FORMAL METHODS FOR SYSTEM VERIFICATION

Examples: Labelled multi-transition systems

Sabina Rossi

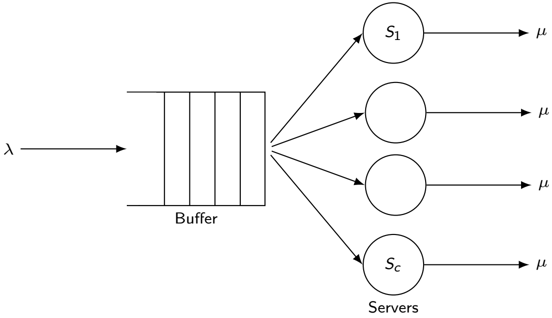
DAIS
Università Ca' Foscari
Venezia

Example 1: Multiple server queue as a single component

Description

We consider a queue with

- c servers
- a buffer with capacity N, where $N > c$
- customers arrive at rate λ
- the service rate of each server is μ



Example 1: Multiple server queue as a single component

PEPA model

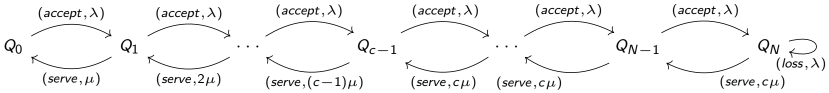
- Let Q_i denote the component representing the behaviour of the queue when there are i costumers present (including those in service).

$$
\begin{array}{lcll}
Q_0 & \stackrel{\text{def}}{=} & (accept, \lambda).Q_1 & \\
\vdots & & & \\
Q_i & \stackrel{\text{def}}{=} & (accept, \lambda).Q_{i+1} + (serve, i\mu).Q_{i-1} & 1 \leq i < c \\
\vdots & & & \\
Q_j & \stackrel{\text{def}}{=} & (accept, \lambda).Q_{j+1} + (serve, c\mu).Q_{j-1} & c \leq j < N \\
\vdots & & & \\
Q_N & \stackrel{\text{def}}{=} & (loss, \lambda).Q_N + (serve, c\mu).Q_{N-1} &
\end{array}
$$

Example 1: Multiple server queue as a single component

Transition diagram

- Following the operational rules we can construct the labelled multi-transition system (also called transition diagram) representing the possible behaviours of a component.
- The transitions are labelled by the activities which they represent.



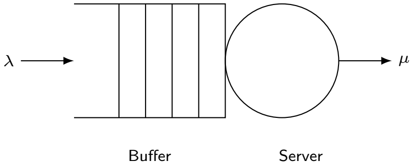
Example 2: Single server queue as two cooperating components

Description

We consider a single server queue with

- buffer capacity N
- customer population N
- customers arrive at rate λ
- the service rate is μ

The arrival process will be suspended when the queue is full as all the costomers will already be present in the queue.



Example 2: Single server queue as two cooperating components

PEPA model of the server

$$\mathit{Server} \stackrel{\mathrm{def}}{=} (\mathit{serve}, \mu).\mathit{Server}$$

Transition diagram

$$\begin{tikzcd} \mathit{Server} \arrow[loop right, "(\mathit{serve}{,}\mu)"] \end{tikzcd}$$

Example 2: Single server queue as two cooperating components

PEPA model of the buffer

$$
\begin{aligned}
Buffer_0 &\stackrel{\text{def}}{=} (accept, \lambda).Buffer_1 \\
&\vdots \\
Buffer_i &\stackrel{\text{def}}{=} (accept, \lambda).Buffer_{i+1} + (serve, \top).Buffer_{i-1} \quad 1 \leq i < N \\
&\vdots \\
Buffer_N &\stackrel{\text{def}}{=} (serve, \top).Buffer_{N-1}
\end{aligned}
$$

