Choreographic Development of Message-Passing Applications

Alex Coto @ GSSI, IT

Roberto Guanciale @ KTH, SE

Emilio Tuosto @ GSSI, IT & UoL, UK

Coordination 2020 15-20 July 2020

Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233.

In the next 90 minutes...

Prologue	An intuitive account
Act I	Some definitions
Act II	A tool
Act III	A little exercise
Epilogue	Work in progress

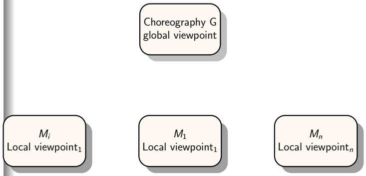
– Prologue –

[An intuitive account]

“Top-down”

Quoting W3C

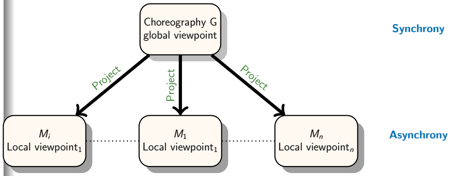
“Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]”



"Top-down"

Quoting W3C

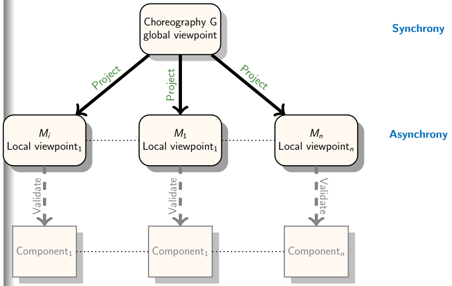
"Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]"



"Top-down"

Quoting W3C

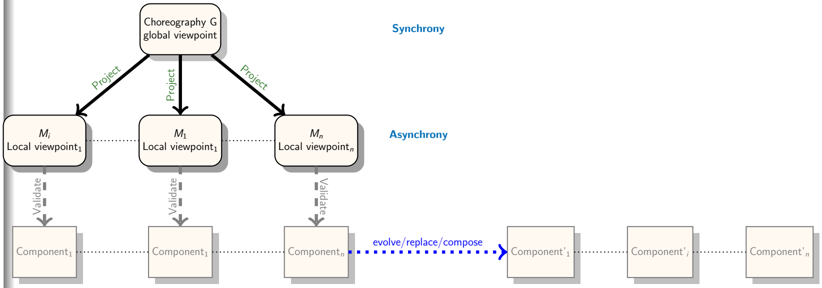
"Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]"



“Top-down” & “Bottom-up”

Quoting W3C

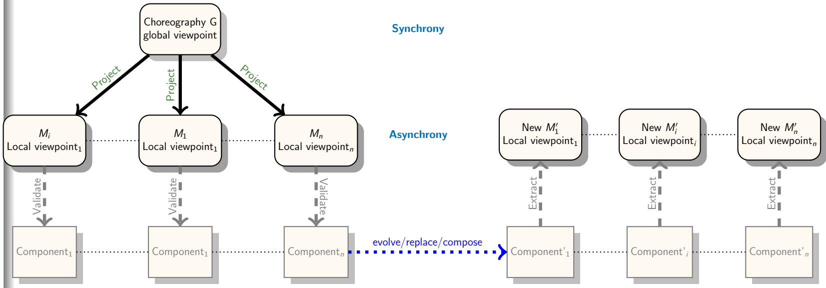
“Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]”



“Top-down” & “Bottom-up”

Quoting W3C

“Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]”



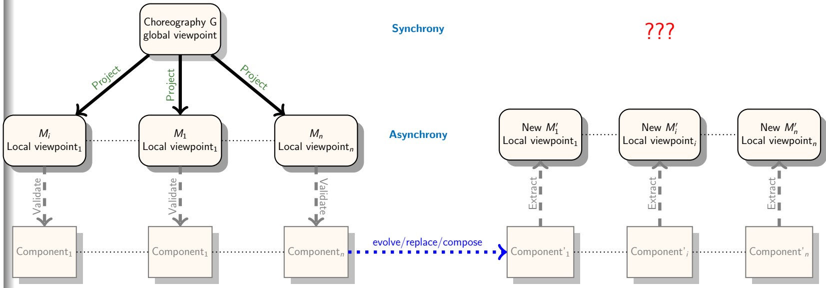
bottom-up

Extract from each component its local viewpoint, combine the local view points in a choreography...if that makes sense [Lange et al., 2015]

“Top-down” & “Bottom-up”

Quoting W3C

“Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]”



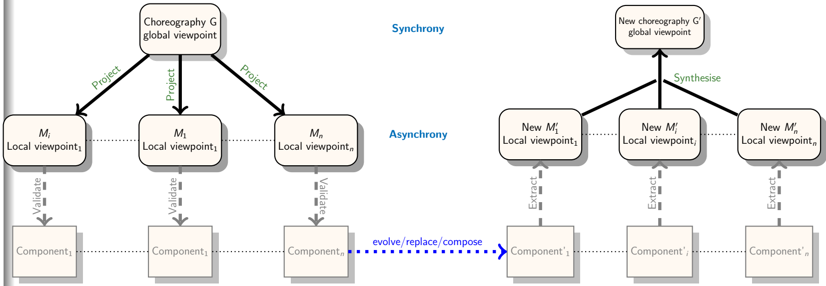
bottom-up

Extract from each component its local viewpoint, combine the local view points in a choreography...if that makes sense [Lange et al., 2015]

"Top-down" & "Bottom-up"

Quoting W3C

"Using the Web Services Choreography specification, a contract containing a global definition of the common ordering conditions and constraints under which messages are exchanged, is produced that describes, from a global viewpoint [...] observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it. The global specification is in turn realised by combination of the resulting local systems [...]"

