the paths to choreography extraction

luís cruz-filipe

(joint work with kim s. larsen and fabrizio montesi)

department of mathematics and computer science
university of southern denmark

fossacs 2017
april 27th, 2017

outline

background

choreography extraction

choreographic programming

context choreographies

- high-level descriptions of communicating systems
- directed communication (from alice to bob)
- automatic compilation to process calculi
- good theoretical properties

choreographic programming

context choreographies

- high-level descriptions of communicating systems
- directed communication (from alice to bob)
- automatic compilation to process calculi
- good theoretical properties

previously core choreographies

- minimal primitives for turing completeness
- captures the "essence" of choreographies

choreographic programming

context choreographies

- high-level descriptions of communicating systems
- directed communication (from alice to bob)
- automatic compilation to process calculi
- good theoretical properties

previously core choreographies

- minimal primitives for turing completeness
- captures the "essence" of choreographies

in this work inverting compilation

- extraction from implementations

core choreographies (i/ii)

syntax

$$C ::= \mathbf{0} \mid \eta; C \mid \mathsf{if}\,(\mathsf{p}.* = \mathsf{q}.*)\,\mathsf{then}\; C_1\;\mathsf{else}\; C_2 \mid \mathsf{def}\; X = C_2\;\mathsf{in}\; C_1 \mid X$$

$$\eta ::= \mathsf{p}.e \to \mathsf{q} \mid \mathsf{p} \to \mathsf{q}[l]$$

$$l ::= \text{labels (at least two distinct)}$$

$$e ::= \text{some set of expressions}$$

core choreographies (ii/ii)

semantics

$$\frac{v = e[\sigma(\mathsf{p})/*]}{\mathsf{p}.e \to \mathsf{q}; C, \sigma \longrightarrow C, \sigma[\mathsf{q} \mapsto v]}$$

$$\frac{}{\mathsf{p} \to \mathsf{q}[l]; C, \sigma \longrightarrow C, \sigma}$$

$$\frac{i = 1 \text{ if } \sigma(\mathsf{p}) = \sigma(\mathsf{q}),\ \ i = 2 \text{ else}}{\text{if } (\mathsf{p}.* = \mathsf{q}.*) \text{ then } C_1 \text{ else } C_2, \sigma \longrightarrow C_i, \sigma}$$

$$\frac{C_1, \sigma \longrightarrow C_1', \sigma'}{\text{def } X = C_2 \text{ in } C_1, \sigma \longrightarrow \text{def } X = C_2 \text{ in } C_1', \sigma'}$$

$$\frac{C_1 \preceq C_1' \qquad C_1', \sigma \longrightarrow C_2', \sigma' \qquad C_2' \preceq C_2}{C_1, \sigma \longrightarrow C_2, \sigma'}$$

(last rule says that
e.g. $\mathsf{p}.e \to \mathsf{q}; \mathsf{r}.e' \to \mathsf{s}, \sigma \longrightarrow \mathsf{p}.e \to \mathsf{q}, \sigma'$)

stateful processes

target language a process calculus with the corresponding primitives:

- send to/receive from a process
- offer a choice to/select an option from a process
- conditional
- recursive definition

epp the endpoint projection of a choreography is a process term that implements the corresponding choreography

example the choreography

$$\mathsf{p}.e \to \mathsf{q}; \mathsf{p}.e' \to \mathsf{r}$$

projects to

$$\mathsf{p} \triangleright \mathsf{q}!e; \mathsf{r}!e' \mid \mathsf{q} \triangleright \mathsf{p}? \mid \mathsf{r} \triangleright \mathsf{p}?$$

outline

the problem

questions

given a process network N:

- is there a choreography C with the same behaviour (bisimilarity)?
- in the affirmative case, can we construct C from N?

the problem

questions given a process network N:

- is there a choreography C with the same behaviour (bisimilarity)?
- in the affirmative case, can we construct C from N?