Transition diagram

$$
\begin{tikzcd}
Buffer_0 \arrow[r, bend left, "{(accept, \lambda)}"] & Buffer_1 \arrow[l, bend left, "{(serve, \top)}"] \arrow[r, bend left, "{(accept, \lambda)}"] & \cdots \arrow[l, bend left, "{(serve, \top)}"] \arrow[r, bend left, "{(accept, \lambda)}"] & Buffer_{N-1} \arrow[l, bend left, "{(serve, \top)}"] \arrow[r, bend left, "{(accept, \lambda)}"] & Buffer_N \arrow[l, bend left, "{(serve, \top)}"]
\end{tikzcd}
$$

Example 2: Single server queue as two cooperating components

PEPA model of the queue

$$Queue_0 \stackrel{\text{def}}{=} Buffer_0 \underset{\{serve\}}{\bowtie} Server$$

Transition diagram

$$\begin{tikzcd}
Queue_0 \arrow[r, bend left, "(accept\,,\lambda)"] & Queue_1 \arrow[l, bend left, "(serve\,,\mu)"] \arrow[r, bend left, "(accept\,,\lambda)"] & \cdots \arrow[l, bend left, "(serve\,,\mu)"] \arrow[r, bend left, "(accept\,,\lambda)"] & Queue_{N-1} \arrow[l, bend left, "(serve\,,\mu)"] \arrow[r, bend left, "(accept\,,\lambda)"] & Queue_N \arrow[l, bend left, "(serve\,,\mu)"]
\end{tikzcd}$$

Example 3: Simple resource usage as two cooperating components

PEPA model

- The system has two components: *Process* and *Resource*.
- The *Process* will undertake two activities consecutively: *use* with some rate r_1 in cooperation with the resource, and *task* at some rate r_2.
- The *Resource* will engage in two activities consecutively: *use* at some rate r_3 and *update* at a rate r_4.

Example 3: Simple resource usage as two cooperating components

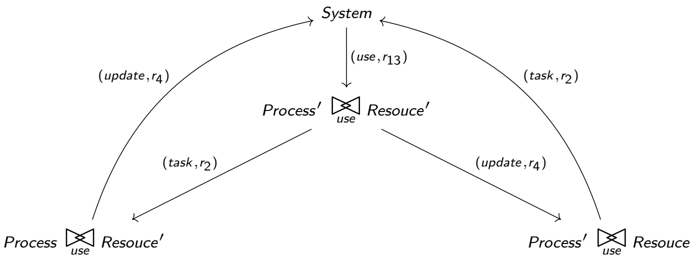
PEPA model of the system

$$
\begin{array}{lcl}
\mathit{Process} & \stackrel{\mathrm{def}}{=} & (\mathit{use}, r_1).\mathit{Process}' \\
\mathit{Process}' & \stackrel{\mathrm{def}}{=} & (\mathit{task}, r_2).\mathit{Process} \\
\mathit{Resource} & \stackrel{\mathrm{def}}{=} & (\mathit{use}, r_3).\mathit{Resource}' \\
\mathit{Resource}' & \stackrel{\mathrm{def}}{=} & (\mathit{update}, r_4).\mathit{Resource} \\
\mathit{System} & \stackrel{\mathrm{def}}{=} & \mathit{Process} \underset{\{\mathit{use}\}}{\bowtie} \mathit{Resource}
\end{array}
$$

Example 3: Simple resource usage as two cooperating components

Transition diagram

Let $r_{13} = min(r_1, r_2)$.



A web server

- Consider a web server which offers html pages for download and only when the transfer is complete will the server be released and available again for acquisition.
- The clients are web browsers, in a domain with a local cache of frequently requested pages.
- Thus any display request may have two possible outcomes: demand for access to data stored at the remote server (with probability p) or demand for access to data available in the local cache (with probability $(1-p)$).
- The browser and the server cooperate when the browser needs to download data which is not available locally.

Exercise 1

A web server: PEPA model

$$
\begin{aligned}
Server &\stackrel{\text{def}}{=} (get, \top).(download, \mu).(rel, \top).Server \\
Browser &\stackrel{\text{def}}{=} (display, p\lambda).(get, g).(download, \top).(rel, r).Browser \\
&\quad + (display, (1-p)\lambda).(cache, m).Browser \\
WEB &\stackrel{\text{def}}{=} Browser \underset{L}{\bowtie} Server
\end{aligned}
$$

where $L = \{get, download, rel\}$.

