PSC 2022/23 (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni
http://www.di.unipi.it/~bruni/

http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/start

27 - PEPA
PEPA
Performance Evaluation Process Algebra
Building models

Conceptualise your system as a Markov chain

Construct your Markov chain (infinitesimal generator matrix)

Solve your equations to derive quantitative information
Building models

Conceptualise your system as a Markov chain

Construct your Markov chain (infinitesimal generator matrix)

Solve your equations to derive quantitative information

Monolithic approach: not suitable for complex systems
the PEPA project started in Edinburgh in 1991
motivated by the performance analysis
of large computer and communication systems
exploit interplay between Process Algebras and CTMC

Process Algebras (PA):
compositional description of complex systems,
formal reasoning (for correctness)

CTMC:
numerical analysis

compositional construction of CTMC
PEPA meets CTMC

interaction designed around CTMC
actions have durations
design of independent components
add rates to labels
cooperation between components
probabilistic branching
explicit interaction
quantitative measures
reusable sub-models
probabilistic model checking
easy to understand models
quantitative logics
space reduction techniques
functional verification
Formal models

 qualitative vs quantitative

reachability: will the system arrive to a particular state? how long will it take the system to arrive to a particular state?

(taken from Jane Hillston’s slides)
Formal models

qualitative vs quantitative

conformance: does system behaviour match its specification?

how likely is that system behaviour will match its specification?

does the frequency profile of the system match that of its specification?

(taken from Jane Hillston’s slides)
Formal models

qualitative vs quantitative

The labelled transition system underlying a process algebra model can be used for functional verification e.g.:
- reachability analysis
- specification matching
- model checking

Will the system arrive in a particular state?

Does system behaviour match its specification?

Does a given property hold within the system?

Does a given property hold within the system with a given probability?

How long is it until a given probability hold?

(taken from Jane Hillston’s slides)
Performance Modelling using CTMC

**SYSTEM**  
**MARKOV**  
**Q** = 

**PROCESS**  
**DIAGRAM**  
**TRANSITION**  
**STATE**  

\[ e.g. \text{throughput, response time, utilisation} \]

**PERFORMANCE MEASURES**

\[ \tau = \left( \begin{array}{c} -\sum \cdots \\ -\sum \cdots \\ -\sum \cdots \end{array} \right) \]

\[ \mathcal{P} = \left( \begin{array}{c} p_1, p_2, p_3, \cdots, p_n \end{array} \right) \]

\[ \mathcal{T} = \left( \begin{array}{c} \text{EQUILIBRIUM PROBABILITY DISTRIBUTION} \end{array} \right) \]

\[ \text{(taken from Jane Hillston’s slides)} \]
Communication style

PEPA parallel composition is based on Hoare’s CSP

<table>
<thead>
<tr>
<th>CCS-style</th>
<th>CSP-style</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions and co-actions</td>
<td>no i/o distinction</td>
</tr>
<tr>
<td>binary synchronisation</td>
<td>multiple cooperation</td>
</tr>
<tr>
<td>conjugate sync</td>
<td>shared name sync</td>
</tr>
<tr>
<td>result in a silent action</td>
<td>result in the same name</td>
</tr>
<tr>
<td>restriction</td>
<td>hiding</td>
</tr>
<tr>
<td>parallel composition</td>
<td>cooperation combinator</td>
</tr>
<tr>
<td>one operator</td>
<td>parametric operator</td>
</tr>
</tbody>
</table>
CSP cooperation combinator

\[ P \otimes_L Q \]

cooperation set

interleaving

\[
\begin{align*}
P_1 & \xrightarrow{\alpha} Q_1 & \alpha \notin L & \quad \text{pure interleaving} \quad P \parallel Q \equiv P \otimes_\emptyset Q \\
P_1 \otimes_L P_2 & \xrightarrow{\alpha} Q_1 \otimes_L P_2 & \alpha \notin L & \quad \text{pure interleaving} \quad P \parallel Q \equiv P \otimes_\emptyset Q
\end{align*}
\]

cooperation

\[
\begin{align*}
P_1 & \xrightarrow{\alpha} Q_1 & \alpha \in L & \quad \text{pure interleaving} \quad P \parallel Q \equiv P \otimes_\emptyset Q \\
P_2 & \xrightarrow{\alpha} Q_2 & \alpha \in L & \quad \text{pure interleaving} \quad P \parallel Q \equiv P \otimes_\emptyset Q
\end{align*}
\]
PEPA
syntax and semantics
PEPA syntax

\[ P, Q ::= \text{nil} \quad \text{inactive process} \]

