Open Multiparty Interaction in the link-calculus

Linda Brodo (Sassari)

joint work with

Chiara Bodei, Roberto Bruni (Pisa)

Roadmap

- Why multi-party interaction ? And open ?
- A new kind of interaction (subsuming CCS)
- Handling message content (subsuming π-calculus)
- Encoding membranes
- Encoding reaction systems
- Conclusion and future work

Interaction

An interaction is an action by which (communicating) processes can influence each other.

Any better abstraction?

Biology
Social networks
Autonomic systems
Internet of Things
...

I/O is the basic form of interaction
but "one size won't fit all"

(it is possibly misleading to think otherwise:
not all interactions are mutual/reciprocal)

**Interactions are not ever binary,
or maybe we are not interested in that level of detail !**

Multiparty interaction

An interaction is multiparty when
it involves two or more processes

Open interaction

An interaction is open when
the number of involved processes is not fixed

Notation

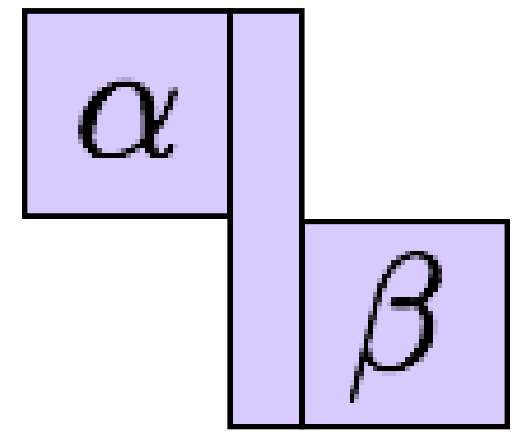
a interaction over channel a

τ silent interaction

$\square$ free “slot”, accepting
any interaction (only in labels)

Link

$\alpha \backslash \beta$ From α to β

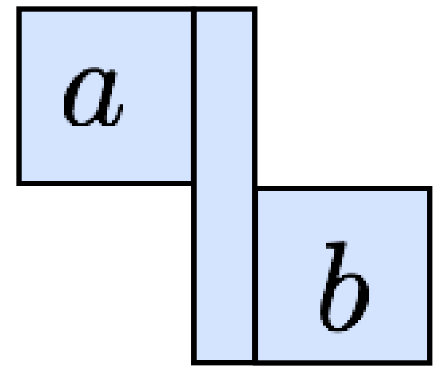


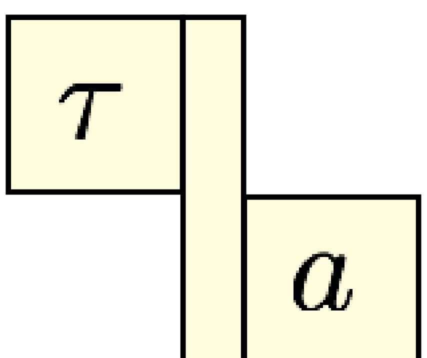
Valid:

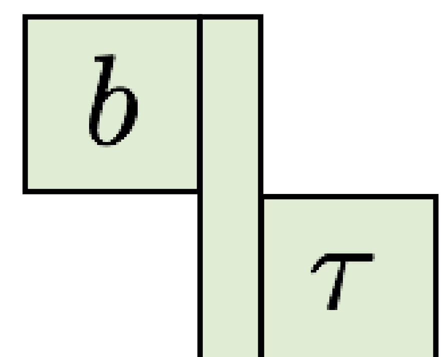
if it is **virtual** $\square \backslash \square$

if it is **solid** $\alpha, \beta \neq \square$

Example: three party

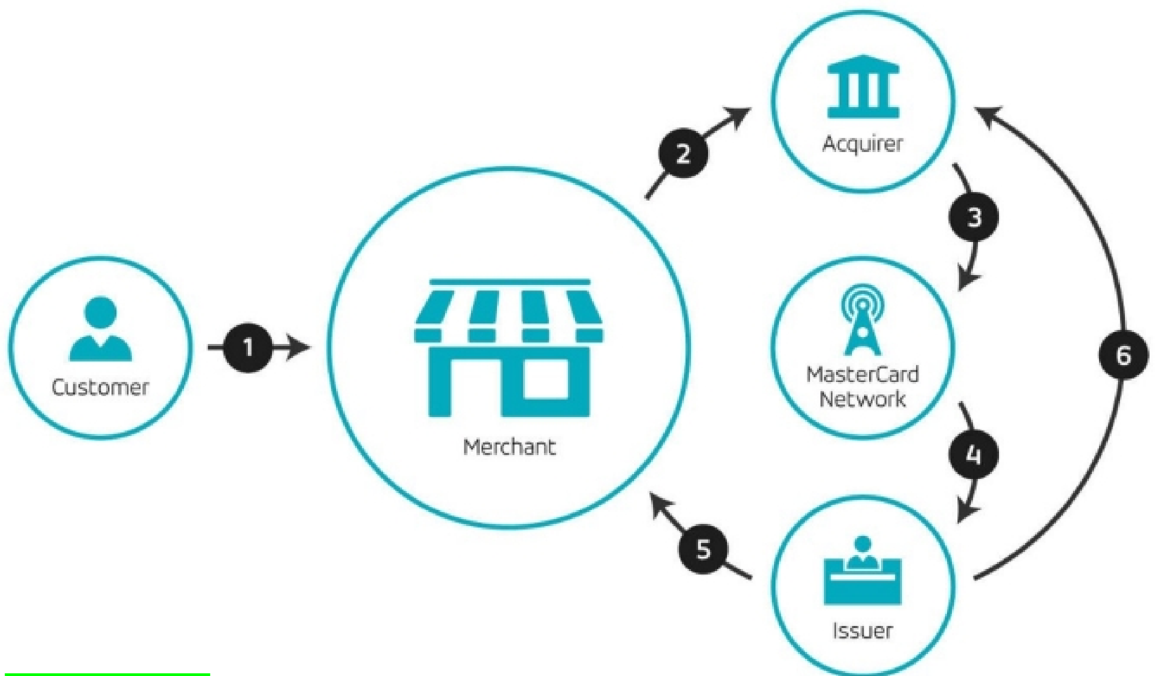


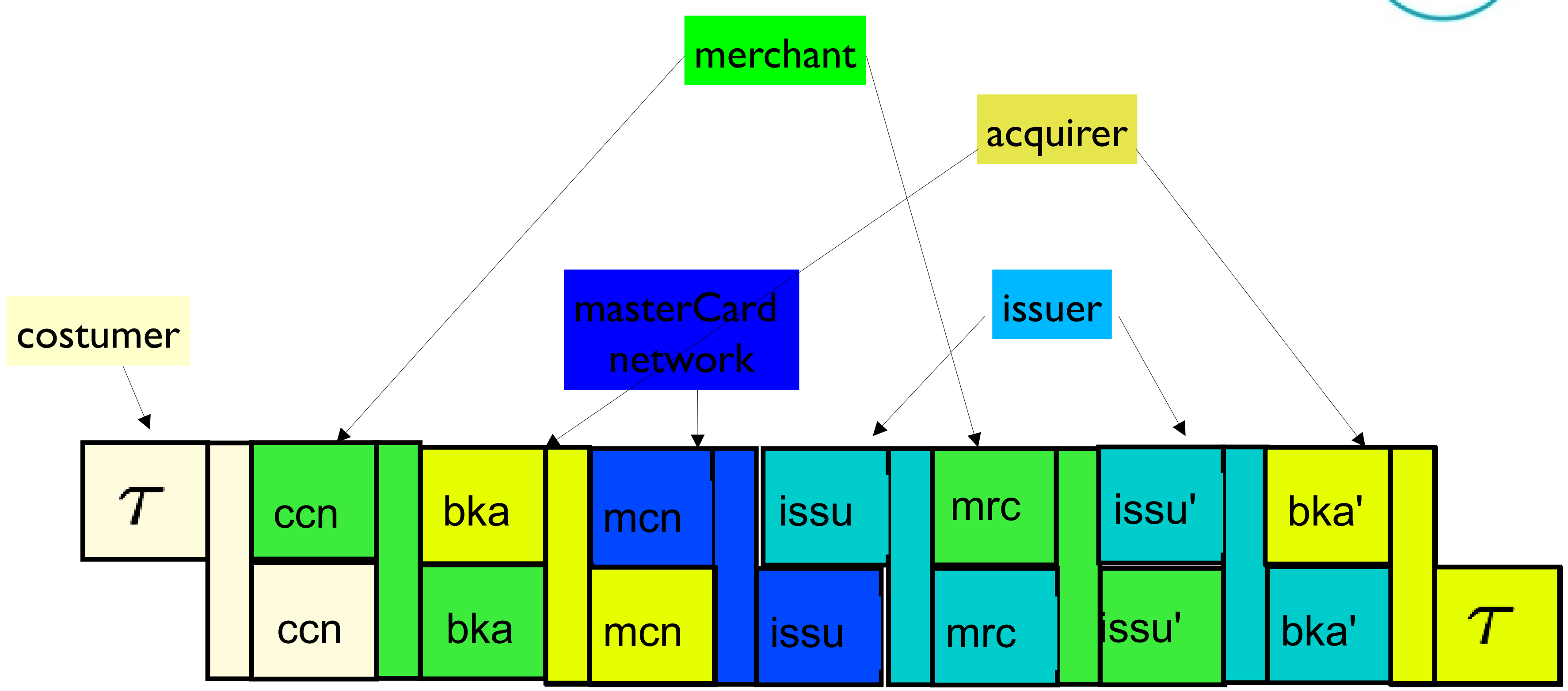




Example: multi-party

mastercard payment





Roadmap

- ✓ Why multi-party interaction ? And open ?
- ✓ A new kind of interaction (subsuming CCS)
- ✓ Handling message content (subsuming π-calculus)
- ✓ Encoding membranes
- ✓ Encoding reaction systems
- ✓ Conclusion and future work

Link chain

$$\alpha_1 \backslash \beta_1 \;\; \alpha_2 \backslash \beta_2 \;\; \cdots \;\; \alpha_n \backslash \beta_n$$

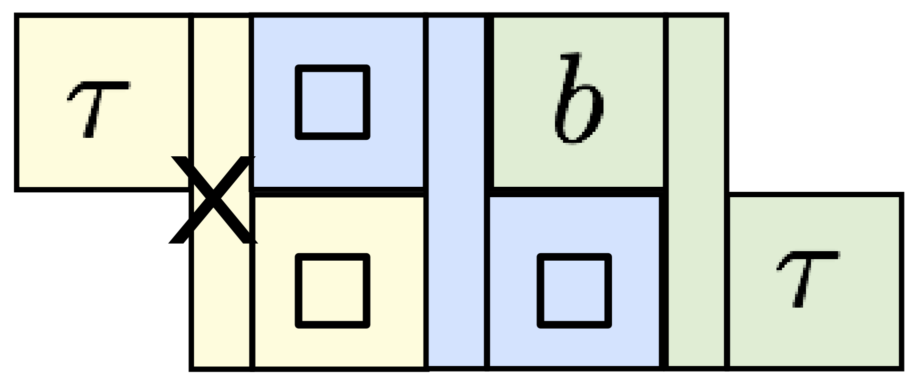
$\mathcal{C}$ is the set of channel names