answer no

- undecidability results prevent perfect solution
- ... but can we solve this for a large enough set of N?

the problem

questions given a process network N:

- is there a choreography C with the same behaviour (bisimilarity)?
- in the affirmative case, can we construct C from N?

answer no

- undecidability results prevent perfect solution
- . . . but can we solve this for a large enough set of N?

new goal given a process network N:

- if we return $\mathtt{yes}$, we can build C bisimilar to N
- we return $\mathtt{yes}$ as much as possible

our approach

idea symbolic execution of N
(abstracting from values, two cases in conditionals)
each “path” corresponds to a choreography

our approach

idea symbolic execution of N
(abstracting from values, two cases in conditionals)
each “path” corresponds to a choreography

$$\begin{tikzcd}
& \mathsf{p} \triangleright \mathsf{q}!e; \mathsf{r}!e' \mid \mathsf{q} \triangleright \mathsf{p}? \mid \mathsf{r} \triangleright \mathsf{if}\ * = \mathsf{p}.*\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{q}? \arrow[d, "\mathsf{p}.e \to \mathsf{q}"] & \\
& \mathsf{p} \triangleright \mathsf{r}!e' \mid \mathsf{q} \triangleright \mathbf{0} \mid \mathsf{r} \triangleright \mathsf{if}\ * = \mathsf{p}.*\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{q}? \arrow[dl, "\mathsf{p}.e' \to \mathsf{r}.\mathsf{then}"'] \arrow[dr, "\mathsf{p}.e' \to \mathsf{r}.\mathsf{else}"] & \\
\mathsf{p} \triangleright \mathbf{0} \mid \mathsf{q} \triangleright \mathbf{0} \mid \mathsf{r} \triangleright \mathbf{0} & & \mathsf{p} \triangleright \mathbf{0} \mid \mathsf{q} \triangleright \mathbf{0} \mid \mathsf{r} \triangleright \mathsf{q}?
\end{tikzcd}$$

our approach

idea

symbolic execution of N
(abstracting from values, two cases in conditionals)
each "path" corresponds to a choreography

$$
\begin{tikzcd}[column sep=tiny]
& \mathsf{p} \triangleright \mathsf{q}!e; \mathsf{r}!e' \mid \mathsf{q} \triangleright \mathsf{p}? \mid \mathsf{r} \triangleright \mathsf{if}\ * = \mathsf{p}.\ *\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{q}? \arrow[d, "\mathsf{p}.e \to \mathsf{q}"] & \\
& \mathsf{p} \triangleright \mathsf{r}!e' \mid \mathsf{q} \triangleright \mathbf{0} \mid \mathsf{r} \triangleright \mathsf{if}\ * = \mathsf{p}.\ *\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{q}? \arrow[dl, "\mathsf{p}.e' \to \mathsf{r}.\mathsf{then}"'] \arrow[dr, "\mathsf{p}.e' \to \mathsf{r}.\mathsf{else}"] & \\
\mathsf{p} \triangleright \mathbf{0} \mid \mathsf{q} \triangleright \mathbf{0} \mid \mathsf{r} \triangleright \mathbf{0} & & \mathsf{p} \triangleright \mathbf{0} \mid \mathsf{q} \triangleright \mathbf{0} \mid \mathsf{r} \triangleright \mathsf{q}?
\end{tikzcd}
$$

extracted choreography

$\mathsf{p}.e \to \mathsf{q};\ \mathsf{if}\ \mathsf{r}.* = \mathsf{p}.\ *\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathbf{1}$
where $\mathbf{1}$ stands for deadlock (equivalent to $\mathbf{0}$)

properties (finite case)