\[ | \quad (\alpha, r).P \quad \text{action prefix} \]

\[ | \quad P + Q \quad \text{choice} \]

\[ | \quad P \parallel_L Q \quad \text{cooperation combinator} \]

\[ | \quad P/L \quad \text{hiding} \]

\[ | \quad C \quad \text{process constant} \]

\[ \alpha \in \Lambda \quad \text{action} \]

\[ L \subseteq \Lambda \quad \text{set of actions} \]

\[ \Delta = \{ C_i \triangleq P_i \}_{i \in I} \quad \text{set of process declarations} \]
ongoing interaction with the environment (with other processes) and its rate

\[ P \xrightarrow{(\alpha, r)} Q \]

a process in its current state

the process state after the interaction

small-step semantics
PEPA semantics (basics)

\[
(\alpha, r).P \xrightarrow{\alpha, r} P
\]

\[
P_1 \xrightarrow{\alpha, r} Q \quad \Rightarrow \quad P_1 + P_2 \xrightarrow{\alpha, r} Q
\]

\[
P_2 \xrightarrow{\alpha, r} Q \quad \Rightarrow \quad P_1 + P_2 \xrightarrow{\alpha, r} Q
\]

\[
C \triangleq P \in \Delta \quad P \xrightarrow{\alpha, r} Q
\]

\[
C \xrightarrow{\alpha, r} Q
\]
Example

Server \triangleq (get, T).(download, \mu).(rel, T).Server

extremely high rate cannot influence the overall rate of interacting components

Browser \triangleq (display, \lambda_1).(cache, m).Browser 
+ (display, \lambda_2).(get, g).(download, T).(rel, r).Browser

a local choice taken with probability \frac{\lambda_i}{\lambda_1 + \lambda_2}
Hiding and interleaving

\[
P \xrightarrow{(\alpha, r)} Q \quad \alpha \notin L
\]

\[
P/L \xrightarrow{(\alpha, r)} Q/L
\]

\[
P_1 \xrightarrow{(\alpha, r)} Q_1 \quad \alpha \notin L
\]

\[
P_1 \otimes P_2 \xrightarrow{(\alpha, r)} Q_1 \otimes P_2
\]

\[
P \xrightarrow{(\alpha, r)} Q \quad \alpha \in L
\]

\[
P/L \xrightarrow{(\tau, r)} Q/L
\]

\[
P_2 \xrightarrow{(\alpha, r)} Q_2 \quad \alpha \notin L
\]

\[
P_2 \otimes P_2 \xrightarrow{(\alpha, r)} P_1 \otimes Q_2
\]
Cooperation

\[ P_1 \xrightarrow{(\alpha, r_1)} Q_1 \quad P_2 \xrightarrow{(\alpha, r_2)} Q_2 \]

\[ P_1 \blacklozenge P_2 \xrightarrow{(\alpha, r)} Q_1 \blacklozenge Q_2 \]

which rate should we put here?
Multiway synchronization

\[ F \overset{\text{def}}{=} (\text{fork}, r_f).(\text{join}, r_j).F' \]
\[ W_1 \overset{\text{def}}{=} (\text{fork}, r_{f_1}).(\text{doWork}_1, r_1).W'_1 \]
\[ W_2 \overset{\text{def}}{=} (\text{fork}, r_{f_2}).(\text{doWork}_2, r_2).W'_2 \]
\[ F' \overset{\text{def}}{=} \ldots, \quad W'_1 \overset{\text{def}}{=} \ldots, \quad W'_2 \overset{\text{def}}{=} \ldots \]
\[ \text{System} \overset{\text{def}}{=} (F \{\text{fork}\} W_1) \{\text{fork}\} W_2 \]

\[
\begin{align*}
F & \xrightarrow{(\text{fork}, r_f)} (\text{join}, r_j)F' \\
& \xrightarrow{(\text{fork}, r'_f)} (\text{join}, r_j).F' \\
W_1 & \xrightarrow{(\text{fork}, r_{f_1})} (\text{doWork}_1, r_1).W'_1 \\
W_2 & \xrightarrow{(\text{fork}, r_{f_2})} (\text{doWork}_2, r_2).W'_2 \\
F & \xrightarrow{(\text{fork}, r'')} (\text{join}, r_j).F' \\
& \xrightarrow{(\text{fork}, r'')} (\text{join}, r_j).F' \\
W_1 & \xrightarrow{(\text{fork}, r_{f_1})} (\text{doWork}_1, r_1)W'_1 \\
W_2 & \xrightarrow{(\text{fork}, r_{f_2})} (\text{doWork}_2, r_2)W'_2 \\
\end{align*}
\]