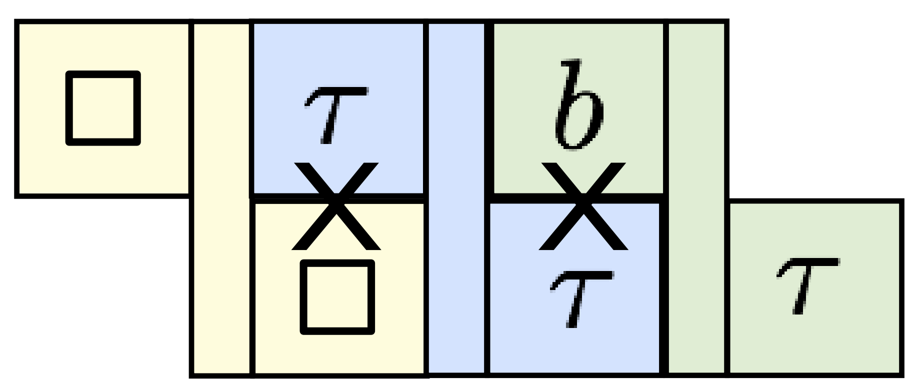
such that:

$$\beta_i, \alpha_{i+1} \in \mathcal{C} \quad \text{implies } \beta_i = \alpha_{i+1}$$

$$\beta_i = \tau \qquad\qquad \text{iff } \alpha_{i+1} = \tau$$

Counter-examples





Link chain: terminology

$$^{\alpha_1}\backslash_{\beta_1}\ ^{\alpha_2}\backslash_{\beta_2}\ \cdots\ ^{\alpha_n}\backslash_{\beta_n}$$

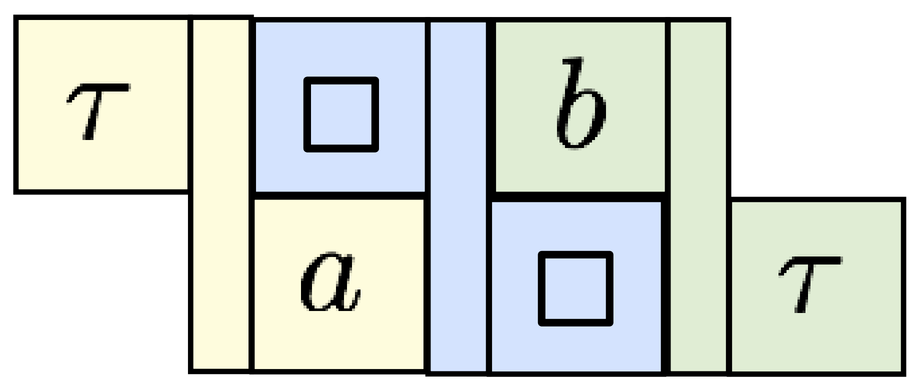
Solid:

if all its links are so

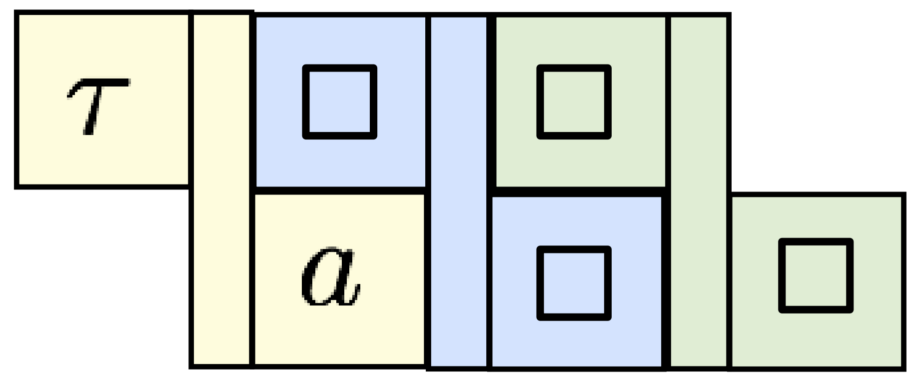
Examples: non solid

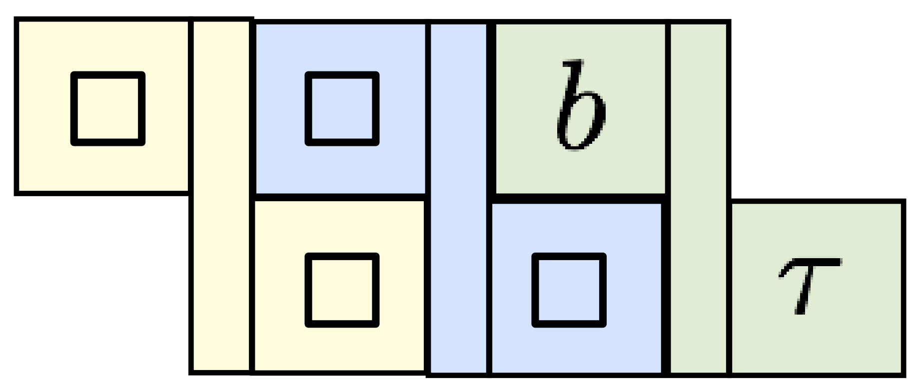
Virtual links
can be read as missing pieces of the puzzle

$\square \backslash \square$



Examples: merge





Merge

$$s = \ell_1 ... \ell_n \qquad\qquad s' = \ell'_1 ... \ell'_n$$

$$s \bullet s' \quad \triangleq \quad (\ell_1 \bullet \ell'_1) \cdots (\ell_n \bullet \ell'_n)$$

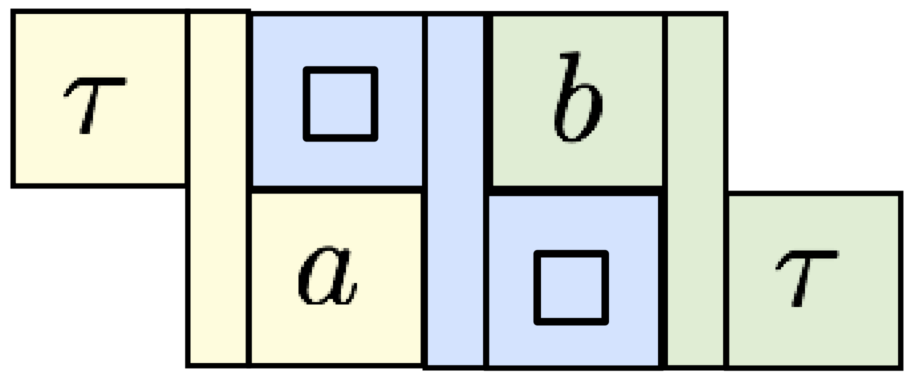
$$^{\alpha}\backslash_{\beta} \bullet {}^{\alpha'}\backslash_{\beta'} \quad \triangleq \quad {}^{(\alpha \bullet \alpha')}\backslash_{(\beta \bullet \beta')}$$

$$\alpha \bullet \beta \quad \triangleq \quad \begin{cases} \alpha & \text{if } \beta = \square \\ \beta & \text{if } \alpha = \square \end{cases}$$

The result is undefined if the outcome is not valid

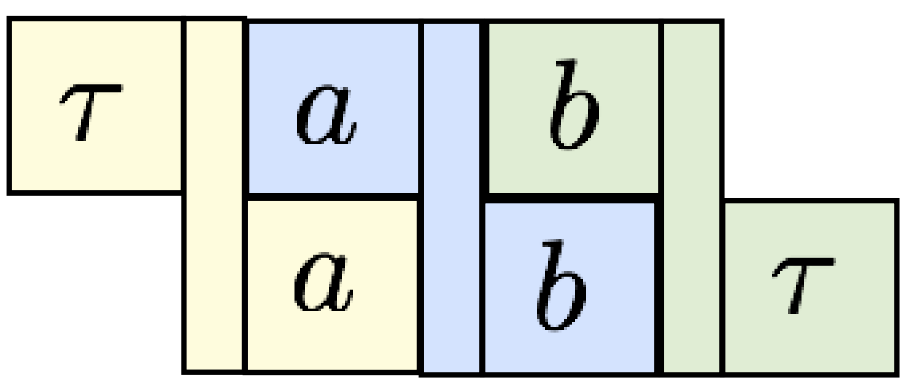
Examples: restriction

(νa)

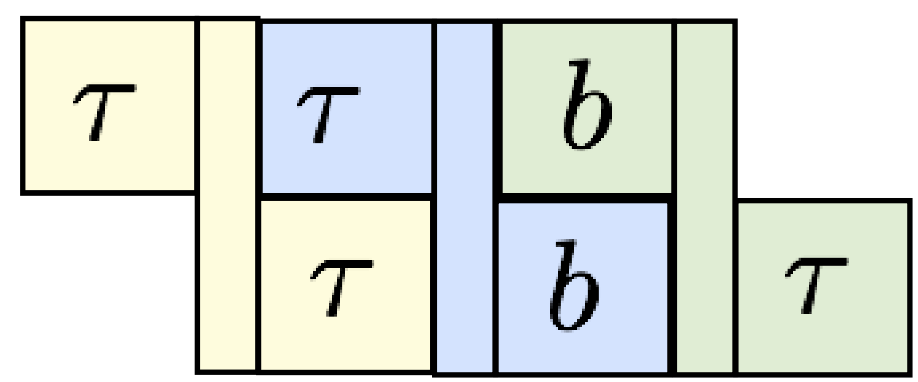


$= \perp$

(νa)



$=$



Restriction

$$(\nu\, a)s \triangleq \begin{cases} ((\nu\, a)\ell_1) \ldots ((\nu\, a)\ell_n) & \text{if } a \text{ is } \textit{matched} \text{ in } s \\ \perp & \text{otherwise} \end{cases}$$

$$(\nu\, a)^{\alpha}\backslash_{\beta} \triangleq {}^{((\nu\, a)\alpha)}\backslash_{((\nu\, a)\beta)}$$

$$(\nu\, a)\alpha \triangleq \begin{cases} \tau & \text{if } \alpha = a \\ \alpha & \text{otherwise} \end{cases}$$

Equivalence relation over link chains

(the black tie)

$$s^{\square}\backslash_{\square} \;\bowtie\; s \qquad\qquad s_1{}^{\square}\backslash^{\square}_{\square}\backslash_{\square} s_2 \;\bowtie\; s_1{}^{\square}\backslash_{\square} s_2$$

$$^{\square}\backslash_{\square} s \;\bowtie\; s \qquad\qquad s_1{}^{\alpha}\backslash^{a}_{a}\backslash_{\beta} s_2 \;\bowtie\; s_1{}^{\alpha}\backslash^{\square}_{a}\backslash^{a}_{\square}\backslash_{\beta} s_2$$

The accordion  lemma

$$If\ P \xrightarrow{s} P'\ \ and\ s \bowtie s',\ then\ P \xrightarrow{s'} P'$$

link-calculus syntax

$$P, Q \;::=\; \mathbf{0} \mid \ell.P \mid P+Q \mid P|Q \mid (\nu\, a)P \mid P[\phi] \mid A$$

null action

prefix
(link prefix)

choice

parallel

restriction

relabelling

recursion

(Relevant) SOS rules

the length of the link chains
(of a transition) is decided
by the semantics

$$\frac{s \bowtie \ell}{\ell.P \xrightarrow{s} P}\ (Act)$$

$$\frac{P \xrightarrow{s} P'}{(\nu\, a)P \xrightarrow{(\nu\, a)s} (\nu\, a)P'}\ (Res)$$

$$\frac{P \xrightarrow{s} P'}{P|Q \xrightarrow{s} P'|Q}\ (Lpar)$$

$$\frac{P \xrightarrow{s} P' \qquad Q \xrightarrow{s'} Q'}{P|Q \xrightarrow{s \bullet s'} P'|Q'}\ (Com)$$