- always terminates
- identifies potential problems by **1**
- bisimilarity *always* holds!
- non-deterministic (up to structural equivalence)
- is sound and (almost) complete
(deadlocks may occur in dead code)

the problem

consider the following networks

- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!{*}; X \mathsf{\ in\ } X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \mathsf{\ in\ } Y \end{array}$
- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!{*}; X \mathsf{\ in\ } \mathsf{q}!{*}; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \mathsf{\ in\ } Y \end{array}$
- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!{*}; \mathsf{q}!{*}; X \mathsf{\ in\ } X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \mathsf{\ in\ } Y \end{array}$
- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!{*}; \mathsf{q}!{*}; X \mathsf{\ in\ } \mathsf{q}!{*}; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; \mathsf{p}?; Y \mathsf{\ in\ } Y \end{array}$

introducing recursion

the problem

consider the following networks

- $\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ X$
  $\mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y$
- $\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X$
  $\mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y$
- $\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ X$
  $\mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y$
- $\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X$
  $\mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; \mathsf{p}?; Y\ \mathsf{in}\ Y$

main intuition

we do not care what the recursive definitions at the processes say!

the idea

do the same as before, but allow loops

introducing recursion

the problem

consider the following networks

- $$\begin{aligned} & \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid\ & \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \qquad \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X \end{aligned}$$
- $$\begin{aligned} & \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid\ & \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \end{aligned}$$
- $$\begin{aligned} & \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid\ & \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \end{aligned}$$
- $$\begin{aligned} & \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid\ & \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; \mathsf{p}?; Y\ \mathsf{in}\ Y \end{aligned}$$

$$\begin{tikzcd} \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \arrow[loop right, "\mathsf{p}.e \to \mathsf{q}"] \end{tikzcd}$$

introducing recursion

the problem

consider the following networks

- $$\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \end{array} \quad \leadsto \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X$$
- $$\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \\ \leadsto \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ \mathsf{p}.e \to \mathsf{q}; X \end{array}$$
- $$\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \end{array}$$
- $$\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; \mathsf{p}?; Y\ \mathsf{in}\ Y \end{array}$$

$$\begin{tikzcd} \mathsf{p} \triangleright \mathsf{q}!*; X \mid \mathsf{q} \triangleright Y \arrow[r, "\mathsf{p}.e \to \mathsf{q}"] & \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \arrow[loop right, "\mathsf{p}.e \to \mathsf{q}"] \end{tikzcd}$$

introducing recursion

the problem

consider the following networks

- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \end{array} \quad \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X$
- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \\ \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ \mathsf{p}.e \to \mathsf{q}; X \end{array}$
- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \\ \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X \end{array}$
- $\begin{array}{l} \quad \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; \mathsf{p}?; Y\ \mathsf{in}\ Y \end{array}$

$$\begin{tikzcd} \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \arrow[r, bend left, "\mathsf{p}.e \to \mathsf{q}"] & \mathsf{p} \triangleright \mathsf{q}!*; X \mid \mathsf{q} \triangleright Y \arrow[l, bend left, "\mathsf{p}.e \to \mathsf{q}"] \end{tikzcd}$$

introducing recursion

the problem

consider the following networks

- $$\begin{array}{l} \;\;\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \end{array} \quad \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X$$
- $$\begin{array}{l} \;\;\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \\ \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ \mathsf{p}.e \to \mathsf{q}; X \end{array}$$
- $$\begin{array}{l} \;\;\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \\ \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X \end{array}$$
- $$\begin{array}{l} \;\;\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; \mathsf{q}!*; X\ \mathsf{in}\ \mathsf{q}!*; X \\ \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; \mathsf{p}?; Y\ \mathsf{in}\ Y \\ \rightsquigarrow \mathsf{def}\ X = \mathsf{p}.e \to \mathsf{q}; \mathsf{p}.e \to \mathsf{q}; X\ \mathsf{in}\ X \end{array}$$

$$\begin{tikzcd} \mathsf{p} \triangleright \mathsf{q}!*; X \mid \mathsf{q} \triangleright Y \arrow[r, bend left, "\mathsf{p}.e \to \mathsf{q}"] & \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright \mathsf{p}?; Y \arrow[l, bend left, "\mathsf{p}.e \to \mathsf{q}"] \end{tikzcd}$$

fairness and starvation

problems not all loops are equal...

