Performance Evaluation with PAFAS
State of the Art and Ongoing Work

Speaker: M. R. Di Berardini

1Dip. di Matematica e Informatica, Università di Camerino

Paco Meeting, L’Aquila 2-3 marzo 2010
Contents

1 Performance Evaluation with PAFAS

2 Two Measures of Performance

3 FASE

4 An I/O Testing Preorder
A Process algebra for Faster Asynchronous Systems

The set of processes is generated by

\[ P ::= \text{nil} \mid x \mid \alpha.P \mid \overline{\alpha}.P \mid P + P \mid P \parallel A P \mid P[\Phi] \mid \text{rec } x. P \]

where \( \alpha \) is a basic action (either visible or \( \tau \)), \( A \subseteq \Delta \) and \( \Phi \) is a relabeling function.

We distinguish between:

- **patient actions**: actions with time bound 1 - \( \alpha \)
- **urgent actions**: actions with time bound 0 - \( \overline{\alpha} \)

Our processes are patient

\[ a.P \xrightarrow{1} \quad \text{but} \quad a.P \parallel \{a\} a.nil \xrightarrow{1} a.P \parallel \{a\} a.nil \]
Transitional Semantics of PAFAS

• Functional Behaviour

\[ P \xrightarrow{\alpha} Q \quad P \text{ evolves into } Q \text{ performing the action } \alpha \]

• Refusal Behaviour

\[ P \xrightarrow{X} Q \quad \text{It is a conditional time step of duration 1} \]
\[ X \text{ is a set of actions that are not just waiting for a synchronization} \]

\[ P \xrightarrow{1} Q \quad \text{whenever } P \xrightarrow{A} Q \text{ - this is a full time step} \]
The Functional Behaviour of PAFAS-terms

\[
\begin{align*}
\text{Act}_p & : \alpha . P \xrightarrow{\alpha} P \\
\text{Act}_u & : \alpha . P \xrightarrow{\alpha} P \\
\text{Sum} & : P + Q \xrightarrow{\alpha} P' \\
\text{Synch} & : \alpha \in A, \ P \xrightarrow{\alpha} P', \ Q \xrightarrow{\alpha} Q' \\
\text{Par} & : \alpha \notin A, \ P \xrightarrow{\alpha} P' \\
\end{align*}
\]

The other rules are as expected
The Refusal Behaviour of PAFAS-terms

\[
\begin{align*}
\text{NIL}_r & \quad \begin{array}{c} \text{nil} \overset{X}{\longrightarrow}_r \text{nil} \\ \end{array} \\
\text{ACT}_{r1} & \quad \begin{array}{c} \alpha.P \overset{X}{\longrightarrow}_r \alpha.P \\ \end{array} \\
\text{ACT}_{r2} & \quad \begin{array}{c} \alpha \not\in X \cup \{\tau\} \quad \alpha.P \overset{X}{\longrightarrow}_r \alpha.P \\ \end{array} \\
\text{SUM}_r & \quad \begin{array}{c} \frac{P \overset{X}{\longrightarrow}_r P', Q \overset{X}{\longrightarrow}_r Q'}{P + Q \overset{X}{\longrightarrow}_r P' + Q'} \\ \end{array} \\
\text{PAR}_r & \quad \begin{array}{c} \frac{P \overset{X_1}{\longrightarrow}_r P', Q \overset{X_2}{\longrightarrow}_r Q', X \subseteq (A \cap (X_1 \cup X_2)) \cup ((X_1 \cap X_2) \setminus A)}{P \parallel_A Q \overset{X}{\longrightarrow}_r P' \parallel_A Q'} \\ \end{array}
\end{align*}
\]

The other rules are as expected.
Notation:

- The set $DL(P) = \{ v \in (A_\tau \cup \{1\})^* \mid P \xrightarrow{v} \}$ contains the **discrete traces** of $P$.

- The set $RT(P) = \{ v \in (A_\tau \cup 2^A)^* \mid P \xrightarrow{\mu} r \}$ contains the **refusal traces** of $P$.

- The **refusal transition system** $RTS(P)$ of $P$ is an arc-labelled graph, an arc being a transition $\xrightarrow{\alpha}$ or $\xrightarrow{X}_r$. 

---

*Performance Evaluation with PAFAS: Two Measures of Performance* 

FASE: An I/O Testing Preorder
Performance Measures

Based on PAFAS, we provide two performance measures:

1. **A faster-than preorder relation**
   - This is a timed variation of the testing preorder
   - $P$ is faster than $P$ if it is successful in more environments than $Q$, or in the same environments within a shorter time

2. **A performance function**
   - Gives the worst-case time a process need to satisfy a given user behaviour (test)
A **timed test** is a pair \((O, D)\) where:

- \(O\) is a test process (perform the success action \(\omega\))
- \(D \in \mathbb{N}_0\) is a time bound

**\(P\) must \((O, D)\)** if for each \(v \in DL(P \parallel O)\) with duration \(\zeta(v) > D\) contains some \(\omega\)