Example

$$P \triangleq {}^{\tau}\backslash_{a}.P_1 | (\nu\, b)Q \text{ , } Q \triangleq {}^{b}\backslash_{\tau}.P_2 | {}^{a}\backslash_{b}.\mathbf{0}$$

$$\dfrac{\dfrac{}{{}^{\tau}\backslash_{a}.P_1 \xrightarrow{\tau\backslash^{\square}_{a}\backslash^{\square}_{\square}\backslash_{\square}} P_1}\,(Act) \qquad \dfrac{\dfrac{\dfrac{}{{}^{b}\backslash_{\tau}.P_2 \xrightarrow{\square\backslash^{\square}_{\square}\backslash^{b}_{\square}\backslash_{\tau}} P_2}\,(Act) \qquad \dfrac{}{{}^{a}\backslash_{b}.\mathbf{0} \xrightarrow{\square\backslash^{a}_{\square}\backslash^{\square}_{b}\backslash_{\square}} \mathbf{0}}\,(Act)}{Q \xrightarrow{\square\backslash^{a}_{\square}\backslash^{b}_{b}\backslash_{\tau}} P_2|\mathbf{0}}\,(Com)}{(\nu\, b)Q \xrightarrow{\square\backslash^{a}_{\square}\backslash^{\tau}_{\tau}\backslash_{\tau}} (\nu\, b)(P_2|\mathbf{0})}\,(Res)}{P \xrightarrow{\tau\backslash^{a}_{a}\backslash^{\tau}_{\tau}\backslash_{\tau}} P_1|(\nu\, b)(P_2|\mathbf{0})}\,(Com)$$

Fact

The process algebra of linked interactions
is a straightforward extension of CCS

It includes CCS as a sub-calculus

Milner's CCS interaction

co-action prefix
(output?)

$$a.P \;\Big|\; \overline{a}.Q$$

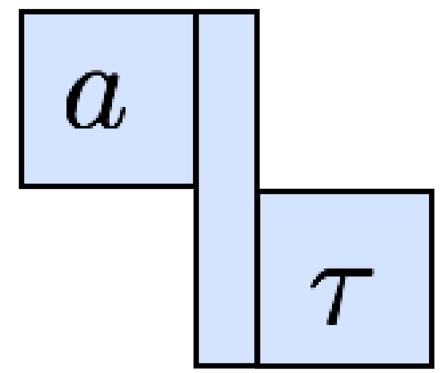
action prefix
(input?)

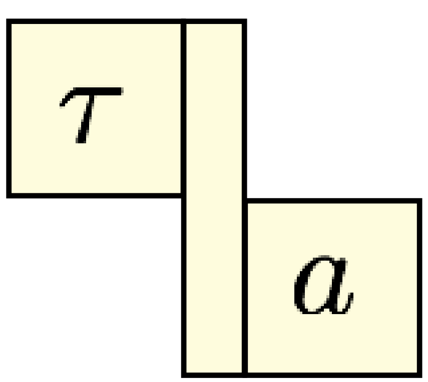
$$\Big\downarrow\; a \bullet \overline{a} = \tau$$

silent action

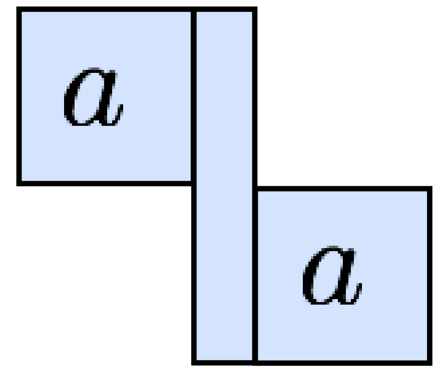
$$P \,|\, Q$$

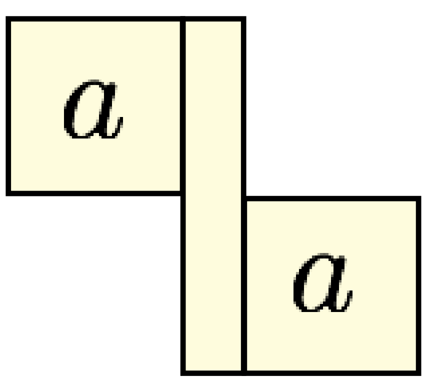
Examples: CCS-like

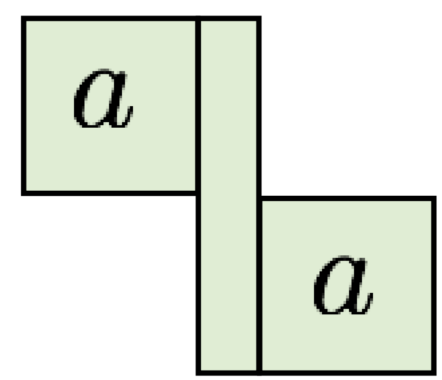




Examples: CSP







Our aim was to

extend the theory of dyadic interactions

as little as possible,

as well as possible,

to deal with open multiparty interaction

Fact

Finer (bisimulation over the) LTS wrt CCS:
three kinds of meaningful observables

$${}^{\tau}\backslash_{a} \qquad\qquad {}^{\tau}\backslash^{\square}_{a}\backslash^{b}_{\square}\backslash_{\tau} \qquad\qquad {}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a}.{}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau}.{}^{\tau}\backslash_{a} \not\sim {}^{\tau}\backslash_{a} \mid {}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a}.{}^{\tau}\backslash_{b} + {}^{\tau}\backslash_{b}.{}^{\tau}\backslash_{a} \sim {}^{\tau}\backslash_{a} \mid {}^{\tau}\backslash_{b}$$

Equivalence relation over link chains

(the white tie)

$$\frac{s \blacktriangleright\!\blacktriangleleft s'}{s \bowtie s'}$$

$$s_1 {}^{\alpha}\backslash^{\tau}_{\tau}\backslash_{\beta} s_2 \bowtie s_1 {}^{\alpha}\backslash_{\beta} s_2$$

Network bisimulation

A *network bisimulation* $\mathbf{R}$ is a binary relation over CNA processes such that, if $P \; \mathbf{R} \; Q$ then:

if $P \xrightarrow{s} P'$, then $\exists \; s'$, Q' such that $s' \bowtie s$, $Q \xrightarrow{s'} Q'$, and $P' \; \mathbf{R} \; Q'$

if $Q \xrightarrow{s} Q'$, then $\exists \; s'$, P' such that $s' \bowtie s$, $P \xrightarrow{s'} P'$, and $P' \; \mathbf{R} \; Q'$

Fact

network bisimulation is a congruence
also with respect to
substitution

$${}^{\tau}\backslash_{a} \qquad\qquad {}^{\tau}\backslash_{a}^{\square}\backslash_{\square}^{b}\backslash_{\tau} \qquad\qquad {}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a}.{}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau}.{}^{\tau}\backslash_{a} \not\sim {}^{\tau}\backslash_{a} \mid {}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a}.{}^{\tau}\backslash_{b} + {}^{\tau}\backslash_{b}.{}^{\tau}\backslash_{a} \sim {}^{\tau}\backslash_{a} \mid {}^{\tau}\backslash_{b}$$

Caveat

there are two kinds of tau:

$$(\nu\, a)({}^{\boxed{\tau}}\backslash_{a}.\mathbf{0} \,|\, {}^{a}\backslash_{\boxed{\tau}}.\mathbf{0})$$

Roadmap

- ✓ Why multi-party interaction ? And open ?
- ✓ A new kind of interaction (subsuming CCS)
- ✓ Handling message content (subsuming π-calculus)
- ✓ Encoding membranes
- ✓ Encoding reaction systems
- ✓ Conclusion and future work

Handling name mobility

Aim: introduce polyadic communication
and reuse/rely on pi as much as possible

One possibility:
each link receive some arguments and
send some names... too complex

$$a(\widetilde{x})\backslash_{b\widetilde{y}}.P$$

Another possibility:
each link in the chain carries the same list of arguments...
but with different (send/receive) capabilities

$$^{a}\backslash_{b}\widetilde{x}.P$$

Separation of concerns

$$P, Q \quad ::= \quad \mathbf{0} \;\mid\; \ell t.P \;\mid \ldots$$

This way we separate

the interaction mechanism ℓ

from

the name passing mechanism t

(We formalize them separately and then fit them together)

No need to reinvent the wheel

We can easily borrow from pi-calculus
the name handling machinery
(and free it from dyadic interaction legacy)

$P \mid a(x).Q$	(waits input from P)	$P' \mid Q[b/x]$
$P \mid \overline{a}x.Q$	(outputs to P)	$P' \mid Q$
$P \mid (\nu x)\overline{a}x.Q$	(extrudes to P)	$(\nu y)P' \mid Q[y/x]$

Tuple

$$t = \langle \widetilde{w} \rangle \qquad w ::= \begin{array}{ll} x & \text{value (output)} \\ \underline{x} & \text{variable (input)} \end{array}$$

variables are instantiated by values

values are used for matching arguments

$$\begin{array}{c} \langle n, m, \underline{x} \rangle \\ \downarrow \qquad = \qquad \uparrow \\ \langle \underline{y}, m, k \rangle \end{array}$$

Assigns n to y

Matches m with m

Assigns k to x

(syntactic) Tuple

$$t = \langle \widetilde{w} \rangle \qquad w ::= \begin{array}{ll} x & \text{value (output)} \\ \underline{x} & \text{variable (input)} \end{array}$$

syntactic tuples, the ones used in the prefixes

(semantic) Tuple

$$g = \langle e_1, \ldots, e_n \rangle \qquad e \;::=\; \begin{array}{ll} x & \text{value (output)} \\ \underline{x} & \text{variable (input)} \\ \hat{x} & \text{value (extruded)} \end{array}$$

semantic tuples, the ones used in the labels

$$\begin{array}{ccc|l} \langle n, & m, & \underline{x} \rangle & \text{Assigns n to y} \\ \downarrow & = & \uparrow & \text{Matches m with m} \\ \langle \underline{y}, & m, & k \rangle & \text{Assigns k to x} \end{array}$$

$$\begin{array}{ccc|l} \langle n, & m, & \underline{x} \rangle & \text{Assigns n to y} \\ \downarrow & = & \uparrow & \text{Matches m with m} \\ \langle y, & m, & \hat{k} \rangle & \text{Assigns extruded k to x} \end{array}$$

Extrusion

an argument in a tuple can be extruded if it is not already annotated

extruded arguments have an hat

$$
\begin{aligned}
(\nu\, a)(sg) &\triangleq ((\nu\, a)s)((\nu\, a)g) \\
(\nu\, a)g &\triangleq \langle(\nu\, a)e_1, ..., (\nu\, a)e_n\rangle
\end{aligned}
$$

$$
(\nu\, a)e \triangleq \begin{cases} e & \text{if } e \neq a, \hat{a}, \underline{a} \\ \hat{a} & \text{if } e = a \end{cases}
$$

Merge

$$g = \langle e_1, ..., e_n \rangle \qquad g' = \langle e'_1, ..., e'_n \rangle$$

$$\begin{array}{lcl} sg \bullet s'g' & \triangleq & (s \bullet s')(g \bullet g') \\ g \bullet g' & \triangleq & \langle e_1 \bullet e'_1, ..., e_n \bullet e'_n \rangle \end{array}$$

$$e \bullet e' \triangleq \begin{cases} e & \text{if } (e = e' = v) \vee (e = e' = \underline{v}) \\ v & \text{if } \{e, e'\} = \{\underline{v}, v\} \\ \hat{v} & \text{if } \{e, e'\} = \{\underline{v}, \hat{v}\} \end{cases}$$

Some notation

$s \not\downarrow$ not ground link chain (i.e. contains virtuals);

$g = [\![\, t \,]\!]_\sigma$ substitution acting on variables, only;

$t \downarrow$ a ground tuple, not variables are present.