$$\begin{aligned} & \mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X \mathsf{\ in\ } X \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \mathsf{\ in\ } Y \\ \mid\ & \mathsf{r} \triangleright \mathsf{def}\ Z = \mathsf{s}!*; Z \mathsf{\ in\ } Z \mid \mathsf{s} \triangleright \mathsf{def}\ W = \mathsf{r}?; W \mathsf{\ in\ } W \end{aligned}$$

$$\underbrace{\mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y}_{\mathsf{p}.* \to \mathsf{q}} \mid \underbrace{\mathsf{r} \triangleright Z \mid \mathsf{s} \triangleright W}_{\mathsf{r}.* \to \mathsf{s}}$$

extracts to $\mathsf{def}\ X = \mathsf{p}.* \to \mathsf{q}; X \mathsf{\ in\ } X$ or $\mathsf{def}\ X = \mathsf{r}.* \to \mathsf{s}; X \mathsf{\ in\ } X$

fairness and starvation

problems not all loops are equal...

$$\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X \ \mathsf{in}\ X \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \ \mathsf{in}\ Y$$
$$\mid \mathsf{r} \triangleright \mathsf{def}\ Z = \mathsf{s}!*; Z \ \mathsf{in}\ Z \mid \mathsf{s} \triangleright \mathsf{def}\ W = \mathsf{r}?; W \ \mathsf{in}\ W$$

solution annotate procedure calls

$$\begin{tikzcd}
 & \mathsf{p} \triangleright X^{\circ} \mid \mathsf{q} \triangleright Y^{\circ} \mid \mathsf{r} \triangleright Z^{\circ} \mid \mathsf{s} \triangleright W^{\circ} \arrow[dl, "\mathsf{p}.*\!\to\!\mathsf{q}"] \arrow[dr, "\mathsf{r}.*\!\to\!\mathsf{s}"] & \\
\mathsf{p} \triangleright X^{\bullet} \mid \mathsf{q} \triangleright Y^{\bullet} \mid \mathsf{r} \triangleright Z^{\circ} \mid \mathsf{s} \triangleright W^{\circ} \arrow[ur, bend left, "\mathsf{r}.*\!\to\!\mathsf{s}"] \arrow[loop below, "\mathsf{p}.*\!\to\!\mathsf{q}"] & & \mathsf{p} \triangleright X^{\circ} \mid \mathsf{q} \triangleright Y^{\circ} \mid \mathsf{r} \triangleright Z^{\bullet} \mid \mathsf{s} \triangleright W^{\bullet} \arrow[ul, bend right, "\mathsf{p}.*\!\to\!\mathsf{q}"'] \arrow[loop below, "\mathsf{r}.*\!\to\!\mathsf{s}"]
\end{tikzcd}$$

fairness and starvation

problems not all loops are equal...

$$\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X \ \mathsf{in}\ X \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \ \mathsf{in}\ Y$$
$$\mid \mathsf{r} \triangleright \mathsf{s}!* \mid \mathsf{s} \triangleright \mathsf{r}?$$

$$\begin{tikzcd} \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \mid \mathsf{r} \triangleright \mathsf{s}!* \mid \mathsf{s} \triangleright \mathsf{r}? \arrow[loop left, "\mathsf{p}.*\to\mathsf{q}"] \arrow[r, "\mathsf{r}.*\to\mathsf{s}"] & \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \arrow[loop right, "\mathsf{p}.*\to\mathsf{q}"] \end{tikzcd}$$

extracts to $\mathsf{def}\ X = \mathsf{p}.* \to \mathsf{q}; X \ \mathsf{in}\ X$ or
$\mathsf{def}\ X = \mathsf{p}.* \to \mathsf{q}; X \ \mathsf{in}\ \mathsf{r}.* \to \mathsf{s}; X$

fairness and starvation

problems not all loops are equal...