Exercise 1

(a) Define the set of current action types $\mathcal{A}(WEB)$

$$
\begin{aligned}
\mathcal{A}(WEB) &= (\mathcal{A}(Browser) \setminus L) \cup (\mathcal{A}(Server) \setminus L) \\
&\quad \cup (\mathcal{A}(Browser) \cap \mathcal{A}(Server) \cap L) \\
&= \{display\} \cup \emptyset \cup \emptyset \\
&= \{display\}\,.
\end{aligned}
$$

Exercise 1

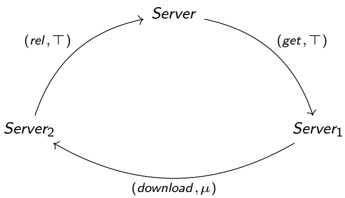
(b) Define the activity multiset $\mathcal{A}ct(WEB)$

$$
\begin{aligned}
\mathcal{A}ct(WEB) &= \mathcal{A}ct_{\setminus L}(Browser) \uplus \mathcal{A}ct_{\setminus L}(Server) \\
&\quad \uplus \{\!|(\alpha, r) |\ \alpha \in L, \\
&\quad \exists(\alpha, r_1) \in \mathcal{A}ct_{\cap L}(Browser), \\
&\quad \exists(\alpha, r_2) \in \mathcal{A}ct_{\cap L}(Server),\ \ r = \\
&\quad \frac{r_1}{r_\alpha(Browser)} \frac{r_2}{r_\alpha(Server)} min(r_\alpha(Browser), r_\alpha(Server))|\!\} \\
&= \{\!|(display, p\lambda), (display, (1-p)\lambda)|\!\} \uplus \emptyset \uplus \emptyset \\
&= \{\!|(display, p\lambda), (display, (1-p)\lambda)|\!\}\,.
\end{aligned}
$$

Exercise 1

(c) Draw the derivation graph of the *Server* component

$$
\begin{aligned}
Server &\stackrel{\text{def}}{=} (get, \top).Server_1 \\
Server_1 &\stackrel{\text{def}}{=} (download, \mu).Server_2 \\
Server_2 &\stackrel{\text{def}}{=} (rel, \top).Server
\end{aligned}
$$



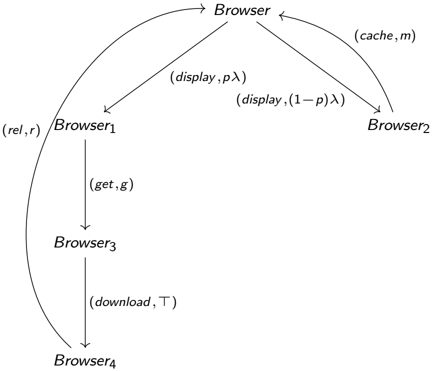
Exercise 1

(*d*) Draw the derivation graph of the *Browser* component

$$
\begin{aligned}
Browser &\stackrel{\mathrm{def}}{=} (display, p\lambda).Browser_1 + (display, (1-p)\lambda).Browser_2 \\
Browser_1 &\stackrel{\mathrm{def}}{=} (get, g).Browser_3 \\
Browser_2 &\stackrel{\mathrm{def}}{=} (cache, m).Browser \\
Browser_3 &\stackrel{\mathrm{def}}{=} (download, \top).Browser_4 \\
Browser_4 &\stackrel{\mathrm{def}}{=} (rel, r).Browser
\end{aligned}
$$

