Symbolic semantics for multiparty interactions in the link-calculus

Linda Brodo
(University of Sassari, Italy)

joint work with
Carlos Olarte
(Universidade Federal do Rio Grande do Norte, Brazil)

Setting

Modelling concurrent communicating systems

Process calculi approach

Symbolic semantics

(some basic knowledge of CCS assumed,
some details omitted)

Roadmap

- A brief introduction to the link-calculus
- Symbolic link chains
- Definition of the symbolic semantics
- Definition of the symbolic bisimulation
- Conclusion and future work

Interaction

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

The starting point: Milner's CCS interaction

co-action prefix

$$a.P \;\mid\; \overline{a}.Q$$

action prefix

silent action

$$P \mid Q$$

The starting point: Milner's CCS interaction

co-action prefix

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

action prefix

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

silent action

$$P | Q$$

Would you...?

...model piano playing using dyadic interaction

Open multiparty interactions are like playing piano
(either bad or good, it does not matter)

Any better abstraction?

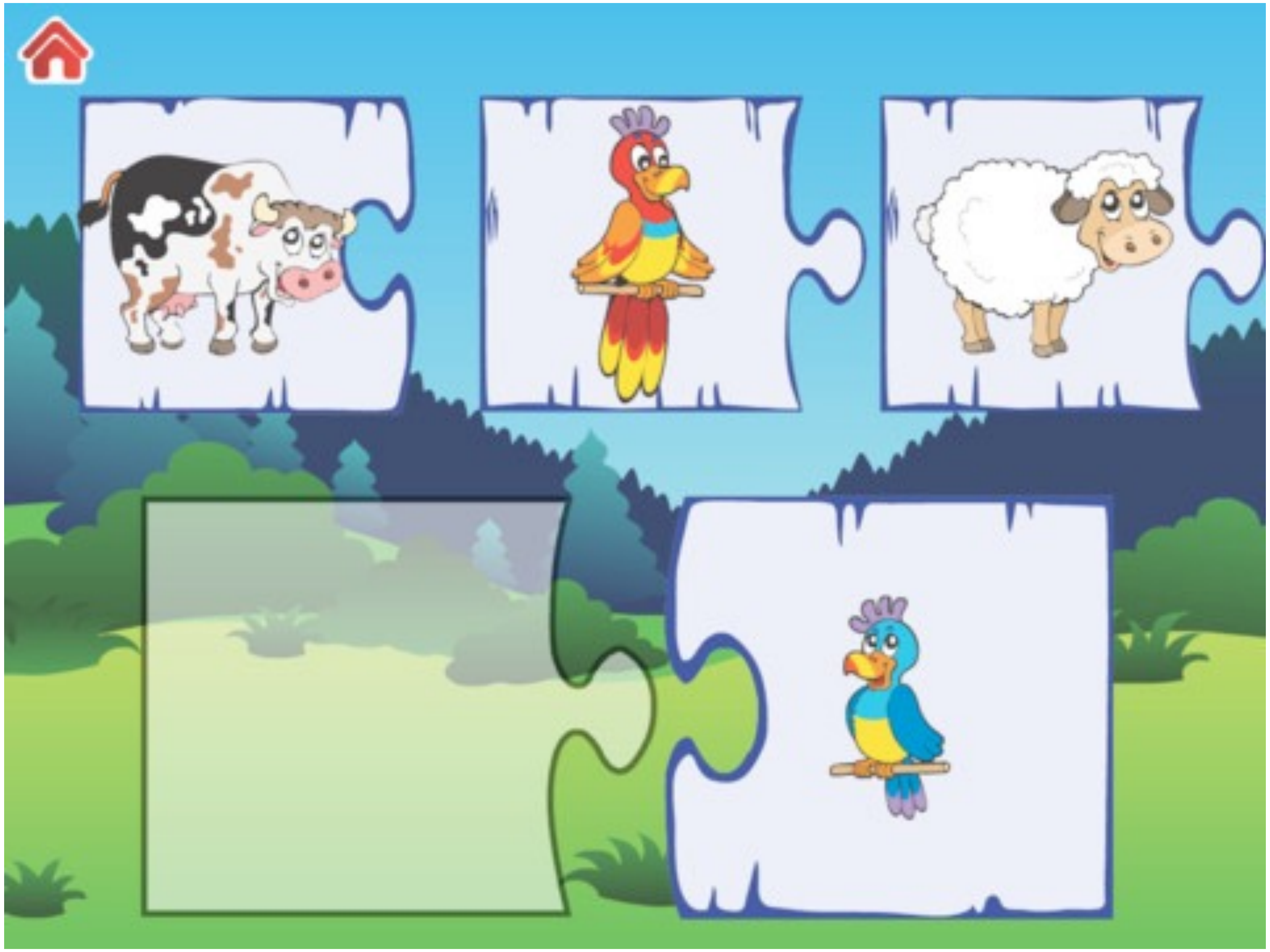
Internet
Biology
Social networks
...

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)

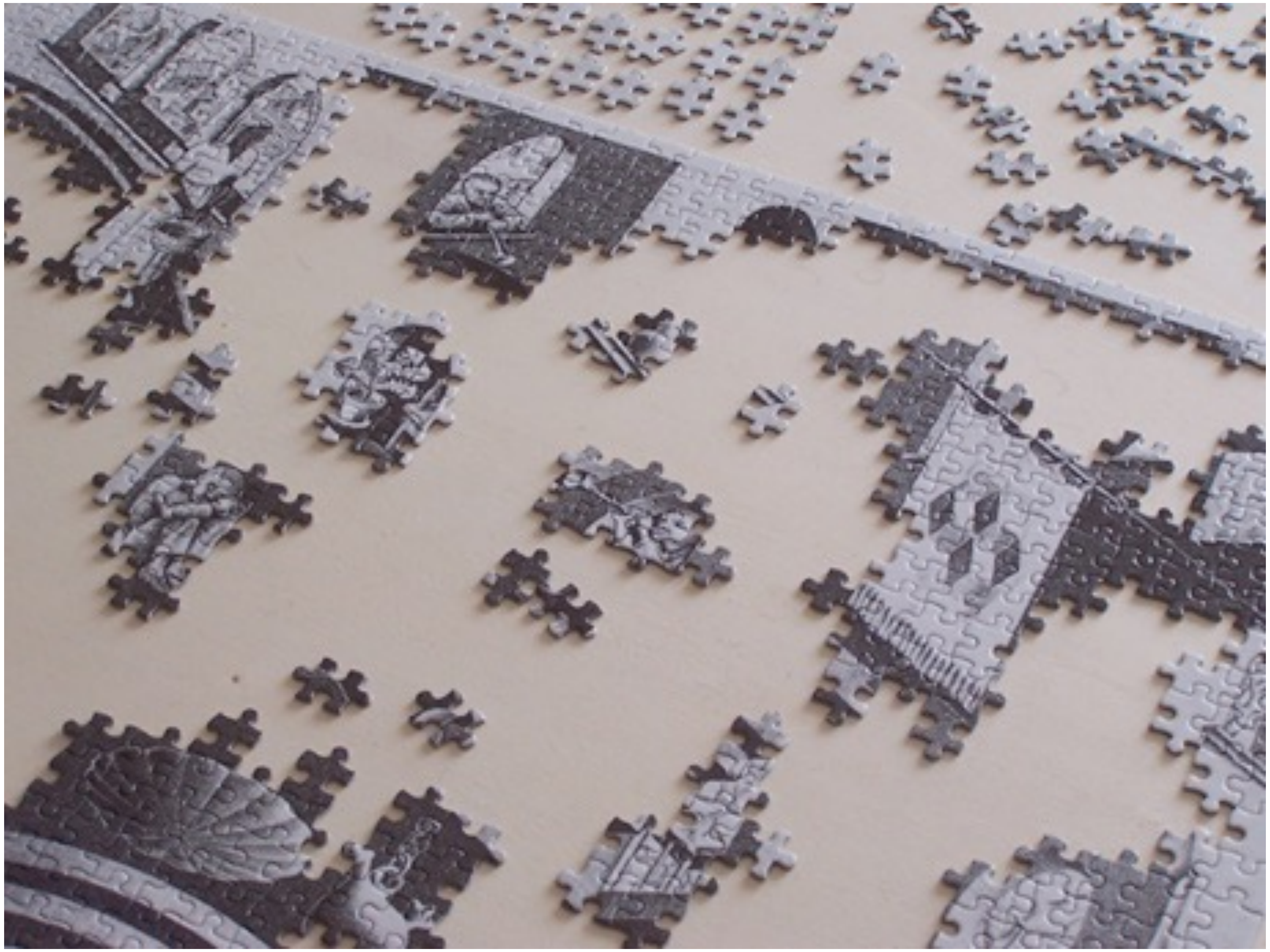
Kid's puzzle

Mutual (binary) interaction



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

Our aim

Extend
in a smooth and coherent way
the theory of dyadic interaction
to deal with
open multipary iteractions

Process algebra ops

$\mathbf{0}$	nil
$\mu.P$	action prefix
$P + Q$	sum
$P \mid Q$	parallel
$(\nu a)P$	restriction
$!P$	replication
X	process variable
rec $X.P$	recursive process
$P[\phi]$	renaming

Linked interaction

We regard an interaction as a chain of links
(still a kid's puzzle after all)

Process algebra ops

$\mathbf{0}$	nil
$\boxed{\mu.P}$	$\boxed{\text{action prefix}}$
$P+Q$	sum
$P \mid Q$	parallel
$(\nu a)P$	restriction
$!P$	replication
X	process variable
rec $X.P$	recursive process
$P[\phi]$	renaming