(Relevant) SOS rules 1/2

$$\frac{\ell \bowtie s \quad s \not\downarrow \quad g = [\![t]\!]_\sigma}{\ell t.P \xrightarrow{sg} P\sigma} \text{(Act1)}$$

$$\frac{t \downarrow}{\ell t.P \xrightarrow{\ell} P} \text{(Act2)}$$

$$\frac{P \xrightarrow{sg} P' \qquad a \in g}{(\nu\, a)P \xrightarrow{(\nu\, a)(sg)} P'} \text{(Open)}$$

$$\frac{P \xrightarrow{sg} P' \qquad a \notin g}{(\nu\, a)P \xrightarrow{(\nu\, a)(sg)} (\nu\, a)P'} \text{(Res)}$$

(analogous to (early) pi rules)

(Relevant) SOS rules 2/2

$$\frac{P \xrightarrow{sg} P' \quad Q \xrightarrow{s'g'} Q' \qquad g\#Q \quad g'\#P \qquad (s \bullet s')\not\downarrow}{P|Q \xrightarrow{(sg)\bullet(s'g')} P'|Q'}\ (\mathrm{Com})$$

$$\frac{P \xrightarrow{sg} P' \qquad Q \xrightarrow{s'g'} Q' \qquad g\#Q \quad g'\#P \quad (s \bullet s')\downarrow \quad (g \bullet g')\downarrow}{P|Q \xrightarrow{s\bullet s'} (\nu\, ex(g \bullet g'))(P'|Q')}\ (\mathrm{Close})$$

(analogous to (early) pi rules)

Fact

The process calculus of linked interactions with name mobility is a straightforward extension of pi-calculus
It includes pi-calculus as a sub-calculus

Finer (bisimilarity over the) LTS pi-calculus
(but it is a congruence)

Milner's pi interaction

$$\bar{a}x.P \mid a(y).Q$$

$$\downarrow \tau$$

$$P \mid Q[x/y]$$

Milner's pi interaction in the link-calculus

$$ {}^{\tau}\backslash_{a}\langle \underline{x} \rangle.P \;\Big|\; {}^{a}\backslash_{\tau}\langle y \rangle.Q $$

$$ \Big\downarrow {}^{\tau}\backslash_{\tau} $$

$$ P \mid Q[y/x] $$

Roadmap

- ✓ Why multi-party interaction ? And open ?
- ✓ A new kind of interaction (subsuming CCS)
- ✓ Handling message content (subsuming π-calculus)
- ✓ Encoding membranes
- ✓ Encoding reaction systems
- ✓ Conclusion and future work

Named, mobile, active, hierarchical ambients

An ambient is a place where computation happens
An ambient defines some sort of boundary

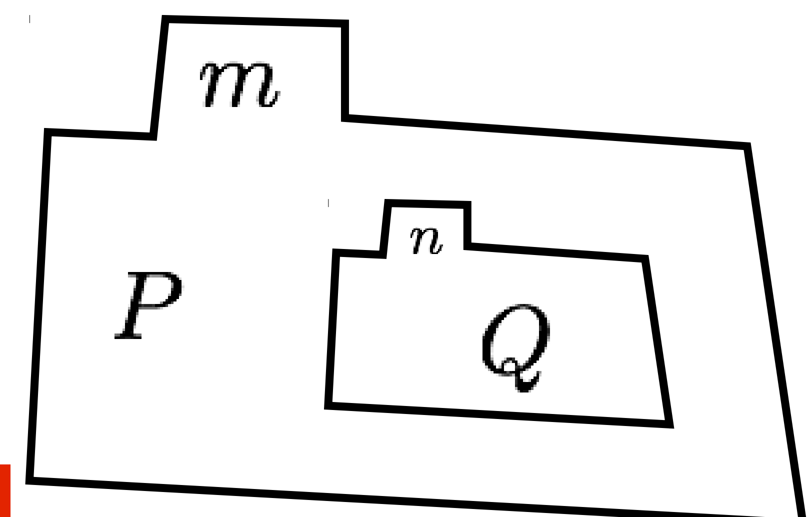
An ambient has a name
An ambient has a collection of local processes
An ambient has a collection of sub-ambients

Ambients are subject to capabilities:
Ambients can move in/out of other ambients
Ambients can dissolve

(Pure) Ambient calculus

$$
\begin{array}{rrl}
P ::= & \mathbf{0} & \text{nil} \\
 & \boxed{m[P]} & \boxed{\text{ambient}} \\
 & \boxed{M.P} & \boxed{\text{exercise a capability}} \\
 & P \mid Q & \text{parallel} \\
 & (\nu a)P & \text{restriction} \\
 & !P & \text{replication}
\end{array}
$$

$$
\begin{array}{rrl}
M ::= & \mathsf{in}\ m & \text{entry capability} \\
 & \mathsf{out}\ m & \text{exit capability} \\
 & \mathsf{open}\ m & \text{open capability}
\end{array}
$$



Ambient calculus: semantics

Structural congruence

$$
\begin{array}{lll}
P \equiv P & Q \equiv P \Rightarrow P \equiv Q & P \equiv Q, Q \equiv R \Rightarrow P \equiv R \\
P \mid \mathbf{0} \equiv P & P \mid Q \equiv Q \mid P & (P \mid Q) \mid R \equiv P \mid (Q \mid R) \\
(\nu n)\mathbf{0} \equiv \mathbf{0} & (\nu n)(\nu m)P \equiv (\nu m)(\nu n)P & P \equiv Q \Rightarrow P|R \equiv Q|R \\
 & (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q, \ \text{if } n \notin fn(P) & P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q \\
!P \equiv P \mid !P & (\nu n)(m[P]) \equiv m[(\nu n)P], \ \text{if } n \neq m & P \equiv Q \Rightarrow n[P] \equiv n[Q]
\end{array}
$$

Ambient calculus: semantics

Structural congruence

$$
\begin{array}{lll}
P \equiv P & Q \equiv P \Rightarrow P \equiv Q & P \equiv Q, Q \equiv R \Rightarrow P \equiv R \\
P \mid \mathbf{0} \equiv P & P \mid Q \equiv Q \mid P & (P \mid Q) \mid R \equiv P \mid (Q \mid R) \\
(\nu n)\mathbf{0} \equiv \mathbf{0} & (\nu n)(\nu m)P \equiv (\nu m)(\nu n)P & P \equiv Q \Rightarrow P|R \equiv Q|R \\
 & (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q, \ \text{if } n \notin fn(P) & P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q \\
!P \equiv P \mid !P & (\nu n)(m[P]) \equiv m[(\nu n)P], \ \text{if } n \neq m & P \equiv Q \Rightarrow n[P] \equiv n[Q]
\end{array}
$$

Reduction semantics

$$\frac{}{n[\,\mathsf{in}\ m.P \mid Q\,] \mid m[\,R\,] \to m[\,n[\,P \mid Q\,] \mid R\,]}\ \boxed{(\mathrm{In})}$$

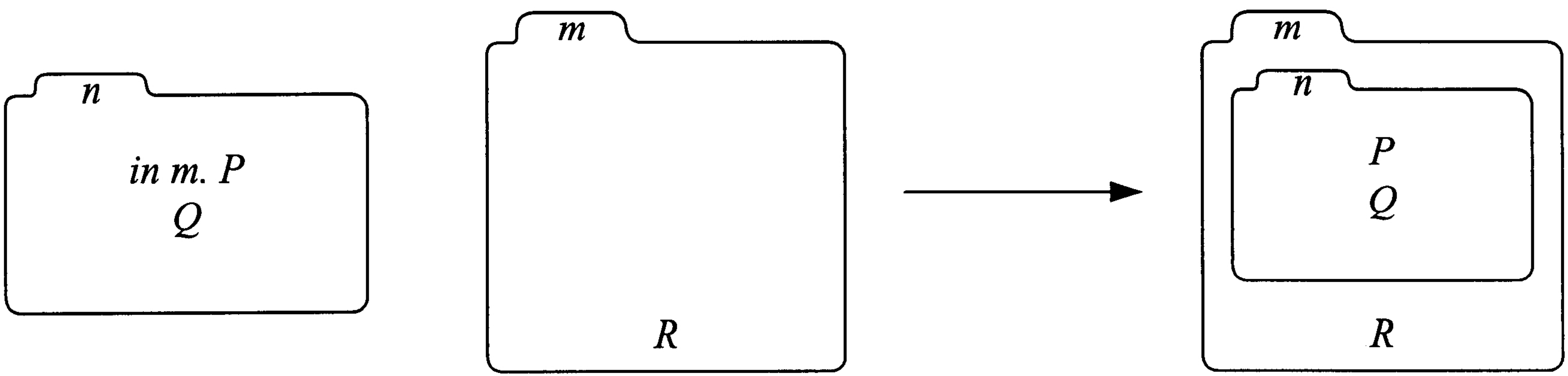
$$\frac{}{m[\,n[\,\mathsf{out}\ m.P \mid Q\,] \mid R\,] \to n[\,P \mid Q\,] \mid m[\,R\,]}\ \boxed{(\mathrm{Out})}$$

$$\frac{}{\mathsf{open}\ n.P \mid n[\,Q\,] \to P \mid Q}\ \boxed{(\mathrm{Open})} \qquad \frac{P \to Q}{(\nu\, n)P \to (\nu\, n)Q}\ (\mathrm{Res}) \qquad \frac{P \to Q}{n[\,P\,] \to n[\,Q\,]}\ (\mathrm{Amb})$$

$$\frac{P \to Q}{P \mid R \to Q \mid R}\ (\mathrm{Par}) \qquad \frac{P' \equiv P \qquad P \to Q \qquad Q \equiv Q'}{P' \to Q'}\ (\mathrm{Cong})$$

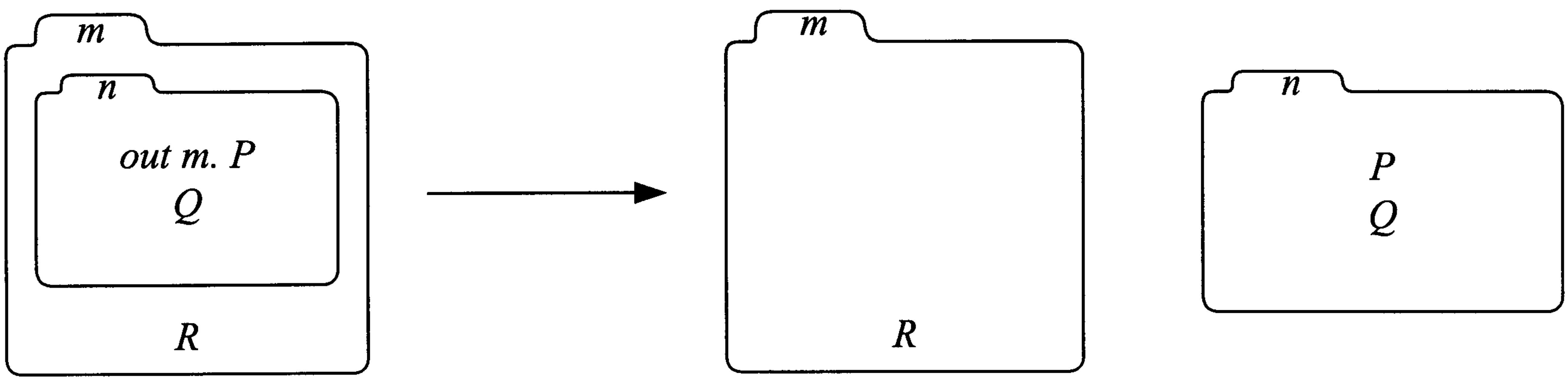
(In)

$$\frac{}{n[\,\mathsf{in}\, m.P \mid Q\,] \mid m[\,R\,] \rightarrow m[\,n[\,P \mid Q\,] \mid R\,]}\ \text{(In)}$$