Exercise 1

(d) Draw the derivation graph of the *Browser* component

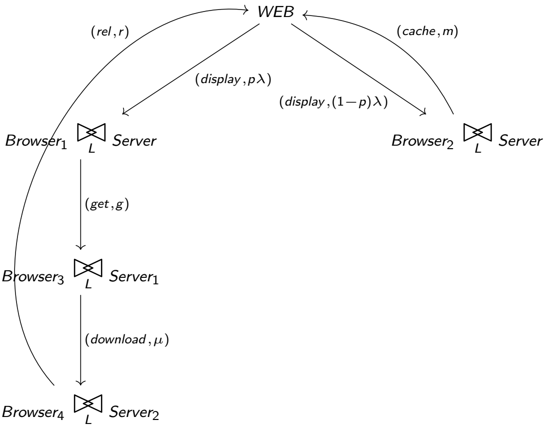


(e) Define the apparent rate $r_{display}(Browser)$

$$
\begin{aligned}
r_{display}(Browser) &= r_{display}((display, p\lambda).Browser_1 + \\
&\qquad (display, (1-p)\lambda).Browser_2)) \\
&= r_{display}((display, p\lambda).Browser_1) + \\
&\quad r_{display}((display, (1-p)\lambda).Browser_2)) \\
&= p\lambda + (1-p)\lambda \\
&= (p + 1 - p)\lambda \\
&= \lambda
\end{aligned}
$$

Exercise 1

(f) Draw the derivation graph of the *WEB* component



Exercise 2

A web server

- Referring to the system above, suppose that we wish to hide the access of the browser to its local cache, i.e.,

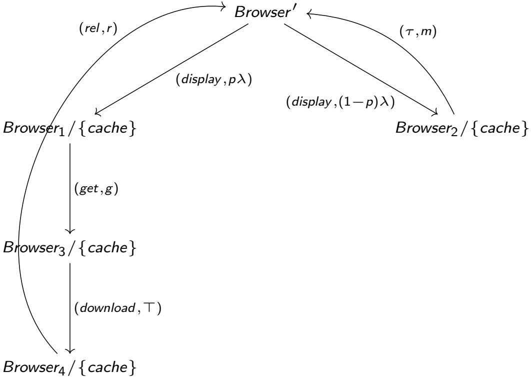
$$Browser' \stackrel{\text{def}}{=} Browser/\{cache\}$$

- A system with two browsers cooperating with the server on action types $L = \{get, download, rel\}$ is represented as:

$$WEB' \stackrel{\text{def}}{=} (Browser' || Browser') \underset{L}{\bowtie} Server$$

Exercise 2

(*a*) Draw the derivation graph of the *Browser′* component



(b) Define the apparent rate $r_{display}(WEB')$

$$
\begin{aligned}
r_{display}(WEB') &= r_{display}((Browser' || Browser') \underset{L}{\bowtie} Server) \\
&\qquad\qquad\qquad\qquad \text{since } display \notin L \\
&= r_{display}(Browser' || Browser') + r_{display}(Server) \\
&= r_{display}(Browser' || Browser') + 0 \\
&= r_{display}(Browser') + r_{display}((Browser') + 0 \\
&= r_{display}(Browser/\{cache\}) + r_{display}(Browser/\{cache\}) + 0 \\
&\qquad\qquad\qquad\qquad \text{since } display \notin \{cache\} \\
&= r_{display}(Browser) + r_{display}((Browser) \\
&= p\lambda + (1-p)\lambda + p\lambda + (1-p)\lambda \\
&= \lambda + \lambda \\
&= 2\lambda
\end{aligned}
$$

Exercise 2

(c) Define the activity multiset $\mathcal{A}ct(WEB')$

$$\begin{aligned}
\mathcal{A}ct(WEB') &= \mathcal{A}ct((Browser' || Browser') \underset{L}{\bowtie} Server) \\
&= \mathcal{A}ct_{\setminus L}(Browser' || Browser') \uplus \mathcal{A}ct_{\setminus L}(Server) \uplus \{|(\alpha, r)| \ \alpha \in L, \\
&\quad \exists(\alpha, r_1) \in \mathcal{A}ct_{\cap L}(Browser' || Browser'), \exists(\alpha, r_2) \in \mathcal{A}ct_{\cap L}(Server), \\
&\quad r = \frac{r_1}{r_\alpha(Browser' || Browser')} \frac{r_2}{r_\alpha(Server)} min(r_\alpha(Browser' || Browser'), r_\alpha(Server))|\} \\
&= \mathcal{A}ct_{\setminus L}(Browser' || Browser') \uplus \emptyset \uplus \emptyset \\
&= \mathcal{A}ct_{\setminus L}(Browser') \uplus \mathcal{A}ct_{\setminus L}(Browser') \\
&= \{|(display, p\lambda), (display, (1-p)\lambda)|\} \uplus \{|(display, p\lambda), (display, (1-p)\lambda)|\} \\
&= \{|(display, p\lambda), (display, (1-p)\lambda), (display, p\lambda), (display, (1-p)\lambda)|\}
\end{aligned}$$