(taken from Mirco Tribastone’s slides)
Exclusive cooperation

\[
\begin{align*}
\text{Premium} & \overset{\text{def}}{=} (\text{dwn}, r_p).\text{Premium}' \\
\text{Basic} & \overset{\text{def}}{=} (\text{dwn}, r_b).\text{Basic}' \\
S & \overset{\text{def}}{=} (\text{dwn}, r_s).S' \\
\ldots \\
\text{System} & \overset{\text{def}}{=} (\text{Premium} \parallel \text{Basic}) \boxtimes S,
\end{align*}
\]

\[L = \{\text{dwn}\}\]

\[
\begin{array}{ccc}
\text{Premium} & \overset{(dwn,r_p)}{\rightarrow} & \text{Premium}' \\
\text{Premium} \parallel \text{Basic} & \overset{(dwn,r_p)}{\rightarrow} & \text{Premium}' \parallel \text{Basic} \\
\text{Premium} \parallel \text{Basic} \boxtimes S & \overset{(dwn,r_p)}{\rightarrow} & \text{Premium}' \parallel \text{Basic} \boxtimes S' \\
\text{System} & \overset{(dwn,r_p)}{\rightarrow} & \text{Premium}' \parallel \text{Basic} \boxtimes S'
\end{array}
\]
Which rate for sync?

stochastic PA differ for the treatment of rates of synchronised actions

- **TIPP**: new rate is product of individual rates
  \[ r = r_1 \times r_2 \]

- **EMPA**: one participant is passive
  \[ r = r_2 \]

- Bounded capacity: new rate is the minimum of the rates
  \[ r = \min(r_1, r_2) \]

(taken from Jane Hillston’s slides)
Each component has a bounded capacity to carry out activities of some type, determined by the apparent rate for that type. Cooperation cannot make a component exceed its bounded capacity. Thus the apparent rate of a cooperation is the minimum of the apparent rates of the co-operands.
PEPA: apparent rates

No component can be made to carry out an action in cooperation faster than its own defined rate for the actions.

Thus shared actions proceed at the minimum of the rates in the participating components.

The apparent rates of independent actions is instead the sum of their rates within independent concurrent components.
PEPA: apparent rate

$r_\alpha(P)$ is the observed rate of action $\alpha$ in $P$

\[
r_\alpha(P) = r_1 + \cdots + r_n
\]
PEPA: apparent rate

$r_\alpha(P)$ is the observed rate of action $\alpha$ in $P$

$r_\alpha(\text{nil}) \triangleq 0$

$r_\alpha((\beta, r).P) \triangleq \begin{cases} r & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases}$

$r_\alpha(P + Q) \triangleq r_\alpha(P) + r_\alpha(Q)$  
(+ is not idempotent!)

$r_\alpha(P/L) \triangleq \begin{cases} r_\alpha(P) & \text{if } \alpha \not\in L \\ 0 & \text{if } \alpha \in L \end{cases}$

$r_\alpha(P \bowtie_L Q) \triangleq \begin{cases} r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \not\in L \\ \min \{r_\alpha(P), r_\alpha(Q)\} & \text{if } \alpha \in L \end{cases}$

$r_\alpha(C) \triangleq r_\alpha(P) \text{ if } C \triangleq P \in \Delta$

actions are interleaved

the slowest must be waited for
Cooperation

\[ P_1 \xrightarrow{(\alpha, r_1)} Q_1 \quad P_2 \xrightarrow{(\alpha, r_2)} Q_2 \]

\[ P_1 \underset{L}{\bowtie} P_2 \xrightarrow{(\alpha, r)} Q_1 \underset{L}{\bowtie} Q_2 \]

\[ r = r_\alpha(P_1 \underset{L}{\bowtie} P_2) \cdot \frac{r_1}{r_\alpha(P_1)} \cdot \frac{r_2}{r_\alpha(P_2)} \]

apparent rate

the sum of the rates of all the \( \alpha \)-transitions that \( P_1 \underset{L}{\bowtie} P_2 \) can do

\[ \alpha \in L \]

probability of specific action \((\alpha, r_i)\)

among the \( \alpha \)-transitions of \( P_i \)
Cooperation: example

For $r_1, r_2$ positive reals,

\[
(\alpha, r_1).P_1 \xrightarrow{(\alpha, r_1)} P_1 \quad (\alpha, r_2).P_2 \xrightarrow{(\alpha, r_2)} P_2,
\]