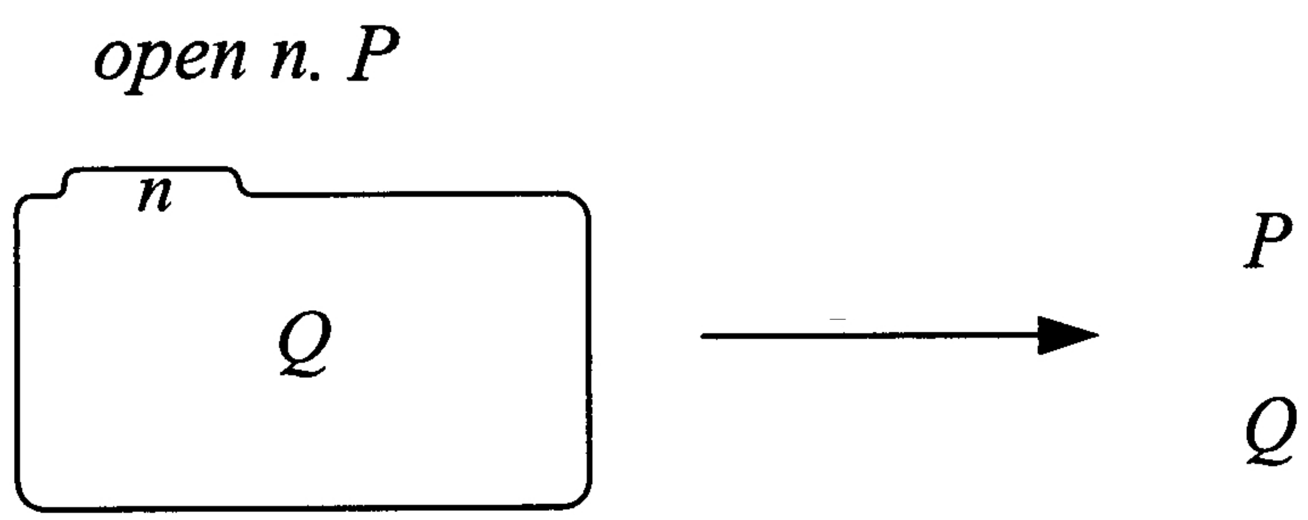
(Out)

$$\frac{}{m[\,n[\,\mathsf{out}\,m.P \mid Q\,] \mid R\,] \rightarrow n[\,P \mid Q\,] \mid m[\,R\,]}\ \text{(Out)}$$



(Open)

$$\frac{}{\mathsf{open}\, n.P \mid n[\,Q\,] \rightarrow P \mid Q} \text{ (Open)}$$



Encoding ambients

Why is it difficult to encode ambients into pi?
(How would you proceed?)

Our guess:
it is just because ambient-like interaction
is inherently non-dyadic!

Motivating example

How to encode Cardelli and Gordon’s mobile ambients
(in ordinary process calculi)?

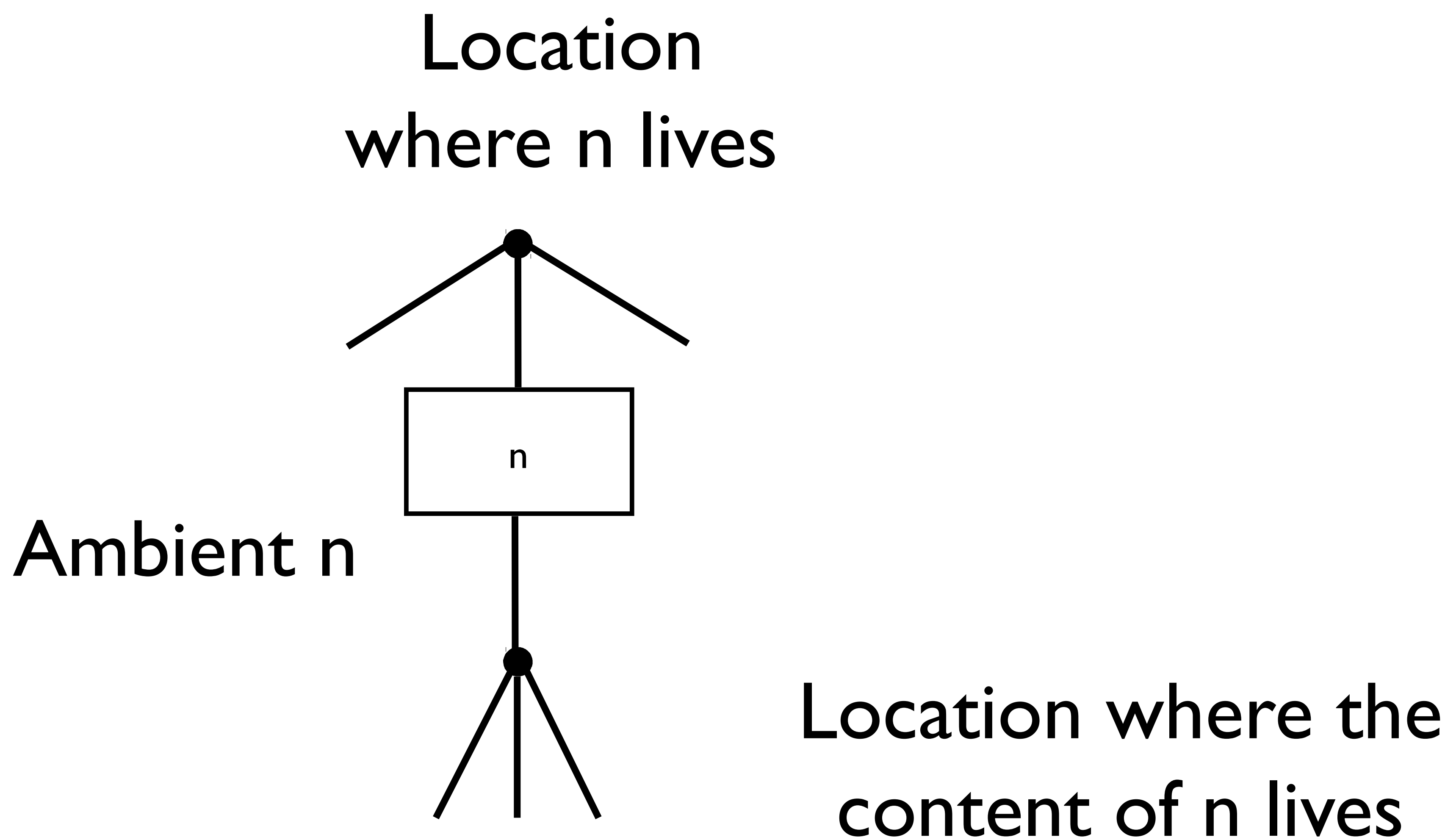
CCS/CSP:
immutable connectivity

pi:
channel mobility

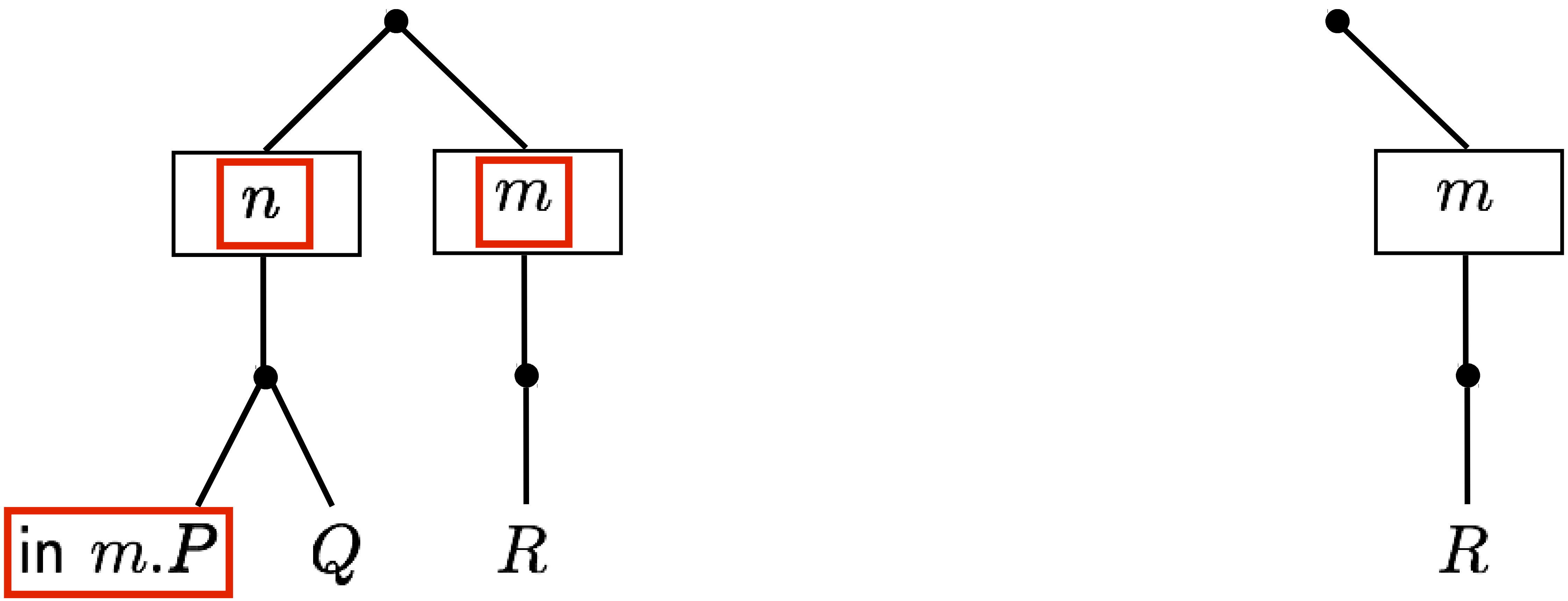
mobile ambients:
mobility of nested processes
(barrier crossing)