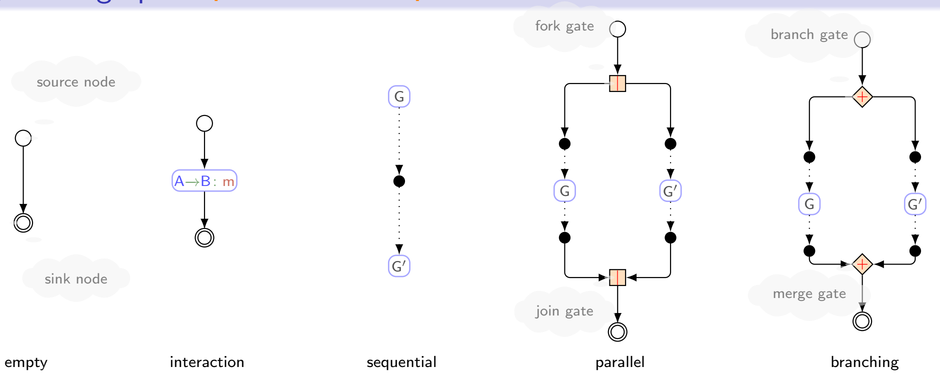


bottom-up

Extract from each component its local viewpoint, combine the local view points in a choreography...if that makes sense [Lange et al., 2015]

Global views, intuitively

g-choreographies [Tuosto and Guanciale, 2018]

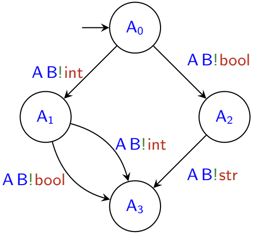


Pomset or (Event Structure[a])

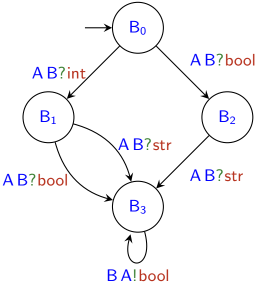
[a] See Ugo de'Liguoro's talk @ ICE 2020

Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



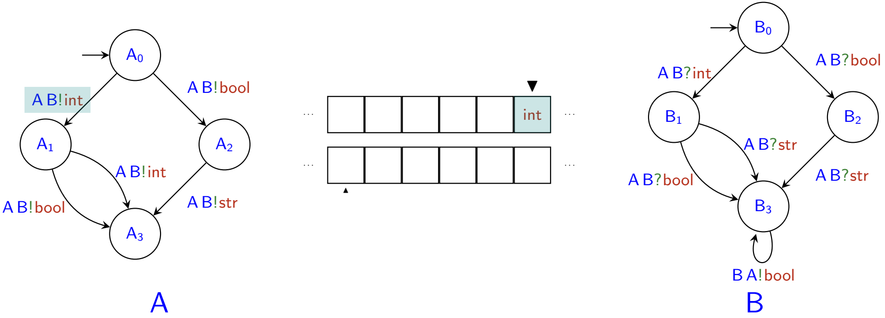
A



B

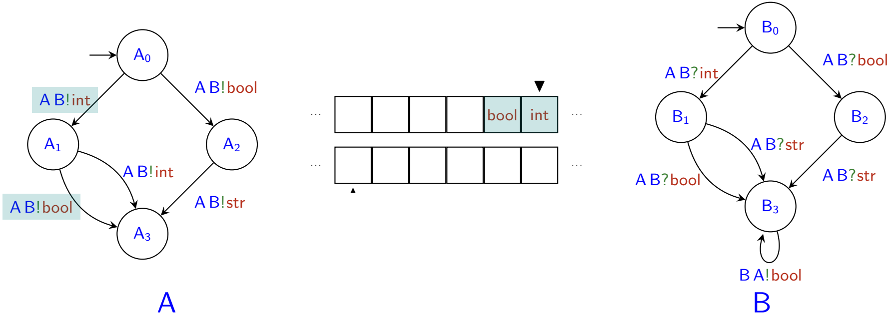
Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



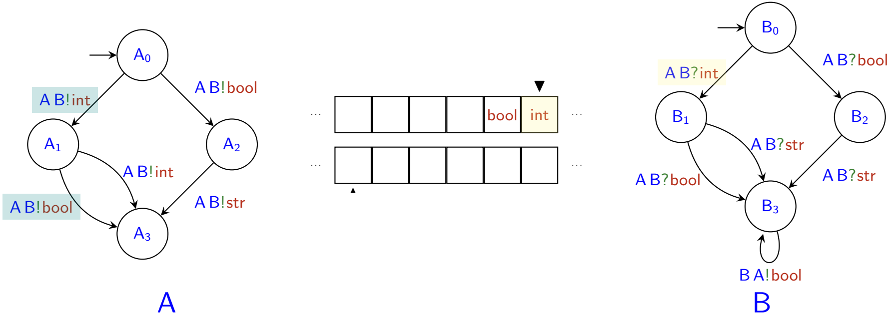
Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



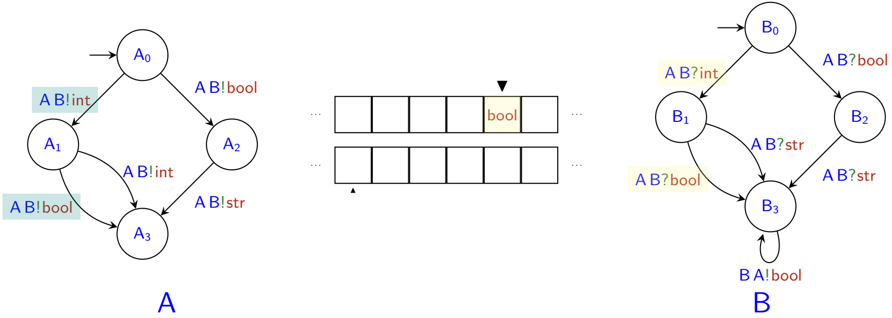
Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



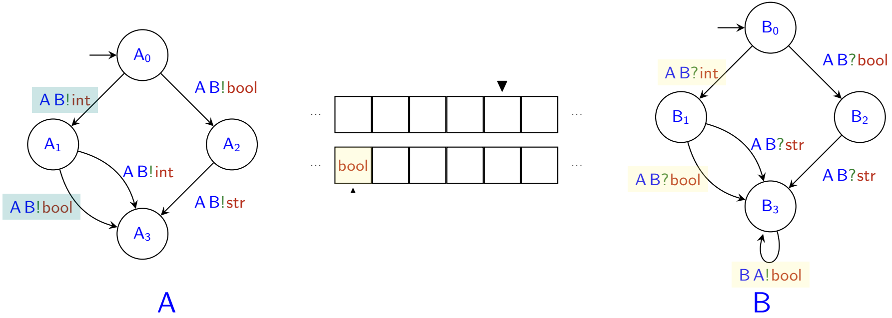
Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