We take as action the offering of a link

Notation

a interaction over a

τ silent interaction

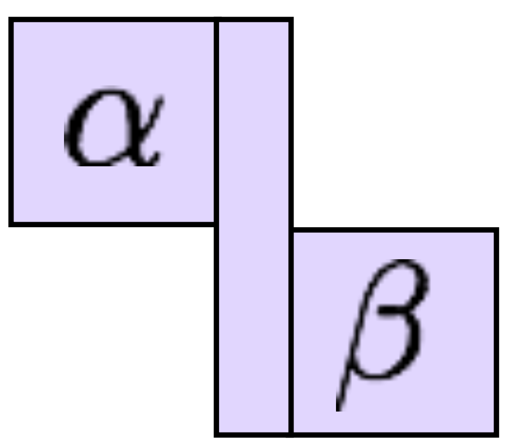
$\square$ any interaction (only in labels)

Link

${}^{\alpha}\backslash_{\beta}$ From α to β

Valid:

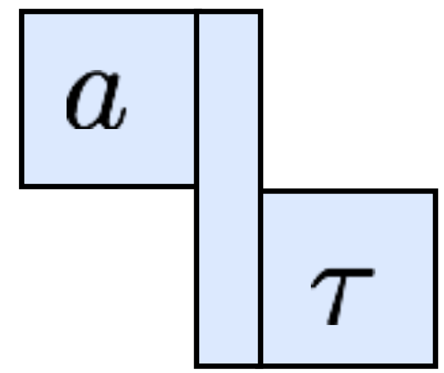
$$\alpha = \beta = \square \text{ or } \alpha, \beta \neq \square$$

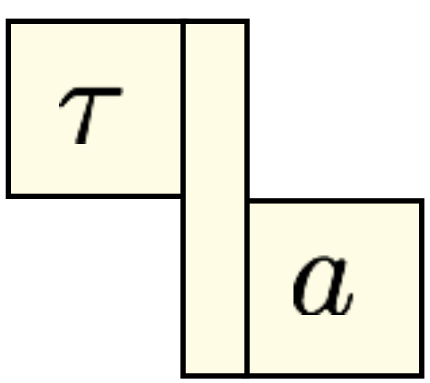


Virtual if ${}^{\square}\backslash_{\square}$

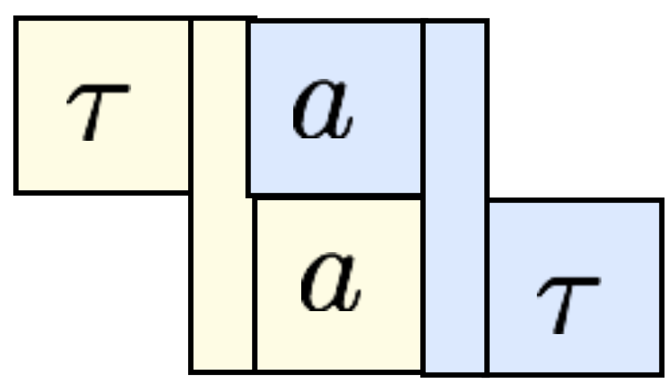
Solid (otherwise)

Examples: CCS-like



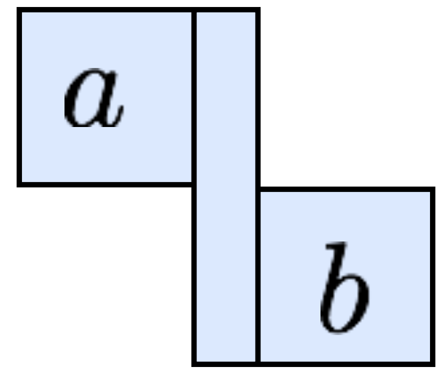


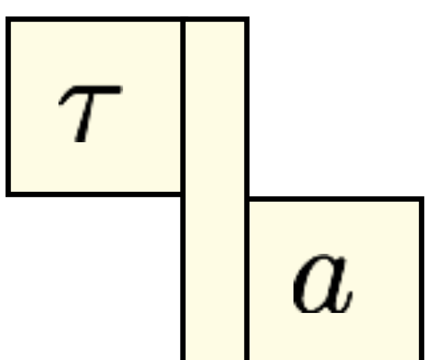
Examples: CCS-like

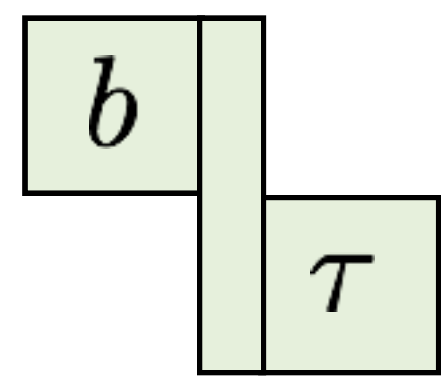


Examples: three party

Swiss-bank box

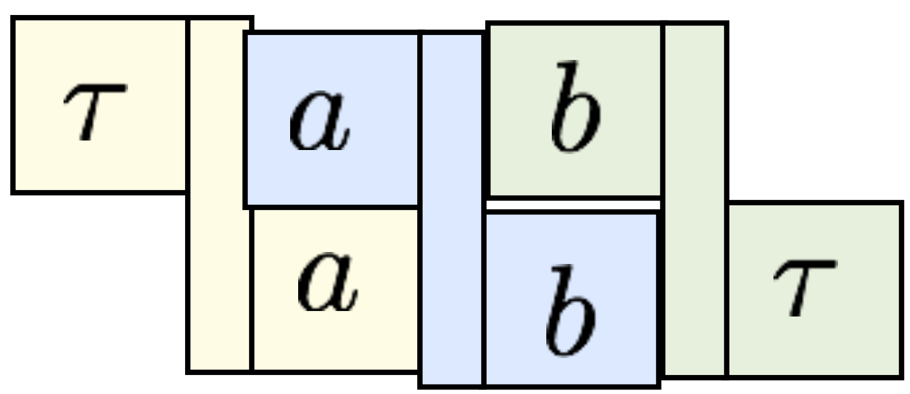






Examples: three party

Swiss-bank box



Link chain

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

such that:

$$\beta_i, \alpha_{i+1} \notin \{\tau, \Box\} \text{ implies } \beta_i = \alpha_{i+1}$$

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

$$\forall i. \alpha_i, \beta_i \in \{\tau, \Box\} \text{ implies } \forall i. \alpha_i = \beta_i = \tau$$

Link chain: terminology

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

Solid:

if all its links are so

Simple:

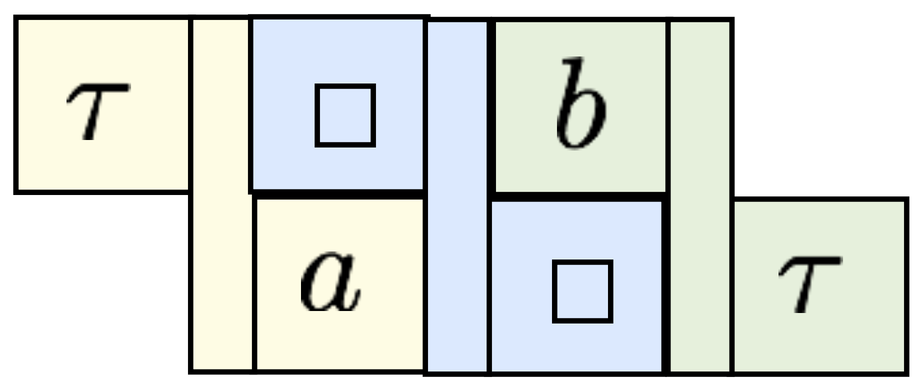
if it contains exactly one solid link

$s \bowtie \ell$:

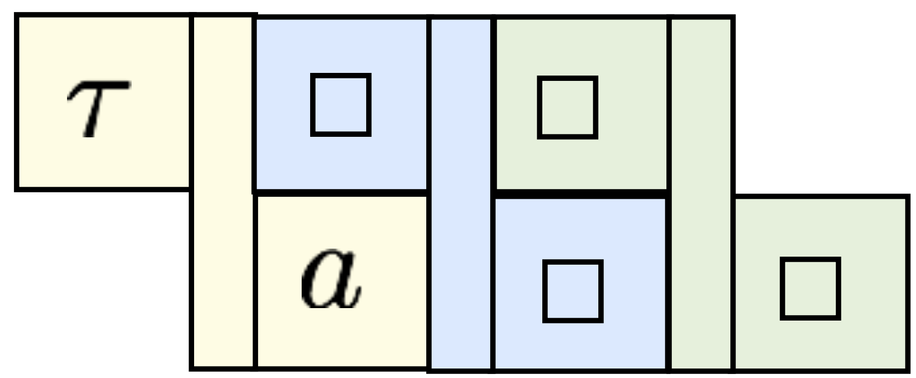
s is simple and ℓ is the only solid link in s

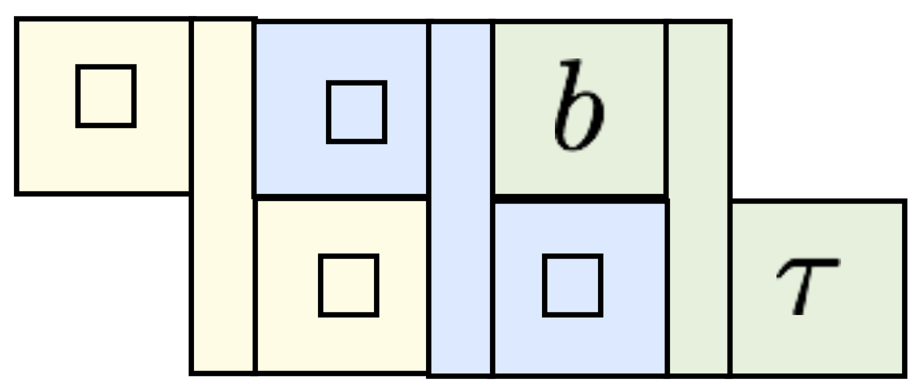
Examples: non solid

Virtual links $\square \backslash \square$
can be read as missing pieces of the puzzle