HOpi:
flat process mobility

Ambients as graphs

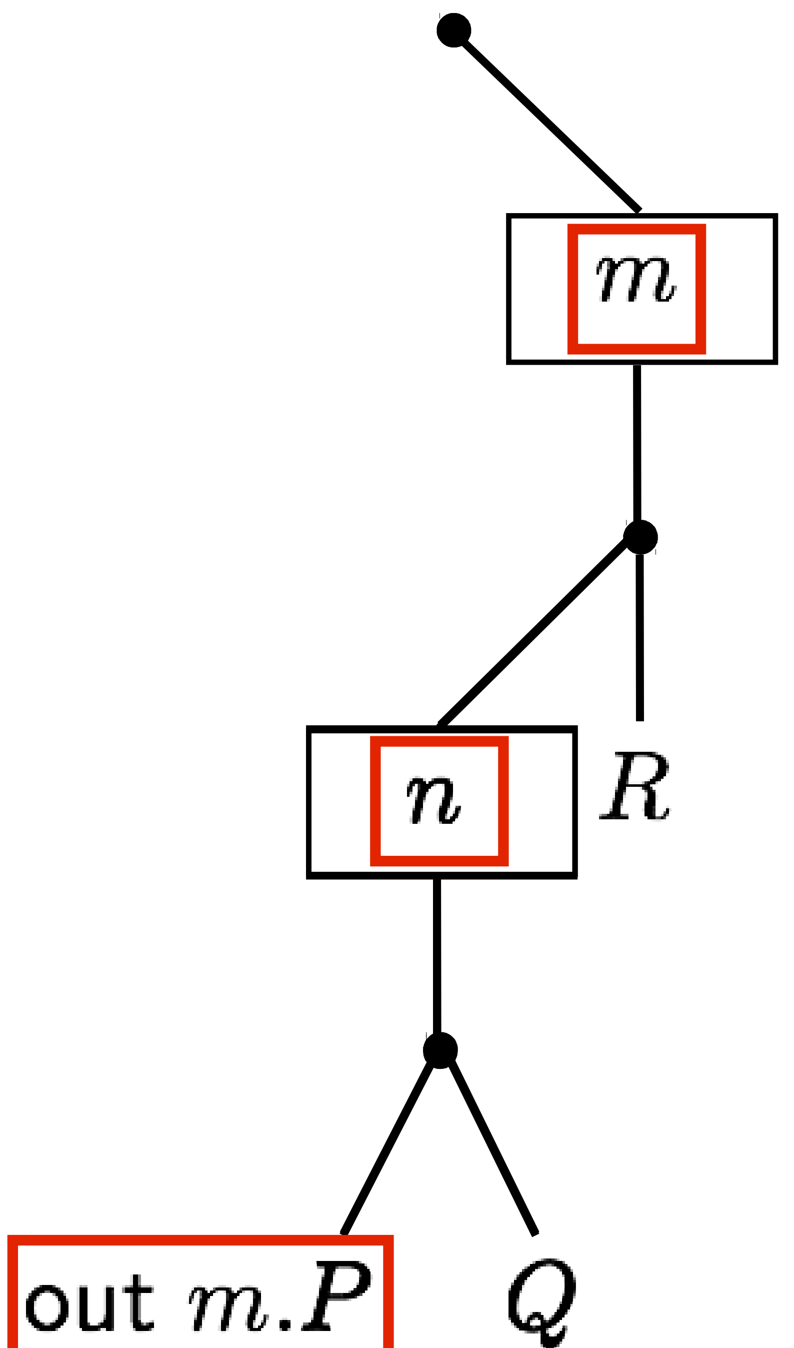


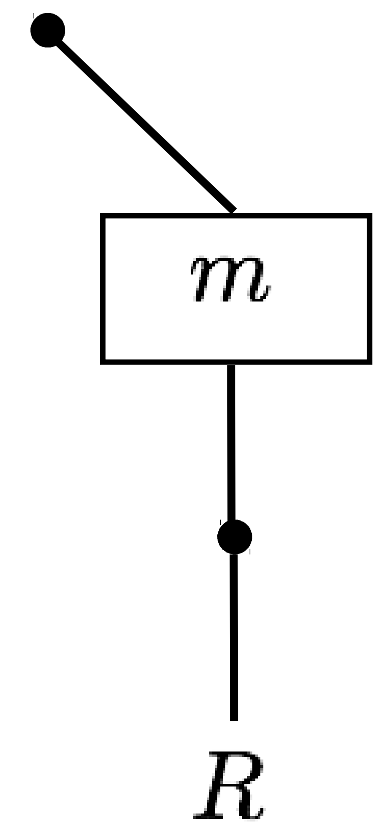
(In)



three-party interaction
(at least)

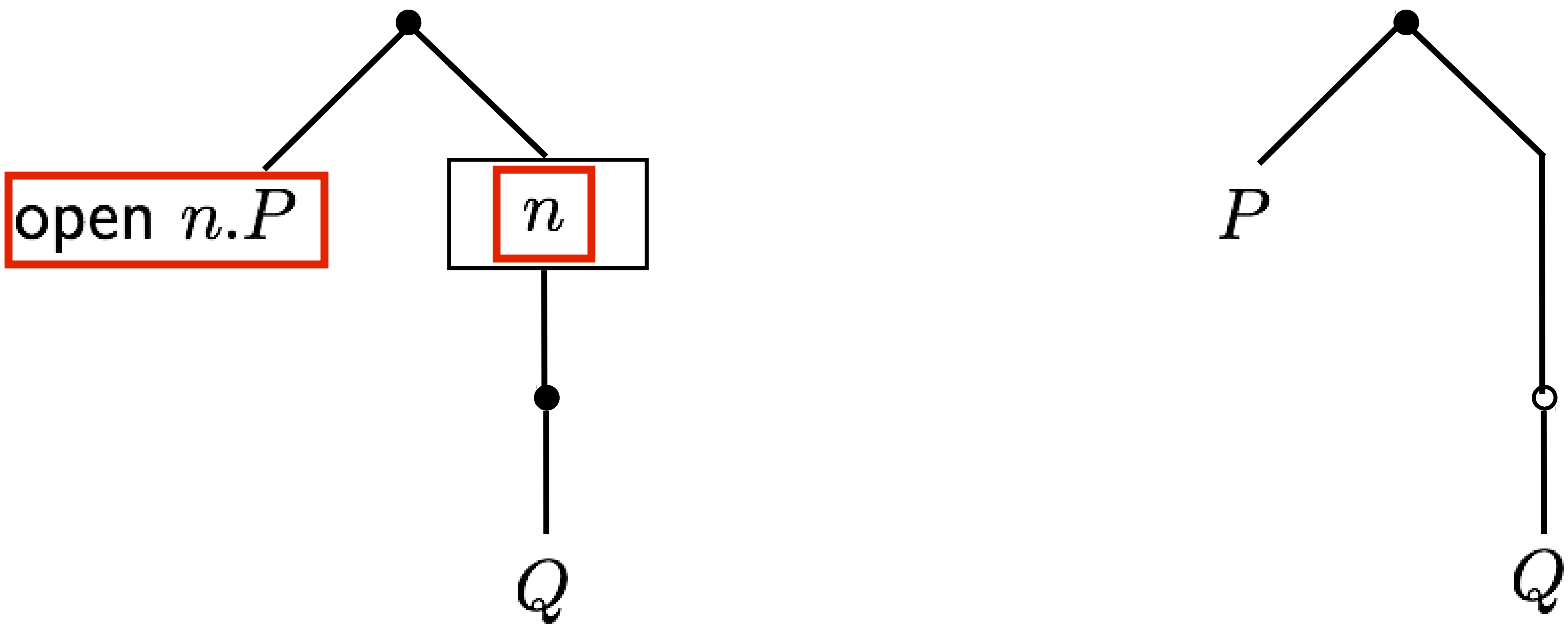
(Out)





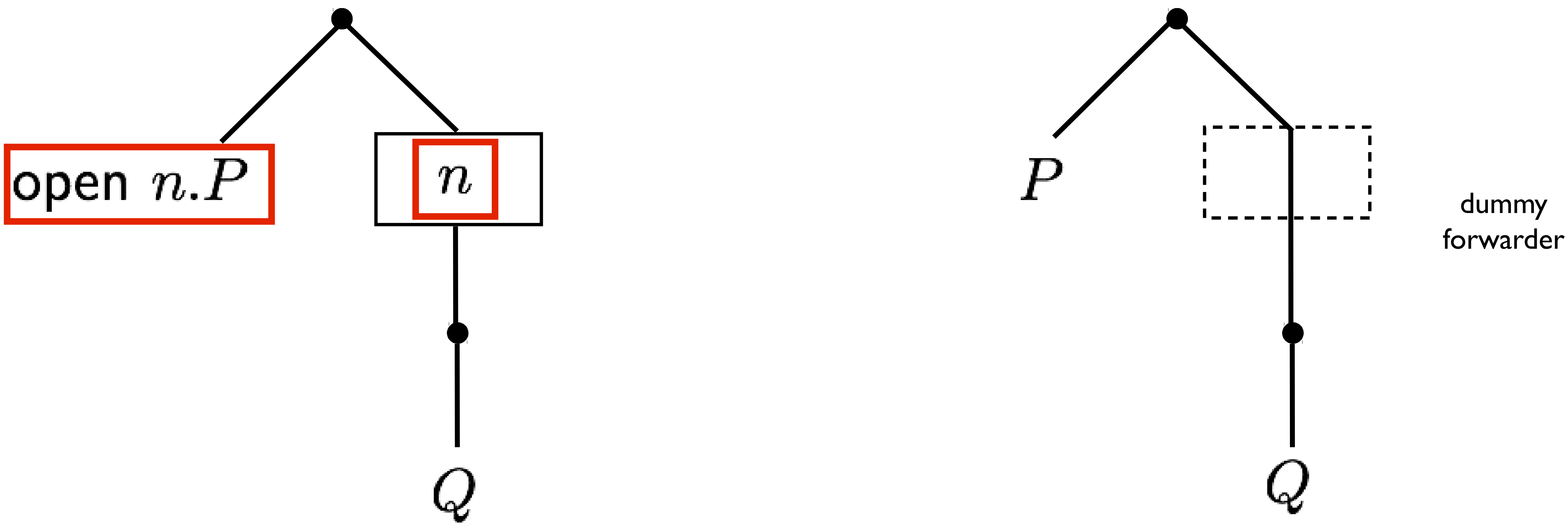
three-party interaction
(at least)

(Open)



looks like a two-party interaction, but it is not!
It is open! (accident of fate):
many processes (Q) change location at once

(Open)



ok, now it is a two-party interaction
But (In) and (Out) become open!
they must involve as many fwd-ers as needed

Some consequences

Proposed encoding are either quite involved
or centralized (unnecessary bottle-necks)

LTS semantics for ambients are ad-hoc
(to say the least)
and based on HO labels

Some references

- Fabio Gadducci, Giacoma Valentina Monreale: A decentralised graphical implementation of mobile ambients. J. Log. Algebr. Program. 80(2): 113-136 (2011)
- Linda Brodo: On the Expressiveness of the pi-Calculus and the Mobile Ambients. TCS : 44-59 (2018)
- Gabriel Ciobanu, Vladimir A. Zakharov: Encoding Mobile Ambients into the pi -Calculus. Ershov Memorial Conference 2006: 148-165
- Linda Brodo, Pierpaolo Degano, Corrado Priami: Reflecting Mobile Ambients into the pi-Calculus. Global Computing 2003: 25-56
- Cédric Fournet, Jean-Jacques Lévy, Alan Schmitt: An Asynchronous, Distributed Implementation of Mobile Ambients. IFIP TCS 2000: 348-364

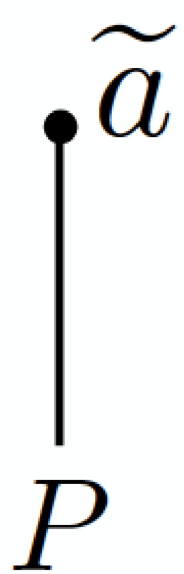
Our inital (very) motivation

Yes, our first initial motivation in defining
the link-calculus
was to define an encoding of the Mobile Ambients
obtaining a one-to-one operational
correspondance
with no central control !