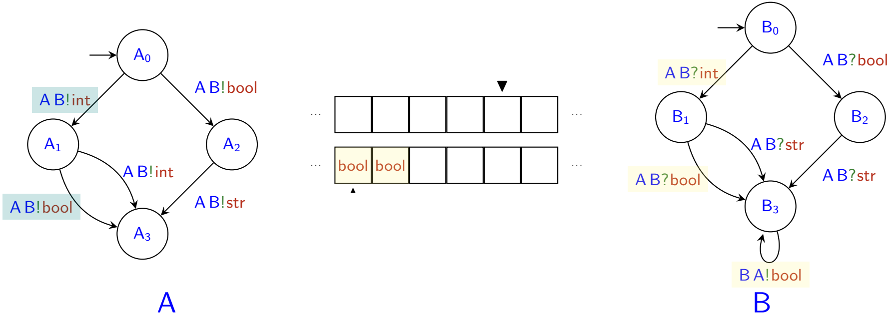
Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



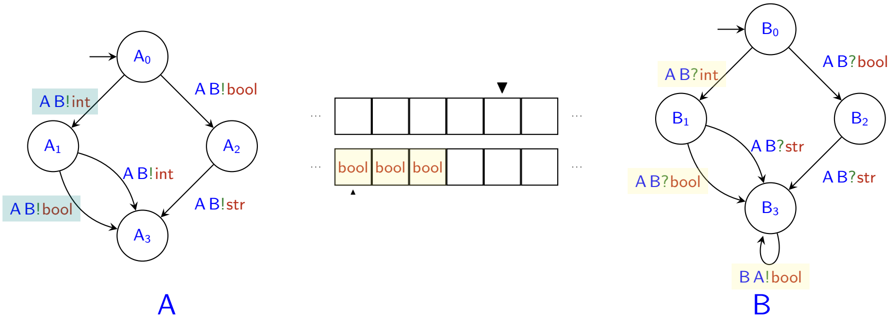
Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



Local views, intuitively

Communicating systems [Brand and Zafiropulo, 1983]



Well-formedness, intuitively

To G or not to G?

Ehm...in a distributed choice $\mathsf{G}_1 + \mathsf{G}_2 + \cdots$

- there should be one active participant
- any non-active participant should be passive decides which branch to take in a choice

Well-formedness, intuitively

To G or not to G?

Ehm...in a distributed choice $G_1 + G_2 + \cdots$

- there should be one active participant
- any non-active participant should be passive decides which branch to take in a choice

Def. A is active when it locally decides which branch to take in a choice

Def. B is passive when

- either B behaves uniformly in each branch
- or B "unambiguously understands" which branch A opted for through the information received on each branch

Well-formedness, intuitively

To G or not to G?

Ehm...in a distributed choice $\mathsf{G}_1 + \mathsf{G}_2 + \cdots$

- there should be one active participant
- any non-active participant should be passive decides which branch to take in a choice

Def. A is active when it locally decides which branch to take in a choice

Def. B is passive when

- either B behaves uniformly in each branch
- or B "unambiguously understands" which branch A opted for through the information received on each branch

Well-branchedness

When the above holds true for each choice, the choreography is well-branched. This enables correctness-by-design.

Class test

Figure out the graphical structure of the following terms and for each of them say which one is well-branched

- $G_1 = \mathsf{A} \rightarrow \mathsf{B} : \mathsf{int} + \mathsf{A} \rightarrow \mathsf{B} : \mathsf{str}$
- $G_2 = \mathsf{A} \rightarrow \mathsf{B} : \mathsf{int} + \mathbf{0}$
- $G_3 = \mathsf{A} \rightarrow \mathsf{B} : \mathsf{int} + \mathsf{A} \rightarrow \mathsf{C} : \mathsf{str}$
- $G_4 = \left(\begin{array}{l} \mathsf{A} \rightarrow \mathsf{C} : \mathsf{int}; \mathsf{A} \rightarrow \mathsf{B} : \mathsf{bool} \\ + \\ \mathsf{A} \rightarrow \mathsf{C} : \mathsf{str}; \mathsf{A} \rightarrow \mathsf{C} : \mathsf{bool}; \mathsf{A} \rightarrow \mathsf{B} : \mathsf{bool} \end{array} \right)$

– Act I –

[Choregraphies, more precisely]

Syntax of g-choreographies

$$
\begin{array}{rcll}
\mathsf{G} & ::= & (\mathsf{o}) & \text{empty} \\
& | & \mathsf{A} \to \mathsf{B} : \mathsf{m} & \text{interaction} \\
& | & \mathsf{G} \mid \mathsf{G} & \text{fork} \\
& | & \texttt{sel}\ \{\mathsf{G} + \cdots + \mathsf{G}\} & \text{choice} \\
& | & \mathsf{G} ; \mathsf{G} & \text{sequential} \\
& | & \texttt{repeat}\ \mathsf{G} & \text{iteration}
\end{array}
$$

Partially-ordered multisets [Pratt, 1986]

Isomorphism class of labelled partially-ordered sets

$$\left\{ \left[\begin{tikzcd} \begin{matrix} e_1 \\ \mathrm{A\,B!x} \end{matrix} \arrow[r] \arrow[d] & \begin{matrix} e_2 \\ \mathrm{A\,B?x} \end{matrix} \arrow[d] \\ \begin{matrix} e_3 \\ \mathrm{A\,B!y} \end{matrix} \arrow[r] & \begin{matrix} e_4 \\ \mathrm{A\,B?y} \end{matrix} \end{tikzcd}\right] , \left[\begin{tikzcd} \begin{matrix} f_1 \\ \mathrm{A\,B!x} \end{matrix} \arrow[r] & \begin{matrix} f_2 \\ \mathrm{A\,B?x} \end{matrix} \\ \begin{matrix} f_3 \\ \mathrm{A\,B!y} \end{matrix} \arrow[r] \arrow[u] & \begin{matrix} f_4 \\ \mathrm{A\,B?y} \end{matrix} \arrow[u] \end{tikzcd}\right] \right\}$$

- specify legit executions
- sets of alternative executions

Language of a pomset

- $e_1\ e_2\ e_3\ e_4 \rightsquigarrow$ A B!x A B?x A B!y A B?y 
- $f_3\ f_1\ f_2\ f_4 \rightsquigarrow$ A B!y A B!x A B?x A B!y 
- $e_1\ e_3\ e_2\ e_4 \rightsquigarrow$ A B!x A B!y A B?x A B?y 