Exercise 3

A web server

- Consider

$$\begin{array}{lcl} \mathit{Browser}'' & \stackrel{\text{def}}{=} & (\mathit{get}, g).(\mathit{download}, \top).(\mathit{rel}, r).\mathit{Browser}'' \\ \mathit{WEB}'' & \stackrel{\text{def}}{=} & (\mathit{Browser}'' \| \mathit{Browser}'') \underset{L}{\bowtie} \mathit{Server} \end{array}$$

where $L = \{\mathit{get}, \mathit{download}, \mathit{rel}\}$.

(a) Determine whether WEB'' is a derivative of WEB

- Recall that

$$WEB = Browser \underset{L}{\bowtie} Server$$

$$WEB'' = (Browser''||Browser'') \underset{L}{\bowtie} Server$$

- We have that $WEB'' \notin ds(WEB)$ since $Browser$ does not contain any cooperation.
- Indeed, each derivative of $Browser \underset{L}{\bowtie} Server$ has the form $Browser_i \underset{L}{\bowtie} Server_i$ where $Browser_i \in ds(Browser)$ and $Server_i \in ds(Server)$.
- Suppose that $WEB'' \in ds(WEB)$, i.e., $(Browser''||Browser'') \underset{L}{\bowtie} Server \in ds(Browser \underset{L}{\bowtie} Server)$. Hence $(Browser''||Browser'') \in ds(Browser)$ but this is not possible since $Browser$ does not contain any cooperation constructor.

(b) Define the set of current action types $\mathcal{A}(WEB'')$

$$
\begin{aligned}
\mathcal{A}(WEB'') &= \mathcal{A}((Browser''||Browser'') \underset{L}{\bowtie} Server) \\
&= (\mathcal{A}(Browser''||Browser'') \setminus L) \cup (\mathcal{A}(Server) \setminus L) \\
&\quad \cup (\mathcal{A}(Browser''||Browser'') \cap \mathcal{A}(Server) \cap L) \\
&= \emptyset \cup \emptyset \cup \{get\} \\
&= \{get\}\,.
\end{aligned}
$$

(c) Define the apparent rate $r_{get}(WEB'')$

$$
\begin{aligned}
r_{get}(WEB'') &= r_{get}((Browser''||Browser'') \underset{L}{\bowtie} Server) \\
& \qquad\qquad\qquad\qquad \text{since } get \in L \\
&= min(r_{get}(Browser''||Browser''),\ r_{get}(Server)) \\
&= min(r_{get}(Browser'' \underset{\emptyset}{\bowtie} Browser''),\ r_{get}(Server)) \\
& \qquad\qquad\qquad\qquad \text{since } get \notin \emptyset \\
&= min(r_{get}(Browser'') + r_{get} Browser''),\ r_{get}(Server)) \\
&= min(g + g, \top) \\
&= g + g \\
&= 2g
\end{aligned}
$$

(d) Define the activity multiset $\mathcal{A}ct(WEB'')$

$$
\begin{aligned}
\mathcal{A}ct(WEB'') &= \mathcal{A}ct((Browser''||Browser'') \underset{L}{\bowtie} Server) \\
&= \mathcal{A}ct_{\setminus L}(Browser''||Browser'') \uplus \mathcal{A}ct_{\setminus L}(Server) \uplus \{|(\alpha, r)|\ \alpha \in L, \\
&\quad \exists(\alpha, r_1) \in \mathcal{A}ct_{\cap L}(Browser''||Browser''), \exists(\alpha, r_2) \in \mathcal{A}ct_{\cap L}(Server), \\
&\quad r = \frac{r_1}{r_\alpha(Browser''||Browser'')} \frac{r_2}{r_\alpha(Server)} min(r_\alpha(Browser''||Browser''), r_\alpha(Server))|\} \\
&= \emptyset \uplus \emptyset \uplus \{|(get, g), (get, g)|\} \\
&= \{|(get, g), (get, g)|\}
\end{aligned}
$$