\[
(\alpha, r_1).P_1 \{\alpha\} P_2 \xrightarrow{(\alpha, R)} P_1 \{\alpha\} P_2
\]

where

\[
R = \frac{\alpha}{r_\alpha((\alpha, r_1).P_1)} \frac{\alpha}{r_\alpha((\alpha, r_2).P_2)} \min \left( r_\alpha((\alpha, r_1).P_1), r_\alpha((\alpha, r_2).P_2) \right)
\]

\[
= \frac{r_1}{r_\alpha((\alpha, r_1).P_1)} \frac{r_2}{r_\alpha((\alpha, r_2).P_2)} \min(r_1, r_2) = \min(r_1, r_2).
\]

We recover the intuitive definition of the minimum between the two rates.

(taken from Mirco Tribastone’s slides)
Cooperation: example

For $r$ a positive real,

\[
(\alpha, r).P_1 \overset{\alpha, r}{\rightarrow} P_1 \quad (\alpha, \top).P_2 \overset{\alpha, \top}{\rightarrow} P_2,
\]

\[
(\alpha, r).P_1 \Join_{\{\alpha\}} (\alpha, \top).P_2 \overset{\alpha, R}{\rightarrow} P_1 \Join_{\{\alpha\}} P_2
\]

where

\[
R = \frac{r}{r_\alpha((\alpha, r).P_1)} \frac{\top}{r_\alpha((\alpha, \top).P_2)} \min \left( r_\alpha((\alpha, r).P_1), r_\alpha((\alpha, \top).P_2) \right)
\]

\[
= \frac{r}{\frac{r_\top}{\top}} \min(r, \top) = r.
\]

We recover the intuitive definition of infinite capacity — the rate of synchronisation is determined by the active component.

(taken from Mirco Tribastone’s slides)
Apparent rates in active cooperation

\[
\begin{align*}
\text{Cli} & \overset{\text{def}}{=} (\alpha, r_d) \cdot \text{Cli}' \\
\text{Ser} & \overset{\text{def}}{=} (\alpha, r_u) \cdot \text{Ser}' \\
\text{Sys} & \overset{\text{def}}{=} (\text{Cli} \parallel \text{Cli}) \boxtimes \text{Ser} \\
\end{align*}
\]

\[
\begin{align*}
(\alpha, r_d) \cdot \text{Cli}' & \xrightarrow{\alpha, r_d} \text{Cli}' \\
\text{Cli} & \xrightarrow{\alpha, r_d} \text{Cli}' \\
\text{Cli} \parallel \text{Cli} & \xrightarrow{\alpha, r_d} \text{Cli}' \parallel \text{Cli} \\
\text{Cli} \parallel \text{Cli} & \boxtimes \text{Ser} \overset{\alpha, R'}{\xrightarrow{\alpha}} \text{Cli}' \parallel \text{cli} \boxtimes \text{Ser}' \\
\end{align*}
\]

\[
R' = \frac{r_d}{r_d + r_d} r_u \min(r_d + r_d, r_u) = \frac{1}{2} \min(r_d + r_d, r_u)
\]

(taken from Mirco Tribastone’s slides)
Apparent rates in active cooperation

\[
\text{Cli} \overset{\text{def}}{=} (\alpha, r_d) \cdot \text{Cli}' \\
\text{Ser} \overset{\text{def}}{=} (\alpha, r_u) \cdot \text{Ser}' \\
\text{Sys} \overset{\text{def}}{=} (\text{Cli} \parallel \text{Cli}) \boxtimes \text{Ser}
\]

\[
\begin{align*}
(\alpha, r_d) \cdot \text{Cli}' &\xrightarrow{(\alpha, r_d)} \text{Cli}' \\
\text{Cli} &\xrightarrow{(\alpha, r_d)} \text{Cli}' \\
\text{Cli} \parallel \text{Cli} &\xrightarrow{(\alpha, r_d)} \text{Cli} \parallel \text{Cli}' \\
(\alpha, r_u) \cdot \text{Ser}' &\xrightarrow{(\alpha, r_u)} \text{Ser}' \\
\text{Ser} &\xrightarrow{(\alpha, r_u)} \text{Ser}' \\
\text{Cli} \parallel \text{Cli} \boxtimes \text{Ser} &\xrightarrow{\{\alpha\}} \text{Cli} \parallel \text{Cli}' \boxtimes \text{Ser}'
\end{align*}
\]

\[
R'' = \frac{r_d}{r_d + r_u} \frac{r_u}{r_d r_u} \min(r_d + r_d, r_u) = \frac{1}{2} \min(r_d + r_d, r_u) = R'
\]