Pomset semantics

The semantics of a g-choreography G

The basic idea

- is a set of pomsets
- each pomset in the set corresponds to a branch of G
- is defined by induction on the structure of G

$$
\begin{aligned}
[\![(\mathrm{o})]\!] &= \{\epsilon\} \\
[\![\mathsf{A} \to \mathsf{B} \colon \mathsf{m}]\!] &= \left\{ \left[\ \mathsf{A}\,\mathsf{B}!\mathsf{m} \longrightarrow \mathsf{A}\,\mathsf{B}?\mathsf{m} \ \right] \right\} \\
[\![\mathtt{repeat}\ \mathsf{G}]\!] &= [\![\mathsf{G}]\!] \\
[\![\mathsf{G} \mid \mathsf{G}']\!] &= \{ \mathsf{par}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!] \} \\
[\![\mathsf{G}; \mathsf{G}']\!] &= \{ \mathsf{seq}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!] \} \\
[\![\mathsf{G} + \mathsf{G}']\!] &= [\![\mathsf{G}]\!] \cup [\![\mathsf{G}']\!]
\end{aligned}
$$

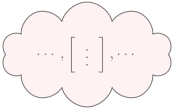
Pomset semantics

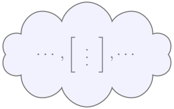
The semantics of a g-choreography G

The basic idea

- is a set of pomsets
- each pomset in the set corresponds to a branch of G
- is defined by induction on the structure of G

$$
\begin{aligned}
[\![(\mathsf{o})]\!] &= \{\epsilon\} \\
[\![\mathsf{A} \to \mathsf{B} : \mathsf{m}]\!] &= \left\{ \left[\ \mathsf{A\,B!m} \longrightarrow \mathsf{A\,B?m}\ \right] \right\} \\
[\![\mathtt{repeat}\ \mathsf{G}]\!] &= [\![\mathsf{G}]\!] \\
[\![\mathsf{G} \mid \mathsf{G}']\!] &= \{ \mathsf{par}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!] \} \\
[\![\mathsf{G} ; \mathsf{G}']\!] &= \{ \mathsf{seq}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!] \} \\
[\![\mathsf{G} + \mathsf{G}']\!] &= [\![\mathsf{G}]\!] \cup [\![\mathsf{G}']\!]
\end{aligned}
$$

$[\![\mathsf{G}]\!] =$ 

$[\![\mathsf{G}']\!] =$ 

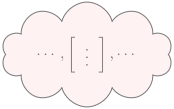
Pomset semantics

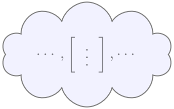
The semantics of a g-choreography G

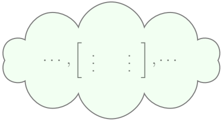
The basic idea

- is a set of pomsets
- each pomset in the set corresponds to a branch of G
- is defined by induction on the structure of G

$$
\begin{aligned}
[\![(o)]\!] &= \{\epsilon\} \\
[\![\mathsf{A} {\to} \mathsf{B} \colon \mathsf{m}]\!] &= \left\{ \left[\ \mathsf{A}\,\mathsf{B}!\mathsf{m} \longrightarrow \mathsf{A}\,\mathsf{B}?\mathsf{m}\ \right] \right\} \\
[\![\mathtt{repeat}\ \mathsf{G}]\!] &= [\![\mathsf{G}]\!] \\
[\![\mathsf{G} \mid \mathsf{G}']\!] &= \{ \mathsf{par}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!] \} \\
[\![\mathsf{G}; \mathsf{G}']\!] &= \{ \mathsf{seq}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!] \} \\
[\![\mathsf{G} + \mathsf{G}']\!] &= [\![\mathsf{G}]\!] \cup [\![\mathsf{G}']\!]
\end{aligned}
$$

$[\![\mathsf{G}]\!] =$ 

$[\![\mathsf{G}']\!] =$ 

$[\![\mathsf{G} \mid \mathsf{G}']\!] =$ 

Pomset semantics

The semantics of a g-choreography G

The basic idea

- is a set of pomsets
- each pomset in the set corresponds to a branch of G
- is defined by induction on the structure of G

$$
\begin{aligned}
[\![(\mathsf{o})]\!] &= \{\epsilon\} \\
[\![\mathsf{A}{\to}\mathsf{B}\colon \mathsf{m}]\!] &= \left\{\left[\ \mathsf{A\,B!m} \longrightarrow \mathsf{A\,B?m}\ \right]\right\} \\
[\![\texttt{repeat}\ \mathsf{G}]\!] &= [\![\mathsf{G}]\!] \\
[\![\mathsf{G} \mid \mathsf{G}']\!] &= \{\mathsf{par}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!]\} \\
[\![\mathsf{G}; \mathsf{G}']\!] &= \{\mathsf{seq}(r, r') \mid (r, r') \in [\![\mathsf{G}]\!] \times [\![\mathsf{G}']\!]\} \\
[\![\mathsf{G} + \mathsf{G}']\!] &= [\![\mathsf{G}]\!] \cup [\![\mathsf{G}']\!]
\end{aligned}
$$

$$[\![\mathsf{G}]\!] = \ldots, [\ \vdots\ ], \ldots \qquad [\![\mathsf{G}']\!] = \ldots, [\ \vdots\ ], \ldots$$

$$[\![\mathsf{G}; \mathsf{G}']\!] = \ldots, [\ \vdots \to \vdots\ ], \ldots$$

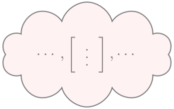
Pomset semantics

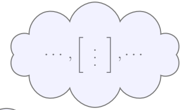
The semantics of a g-choreography G

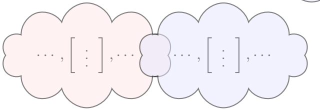
The basic idea

- is a set of pomsets
- each pomset in the set corresponds to a branch of G
- is defined by induction on the structure of G

$$
\begin{aligned}
\llbracket (\mathsf{o}) \rrbracket &= \{\epsilon\} \\
\llbracket \mathsf{A} \to \mathsf{B} \colon \mathsf{m} \rrbracket &= \left\{ \left[\ \mathsf{A\,B!m} \longrightarrow \mathsf{A\,B?m}\ \right] \right\} \\
\llbracket \mathtt{repeat}\ \mathsf{G} \rrbracket &= \llbracket \mathsf{G} \rrbracket \\
\llbracket \mathsf{G} \mid \mathsf{G}' \rrbracket &= \{ \mathsf{par}(r, r') \mid (r, r') \in \llbracket \mathsf{G} \rrbracket \times \llbracket \mathsf{G}' \rrbracket \} \\
\llbracket \mathsf{G}; \mathsf{G}' \rrbracket &= \{ \mathsf{seq}(r, r') \mid (r, r') \in \llbracket \mathsf{G} \rrbracket \times \llbracket \mathsf{G}' \rrbracket \} \\
\llbracket \mathsf{G} + \mathsf{G}' \rrbracket &= \llbracket \mathsf{G} \rrbracket \cup \llbracket \mathsf{G}' \rrbracket
\end{aligned}
$$