$$\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!{*}; X \ \mathsf{in}\ X \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y \ \mathsf{in}\ Y$$
$$\mid \mathsf{r} \triangleright \mathsf{s}!{*} \mid \mathsf{s} \triangleright \mathsf{r}?$$

$$\begin{tikzcd}
\mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \mid \mathsf{r} \triangleright \mathsf{s}!{*} \mid \mathsf{s} \triangleright \mathsf{r}? \arrow[loop left, "\mathsf{p}.{*}{\to}\mathsf{q}"] \arrow[r, "\mathsf{r}.{*}{\to}\mathsf{s}"] & \mathsf{p} \triangleright X \mid \mathsf{q} \triangleright Y \arrow[loop right, "\mathsf{p}.{*}{\to}\mathsf{q}"]
\end{tikzcd}$$

extracts to $\mathsf{def}\ X = \mathsf{p}.{*} \to \mathsf{q}; X \ \mathsf{in}\ X$ or
$\mathsf{def}\ X = \mathsf{p}.{*} \to \mathsf{q}; X \ \mathsf{in}\ \mathsf{r}.{*} \to \mathsf{s}; X$

solution no finite behaviour in loops (except deadlocks)

fairness and starvation

problems not all loops are equal. . .

$$\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!*; X\ \mathsf{in}\ X \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y$$
$$\mid \mathsf{r} \triangleright \mathsf{def}\ Z = \mathsf{q}!*; Z\ \mathsf{in}\ Z$$

$$\mathsf{p} \triangleright X^\circ \mid \mathsf{q} \triangleright Y^\circ \mid \mathsf{r} \triangleright Z^\circ \xrightarrow{\mathsf{p}.*\to\mathsf{q}} \mathsf{p} \triangleright X^\bullet \mid \mathsf{q} \triangleright Y^\bullet \mid \mathsf{r} \triangleright Z^\circ \quad \mathsf{p}.*\to\mathsf{q}$$

oops not extractable (but r is deadlocked)

fairness and starvation

problems not all loops are equal. . .

$$
\begin{aligned}
&\mathsf{p} \triangleright \mathsf{def}\ X = \mathsf{q}!{*}; X\ \mathsf{in}\ X \mid \mathsf{q} \triangleright \mathsf{def}\ Y = \mathsf{p}?; Y\ \mathsf{in}\ Y \\
\mid\ &\mathsf{r} \triangleright \mathsf{def}\ Z = \mathsf{q}!{*}; Z\ \mathsf{in}\ Z
\end{aligned}
$$

$$
\begin{tikzcd}
\mathsf{p} \triangleright X^\circ \mid \mathsf{q} \triangleright Y^\circ \mid \mathsf{r} \triangleright Z^\circ \arrow[r, "\mathsf{p}.{*}\to\mathsf{q}"] & \mathsf{p} \triangleright X^\bullet \mid \mathsf{q} \triangleright Y^\bullet \mid \mathsf{r} \triangleright Z^\circ \arrow[loop below, "\mathsf{p}.{*}\to\mathsf{q}"]
\end{tikzcd}
$$

oops not extractable (but r is deadlocked)

in general some networks are not extractable

results

- if symbolic execution does not generate a node from which some process is always deadlocked, then N is extractable
- if C is extracted from N, then C and N are bisimilar (C may contain deadlocks)
- extraction terminates in time $O\left(n \times e^{2n/e}\right)$
- works for synchronous semantics
- can be adapted to/extended in the asynchronous case

conclusions & future directions

- showed how to extract choreographies from implementations
- significant improvement in complexity wrt previous work
- prototype implementation nearly ready
- extension to process spawning in progress

thank you!