Producer-Consumer

- Consider the following PEPA model which describes a producer which puts goods into one of two buffers. These goods are extracted from each of the buffers by the consumer which is associated with that buffer. Thus the producer and the buffers cooperate on the *put* action and the consumers cooperate with their buffer on the *get* action.

$$\begin{array}{lcl}
\mathit{Producer} & \stackrel{\text{def}}{=} & (\mathit{put}, \lambda).\mathit{Producer} \\
\mathit{Consumer} & \stackrel{\text{def}}{=} & (\mathit{get}, \mu).\mathit{Consumer} \\
\mathit{Buffer} & \stackrel{\text{def}}{=} & (\mathit{put}, \top).(\mathit{get}, \top).\mathit{Buffer} \\
\mathit{System} & \stackrel{\text{def}}{=} & \mathit{Producer} \underset{\{\mathit{put}\}}{\bowtie} \big((\mathit{Buffer} \underset{\{\mathit{get}\}}{\bowtie} \mathit{Consumer}) \| (\mathit{Buffer} \underset{\{\mathit{get}\}}{\bowtie} \mathit{Consumer})\big)
\end{array}$$

For simplicity, we use the following abbrevations:

$$\begin{array}{lcl}
\mathit{BC} & \stackrel{\text{def}}{=} & \mathit{Buffer} \underset{\{\mathit{get}\}}{\bowtie} \mathit{Consumer} \\
\mathit{2BC} & \stackrel{\text{def}}{=} & (\mathit{Buffer} \underset{\{\mathit{get}\}}{\bowtie} \mathit{Consumer}) \| (\mathit{Buffer} \underset{\{\mathit{get}\}}{\bowtie} \mathit{Consumer})
\end{array}$$

Exercise 4

(a) Define the set of current action types $\mathcal{A}(2BC)$

$$
\begin{aligned}
\mathcal{A}(2BC) &= \mathcal{A}(BC) \cup \mathcal{A}(BC) \\
&= (\mathcal{A}(Buffer) \setminus \{get\}) \cup (\mathcal{A}(Consumer) \setminus \{get\}) \cup (\mathcal{A}(Buffer) \cap \mathcal{A}(Consumer) \cap \{get\}) \\
&\quad \cup (\mathcal{A}(Buffer) \setminus \{get\}) \cup (\mathcal{A}(Consumer) \setminus \{get\}) \cup (\mathcal{A}(Buffer) \cap \mathcal{A}(Consumer) \cap \{get\}) \\
&= (\{put\} \cup \emptyset \cup \emptyset) \cup (\{put\} \cup \emptyset \cup \emptyset) \\
&= \{put\}
\end{aligned}
$$

Exercise 4

(b) Define the set of current action types $\mathcal{A}(System)$

$$
\begin{aligned}
\mathcal{A}(System) &= \mathcal{A}\Big(Producer \underset{\{put\}}{\bowtie} \Big((Buffer \underset{\{get\}}{\bowtie} Consumer) \| (Buffer \underset{\{get\}}{\bowtie} Consumer)\Big)\Big) \\
&= (\mathcal{A}(Producer) \setminus \{put\}) \cup (\mathcal{A}(2BC) \setminus \{put\}) \cup (\mathcal{A}(Producer) \cap \mathcal{A}(2BC) \cap \{put\}) \\
&= \emptyset \cup \emptyset \cup \{put\} \\
&= \{put\}
\end{aligned}
$$