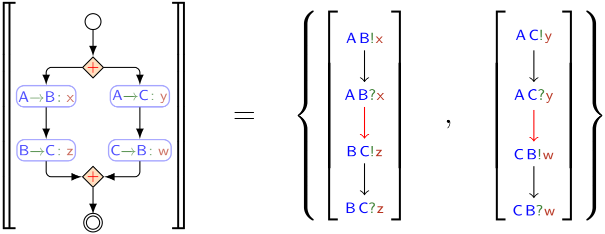
$\llbracket \mathsf{G} \rrbracket =$ 

$\llbracket \mathsf{G}' \rrbracket =$ 

$\llbracket \mathsf{G} + \mathsf{G}' \rrbracket =$ 

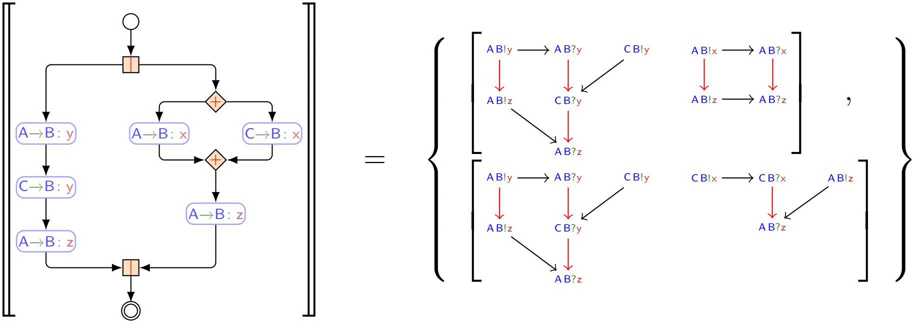
Some examples

Choice & Sequential



Some examples

Parallel & choice



Realisability

Put simply...

A set of pomsets R is *realizable* if there is a deadlock-free[a] communicating system whose language is $\mathcal{L}(R)$.

[a] A system S is *deadlock-free* if none of its reachable configurations s is a deadlock, that is $s \not\to$ and either some buffers are not empty or some CFSMs have transitions from their state in s.

Realisability

Put simply...

A set of pomsets R is *realizable* if there is a deadlock-free[a] communicating system whose language is $\mathcal{L}(R)$.

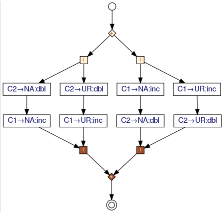
[a] A system S is *deadlock-free* if none of its reachable configurations s is a deadlock, that is $s \not\rightarrow$ and either some buffers are not empty or some CFSMs have transitions from their state in s.

Trivial non-realisability

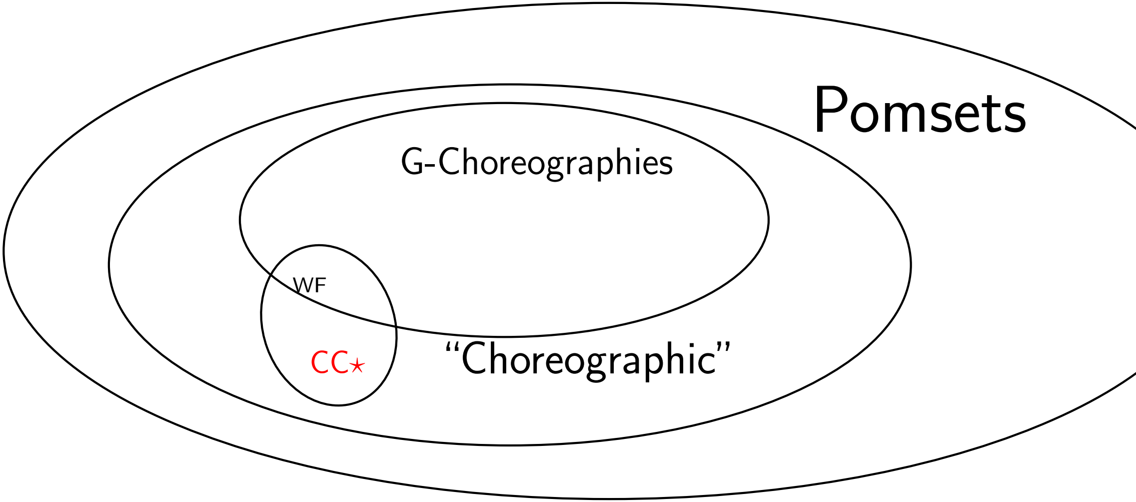
A B?m $\longrightarrow$ B C?n

Communicating systems "start" with outputs!

Non-trivial non-realisability [Alur et al., 2003]



A taxonomy of global views



Closures

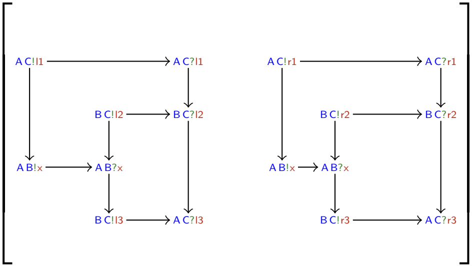
$$\begin{bmatrix}
\begin{tikzcd}
\text{A C!l1} \arrow[rr] \arrow[dd] & & \text{A C?l1} \arrow[d] \\
& \text{B C!l2} \arrow[r] \arrow[d] & \text{B C?l2} \arrow[dd] \\
\text{A B!x} \arrow[r] & \text{A B?x} \arrow[d] & \\
& \text{B C!l3} \arrow[r] & \text{A C?l3}
\end{tikzcd}
\qquad
\begin{tikzcd}
\text{A C!r1} \arrow[rr] \arrow[dd] & & \text{A C?r1} \arrow[d] \\
& \text{B C!r2} \arrow[r] \arrow[d] & \text{B C?r2} \arrow[dd] \\
\text{A B!x} \arrow[r] & \text{A B?x} \arrow[d] & \\
& \text{B C!r3} \arrow[r] & \text{A C?r3}
\end{tikzcd}
\end{bmatrix}$$

