{=} (\textit{break}, \top). \textit{Probe}_{\textit{broken}} \\ &+ \underline{(\textit{repair}, \top). \textit{Probe}} \\ \textit{Probe}_{\textit{broken}} &\stackrel{\textit{def}}{=} (\textit{repair}, \top). \textit{Probe} \\ &+ \underline{(\textit{break}, \top). \textit{Probe}_{\textit{broken}}} \end{aligned}$$

"A probe graph"



"A more complex probe graph"



"The probe definitions"

$$\begin{aligned} \textit{Stopped} &\stackrel{\textit{def}}{=} (a, \top).\textit{Stopped1} \\ &+ \underline{(b, \top).\textit{Stopped}} \end{aligned}$$

$$\begin{aligned} \textit{Stopped1} &\stackrel{\textit{def}}{=} (a, \top).\textit{Stopped2} \\ &+ (b, \top).\textit{Stopped} \end{aligned}$$

$$\begin{aligned} \textit{Stopped2} &\stackrel{\textit{def}}{=} (a, \top).\textit{Running} \\ &+ (b, \top).\textit{Stopped} \end{aligned}$$

$$\begin{aligned} \textit{Running} &\stackrel{\textit{def}}{=} (b, \top).\textit{Stopped} \\ &+ \underline{(a, \top).\textit{Running}} \end{aligned}$$

"A regular-expression-like syntax for probes"

R	$:=$	$activity$	Observe action
		R_1, R_2	sequence
		$R_1 \mid R_2$	choice
		$R : label$	labelled
		$R n$	iterate n times
		$R\{m, n\}$	iterate between m and n times
		$R+$	one or more
		R^*	zero or more
		$R?$	zero or one
		$R/activity$	observe R without the activity

"A regular-expression-like syntax for probes"

$(a, a, a)/b : start, b : stop$

$(a\ 3)/b : start, b : stop$

$(a | b | c) : start, (s | t) : stop$

$a : start, (b | c | d), e : stop$

$a : start, (b, c)^*, d : stop$

"The Possible Solution"

$$\begin{aligned} \text{Tr}(P, Q, a) &= P \stackrel{\text{def}}{=} (a, \top).Q \\ \text{Tr}(P, Q, (R_1, R_2)) &= \text{Tr}(P, Q_1, R_1) \cup \text{Tr}(Q_1, Q, R_2) \\ \text{Tr}(P, Q, (R_1 \mid R_2)) &= P \stackrel{\text{def}}{=} P_1 + P_2 \\ &\cup \text{Tr}(P_1, Q, R_1) \\ &\cup \text{Tr}(P_2, Q, R_2) \end{aligned}$$

"The Possible Solution"

$(a, b) \mid (a, c)$

$$P \stackrel{\text{def}}{=} P_1 + P_2$$

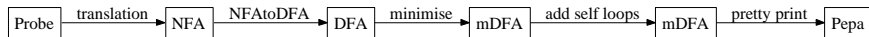
$$P_1 \stackrel{\text{def}}{=} (a, \top).Q_1$$

$$Q_1 \stackrel{\text{def}}{=} (b, \top).Q$$

$$P_2 \stackrel{\text{def}}{=} (a, \top).Q_2$$

$$Q_2 \stackrel{\text{def}}{=} (c, \top).Q$$

"The Probe Implementation"



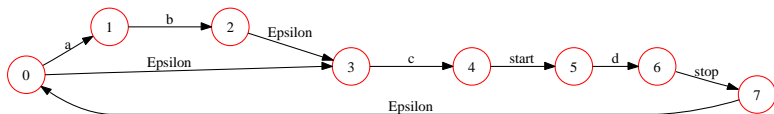
"Another Example"

"Example"

$(a, b)?, c : start, d : stop$

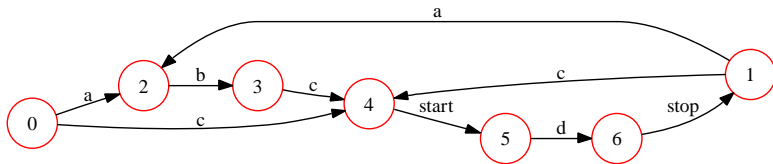
"Non-deterministic Finite Automata"

$(a, b)?, c : start, d : stop$



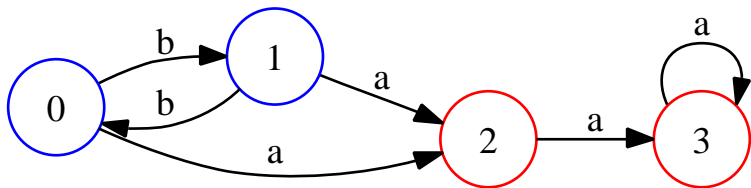
"Deterministic Finite Automata"

$(a, b)?, c : start, d : stop$



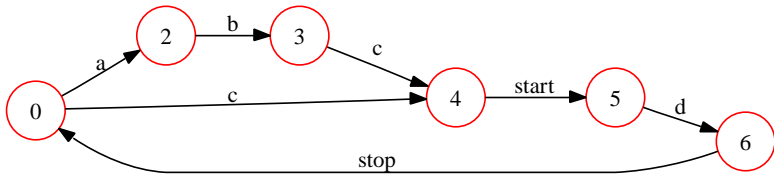
"Minimising the DFA"

- ▶ Two states may be merged if their **outgoing** edges are the same.
- ▶ For the purpose of determining this, either of the two states under consideration are taken to be equal.
- ▶ Here the two blue states and the two red states may be merged.
- ▶ Once merged all transitions which target either state must be retargeted to the merged state.



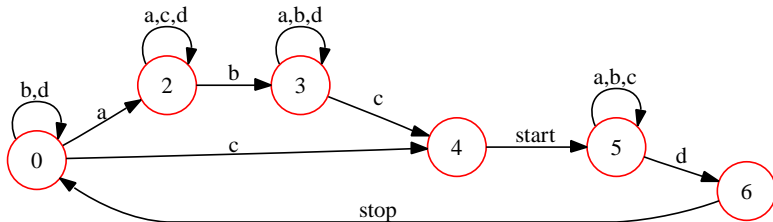
"Minimised Dfa"

$(a, b)?, c : start, d : stop$



"Self-loops Added"

$(a, b)?, c : \text{start}, d : \text{stop}$



"The pretty printed probe definitions"

$$\begin{aligned} \text{Probe}_0 &\stackrel{\text{def}}{=} (a, \top). \text{Probe}_2 \\ &+ (c, \top). \text{Probe}_4 \\ &+ \underline{(b, \top). \text{Probe}_0} \\ &+ \underline{(d, \top). \text{Probe}_0} \end{aligned}$$

$$\begin{aligned} \text{Probe}_2 &\stackrel{\text{def}}{=} (b, \top). \text{Probe}_3 \\ &+ \underline{(a, \top). \text{Probe}_2} \\ &+ \underline{(c, \top). \text{Probe}_2} \\ &+ \underline{(d, \top). \text{Probe}_2} \end{aligned}$$

...

"Future Work"

- ▶ Current we can reset a probe, but not reset a clock:
- ▶ $(a : \textit{start}, b, c) / d, e : \textit{stop}$
- ▶ Matching parentheses is pretty difficult in a regular-expression-like language.

"Any questions?"

google: ipclib pepaprobe

Any Questions?