Exercise 4

(c) Define the activity multiset $\mathcal{A}ct(2BC)$

$$
\begin{aligned}
\mathcal{A}ct(2BC) &= \mathcal{A}ct\Big((Buffer \underset{\{get\}}{\bowtie} Consumer) \| (Buffer \underset{\{get\}}{\bowtie} Consumer)\Big) \\
&= \mathcal{A}ct(Buffer \underset{\{get\}}{\bowtie} Consumer) \uplus \mathcal{A}ct(Buffer \underset{\{get\}}{\bowtie} Consumer) \\
&= \mathcal{A}ct_{\backslash\{get\}}(Buffer) \uplus \mathcal{A}ct_{\backslash\{get\}}(Consumer) \uplus \emptyset \\
&\quad \uplus\ \mathcal{A}ct_{\backslash\{get\}}(Buffer) \uplus \mathcal{A}ct_{\backslash\{get\}}(Consumer) \uplus \emptyset \\
&= \{\!|(put, \top)|\!\} \uplus \emptyset \uplus \emptyset \uplus \{\!|(put, \top)|\!\} \uplus \emptyset \uplus \emptyset \\
&= \{\!|(put, \top), (put, \top)|\!\}
\end{aligned}
$$

Exercise 4

(d) Define the activity multiset $\mathcal{A}ct(System)$

$$
\begin{aligned}
\mathcal{A}ct(System) \quad &= \quad \mathcal{A}ct\Big(Producer \underset{\{put\}}{\bowtie} \Big((Buffer \underset{\{get\}}{\bowtie} Consumer)\|(Buffer \underset{\{get\}}{\bowtie} Consumer)\Big)\Big) \\
&= \quad \mathcal{A}ct_{\setminus\{put\}}(Producer) \uplus \mathcal{A}ct_{\setminus\{put\}}(2BC) \uplus \{|(\alpha, r)| \ \alpha \in \{put\}, \\
&\quad\quad \exists(\alpha, r_1) \in \mathcal{A}ct_{\cap\{put\}}(Producer), \exists(\alpha, r_2) \in \mathcal{A}ct_{\cap\{put\}}(2BC), \\
&\quad\quad r = \frac{r_1}{r_\alpha(Producer)} \frac{r_2}{r_\alpha(2BC)} min(r_\alpha(Producer), r_\alpha(2BC))|\} \\
&= \quad \emptyset \uplus \emptyset \uplus \{|(put, r)| \ \exists(put, r_1) \in \mathcal{A}ct_{\cap\{put\}}(Producer), \exists(put, r_2) \in \mathcal{A}ct_{\cap\{put\}}(2BC), \\
&\quad\quad r = \frac{r_1}{r_{put}(Producer)} \frac{r_2}{r_{put}(2BC)} min(r_{put}(Producer), r_{put}(2BC))|\} \\
&= \quad \{|(put, r)| \ \exists(put, r_1) \in \mathcal{A}ct_{\cap\{put\}}(Producer), \exists(put, r_2) \in \mathcal{A}ct_{\cap\{put\}}(2BC), \\
&\quad\quad r = \frac{\lambda}{\lambda} \cdot \frac{\top}{2\top} \cdot min(\lambda, 2\top)|\} \\
&\quad\quad \{|(put, \tfrac{\lambda}{2}), (put, \tfrac{\lambda}{2})|\}
\end{aligned}
$$

(e) Define $r_{get}(2BC)$

$$
\begin{aligned}
r_{get}(2BC) &= r_{get}(BC) + r_{get}(BC) \\
&= r_{get}(Buffer \underset{\{get\}}{\bowtie} Consumer) + r_{get}(Buffer \underset{\{get\}}{\bowtie} Consumer) \\
&= min(r_{get}(Buffer), r_{get}(Consumer)) + min(r_{get}(Buffer), r_{get}(Consumer)) \\
&= min(0, \mu) + min(0, \mu) \\
&= 0
\end{aligned}
$$

Exercise 4

(f) Define $r_{put}(2BC)$

$$
\begin{aligned}
r_{put}(2BC) &= r_{put}(BC) + r_{put}(BC) \\
&= r_{put}(Buffer \underset{\{get\}}{\bowtie} Consumer) + r_{put}(Buffer \underset{\{get\}}{\bowtie} Consumer) \\
&= r_{put}(Buffer) + r_{put}(Consumer) + r_{put}(Buffer) + r_{put}(Consumer) \\
&= \top + 0 + \top + 0 \\
&= 2\top
\end{aligned}
$$