Encoding mobile ambients

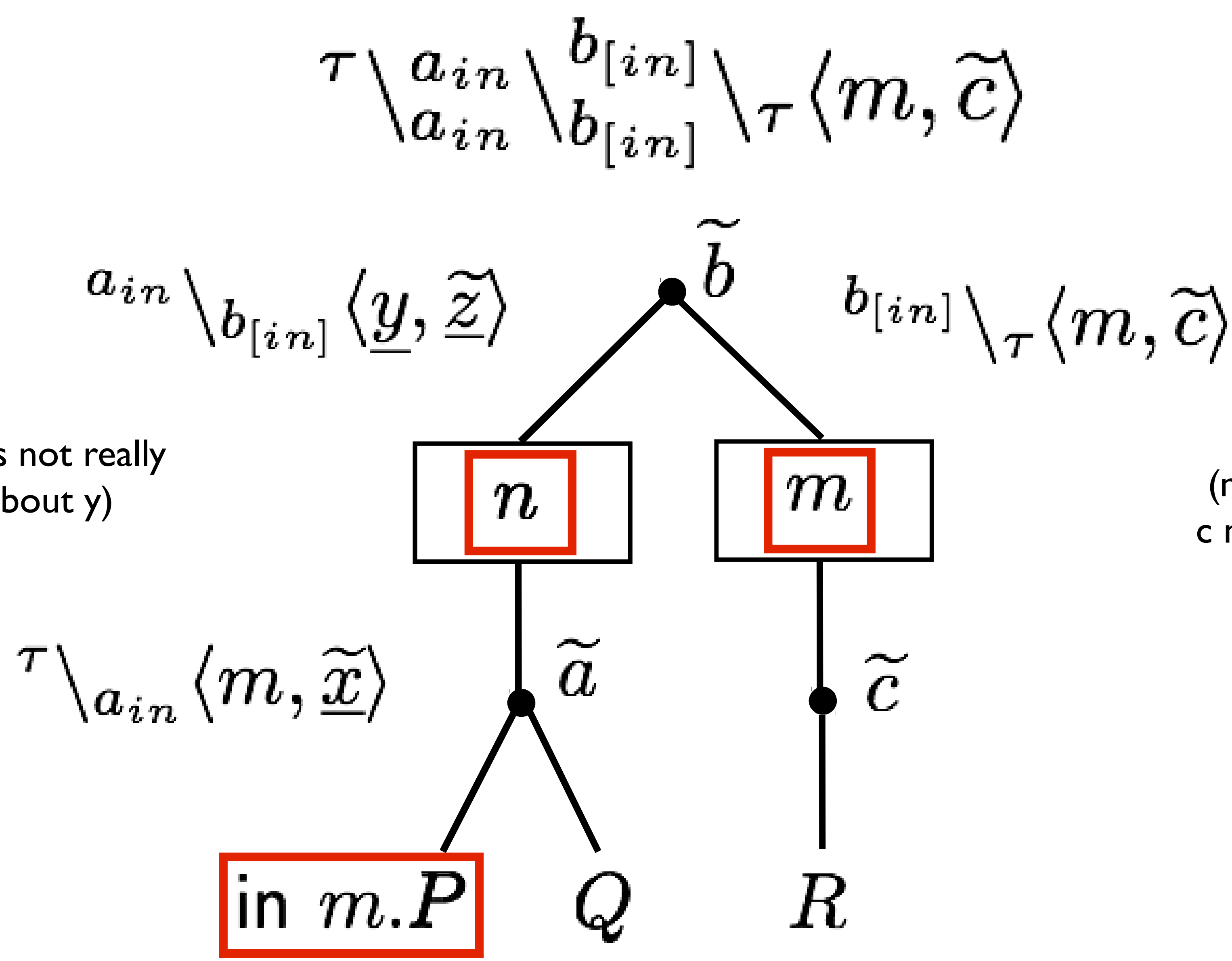
$[\![P]\!]_{\tilde{a}}$

a_{in}	requests from in capability
$a_{[in]}$	requests from an ambient with in capability inside
a_{out}	requests from out capability
$a_{[out]}$	requests from an ambient with out capability inside
a_{opn}	requests from open capability



Sketch of the idea

(n[ ] does not really care about y)



(m must match, c needed by n[ ])

(P does not really care about x)

c (and a) are typically restricted: c must be extruded

Desiderata

$P \to P'$ implies $[\![P]\!] \to [\![P']\!]$

$[\![P]\!] \to Q$ implies $\exists P' \quad Q = [\![P']\!] \quad P \to P'$

But both statements fail because of forwarders!

Roundabout

Extend ambients with parentheses

$$P \quad ::= \quad \cdots \mid (\!| P |\!)$$

They are introduced when an ambient is dissolved

The encoding

$$
\begin{aligned}
[\![\mathbf{0}]\!]_{\tilde{a}} &\triangleq \mathbf{0} \\
[\![n[P]]\!]_{\tilde{a}} &\triangleq (\nu \tilde{b})(Amb(n,\tilde{b},\tilde{a}) | [\![P]\!]_{\tilde{b}}) \\
[\![(\!| P |\!)]\!]_{\tilde{a}} &\triangleq (\nu \tilde{b})(Fwd(\tilde{b},\tilde{a}) | [\![P]\!]_{\tilde{b}}) \\
[\![\mathsf{in}\, m.P]\!]_{\tilde{a}} &\triangleq {}^{\tau}\backslash_{a_{in}} \langle m, \underline{\tilde{x}} \rangle . [\![P]\!]_{\tilde{a}} \\
[\![\mathsf{out}\, m.P]\!]_{\tilde{a}} &\triangleq {}^{\tau}\backslash_{a_{out}} \langle m, \underline{\tilde{x}} \rangle . [\![P]\!]_{\tilde{a}} \\
[\![\mathsf{open}\, n.P]\!]_{\tilde{a}} &\triangleq {}^{\tau}\backslash_{a_{opn}} \langle n \rangle . [\![P]\!]_{\tilde{a}}
\end{aligned}
$$

$$
\begin{aligned}
[\![P|Q]\!]_{\tilde{a}} &\triangleq [\![P]\!]_{\tilde{a}} | [\![Q]\!]_{\tilde{a}} \\
[\![(\nu n)P]\!]_{\tilde{a}} &\triangleq (\nu n)[\![P]\!]_{\tilde{a}} \\
[\![!P]\!]_{\tilde{a}} &\triangleq \mathsf{rec}X.([\![P]\!]_{\tilde{a}} | X)
\end{aligned}
$$

$$
\begin{aligned}
Amb(n,\tilde{a},\tilde{f}) \triangleq\ & {}^{a_{in}}\backslash_{f_{[in]}} \langle \underline{m}, \underline{\tilde{z}} \rangle . Amb(n,\tilde{a},\tilde{z}) + {}^{f_{[in]}}\backslash_{\tau} \langle n, \tilde{a} \rangle . Amb(n,\tilde{a},\tilde{f}) + \\
& {}^{a_{out}}\backslash_{f_{[out]}} \langle \underline{m}, \underline{\tilde{z}} \rangle . Amb(n,\tilde{a},\tilde{z}) + {}^{a_{[out]}}\backslash_{\tau} \langle n, \tilde{f} \rangle . Amb(n,\tilde{a},\tilde{f}) + \\
& {}^{f_{opn}}\backslash_{\tau} \langle n \rangle . Fwd(\tilde{a},\tilde{f}) \\
Fwd(\tilde{a},\tilde{f}) \triangleq\ & {}^{a_{in}}\backslash_{f_{in}} \langle \underline{n}, \underline{\tilde{x}} \rangle . Fwd(\tilde{a},\tilde{f}) + {}^{a_{[in]}}\backslash_{f_{[in]}} \langle \underline{n}, \underline{\tilde{x}} \rangle . Fwd(\tilde{a},\tilde{f}) + {}^{f_{[in]}}\backslash_{a_{[in]}} \langle \underline{n}, \underline{\tilde{x}} \rangle . Fwd(\tilde{a},\tilde{f}) + \\
& {}^{a_{out}}\backslash_{f_{out}} \langle \underline{n}, \underline{\tilde{x}} \rangle . Fwd(\tilde{a},\tilde{f}) + {}^{a_{[out]}}\backslash_{f_{[out]}} \langle \underline{n}, \underline{\tilde{x}} \rangle . Fwd(\tilde{a},\tilde{f}) + \\
& {}^{a_{opn}}\backslash_{f_{opn}} \langle \underline{n} \rangle . Fwd(\tilde{a},\tilde{f}) + {}^{f_{opn}}\backslash_{a_{opn}} \langle \underline{n} \rangle . Fwd(\tilde{a},\tilde{f})
\end{aligned}
$$

Desiderata (we got it!)

$P \to P'$ implies $[\![P]\!] \to [\![P']\!]$

$[\![P]\!] \to Q$ implies $\exists P' \quad Q = [\![P']\!] \quad P \to P'$

We claim that:

- ✓ membranes play an active role in blocking/allowing interactions;
- ✓ we encode them as processes;
- ✓ all the calculi equipped with such membranes are multi-party calculi;
- ✓ more in general, location can be model as a process.

Roadmap

- ✓ Why multi-party interaction ? And open ?
- ✓ A new kind of interaction (subsuming CCS)
- ✓ Handling message content (subsuming π-calculus)
- ✓ Encoding membranes
- ✓ Encoding reaction systems
- ✓ Conclusion and future work

Reaction Systems

A reaction system is a set of rules of the type:

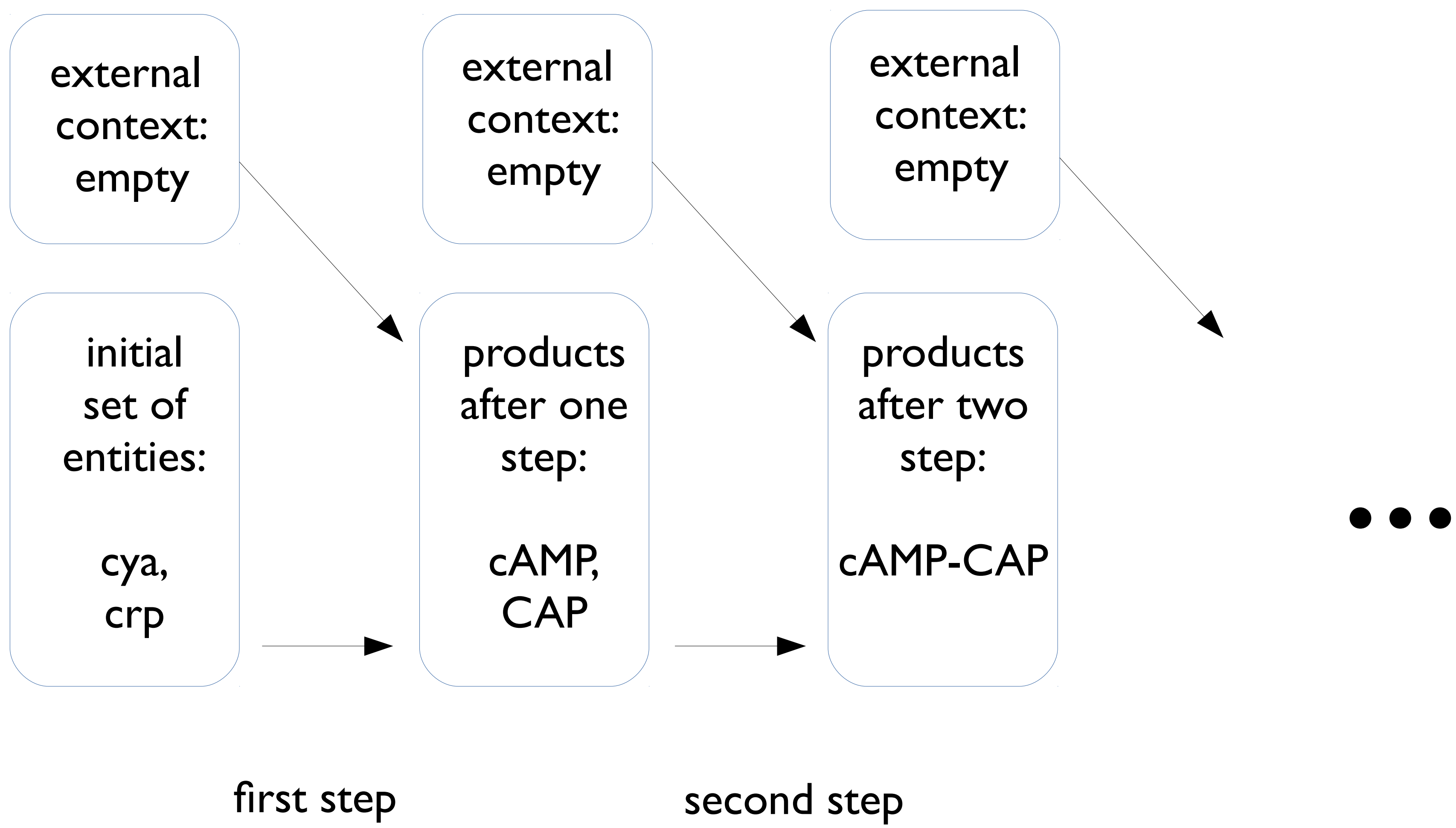
$$a = (R, I, P)$$

Reactants Inhibitors Products

$$(\{cAMP, CAP\}\ ,\ \{glucose\}\ ,\ \{cAMP\text{–}CAP\})$$

R. Brijder, A. Ehrenfeucht, M. Main, and G. Rozenberg. A tour of reaction systems. International Journal of Foundations of Computer Science, 22(07): 1499--1517, 2011.