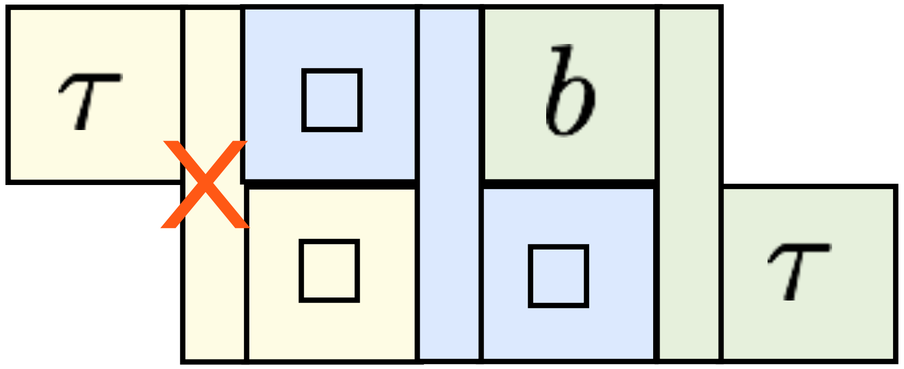


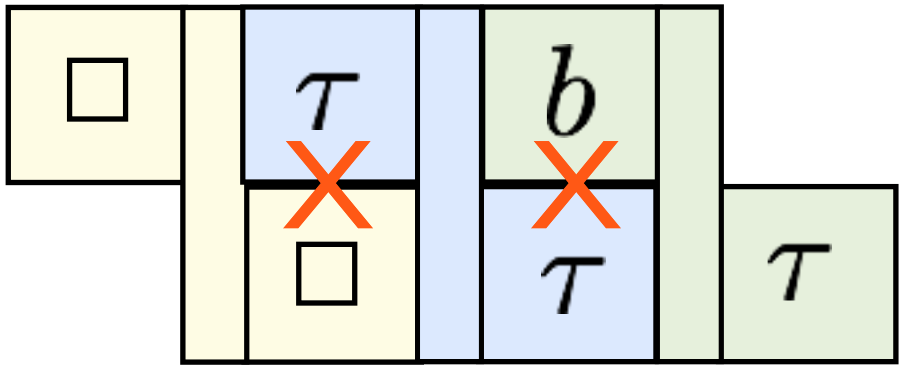
Examples: simple





Counter-examples





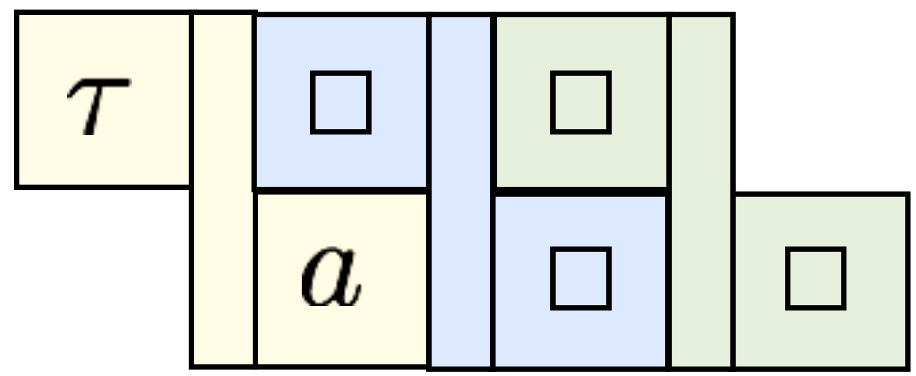
(Relevant) SOS rules

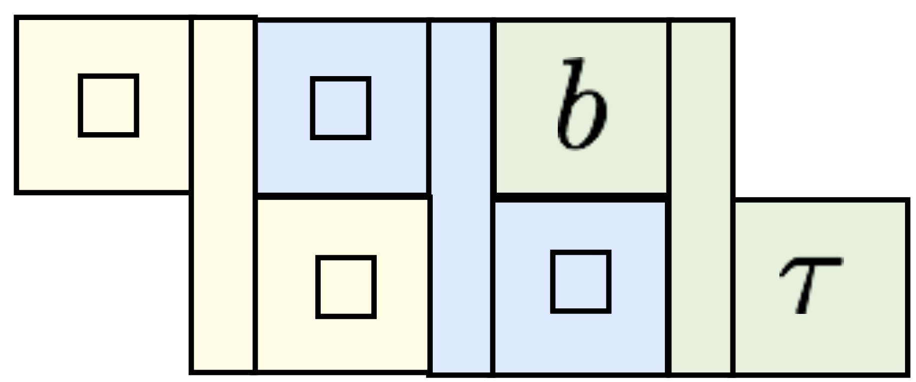
$$\frac{s \bowtie \ell}{\ell.P \xrightarrow{s} P} \ (\mathrm{Act})$$

equivalence relation

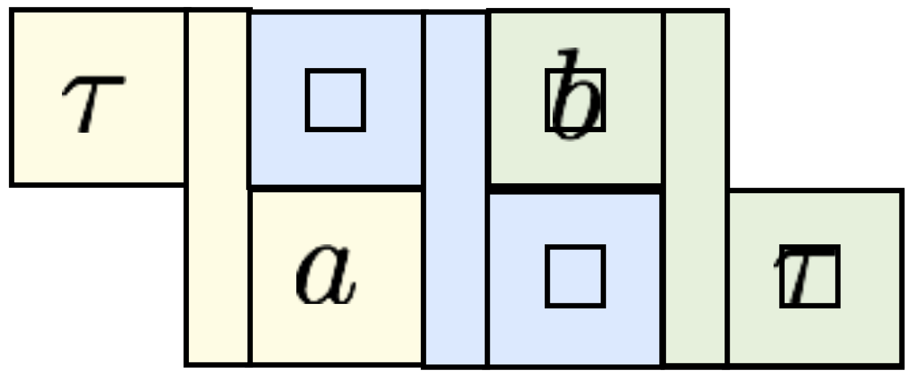
$$
\begin{array}{lcl}
s^{\Box}\backslash_{\Box} & \bowtie & s \\
{}^{\Box}\backslash_{\Box} s & \bowtie & s
\end{array}
\qquad
\begin{array}{rcl}
s_1{}^{\Box}\backslash^{\Box}_{\Box}\backslash_{\Box} s_2 & \bowtie & s_1{}^{\Box}\backslash_{\Box} s_2 \\
s_1{}^{\alpha}\backslash^{\Box}_{a}\backslash^{a}_{\Box}\backslash_{\beta} s_2 & \bowtie & s_1{}^{\alpha}\backslash^{a}_{a}\backslash_{\beta} s_2
\end{array}
$$

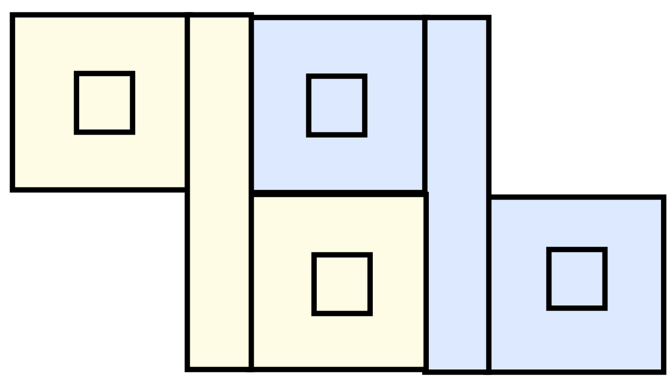
Examples: merge





Examples: merge





Merge

(All the ops we show are strict)

$$^{\alpha}\backslash^{\alpha_1}_{\beta}\backslash^{\alpha_2}_{\beta_1}\cdots\backslash^{\beta_{n-1}}_{\alpha_{n-1}}\backslash_{\alpha_n} \bullet {}^{\alpha'}\backslash^{\alpha'_1}_{\beta'}\backslash^{\alpha'_2}_{\beta'_1}\cdots\backslash^{\beta_{n-1}}_{\alpha'_{n-1}}\backslash_{\alpha'_n}$$

$$^{\alpha}\backslash_{\beta} \bullet {}^{\alpha'}\backslash_{\beta'} \triangleq \begin{cases} ^{(\alpha\bullet\alpha')}\backslash_{(\beta\bullet\beta')} & \text{if } \alpha\bullet\alpha', \beta\bullet\beta' \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

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

The definition extends to chains element-wise (the result is undefined if the outcome is not valid)

Restriction

matched action

$$(\nu a)({}^{\alpha_1}\backslash_{\beta_1}\ {}^{\alpha_2}\backslash_{\beta_2}\ \dots\ {}^{\alpha_n}\backslash_{\beta_n}\ )$$

1. $a \neq \alpha_1, \beta_n$, and
2. for any $i \in [1, n-1]$, either $\beta_i = \alpha_{i+1} = a$ or $\beta_i, \alpha_{i+1} \neq a$.