**\(P \sqsubseteq Q\)** if, for each \((O, D)\), \(Q\) must \((O, D)\) implies \(P\) must \((O, D)\)

**Theorem** (Characterization of the testing preorder):

\[ P \sqsubseteq Q \text{ if and only if } RT(P) \subseteq RT(Q) \]
The *performance function* $p(P, O)$ gives the worst-case time the process $P$ needs to satisfy $O$

$$p(P, O) = \sup \{ \zeta(v) \mid v \in DL(P \parallel O) \text{ and } v \text{ does not contain } \omega \}$$

**Proposition** (Quantitative formulation of the testing preorder):

$P \sqsupseteq Q$ if and only if $p(P, O) \leq p(Q, O)$ for all tests $O$
Response Performance

- Consider

\[
\text{Seq} = \text{rec } x. \; \text{in}. \tau. \text{out}. x, \; \text{and}
\]

\[
\text{Pipe} = (\text{rec } x. \; \text{in}. s. x \parallel \{s\} \; \text{rec } x. s. \text{out}. x) / s
\]

- One would expect that Pipe is faster than Seq, but this is not true.
- Pipe **must** \((in.in.\omega, 1)\), but Seq \frown\textbf{must}(in.in.\omega, 1).
- The expectation that Pipe is faster than Seq is based on some assumption about the users.
- We want to compare these processes w.r.t. their ability to answer a given number of requests as fast as possible.
Response Performance

We compare processes w.r.t. the class of users $\mathcal{U} = \{U_n | n \geq 1\}$

$$U_1 = \text{in.out.}\omega$$
$$U_{n+1} = U_n \parallel \omega \text{ in.out.}\omega$$

Given this class of users, the response performance function is a function $rp_P : \mathbb{N} \longrightarrow \mathbb{N}_0 \cup \{\infty\}$ such that

$$rp_P(n) = \sup\{\zeta(v) | v \in DL(P \parallel U_n) \text{ and } v \text{ does not contain } \omega\}$$

A process $P$ is a response process if:

- it can only perform $\text{in}$ and $\text{out}$ as visible actions,
- it is always able to perform the required number of $\text{out}$ actions and never performs too many

Ex: $\text{Seq} = \text{rec } x.\text{ in.}\tau.\text{out.}x$, but also Pipe
In [CV2005], it has been shown how to calculate the response performance for a response process starting from the refusal transition system $RTS(P)$.

- Remove from $RTS(P)$ all conditional time steps (and all processes that are not reachable anymore) that cannot participate in a full time when considering the behavior of $P \parallel U_n - rRTS(P)$.

Worst-Cycles:
- Catastrophic Cycles
- Bad Cycles
In [CV2005], it has been shown how to calculate the response performance for a response process starting from the refusal transition system $\text{RTS}(P)$.

- Remove from $\text{RTS}(P)$ all conditional time steps (and all processes that are not reachable anymore) that cannot participate in a full time when considering the behavior of $P \parallel U_n - r\text{RTS}(P)$.

- Consider a special subset of paths in $r\text{RTS}(P)$ called $n$-critical paths $\nu \in D\ell(P \parallel U_n)$ that does not contain $\omega$ iff there is an $n$-critical path $\nu$ of $P$ such $\zeta(\nu) = \zeta(\nu)$.

$$r_{P}(n) = \sup\{\zeta(\nu) \mid \nu \in D\ell(P \parallel U_n) \text{ and } \nu \text{ not contain } \omega\}$$

$$= \sup\{\zeta(\nu) \mid \nu \text{ not contain } \omega\}$$

Worst-Cycles:
Catastrophic Cycles
Bad Cycles
Response Performance Calculation

In [CV2005], it has been shown how to calculate the response performance for a response process starting from the refusal transition system $\text{RTS}(P)$.

- Remove from $\text{RTS}(P)$ all conditional time steps (and all processes that are not reachable anymore) that cannot participate in a full time when considering the behavior of $P \parallel U_n - \text{rRTS}(P)$.

- Consider a special subset of paths in $\text{rRTS}(P)$ called $n$-critical paths $\nu \in DL(P \parallel U_n)$ that does not contain $\omega$ iff there is an $n$-critical path $\nu$ of $P$ such $\zeta(\nu) = \zeta(\nu)$.

$$rp_P(n) = \sup\{\zeta(\nu) \mid \nu \in DL(P \parallel U_n) \text{ and } \nu \text{ not contain } \omega\}$$

- Worst-Cycles: **Catastrophic Cycles** and **Bad Cycles**
Catastrophic Cycles

- A cycle in rRTS($P$) is **catastrophic** if it contains a positive number of time steps but no in’s and no out’s.

- Let $P = \text{in.rec } x. (\tau.x + \text{out})$

\[
P \xrightarrow{\text{in}} Q = \text{rec } x. (\tau.x + \text{out}) \xrightarrow{A} r \text{ rec } x. (\tau.x + \text{out}) \xrightarrow{\tau} Q
\]

- Along this cycle time increases without limits, but no ‘useful’ actions are performed.