Closures

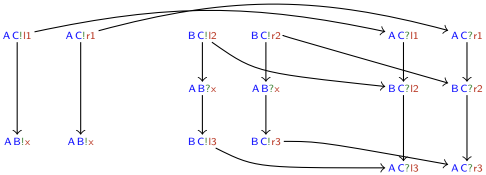
$$
\left[
\begin{tikzcd}
\text{A C!l1} \arrow[rr] \arrow[dd] & & \text{A C?l1} \arrow[d] \\
& \text{B C!l2} \arrow[r] \arrow[d] & \text{B C?l2} \arrow[dd] \\
\text{A B!x} \arrow[r] & \text{A B?x} \arrow[d] & \\
& \text{B C!l3} \arrow[r] & \text{A C?l3}
\end{tikzcd}
\quad
\begin{tikzcd}
\text{A C!r1} \arrow[rr] \arrow[dd] & & \text{A C?r1} \arrow[d] \\
& \text{B C!r2} \arrow[r] \arrow[d] & \text{B C?r2} \arrow[dd] \\
\text{A B!x} \arrow[r] & \text{A B?x} \arrow[d] & \\
& \text{B C!r3} \arrow[r] & \text{A C?r3}
\end{tikzcd}
\right]
$$

$$
\begin{tikzcd}
\text{A C!l1} \arrow[d] & \text{A C!r1} \arrow[d] \\
\text{A B!x} & \text{A B!x}
\end{tikzcd}
\qquad
\begin{tikzcd}
\text{B C!l2} \arrow[d] & \text{B C!r2} \arrow[d] \\
\text{A B?x} \arrow[d] & \text{A B?x} \arrow[d] \\
\text{B C!l3} & \text{B C!r3}
\end{tikzcd}
\qquad
\begin{tikzcd}
\text{A C?l1} \arrow[d] & \text{A C?r1} \arrow[d] \\
\text{B C?l2} \arrow[d] & \text{B C?r2} \arrow[d] \\
\text{A C?l3} & \text{A C?r3}
\end{tikzcd}
$$

Closures

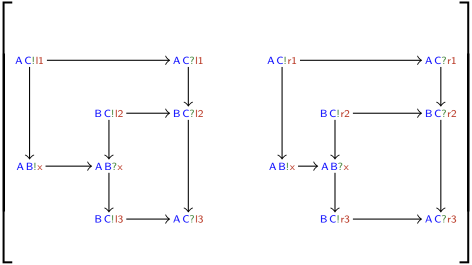


A C!l1
A C?l1
B C!l2
B C?l2
A B!x
A B?x
B C!l3
A C?l3
A C!r1
A C?r1
B C!r2
B C?r2
A B!x
A B?x
B C!r3
A C?r3

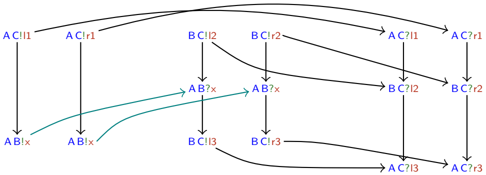


A C!l1
A C!r1
B C!l2
B C!r2
A C?l1
A C?r1
A B?x
A B?x
B C?l2
B C?r2
A B!x
A B!x
B C!l3
B C!r3
A C?l3
A C?r3

Closures

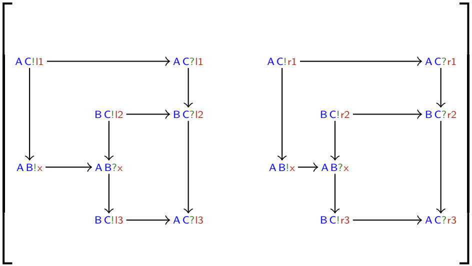


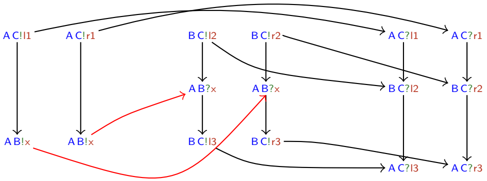
A C!l1
A C?l1
B C!l2
B C?l2
A B!x
A B?x
B C!l3
A C?l3
A C!r1
A C?r1
B C!r2
B C?r2
A B!x
A B?x
B C!r3
A C?r3



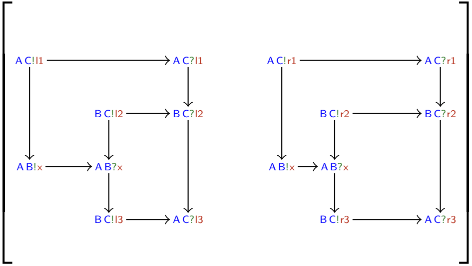
A C!l1
A C!r1
B C!l2
B C!r2
A C?l1
A C?r1
A B?x
A B?x
B C?l2
B C?r2
A B!x
A B!x
B C!l3
B C!r3
A C?l3
A C?r3

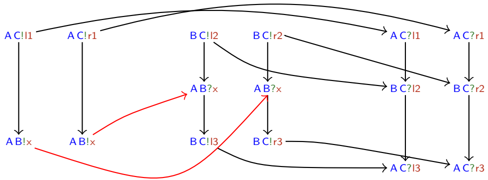
Closures


A C!l1
A C?l1
B C!l2
B C?l2
A B!x
A B?x
B C!l3
A C?l3
A C!r1
A C?r1
B C!r2
B C?r2
A B!x
A B?x
B C!r3
A C?r3


A C!l1
A C!r1
B C!l2
B C!r2
A C?l1
A C?r1
A B?x
A B?x
B C?l2
B C?r2
A B!x
A B!x
B C!l3
B C!r3
A C?l3
A C?r3

Closures





CC*-POM

Take a set of pomsets R

Choose a pomset $\bar{r}^{A} \in R$ for each participant

less permissive

Def. R is CC2-POM if $\forall r \in \Box((r^{A}\lfloor_{A})_{A \in \mathcal{P}}) \ : \ \exists r' \in R \ : \ r \sqsubseteq r'$

Choose a prefix $\bar{r}^{A}$ of a pomset in R for each participant A

Def. R is CC3-POM if $\forall \bar{r} \in \Box((\bar{r}^{A}\lfloor_{A})_{A \in \mathcal{P}}) \ : \ \exists r' \in R, \bar{r}'$ prefix of $r' \ : \ \bar{r} \sqsubseteq \bar{r}'$

Class test : solutions

Which of the following g-choreographies is well-branched?