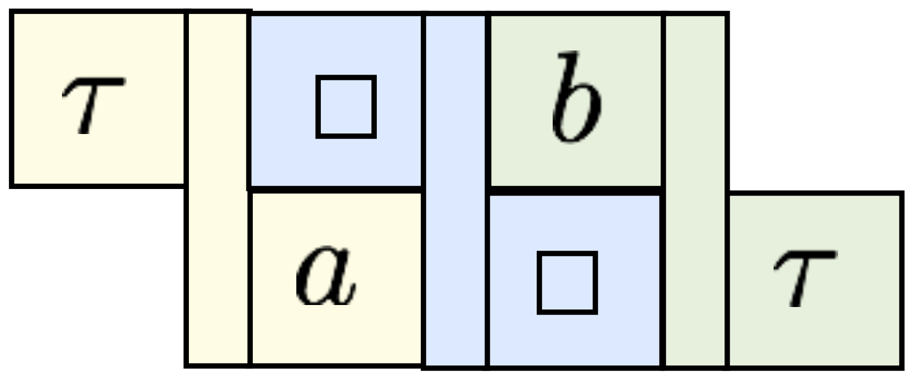
restriction

$$(\nu a)({}^{\alpha_1}\backslash_{\beta_1}\ {}^{\alpha_2}\backslash_{\beta_2}\ \dots\ {}^{\alpha_n}\backslash_{\beta_n}\ ) \triangleq \quad {}^{((\nu\, a)\alpha)}\backslash_{((\nu\, a)\beta)}$$

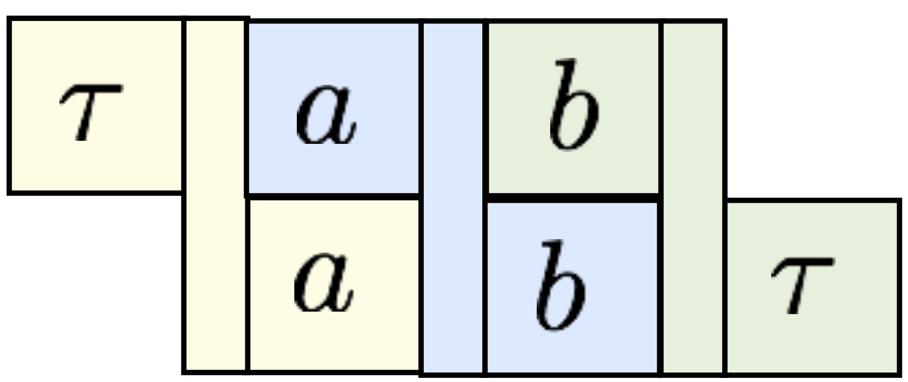
$$(\nu\, a)\alpha \quad \triangleq \quad \begin{cases} \tau & \text{if } \alpha = a \\ \alpha & \text{otherwise} \end{cases}$$

Examples: restriction

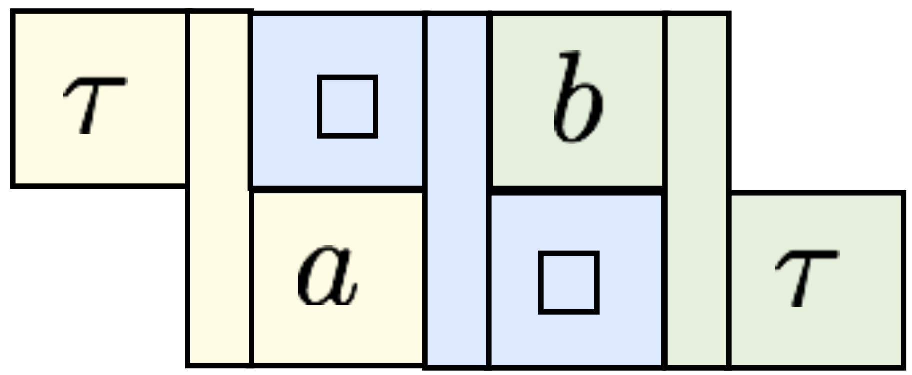
(νa)

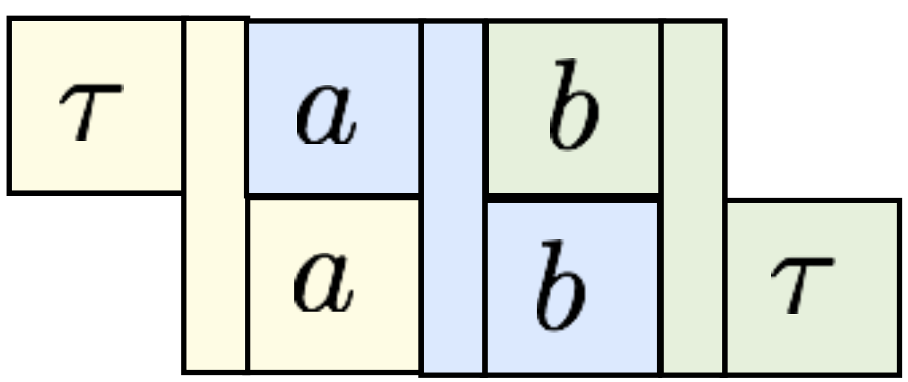


(νa)

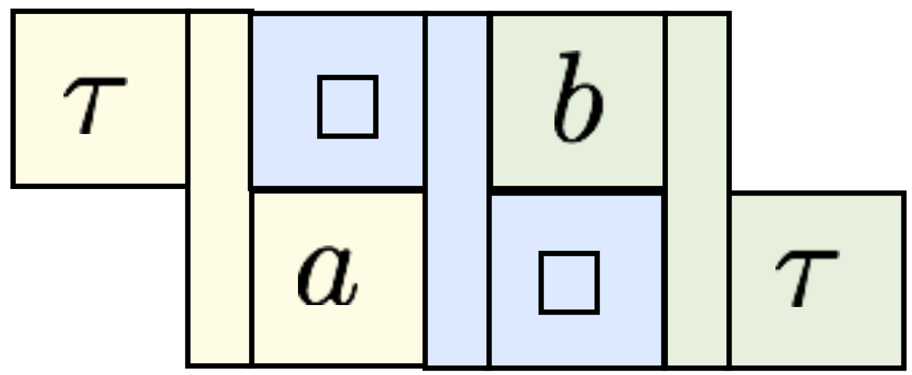


Examples: restriction

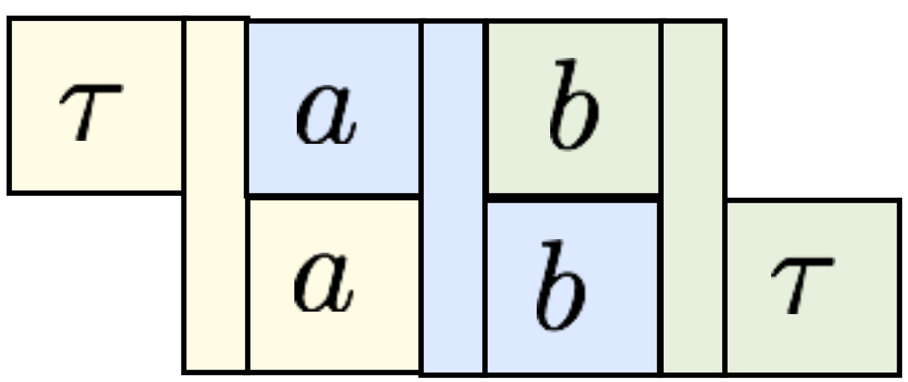
(νa)  $= \perp$

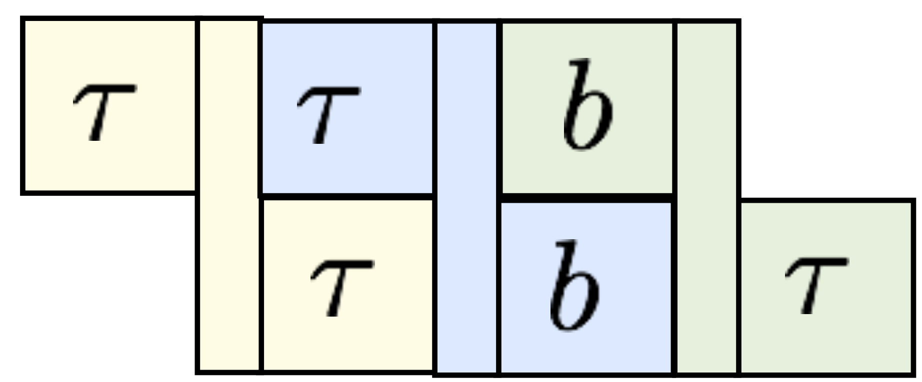
(νa) 

Examples: restriction

(νa) 

$= \perp$

(νa) 

$=$ 

(Relevant) SOS rules

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

equivalence relation

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

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

(Relevant) SOS rules

$$\frac{s \bowtie \ell}{\ell.P \xrightarrow{s} P} \text{(Act)} \qquad \frac{P \xrightarrow{s} P'}{(\nu\, a)P \xrightarrow{(\nu\, a)s} (\nu\, a)P'} \text{(Res)}$$

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

(look as ordinary CCS rules)

Example 1

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

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

Bisimulation

The process algebra of linked interactions
is a straightforward extension of CCS.
It includes CCS as a sub-calculus.

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

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

Bisimulation

The process algebra of linked interactions
is a straightforward extension of CCS.
It includes CCS as a sub-calculus.

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

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

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

Bisimulation

The process algebra of linked interactions
is a straightforward extension of CCS.
It includes CCS as a sub-calculus.