(taken from Mirco Tribastone’s slides)
Apparent rates in active cooperation

\[ Cli \overset{\text{def}}{=} (\alpha, r_d).Cli' \]
\[ Ser \overset{\text{def}}{=} (\alpha, r_u).Ser' \]
\[ Sys \overset{\text{def}}{=} (Cli \parallel Cli) \{\alpha\} Ser \]

\[ Cli \parallel Cli \{\alpha\} Ser \]

\[ (\alpha, 1/2 \min(2r_d, r_u)) \]

\[ Cli' \parallel Cli \{\alpha\} Ser' \]

\[ (\alpha, 1/2 \min(2r_d, r_u)) \]

\[ Cli \parallel Cli' \{\alpha\} Ser' \]

(taken from Mirco Tribastone’s slides)
Careful with that cooperation set

\[
\begin{align*}
\text{1. } & ((\alpha, r).P \boxdot (\alpha, s).Q) \boxdot (\alpha, t).R \\
\text{2. } & ((\alpha, r).P \parallel (\alpha, s).Q) \boxdot (\alpha, t).R \\
\text{3. } & ((\alpha, r).P \boxdot (\alpha, s).Q) \parallel (\alpha, t).R
\end{align*}
\]

(taken from Jane Hillston’s slides)
Example

Server \( \triangleq (get, \top). (download, \mu). (rel, \top). Server \)
\[ S \triangleq (get, \top). S1 \]
\[ S1 \triangleq (dnd, \mu). S2 \]
\[ S2 \triangleq (rel, \top). S \]

Browser \( \triangleq (display, \lambda_1). (cache, m). Browser \)
\[ + (display, \lambda_2). (get, g). (download, \top). (rel, r). Browser \)
\[ B \triangleq (dis, \lambda_1). B1 + (dis, \lambda_2). B2 \]
\[ B1 \triangleq (cac, m). B \]
\[ B2 \triangleq (get, g). B3 \]
\[ B3 \triangleq (dnd, \top). B4 \]
\[ B4 \triangleq (rel, r). B \]
Example

\[
S \triangleq (\text{get}, \top).S1 \\
S1 \triangleq (\text{dnd}, \mu).S2 \\
S2 \triangleq (\text{rel}, \top).S \\
B \triangleq (\text{dis}, \lambda_1).B1 + (\text{dis}, \lambda_2).B2 \\
B1 \triangleq (\text{cac}, m).B \\
B2 \triangleq (\text{get}, g).B3 \\
B3 \triangleq (\text{dnd}, \top).B4 \\
B4 \triangleq (\text{rel}, r).B
\]

\[
L = \{ \text{get}, \text{dnd}, \text{rel} \}
\]

\[
B \bowtie_L S
\]

independent

\[
\begin{array}{c}
B1 \bowtie_L S \\
B2 \bowtie_L S \\
B3 \bowtie_L S1 \\
B4 \bowtie_L S2
\end{array}
\]

shared
Example

\[ S \triangleq (get, \top).S1 \]

\[ \mathbf{B} \triangleq (\text{dis}, \lambda_1).\mathbf{B1} + (\text{dis}, \lambda_2).\mathbf{B2} \]

\[ S1 \triangleq (\text{dnd}, \mu).S2 \]

\[ \mathbf{B1} \triangleq (\text{cac}, m).\mathbf{B} \]

\[ S2 \triangleq (\text{rel}, \top).\mathbf{S} \]

\[ \mathbf{B2} \triangleq (get, g).\mathbf{B3} \]

\[ \mathbf{B3} \triangleq (\text{dnd}, \top).\mathbf{B4} \]

\[ \mathbf{B4} \triangleq (\text{rel}, r).\mathbf{B} \]

\[ L = \{ \text{get, dnd, rel} \} \]

\[
(\mathbf{B} \parallel \mathbf{B}) \ltimes_{L} \mathbf{S} \\
(\mathbf{B} \parallel \mathbf{B}) \ltimes_{L} \mathbf{S} \xrightarrow{(\text{dis}, \lambda_2)} (\mathbf{B2} \parallel \mathbf{B}) \ltimes_{L} \mathbf{S} \xrightarrow{(\text{dis}, \lambda_2)} (\mathbf{B2} \parallel \mathbf{B2}) \ltimes_{L} \mathbf{S} \\
(\mathbf{B2} \parallel \mathbf{B2}) \ltimes_{L} \mathbf{S} \xrightarrow{(\text{get}, g)} (\mathbf{B3} \parallel \mathbf{B2}) \ltimes_{L} \mathbf{S1} \\
(\mathbf{B2} \parallel \mathbf{B3}) \ltimes_{L} \mathbf{S1} \]

\[ r_{\text{get}}(\mathbf{B2}) = g \]

\[ r_{\text{get}}(\mathbf{B2} \parallel \mathbf{B2}) = 2g \]

\[ r_{\text{get}}(\mathbf{S}) = \top \]

\[ r_{\text{get}}((\mathbf{B2} \parallel \mathbf{B2}) \ltimes_{L} \mathbf{S}) = 2g \]
Consumer/producer