- $G_1 = A \to B : \text{int} + A \to B : \text{str}$
- $G_2 = A \to B : \text{int} + \mathbf{0}$
- $G_3 = A \to B : \text{int} + A \to C : \text{str}$
- $G_4 = \left(\begin{array}{l} A \to C : \text{int}; A \to B : \text{bool} \\ \;+ \\ A \to C : \text{str}; A \to C : \text{bool}; A \to B : \text{bool} \end{array} \right)$

Class test : solutions

Which of the following g-choreographies is well-branched?

- $\mathsf{G}_1 = \mathsf{A}{\to}\mathsf{B} : \mathsf{int} + \mathsf{A}{\to}\mathsf{B} : \mathsf{str}$
- $\mathsf{G}_2 = \mathsf{A}{\to}\mathsf{B} : \mathsf{int} + \mathbf{0}$
- $\mathsf{G}_3 = \mathsf{A}{\to}\mathsf{B} : \mathsf{int} + \mathsf{A}{\to}\mathsf{C} : \mathsf{str}$
- $\mathsf{G}_4 = \left(\begin{array}{l} \mathsf{A}{\to}\mathsf{C} : \mathsf{int}; \mathsf{A}{\to}\mathsf{B} : \mathsf{bool} \\ \;+ \\ \mathsf{A}{\to}\mathsf{C} : \mathsf{str}; \mathsf{A}{\to}\mathsf{C} : \mathsf{bool}; \mathsf{A}{\to}\mathsf{B} : \mathsf{bool} \end{array} \right)$

Class test : solutions

Which of the following g-choreographies is well-branched?

- $\mathsf{G}_1 = \mathsf{A} \rightarrow \mathsf{B} : \mathsf{int} + \mathsf{A} \rightarrow \mathsf{B} : \mathsf{str}$
- $\mathsf{G}_2 = \mathsf{A} \rightarrow \mathsf{B} : \mathsf{int} + \mathbf{0}$
- $\mathsf{G}_3 = \mathsf{A} \rightarrow \mathsf{B} : \mathsf{int} + \mathsf{A} \rightarrow \mathsf{C} : \mathsf{str}$
- $\mathsf{G}_4 = \left(\begin{array}{l} \mathsf{A} \rightarrow \mathsf{C} : \mathsf{int}; \mathsf{A} \rightarrow \mathsf{B} : \mathsf{bool} \\ \ + \\ \mathsf{A} \rightarrow \mathsf{C} : \mathsf{str}; \mathsf{A} \rightarrow \mathsf{C} : \mathsf{bool}; \mathsf{A} \rightarrow \mathsf{B} : \mathsf{bool} \end{array} \right)$

Class test : solutions

Which of the following g-choreographies is well-branched?

- $G_1 = A \to B : \text{int} + A \to B : \text{str}$
- $G_2 = A \to B : \text{int} + \mathbf{0}$
- $G_3 = A \to B : \text{int} + A \to C : \text{str}$
- $G_4 = \left(\begin{array}{l} A \to C : \text{int}; A \to B : \text{bool} \\ + \\ A \to C : \text{str}; A \to C : \text{bool}; A \to B : \text{bool} \end{array} \right)$

Class test : solutions

Which of the following g-choreographies is well-branched?

- $\mathsf{G}_1 = \mathsf{A} {\rightarrow} \mathsf{B} : \mathsf{int} + \mathsf{A} {\rightarrow} \mathsf{B} : \mathsf{str}$ 
- $\mathsf{G}_2 = \mathsf{A} {\rightarrow} \mathsf{B} : \mathsf{int} + \mathbf{0}$ 
- $\mathsf{G}_3 = \mathsf{A} {\rightarrow} \mathsf{B} : \mathsf{int} + \mathsf{A} {\rightarrow} \mathsf{C} : \mathsf{str}$ 
- $\mathsf{G}_4 = \left(\begin{array}{l} \mathsf{A} {\rightarrow} \mathsf{C} : \mathsf{int} ; \mathsf{A} {\rightarrow} \mathsf{B} : \mathsf{bool} \\ \;+ \\ \mathsf{A} {\rightarrow} \mathsf{C} : \mathsf{str} ; \mathsf{A} {\rightarrow} \mathsf{C} : \mathsf{bool} ; \mathsf{A} {\rightarrow} \mathsf{B} : \mathsf{bool} \end{array} \right)$ 

Class test : solutions

Which of the following g-choreographies is well-branched?

- $G_1 = A \to B : \mathsf{int} + A \to B : \mathsf{str}$ 
- $G_2 = A \to B : \mathsf{int} + \mathbf{0}$ 
- $G_3 = A \to B : \mathsf{int} + A \to C : \mathsf{str}$ 
- $G_4 = \left(\begin{array}{l} A \to C : \mathsf{int}; A \to B : \mathsf{bool} \\ \quad + \\ A \to C : \mathsf{str}; A \to C : \mathsf{bool}; A \to B : \mathsf{bool} \end{array} \right)$ 

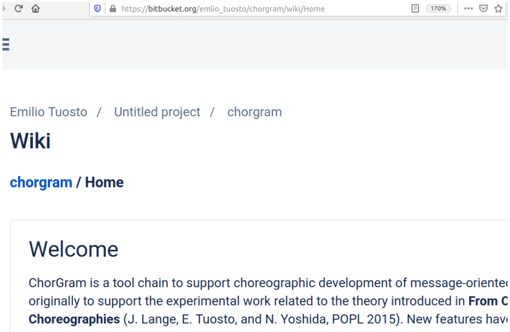
Find out which closure conditions the non well-branched properties violate

– Act II –

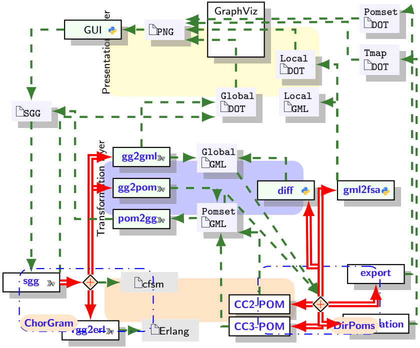
[An exercise: prototype tool support]

The ChorGram prototype [Coto et al., , Guanciale and Tuosto, 2020, Guanciale, 2019, Lange et al., 2017]

Supporting well-formedness analysis



Legend. Components connect to each other either via some files (dashed lines) or by invoking each others' functionalities (doubled lines)



A Simple Exercise in BehAPI

Given B, a bank's API s.t.

- GET authReq :: authenticate; return authFail or granted
- GET authWithdrawal :: request cash; return allow or deny
- GET getBalance :: get balance; return balance

A Simple Exercise in BehAPI

Given B, a bank's API s.t.