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

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

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

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

White tie

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

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

To be used in the network bisimulation
(the bisimulation of the link-calculus)

Bisimulation, definition

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

Bisimulation, definition

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

$^{\tau} \backslash_{a} \cdot^{b} \backslash_{\tau} +^{b} \backslash_{\tau} \cdot^{\tau} \backslash_{a}$

Bisimulation, definition

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

$^{\tau}\backslash_{a} \cdot^{b} \backslash_{\tau} +^{b} \backslash_{\tau} \cdot^{\tau}\backslash_{a}$

$^{\tau}\backslash_{a} |^{b}\backslash_{\tau}$

Bisimulation, definition

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

${}^{\tau}\backslash_{a} \cdot^{b} \backslash_{\tau} +^{b} \backslash_{\tau} \cdot^{\tau}\backslash_{a}$

${}^{\tau}\backslash_{a} \,|^{b}\backslash_{\tau}$

Bisimulation, definition

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

$${}^{\tau}\backslash_{a} \cdot {}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau} \cdot {}^{\tau}\backslash_{a} \qquad\qquad {}^{\tau}\backslash_{a} \,|\, {}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a} \big\downarrow$$

Bisimulation, definition

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

$^{\tau}\backslash_a \cdot^{b}\backslash_{\tau} +^{b}\backslash_{\tau} \cdot^{\tau}\backslash_a$

$^{\tau}\backslash_a \,|^{b}\backslash_{\tau}$

$^{\tau}\backslash_a$

Bisimulation, definition

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

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

$${}^{\tau}\backslash_{a} \big\downarrow$$

$${}^{b}\backslash_{\tau} \big\downarrow$$

Bisimulation, definition

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

$^{\tau}\backslash_{a} \cdot {}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau} \cdot {}^{\tau}\backslash_{a}$

$^{\tau}\backslash_{a} \,|\, {}^{b}\backslash_{\tau}$

$\big\downarrow\, {}^{\tau}\backslash_{a}$

$\big\downarrow\, {}^{b}\backslash_{\tau}$

Bisimulation, definition

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

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

$$\downarrow {}^{\tau}\backslash_{a} \qquad\qquad\qquad\qquad \downarrow {}^{\tau}\backslash_{a}$$

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

Bisimulation, definition

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

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

$${}^{\tau}\backslash_{a}$$

$${}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a} \mid {}^{b}\backslash_{\tau}$$

$${}^{\tau}\backslash_{a}$$

Bisimulation, definition

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

$$
\begin{array}{lcl}
{}^{\tau}\backslash_{a} . {}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau} . {}^{\tau}\backslash_{a} & \qquad\qquad & {}^{\tau}\backslash_{a} \,|\, {}^{b}\backslash_{\tau} \\
{}^{\tau}\backslash_{a} \big\downarrow & & {}^{\tau}\backslash_{a} \big\downarrow \\
{}^{b}\backslash_{\tau} \big\downarrow & & {}^{b}\backslash_{\tau} \big\downarrow
\end{array}
$$

Bisimulation, definition

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

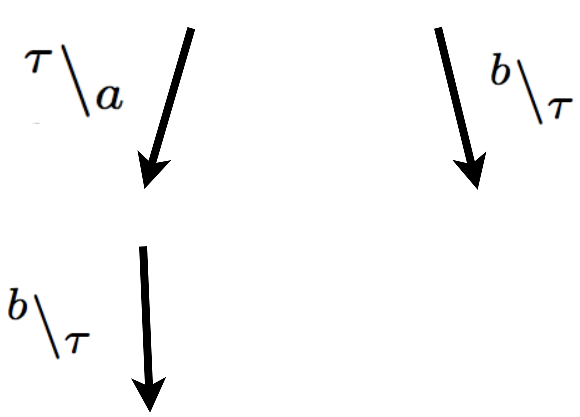
$$
\begin{array}{ccc}
{}^{\tau}\backslash_{a} \cdot {}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau} \cdot {}^{\tau}\backslash_{a} & \qquad\qquad & {}^{\tau}\backslash_{a} \,|\, {}^{b}\backslash_{\tau} \\
{}^{\tau}\backslash_{a} \downarrow \qquad \downarrow & & {}^{\tau}\backslash_{a} \downarrow \\
{}^{b}\backslash_{\tau} \downarrow \qquad\quad & & {}^{b}\backslash_{\tau} \downarrow
\end{array}
$$

Bisimulation, definition

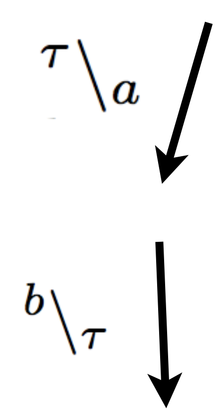
A network bisimulation **R** is a binary relation over CNA processes such that, if P **R** Q then

- 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 $P'\ \mathbf{R}\ Q'$.

${}^{\tau}\backslash_{a} \cdot {}^{b}\backslash_{\tau} + {}^{b}\backslash_{\tau} \cdot {}^{\tau}\backslash_{a}$



${}^{\tau}\backslash_{a}\ |\ {}^{b}\backslash_{\tau}$

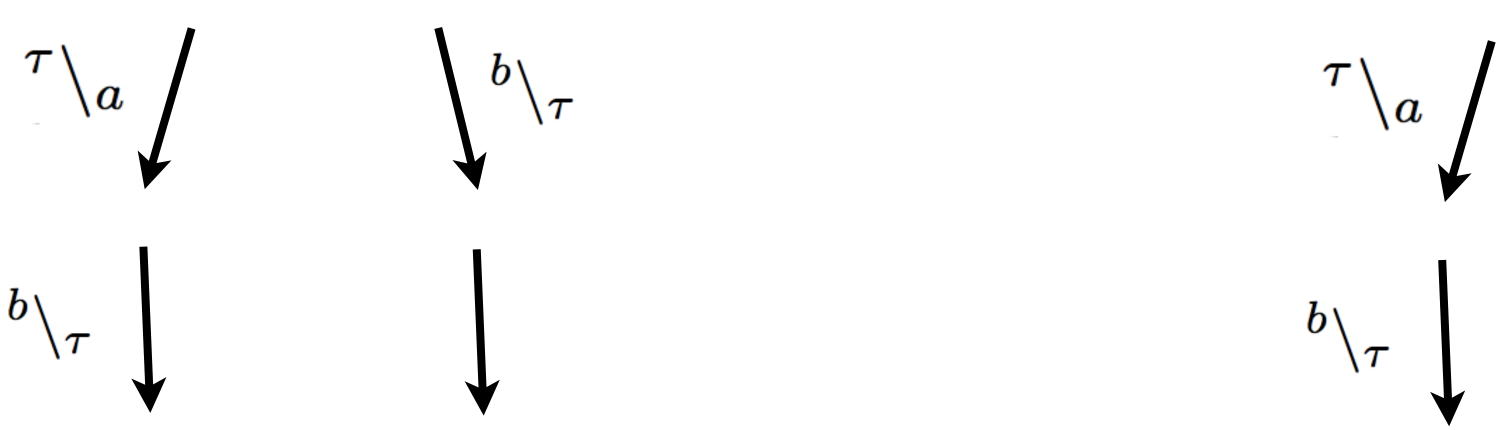


Bisimulation, definition

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

${}^{\tau}\backslash_a . {}^{b}\backslash_\tau + {}^{b}\backslash_\tau . {}^{\tau}\backslash_a$ ${}^{\tau}\backslash_a \,|\, {}^{b}\backslash_\tau$

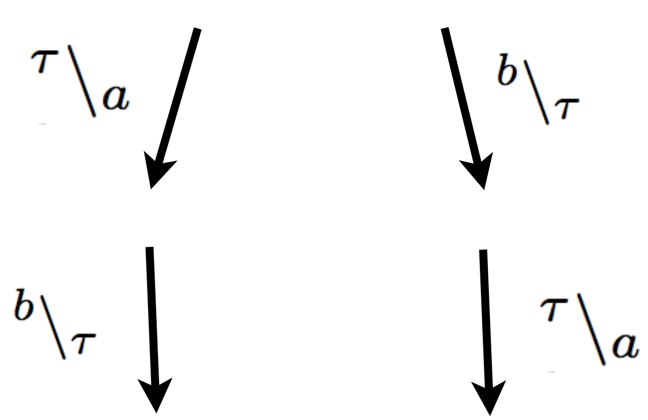


Bisimulation, definition

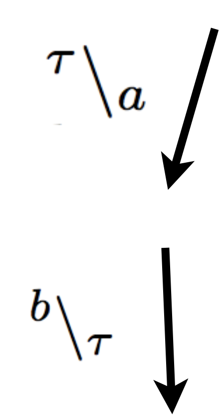
A network bisimulation **R** is a binary relation over CNA processes such that, if P **R** Q then

- 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 $P'\ \mathbf{R}\ Q'$.

$^{\tau}\backslash_a .^{b}\backslash_\tau +^{b}\backslash_\tau .^{\tau}\backslash_a$



$^{\tau}\backslash_a |^{b}\backslash_\tau$

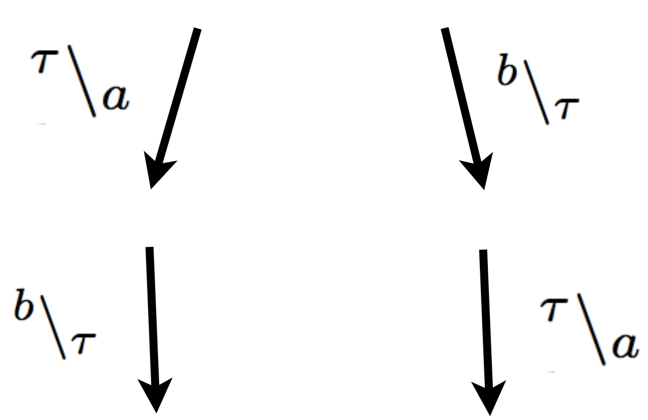


Bisimulation, definition

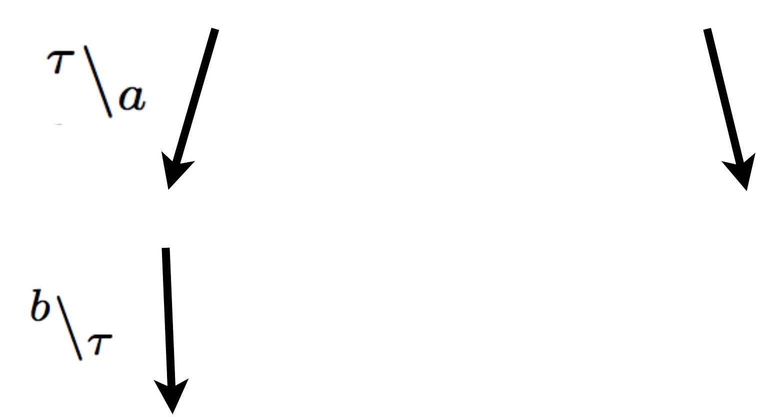
A network bisimulation **R** is a binary relation over CNA processes such that, if P **R** Q then

- 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 $P'\ \mathbf{R}\ Q'$.