- As a consequence, $rp_P(n) = \infty$ for each $n$. 
Bad cycles

- For a $P$ without catastrophic cycles, we consider cycles that may be reached from $P$ by a path where all time steps are full and which themselves contains only full time steps.

- The average performance of such a cycle is

\[
\text{the number of time steps} \over \text{the number of the } in\text{'s}
\]

- We call a cycle bad if it is a cycle of maximal average performance in rRTS($P$)
An Example

$$rRTS(\text{Seq})$$

Figure: A cycle with average performance 1
An Example

\[ rRTS(\text{Seq}) \]

\[ \text{Seq} \xrightarrow{\text{in}} \tau.\text{out}.\text{Seq} \xrightarrow{\tau, \Delta \tau} \text{out}.\text{Seq} \xrightarrow{\Delta} \text{out}.\text{Seq} \]

**Figure:** A cycle with average performance 2

**Theorem** (Bad cycles theorem) [CV2005]: For \( P \) without catastrophic cycles, the response performance of \( P \) is asymptotically linear and its asymptotic factor is the average performance of a bad cycle

\[ rps_{\text{Seq}} = 2n + \Theta(1) \]
FASE (Faster Asynchronous Systems Evaluation)

FASE is a tool that support us in the evaluation of the evaluation of the response performance function. Main features:

- **Catastrophic-Cycles Detection**
  The problem has been reduced to that of finding shortest paths in a weighted modification of rRTS($P$)
  Floyd-Warshall algorithm - $O(N^3)$

- **Bad-Cycles Detection**
  The problem has been reduced to that of finding a cycle with minimal average throughput (inverse of the average performance)
  Karp’s algorithm - $O(N^3)$

- **$n$-critical path with maximal duration**
  Currently, FASE adopts an exhaustive research on the state space of rRTS($P$) to find the maximal $n$-critical for a given $n$
FASE (Faster Asynchronous Systems Evaluation)

FASE is a tool that support us in the evaluation of the response performance function. Main features:

- **Catastrophic-Cycles Detection**
- **Bad-Cycles Detection**
  
  The algorithms for Catastrophic-Cycles and Bad-Cycles Detection have been improved

- **n-critical path with maximal duration**
  
  Currently, FASE adopts an exhaustive research on the state space of rRTS(P) to find the maximal n-critical for a given n
Relationship among our Performance Measures

- $P \sqsupseteq Q$ implies $rp_P \leq rp_Q$, but
- $rp_P \leq rp_Q$ does not necessarily imply that $P \sqsupseteq Q$


Relationship among our Performance Measures

- $P \sqsupseteq Q$ implies $rp_P \leq rp_Q$, but
- $rp_P \leq rp_Q$ does not necessarily imply that $P \sqsupseteq Q$


The response performance function only considers a **restricted class of test environments**
Relationship among our Performance Measures

- \( P \sqsubseteq Q \) implies \( rp_P \leq rp_Q \), but
- \( rp_P \leq rp_Q \) does not necessarily imply that \( P \sqsubseteq Q \)


- The response performance function only considers a restricted class of test environments
- We can reasonably expect that if we use such a restricted class of environments also when comparing processes efficiency, (i.e., if we introduce a new efficiency preorder \( \sqsubseteq_{io} \)) then
  \[
  P \sqsubseteq_{io} Q \text{ iff } rp_P \leq rp_Q
  \]
A **i/o timed test** is a pair \((U_n, D)\) where:

- \(U_n\) is a test process in \(\mathcal{U}\)
- \(D \in \mathbb{N}_0\) is a time bound

If \(P\) is a response process we say that \(P\) **must** \((U_n, D)\) if for each \(v \in DL(P \parallel U_n)\) with duration \(\zeta(v) > D\) contains some \(\omega\)

\(P \sqsupseteq_{io} Q\) if, \(n \in \mathbb{N}\) and for each \(D \in \mathbb{N}_0\), \(Q\) **must** \((U_n, D)\) implies \(P\) **must** \((U_n, D)\)
A First Result

**Theorem:** \( P \sqsupseteq_{io} Q \) if and only if \( rp_P \leq rp_Q \)

**Sketch of the proof**

\( P \sqsupseteq_{io} Q \) iff, for each \( n \in \mathbb{N} \),

\[
\sup\{\zeta v \in DL(\tau.P \parallel U_n) \text{ that does not contain } \omega \} \leq \sup\{\zeta v \in DL(\tau.Q \parallel U_n) \text{ that does not contain } \omega \}
\]

iff

\[
\sup\{\zeta(w) \mid w \text{ is an } n\text{-critical path of } P \leq \sup\{\zeta(w) \mid w \text{ is an } n\text{-critical path of } Q\}
\]

iff \( rp_P(n) \leq rp_Q(n) \)
Future Work

- It still missing a finite characterization of the preorder $\sqsubseteq_{io}$
- This finite characterization may also be useful to improve the algorithm that calculate the $n$-critical path with maximal duration
- Instantiating this framework in the timed automata setting
- Possibly with a finite characterization
Thank you for your attention