(g) Define $r_{put}(System)$

$$
\begin{aligned}
r_{put}(System) &= min(r_{put}(Producer), r_{put}(2BC)) \\
&= min(\lambda, 2T) \\
&= \lambda
\end{aligned}
$$

Exercise 4

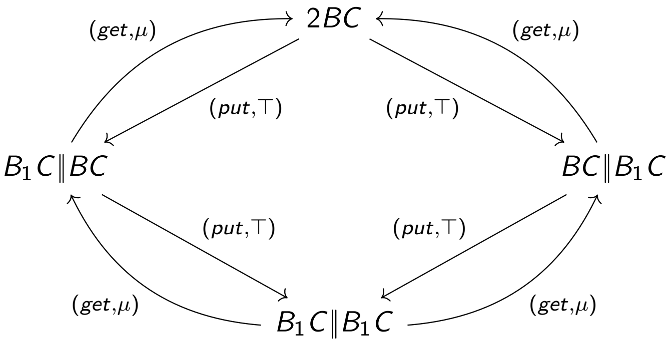
(h) Draw the derivation graph of the BC component

$$\begin{tikzcd} BC \arrow[rr, bend left, "(put{,}\top)"] & & B_1C \arrow[ll, bend left, "(get{,}\mu)"] \end{tikzcd}$$

where

$$\begin{aligned} BC &\stackrel{\text{def}}{=} Buffer \underset{\{get\}}{\bowtie} Consumer \\ B_1C &\stackrel{\text{def}}{=} (get, \top).Buffer \underset{\{get\}}{\bowtie} Consumer \end{aligned}$$

(*h*) Draw the derivation graph of the $2BC$ component



(h) Draw the derivation graph of the *System* component

$$
\begin{tikzcd}[column sep=large, row sep=huge]
 & System \arrow[dl, "(put{,}\frac{\lambda}{2})"] \arrow[dr, "(put{,}\frac{\lambda}{2})"'] & \\
P \underset{\{put\}}{\bowtie} B_1C \| BC \arrow[ur, bend left, "(get{,}\mu)"] \arrow[dr, "(put{,}\lambda)"] & & P \underset{\{put\}}{\bowtie} BC \| B_1C \arrow[ul, bend right, "(get{,}\mu)"'] \arrow[dl, "(put{,}\lambda)"'] \\
 & P \underset{\{put\}}{\bowtie} B_1C \| B_1C \arrow[ul, bend left, "(get{,}\mu)"] \arrow[ur, bend right, "(get{,}\mu)"'] &
\end{tikzcd}
$$

Exercise 5

Two independet parallel processes

- Consider the PEPA component:

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} (start, r_1).P_2 \\ P_2 &\stackrel{\text{def}}{=} (run, r_2).P_3 \\ P_3 &\stackrel{\text{def}}{=} (stop, r_3).P_1 \end{aligned}$$

Exercise 5

(a) Determine the derivative set of $P_1 \| P_1$, $ds(P_1 \| P_1)$

$$
\begin{aligned}
ds(P_1 \| P_1) \quad = \quad \{ \; & P_1 \| P_1 \,, \\
& P_1 \| P_2 \,, \\
& P_1 \| P_3 \,, \\
& P_2 \| P_1 \,, \\
& P_2 \| P_2 \,, \\
& P_2 \| P_3 \,, \\
& P_3 \| P_1 \,, \\
& P_3 \| P_2 \,, \\
& P_3 \| P_3 \}
\end{aligned}
$$

Exercise 5

(b) Define the apparent rates $r_{run}(P_2 \| P_2)$ and $r_{run}(P_2 \underset{\{run\}}{\bowtie} P_2)$

$$
\begin{aligned}
r_{run}(P_2 \| P_2) &= r_{run}(P_2) + r_{run}(P_2) \\
&= r_2 + r_2 \\
&= 2r_2\,.
\end{aligned}
$$

$$
\begin{aligned}
r_{run}(P_2 \underset{\{run\}}{\bowtie} P_2) &= min(r_{run}(P_2), r_{run}(P_2)) \\
&= r_{run}(P_2) \\
&= r_2\,.
\end{aligned}
$$