$^{\tau}\backslash_a \,.^{b}\backslash_\tau +^{b}\backslash_\tau \,.^{\tau}\backslash_a$



$^{\tau}\backslash_a \,|^{b}\backslash_\tau$

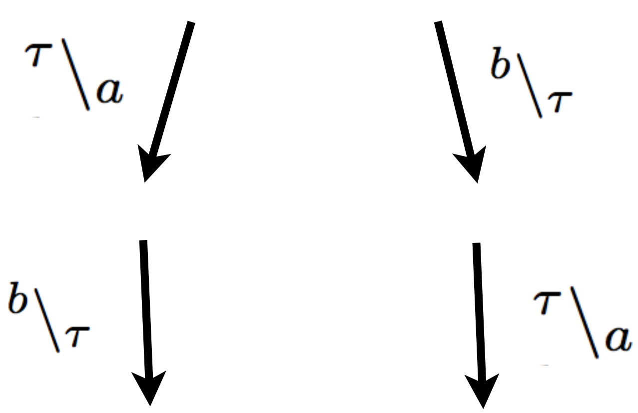


Bisimulation, definition

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

$^{\tau}\backslash_{a} \cdot^{b}\backslash_{\tau} +^{b}\backslash_{\tau} \cdot^{\tau}\backslash_{a}$

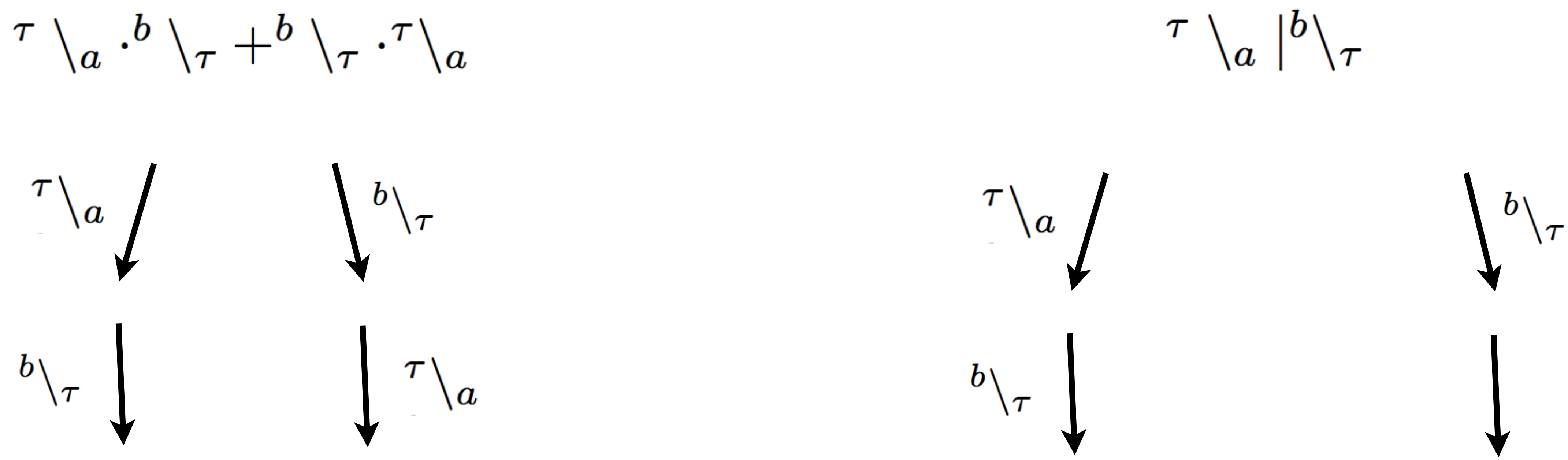


$^{\tau}\backslash_{a} |^{b}\backslash_{\tau}$

Bisimulation, definition

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



Bisimulation, definition

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

$$\begin{tikzcd}
& {}^{\tau}\backslash_a \cdot {}^{b}\backslash_\tau + {}^{b}\backslash_\tau \cdot {}^{\tau}\backslash_a \arrow[dl, "{}^{\tau}\backslash_a"'] \arrow[dr, "{}^{b}\backslash_\tau"] & \\
{} \arrow[d, "{}^{b}\backslash_\tau"'] & & {} \arrow[d, "{}^{\tau}\backslash_a"] \\
{} & & {}
\end{tikzcd}
\qquad
\begin{tikzcd}
& {}^{\tau}\backslash_a \mid {}^{b}\backslash_\tau \arrow[dl, "{}^{\tau}\backslash_a"'] \arrow[dr, "{}^{b}\backslash_\tau"] & \\
{} \arrow[d, "{}^{b}\backslash_\tau"'] & & {} \arrow[d, "{}^{\tau}\backslash_a"] \\
{} & & {}
\end{tikzcd}$$

Bisimulation, definition

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

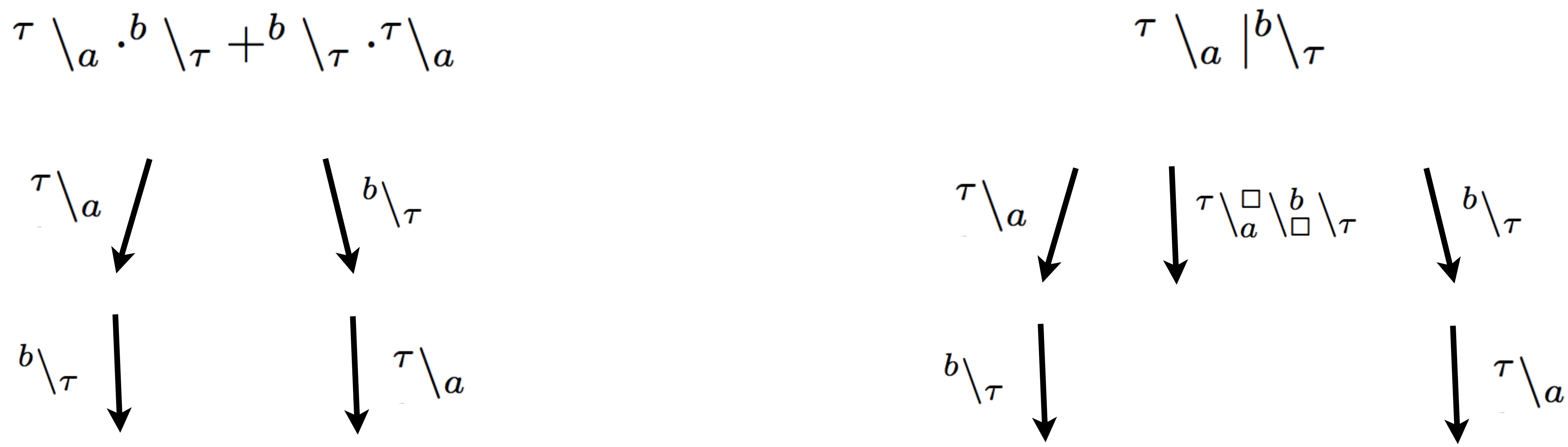
${}^{\tau}\backslash_a \cdot {}^{b}\backslash_\tau + {}^{b}\backslash_\tau \cdot {}^{\tau}\backslash_a$

${}^{\tau}\backslash_a \mid {}^{b}\backslash_\tau$

Bisimulation, definition

A network bisimulation **R** is a binary relation over CNA processes such that, if P **R** Q then

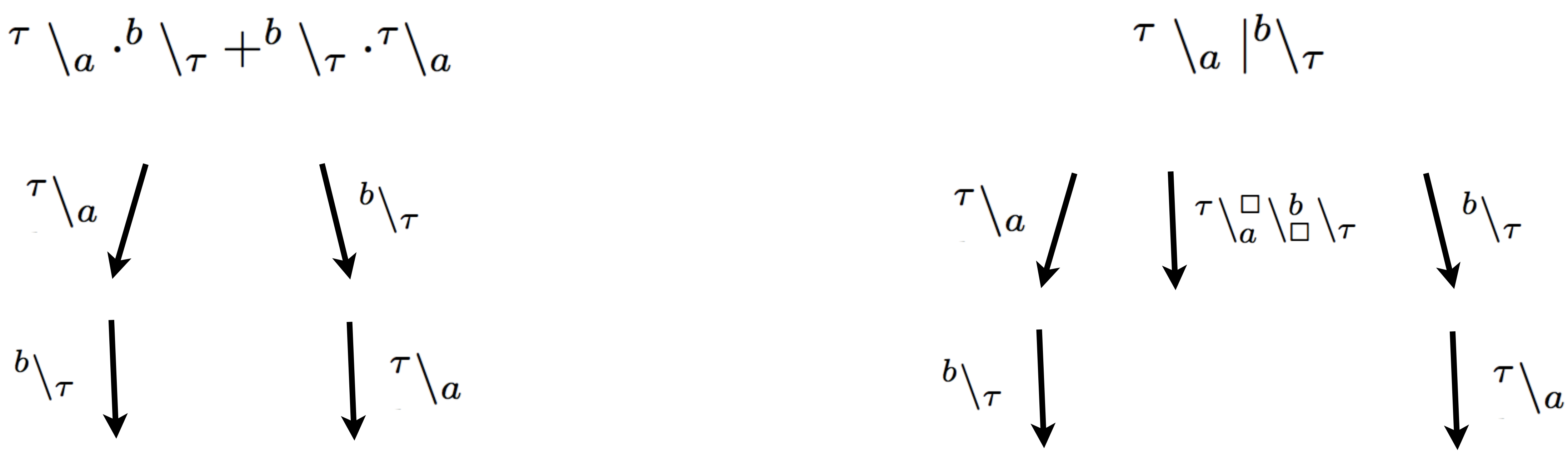
- if $P \xrightarrow{s} P'$, then $\exists\ s', Q'$ such that $s' \bowtie s$, $Q \xrightarrow{s'} Q'$, and $P'\ \mathbf{R}\ Q'$;
- if $Q \xrightarrow{s} Q'$, then $\exists\ s', P'$ such that $s' \bowtie s$, $P \xrightarrow{s'} P'$, and $P'\ \mathbf{R}\ Q'$.