Reaction Systems

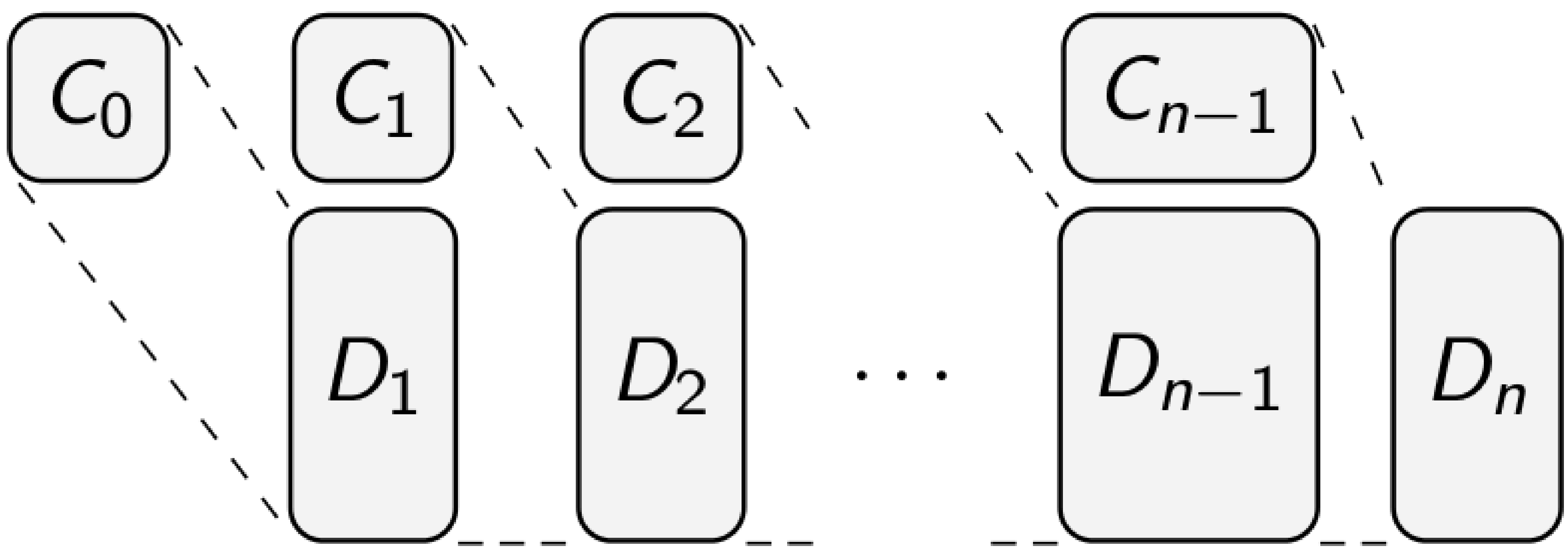


always are applied
(when possible)
all together

$$(\{cya\}, \{...\}, \{cAMP\})$$
$$(\{crp\}, \{...\}, \{CAP\})$$
$$(\{cAMP, CAP\}, \{glucose\}, \{cAMP\text{-}CAP\})$$

Reaction Systems

$$
\begin{aligned}
a_1 &= (\{lac\}, \{...\}, \{lac\}), & a_6 &= (\{cya\}, \{...\}, \{cAMP\}),\\
a_2 &= (\{lacI\}, \{...\}, \{lacI\}), & a_7 &= (\{crp\}, \{...\}, \{crp\}),\\
a_3 &= (\{lacI\}, \{...\}, \{I\}), & a_8 &= (\{crp\}, \{...\}, \{CAP\}),\\
a_4 &= (\{I\}, \{lactose\}, \{I\text{-}OP\}), & a_9 &= (\{cAMP, CAP\}, \{glucose\}, \{cAMP\text{-}CAP\}),\\
a_5 &= (\{cya\}, \{...\}, \{cya\}), & a_{10} &= (\{lac, cAMP\text{-}CAP\}, \{I\text{-}OP\}, \{Z, Y, A\}).
\end{aligned}
$$



C_i are the entities provided by the biological external context:

The *chained* link-calculus

Is a version of the link-calculus where prefixes are link chains.

syntax $P, Q ::= \sum_{i \in I} v_i.P_i \mid P|Q \mid (\nu a)P \mid P[\phi] \mid A$

link chain prefix $v = \ell_1 ... \ell_n$

relevant semantic rule

$$\frac{v \bowtie v_j}{\sum_{i \in I} v_i.P_i \xrightarrow{v} P_j} \,(Sum)$$

The enconding

(Sketch of the idea)

assuming a rs with only 2 reactions, and 5 entities:

reaction 1 $(\{cya\}, \{...\}, \{cAMP\})$

reaction 2 $(\{cAMP, CAP\}, \{glucose\}, \{cAMP\text{-}CAP\})$

encoding the two reactions

reaction 1 $P_1 \triangleq \tau \backslash^{\square}_{cya} \backslash^{cya}_{\square} \backslash^{\square}_{\widetilde{cAMP}} \backslash^{\widetilde{cAMP}}_{\square} \backslash_{r_2} . P_1 + \ \ldots$

reaction 2

$$P_2 \triangleq r_2 \backslash^{\square}_{cAMP_i} \backslash^{cAMP_o}_{\square} \backslash^{\square}_{CAP_i} \backslash^{CAP_o}_{\square} \backslash^{\square}_{\overline{glucose_i}} \backslash^{\overline{glucose_o}}_{\square} \backslash^{\square}_{\widetilde{cAMP\text{-}CAP_i}} \backslash^{\widetilde{cAMP\text{-}CAP_o}}_{\square} \backslash_{\tau} . P_2 + \ \ldots$$

The enconding

(Sketch of the idea)

the link chain prefixes of the two reactions can be linked
(*forming a sort of communication backbone*):

$$\tau \backslash^{\square}_{cya_i} \backslash^{cya_o}_{\square} \backslash^{\square}_{\widetilde{cAMP}_i} \backslash^{\widetilde{cAMP}_o}_{\square} \backslash r_2 \qquad r_2 \backslash^{\square}_{cAMP_i} \backslash^{cAMP_o}_{\square} \backslash^{\square}_{CAP_i} \backslash^{CAP_o}_{\square} \backslash^{\square}_{\overline{glucose}_i} \backslash^{\overline{glucose}_o}_{\square} \backslash^{\square}_{c\widetilde{AMP\text{-}CAP}_i} \backslash^{c\widetilde{AMP\text{-}CAP}_o}_{\square} \backslash \tau$$

what is still missing is the contribution of the single entities (molecules)

The enconding

(Sketch of the idea)

$$\tau\backslash^{\Box}_{cya_i}\backslash^{cya_o}_{\Box}\backslash^{\Box}_{\widetilde{cAMP}_i}\backslash^{\widetilde{cAMP}_o}_{\Box}\backslash_{r_2} \quad {}^{r_2}\backslash^{\Box}_{cAMP_i}\backslash^{cAMP_o}_{\Box}\backslash^{\Box}_{CAP_i}\backslash^{CAP_o}_{\Box}\backslash^{\Box}_{\overline{glucose}_i}\backslash^{\overline{glucose}_o}_{\Box}\backslash^{\Box}_{\widetilde{cAMP\text{-}CAP}_i}\backslash^{\widetilde{cAMP\text{-}CAP}_o}_{\Box}\backslash_{\tau}$$

$$P_{CAP} \triangleq {}^{CAP_i}\backslash_{CAP_o}.P_{CAP}$$

$$P_{\overline{glucose}} \triangleq {}^{\overline{glucose}_i}\backslash_{\overline{glucose}_o}.P_{\overline{glucose}}$$

$$P_{cya} \triangleq {}^{cya_i}\backslash_{cya_o}.P_{cya}$$

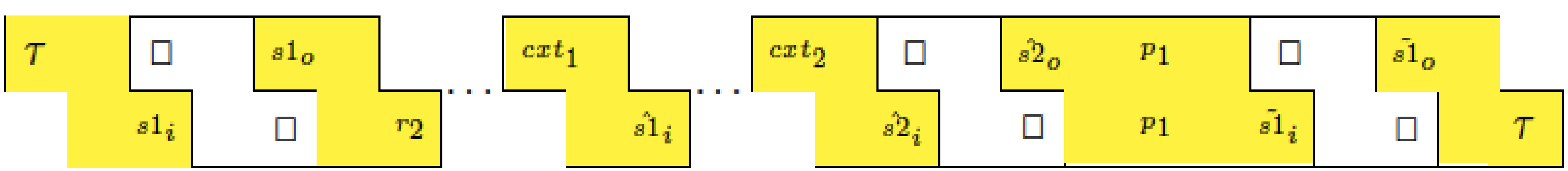
$$P_{cAMP} \triangleq {}^{\widetilde{cAMP}_i}\backslash^{\Box}_{\widetilde{cAMP}_o}\backslash^{cAMP_i}_{\Box}\backslash_{cAMP_o}.P_{cAMP}$$

$$P_{cAMP\text{-}CAP} \triangleq {}^{\widetilde{cAMP\text{-}CAP}_i}\backslash_{\widetilde{cAMP\text{-}CAP}_o}.P_{cAMP\text{-}CAP}$$

encoding the entities

What we gain:

- recursive contexts
- modeling mutating entities
- communicating reaction systems: for example, the lac operon system (that depends on the presence or absence of the glucose) can be connected with the system producing the lactose.
- modeling style: backbone + resources: the processes encoding the reactions and the context form the backbone; processes encoding entities provide the resources.



Bibliography

Bodei, C., Brodo, L. , Bruni, R.
A Formal Approach to Open Multiparty Interactions
(2019) Theoretical Computer Science,
Volume 763: pp. 38-65

Brodo, L., Olarte, C.
Symbolic semantics for multiparty interactions in the link-calculus
(2017) Lecture Notes in Computer Science,
volume 10139 LNCS, pp. 62-75

Bodei, C., Brodo, L., Bruni, R., Chiarugi, D.
A flat process calculus for nested membrane interactions
(2014) Scientific Annals of Computer Science
Volume 24, Issue 1, pp. 91-136

Bodei, C., Brodo, L., Bruni, R.
Open multiparty interaction
(2013) Lecture Notes in Computer Science,
volume 7841 LNCS, pp. 1-23

Bodei, C., Brodo, L. , Bruni, R.
The link-calculus for Open Multiparty Interactions
submitted to Information and Computation

Brodo, L., Olarte, C.
Verification techniques for a network algebra
submitted to Fundamenta Informaticae

Brodo, L., Bruni, R., Falaschi, M.
Embedding reaction systems into link-calculus
submitted to Second International Workshop on Reaction Systems

The link-calculus homepage:
http://linkcalculus.di.unipi.it

Future work

We would like to:

- ✓ define quantitative extensions of the calculus;
- ✓ exploit the nature of link interaction for changing the abstraction level in modeling distributed system;
- ✓ explore the feasibility for modeling IoT;
- ✓ ...

✓


THANKS
FOR
YOUR
ATTENTION!