Possible variants:

- A buffer with $n$ places:

  $Buf_n \overset{\text{def}}{=} (\text{get}, \top).Buf_{n-1}$

  $Buf_i \overset{\text{def}}{=} (\text{get}, \top).Buf_{i-1} + (\text{put}, \top).Buf_{i+1}$, for $1 \leq i \leq n - 1$

  $Buf_0 \overset{\text{def}}{=} (\text{put}, \top).Buf_1$

- and $k$ consumers:

  $\underbrace{\text{Cons}_1 \parallel \text{Cons}_1 \parallel \ldots \parallel \text{Cons}_1}_k$

  $\underbrace{\text{Buf}_n \parallel \text{Prod}_1}_{\{\text{get}\}}$}

(taken from Mirco Tribastone’s slides)
Consumer/producer

\[
\begin{align*}
\text{Cons}_1 & \overset{\text{def}}{=} (\text{get}, r_g).\text{Cons}_2 & \text{Prod}_1 & \overset{\text{def}}{=} (\text{make}, r_m).\text{Prod}_2 \\
\text{Cons}_2 & \overset{\text{def}}{=} (\text{cons}, r_c).\text{Cons}_1 & \text{Prod}_2 & \overset{\text{def}}{=} (\text{put}, r_p).\text{Prod}_1 \\
\text{Buf}_2 & \overset{\text{def}}{=} (\text{get}, \top).\text{Buf}_1 & \text{Buf}_1 & \overset{\text{def}}{=} (\text{get}, \top).\text{Buf}_0 + (\text{put}, \top).\text{Buf}_2 \\
\text{Buf}_0 & \overset{\text{def}}{=} (\text{put}, \top).\text{Buf}_1 & \text{Sys} & \overset{\text{def}}{=} \text{Cons}_1 \ltimes \text{Buf}_2 \ltimes \text{Prod}_1
\end{align*}
\]

(taken from Mirco Tribastone’s slides)
### Consumer/producer

#### Definitions:

- **Cons**:
  - $\text{Cons}_1 \overset{\text{def}}{=} (\text{get}, r_g).\text{Cons}_2$
  - $\text{Cons}_2 \overset{\text{def}}{=} (\text{cons}, r_c).\text{Cons}_1$
- **Prod**:
  - $\text{Prod}_1 \overset{\text{def}}{=} (\text{make}, r_m).\text{Prod}_2$
  - $\text{Prod}_2 \overset{\text{def}}{=} (\text{put}, r_p).\text{Prod}_1$
- **Buf**:
  - $\text{Buf}_2 \overset{\text{def}}{=} (\text{get}, \top).\text{Buf}_1$
  - $\text{Buf}_0 \overset{\text{def}}{=} (\text{put}, \top).\text{Buf}_1$
- **Sys**:
  - $\text{Sys} \overset{\text{def}}{=} \text{Cons}_1 \blacklozenge \text{Buf}_2 \blacklozenge \text{Prod}_1$

#### Transitions:

- $\text{Prod}_1 \xrightarrow{(\text{make}, r_m)} \text{Prod}_2$
- $\text{Sys} \xrightarrow{(\text{make}, r_m)} \text{Cons}_1 \blacklozenge \text{Buf}_2 \blacklozenge \text{Prod}_2$