Bisimulation, definition

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



it is a congruence

Some references

- Chiara Bodei, Linda Brodo, Roberto Bruni: Open Multiparty Interaction. Workshop on Algebraic Development Techniques 2012: 1-23.
- Chiara Bodei, Linda Brodo, Roberto Bruni, Davide Chiarugi: A Flat Process Calculus for Nested Membrane Interactions. Sci. Ann. Comp. Sci. 24(1): 91-136 (2014).
- Chiara Bodei, Linda Brodo, Roberto Bruni: A Formal Approach to Open Multiparty Interactions. Submitted to Theoretical Computer Science.

Roadmap

- A brief introduction to the link-calculus
- Symbolic link chains
- Definition of the symbolic semantics
- Definition of the symbolic bisimulation
- Conclusion and future work

Example 2

$$P =^{a} \backslash_{b}.P' \quad Q =^{b} \backslash_{a}.Q'$$

Example 2

$$P =^{a} \backslash_{b}.P' \quad Q =^{b} \backslash_{a}.Q'$$

$$P|Q \xrightarrow{{}^{a}\backslash^{b}_{b}\backslash_{a}} P'|Q'$$

Example 2

$$P =^{a} \backslash_{b}.P' \quad Q =^{b} \backslash_{a}.Q'$$

$$P|Q \xrightarrow{a \backslash^{b}_{b} \backslash_{a}} P'|Q'$$

$$P|Q \xrightarrow{b \backslash^{a}_{a} \backslash_{b}} P'|Q'$$

Example 2

$$P =^{a} \backslash_{b}.P' \quad Q =^{b} \backslash_{a}.Q'$$

$$P|Q \xrightarrow{{}^{a}\backslash^{b}_{b}\backslash_{a}} P'|Q'$$

$$P|Q \xrightarrow{{}^{b}\backslash^{a}_{a}\backslash_{b}} P'|Q'$$

$$P|Q \xrightarrow{{}^{a}\backslash^{\Box}_{b}\backslash_{\Box}\cdots{}^{\Box}\backslash^{b}_{\Box}\backslash_{a}} P'|Q'$$

Example 2

$$P =^{a} \backslash_{b}.P' \quad Q =^{b} \backslash_{a}.Q'$$

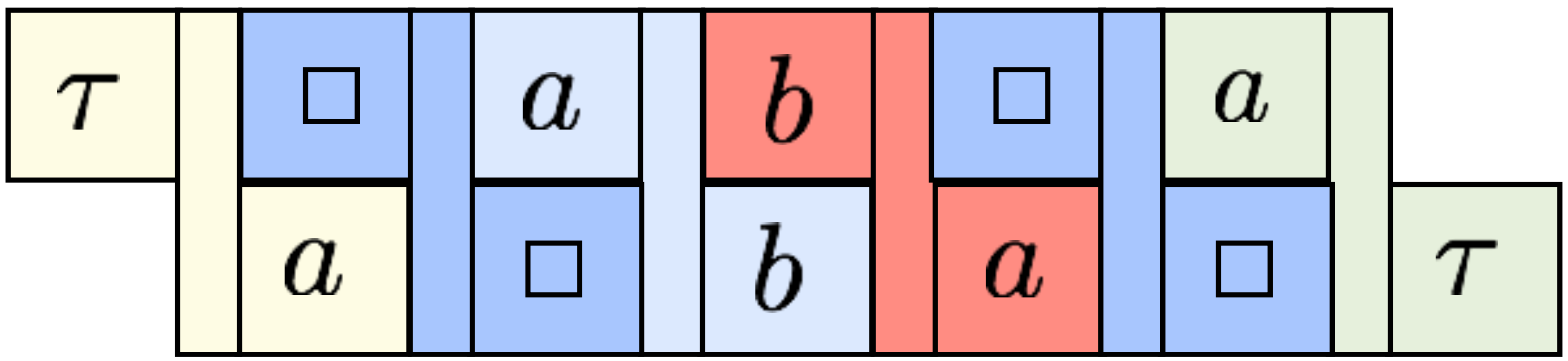
$$P|Q \xrightarrow{{}^{a}\backslash^{b}_{b}\backslash_{a}} P'|Q'$$

$$P|Q \xrightarrow{{}^{b}\backslash^{a}_{a}\backslash_{b}} P'|Q'$$

$$P|Q \xrightarrow{{}^{a}\backslash^{\square}_{b}\backslash_{\square}\cdots{}^{\square}\backslash^{b}_{\square}\backslash_{a}} P'|Q'$$

$$P|Q \xrightarrow{{}^{b}\backslash^{\square}_{a}\backslash_{\square}\cdots{}^{\square}\backslash^{a}_{\square}\backslash_{b}} P'|Q'$$