- GET authReq :: authenticate; return authFail or granted
- GET authWithdrawal :: request cash; return allow or deny
- GET getBalance :: get balance; return balance

Develop A, the sw for an ATM machine

- GET auth :: authentication request; return authFail or granted
- GET withdraw :: request cash; return money or bye
- GET checkBalance :: check balance request; return balance
- ...

A Simple Exercise in BehAPI

Given B, a bank's API s.t.

- GET authReq :: authenticate; return authFail or granted
- GET authWithdrawal :: request cash; return allow or deny
- GET getBalance :: get balance; return balance

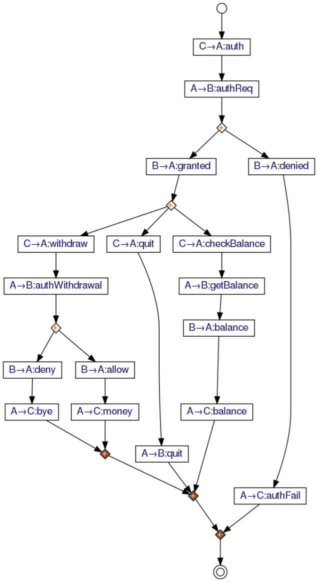
Develop A, the sw for an ATM machine

- GET auth :: authentication request; return authFail or granted
- GET withdraw :: request cash; return money or bye
- GET checkBalance :: check balance request; return balance
- ...

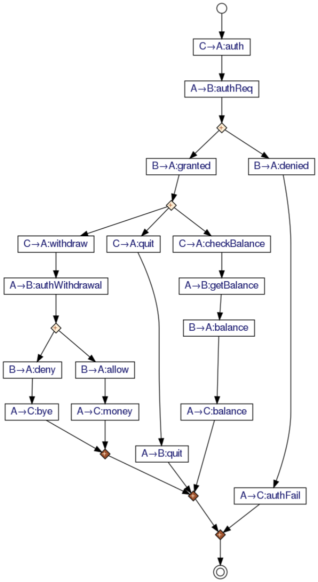
Modelling C, a fictional customer

- ...

Define the global view

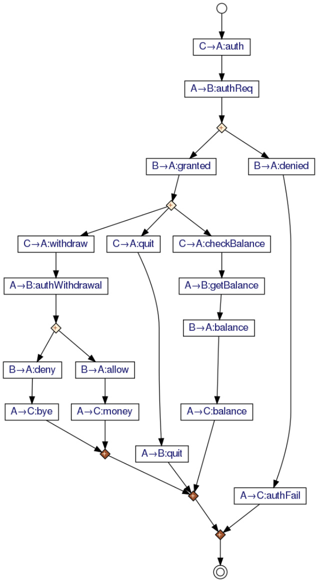


Define the global view



Is this g-choreography well-branched?

Define the global view



Is this g-choreography well-branched?

Let's try **ChorGram**

– Epilogue –

[Work in progress]

The missing bits

What we didn't show

- Going bottom-up

The missing bits

What we didn't show

- Going bottom-up
- Termination awareness

The missing bits

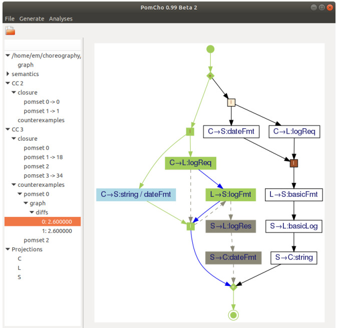
What we didn't show

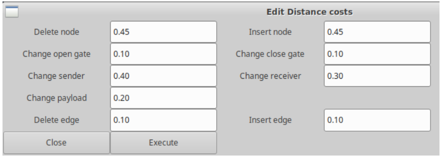
- Going bottom-up
- Termination awareness
- Run-time support (code & monitor generation)

The missing bits

What we didn't show

- Going bottom-up
- Termination awareness
- Run-time support (code & monitor generation)
- An experimental "debugging" mechanism





What we are doing

Theory

- Choreographic Testing

  Alex & Roberto: see Alex's talk@ICE this Fri

- (De-)Composition of choreographies

  Mariangiola, Franco, & Ivan: see Franco's talk@COORDINATION this Tue

- New communication frameworks

  Hernán: see my talk@COORDINATION this Tue

- Refinement of choreographies

  Hernán & Ugo: see Ugo's talk@ICE this Fri

What we are doing

Theory

- Choreographic Testing

  Alex & Roberto: see Alex's talk@ICE this Fri
- (De-)Composition of choreographies

  Mariangiola, Franco, & Ivan: see Franco's talk@COORDINATION this Tue
- New communication frameworks

  Hernán: see my talk@COORDINATION this Tue
- Refinement of choreographies

  Hernán & Ugo: see Ugo's talk@ICE this Fri

Practice

- Better integration of top-down & bottom-up
- Code generation / Code testing
- Keep working on **ChorGram**
  - existing features (e.g., "debugging", pom2gg,...)
  - new features (e.g., test generation, modularity,...)
  - usability (the most boring yet important part)

Thank you

References

1. Alur, R., Etessami, K., and Yannakakis, M. (2003). *Inference of Message Sequence Charts*. IEEE Trans. Software Eng., 29(7):623–633.
2. Brand, D. and Zafiropulo, P. (1983). *On Communicating Finite-State Machines*. JACM, 30(2):323–342.
3. Coto, A., Guanciale, R., Lange, J., and Tuosto, E. **ChorGram***: tool support for choreographic deveelopment*. Available at `https://bitbucket.org/emlio_tuosto/chorgram/wiki/Home`.
4. Guanciale, R. (2019). *Dirpoms: Automatic checker of distributed realizability of pomsets*. In *Coordination 2019*
5. Guanciale, R. and Tuosto, E. (2016). *An abstract semantics of the global view of choreographies*. In *ICE 2016*
6. Guanciale, R. and Tuosto, E. (2020). *Pomcho: a tool chain for choreographic design*. OSP. To appear.
7. Lange, J., Tuosto, E., and Yoshida, N. (2015). *From Communicating Machines to Graphical Choreographies*. In *POPL15*.
8. Lange, J., Tuosto, E., and Yoshida, N. (2017). *A tool for choreography-based analysis of message-passing software*. In *Behavioural Types: from Theory to Tools*
9. Pratt, V. (1986). *Modeling concurrency with partial orders*. International Journal of Parallel Programming, 15(1):33–71.
10. Tuosto, E. and Guanciale, R. (2018). *Semantics of global view of choreographies*. JLAMP, 95:17–40.