(taken from Mirco Tribastone’s slides)
Consumer/producer

we may denote a state by \( \langle i, j, k \rangle \) to indicate \( \text{Cons}_i \ \{\text{get}\} \ \text{Buf}_j \ \{\text{put}\} \ \text{Prod}_k \)

\begin{itemize}
  \item \( \langle 2, 0, 2 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
      \item \( \text{(get, } r_g \) \)
      \item \( \text{(cons, } r_c \) \)
    \end{itemize}
  \item \( \langle 1, 0, 2 \rangle \):
    \begin{itemize}
      \item \( \text{(put, } r_p \) \)
      \item \( \text{(make, } r_m \) \)
      \item \( \text{(cons, } r_c \) \)
    \end{itemize}
  \item \( \langle 1, 0, 1 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
      \item \( \text{(get, } r_g \) \)
      \item \( \text{(cons, } r_c \) \)
    \end{itemize}
  \item \( \langle 2, 1, 1 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
      \item \( \text{(cons, } r_c \) \)
    \end{itemize}
  \item \( \langle 2, 1, 2 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
      \item \( \text{(cons, } r_c \) \)
    \end{itemize}
  \item \( \langle 1, 2, 1 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
      \item \( \text{(cons, } r_c \) \)
    \end{itemize}
  \item \( \langle 1, 2, 2 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
    \end{itemize}
  \item \( \langle 2, 2, 1 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
    \end{itemize}
  \item \( \langle 2, 2, 2 \rangle \):
    \begin{itemize}
      \item \( \text{(make, } r_m \) \)
    \end{itemize}
\end{itemize}

(taken from Mirco Tribastone’s slides)
Consumer/producer

we may denote a state by \( \langle i, j, k \rangle \) to indicate \( \text{Cons}_i \overset{\text{get}}{\rightarrow} \text{Buf}_j \overset{\text{put}}{\rightarrow} \text{Prod}_k \)

(taken from Mirco Tribastone’s slides)
Bus maps

A third party app receives requests from users for live bus positioning information. It sends requests to the the Google Map API and the TfE Bus Info API and then aggregates the results to present a map view of the bus data which is returned to the user.

Construct a PEPA model to represent this system.

(taken from Jane Hillston’s slides)
Bus maps

A third party app receives requests from users for live bus positioning information. It sends requests to the the Google Map API and the TfE Bus Info API and then aggregates the results to present a map view of the bus data which is returned to the user.

\[
\begin{align*}
\text{User} & \overset{\text{def}}{=} (\text{bus}_{\text{pos}}_{\text{req}}, \text{r}).(\text{bus}_{\text{pos}}_{\text{resp}}, \top).\text{User} \\
\text{Map}\_\text{finder} & \overset{\text{def}}{=} (\text{bus}_{\text{pos}}_{\text{req}}, \text{r}).(\text{google}_{\text{req}}, \lambda_1).\ \\
& \hspace{1cm} (\text{google}_{\text{resp}}, \top).(\text{bus}_{\text{pos}}_{\text{resp}}, \top).\text{Map}\_\text{finder} \\
\text{Bus}\_\text{finder} & \overset{\text{def}}{=} (\text{bus}_{\text{pos}}_{\text{req}}, \text{r}).(\text{tfe}_{\text{req}}, \lambda_2).\ \\
& \hspace{1cm} (\text{tfe}_{\text{resp}}, \top).(\text{bus}_{\text{pos}}_{\text{resp}}, \top).\text{Bus}\_\text{finder} \\
\text{Google} & \overset{\text{def}}{=} (\text{google}_{\text{req}}, \top).(\text{google}_{\text{resp}}, \mu_1).\text{Google} \\
\text{TfE} & \overset{\text{def}}{=} (\text{tfe}_{\text{req}}, \top).(\text{tfe}_{\text{resp}}, \mu_2).\text{Tfe} \\
\text{System} & \overset{\text{def}}{=} \text{User} \boxtimes_L \left( \text{Bus}\_\text{finder} \boxtimes_L \text{Map}\_\text{finder} \right) \boxtimes_K (\text{Google} \parallel \text{TfE}) \\
\end{align*}
\]

where \( L = \{ \text{bus}_{\text{pos}}_{\text{req}}, \text{bus}_{\text{pos}}_{\text{resp}} \} \) and \( K = \{ \text{google}_{\text{req}}, \text{google}_{\text{resp}}, (\text{tfe}_{\text{req}}, \top).(\text{tfe}_{\text{resp}}, \mu_2) \} \).

(taken from Jane Hillston’s slides)
Example

\[\begin{align*}
\text{Proc}_0 & \overset{\text{def}}{=} (\text{task1, } r_1).\text{Proc}_1 \\
\text{Proc}_1 & \overset{\text{def}}{=} (\text{task2, } r_2).\text{Proc}_0 \\
\text{Res}_0 & \overset{\text{def}}{=} (\text{task1, } r_3).\text{Res}_1 \\
\text{Res}_1 & \overset{\text{def}}{=} (\text{reset, } r_4).\text{Res}_0
\end{align*}\]

\[
\begin{pmatrix}
-R & R & 0 & 0 \\
0 & -(r_2 + r_4) & r_4 & r_2 \\
r_2 & 0 & -r_2 & 0 \\
r_4 & 0 & 0 & -r_4
\end{pmatrix}
\]

\[
\begin{align*}
p \cdot Q &= 0 \\
N \sum_{i=1}^{N} p_i &= 1
\end{align*}
\]

(taken from Jane Hillston’s slides)
Example

\[
Q = \begin{pmatrix}
-R & R & 0 & 0 \\
0 & -(r_2 + r_4) & r_4 & r_2 \\
r_2 & 0 & -r_2 & 0 \\
r_4 & 0 & 0 & -r_4
\end{pmatrix}
\]

\[
p \cdot Q = 0 \\
\sum_{i=1}^{N} p_i = 1 \\
R = \min\{r_1, r_3\} = 2
\]

\[
r_1 = 2 \quad r_2 = 2 \quad r_3 = 6 \quad r_4 = 8
\]

\[
p_1 = \frac{20}{41} \quad p_2 = \frac{4}{41} \quad p_3 = \frac{1}{41} \quad p_4 = \frac{16}{41}
\]
Reward structure

\( C \) a set of PEPA components

\( \rho : C \to \mathbb{R} \) a reward structure

\( p \) a steady state distribution

\[
R_\rho \triangleq \sum_i p_i \cdot \rho(C_i)
\]

sometimes rewards are defined in terms of activities

\[
\rho : L \to \mathbb{R}
\]

\[
\rho(C') = \sum_{C \xrightarrow{(\alpha, r)} Q} \rho(\alpha)
\]
Example: throughput

\[ Q = \begin{pmatrix}
-R & R & 0 & 0 \\
0 & -(r_2 + r_4) & r_4 & r_2 \\
0 & 0 & -r_2 & 0 \\
0 & 0 & 0 & -r_4 \\
\end{pmatrix} \]

\[ p_1 = \frac{20}{41} \quad p_2 = \frac{4}{41} \]

\[ p_3 = \frac{1}{41} \quad p_4 = \frac{16}{41} \]

\[ \rho(task_i) = 1 \quad \rho(reset) = 0 \]

\[ \rho(C_1) = \rho(C_2) = \rho(C_3) = 1 \quad \rho(C_4) = 0 \]

\[ R = \frac{20 + 4 + 1}{41} = \frac{25}{41} = 61\% \]
PEPA

further considerations
The importance of being Exp

\[(\alpha, r).Stop \parallel (\beta, s).Stop\]

\[(\alpha, r) \quad (\beta, s)\]

\[Stop \parallel (\beta, s).Stop \quad (\alpha, r).Stop \parallel Stop\]

\[(\beta, s) \quad (\alpha, r)\]

\[Stop \parallel Stop\]

We retain the expansion law of classical process algebra:

\[(\alpha, r).Stop \parallel (\beta, s).Stop = (\alpha, r).(\beta, s).(Stop \parallel Stop) + (\beta, s).(\alpha, r).(Stop \parallel Stop)\]

only if the negative exponential distribution is assumed.

(taken from Jane Hillston’s slides)
Model aggregation

we can exploit CTMC bisimulation to reduce the state space (notion of lumpable partition)

it is the only equivalence that preserves the Markov property

(taken from Jane Hillston’s slides)
Compositionality

PEPA MODEL

\[
\begin{pmatrix}
-\sum & \cdots \\
-\sum & -\sum & \cdots \\
-\sum & -\sum & -\sum & \cdots \\
\dots & \dots & \cdots & \cdots & \cdots \\
-\sum & -\sum & \cdots & \cdots & -\sum \\
\end{pmatrix}
\]

matrix size can grow very large

\[
\lambda = \left( \begin{array}{c}
\cdots \\
-\sum \\
\cdots \\
\end{array} \right)
\]

\[
\Pi = \left( \begin{array}{c}
\cdots \\
\cdots \\
\cdots \end{array} \right)
\]

(taken from Jane Hillston’s slides)
Certain structures in the matrix are known to be amenable to efficient, decomposed solution.

(taken from Jane Hillston’s slides)
Compositionality

lift independent structures to the PEPA model!

(taken from Jane Hillston’s slides)
The final exam of a course consists of a list of 30 questions and a list of 30 answers. Each student has to draw a bijective correspondence between the two lists, linking each question to its answer. The teacher will assign 1 point to each correct link and 0 to each wrong link.

Unfortunately, many students had no time to prepare for the exam, because they had a tight deadline to deliver a project and they will answer completely random.

1. What is the average score for such students?

2. Would the average score be improved by adding more questions and answers?