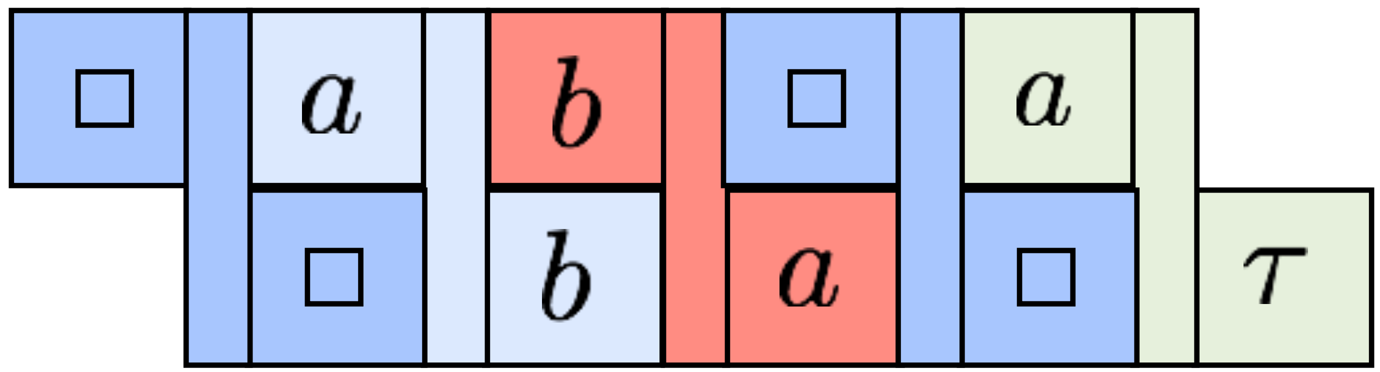
Symbolic configurations



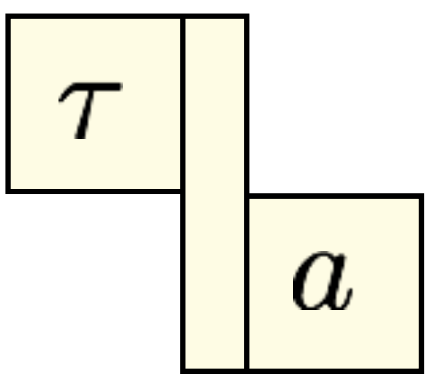
link chain

symbolic configuration

Symbolic configurations

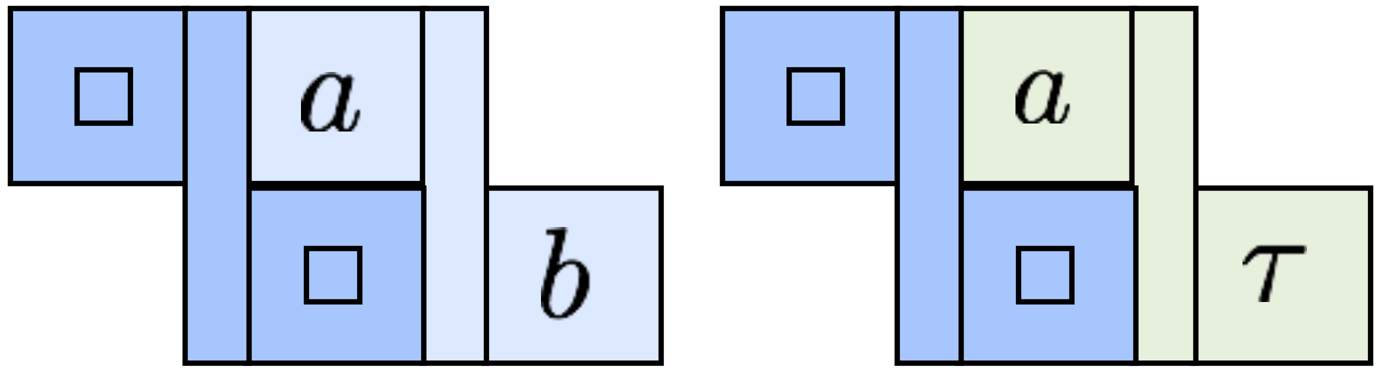


link chain

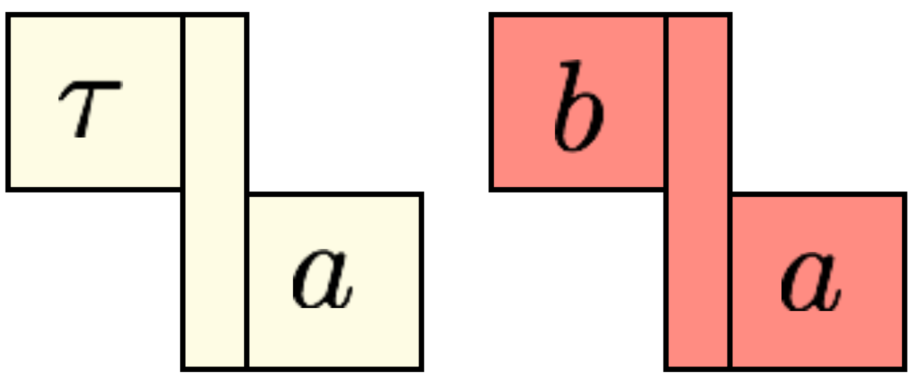


symbolic configuration

Symbolic configurations

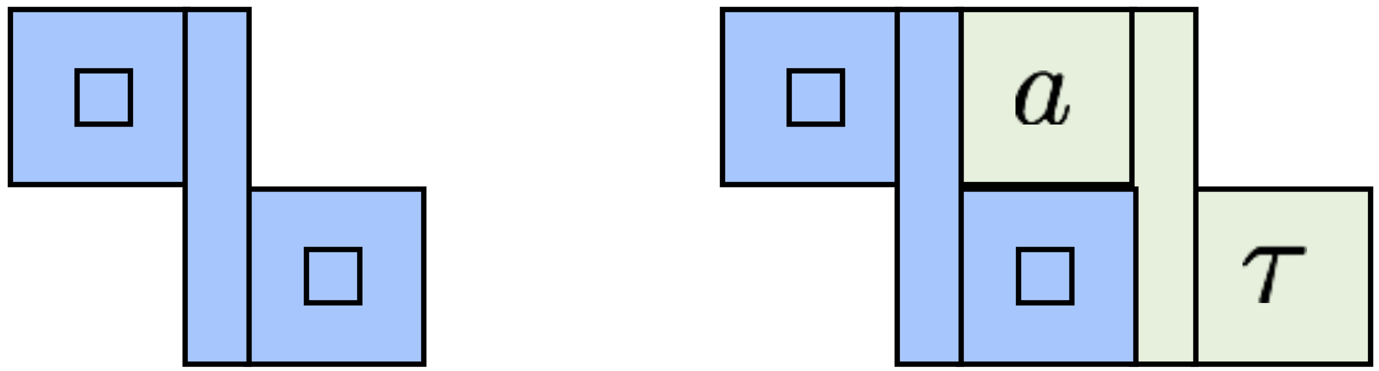


link chain

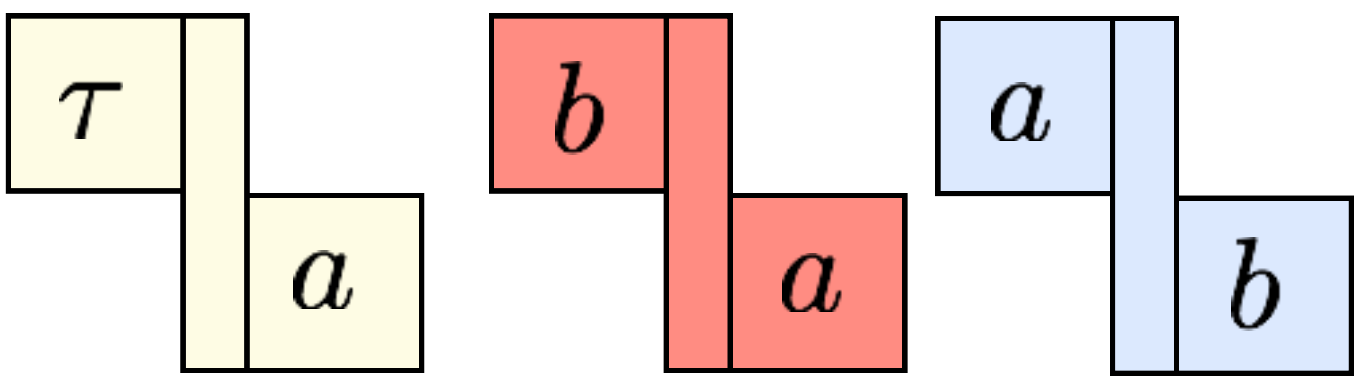


symbolic configuration

Symbolic configurations

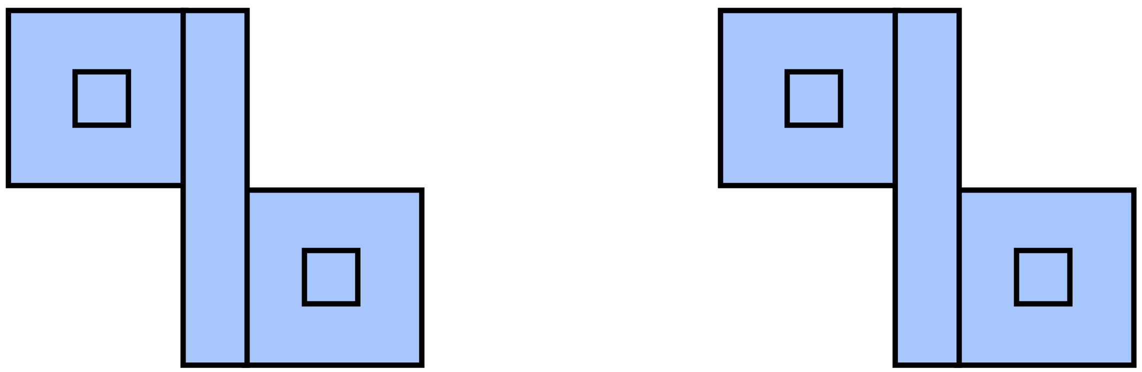


link chain

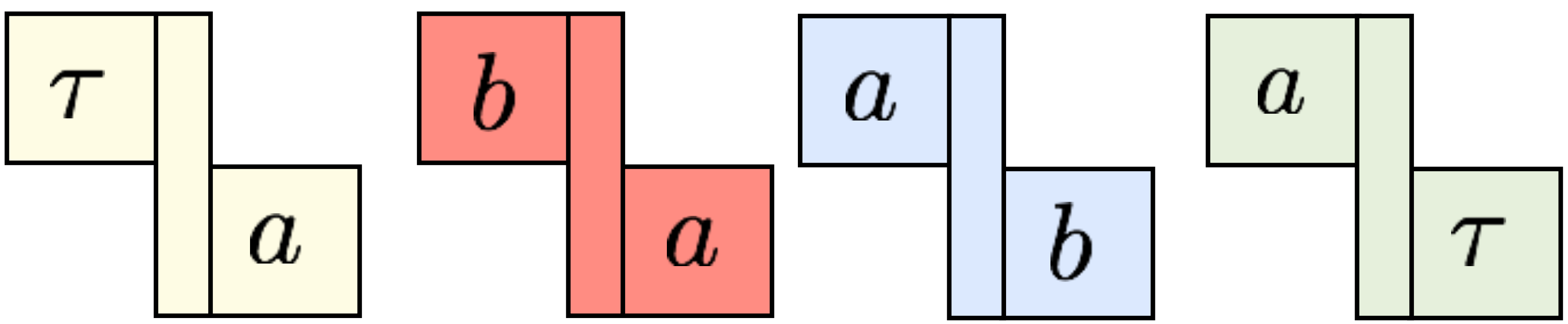


symbolic configuration

Symbolic configurations

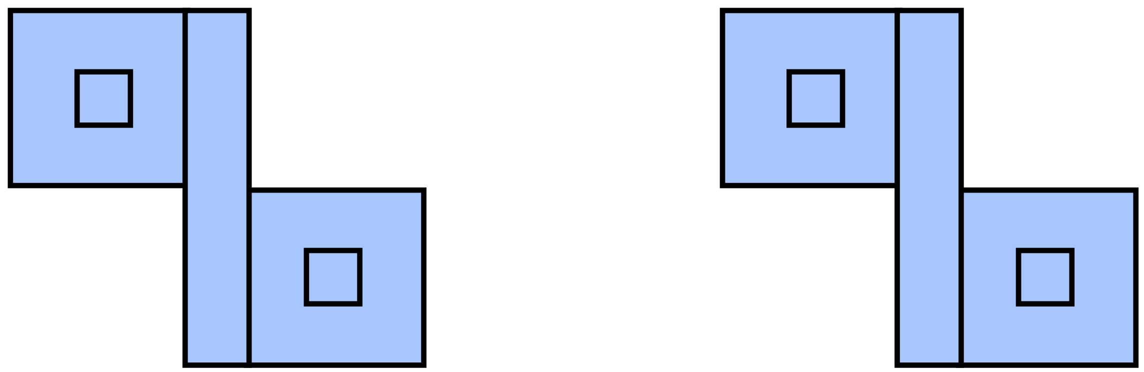


link chain

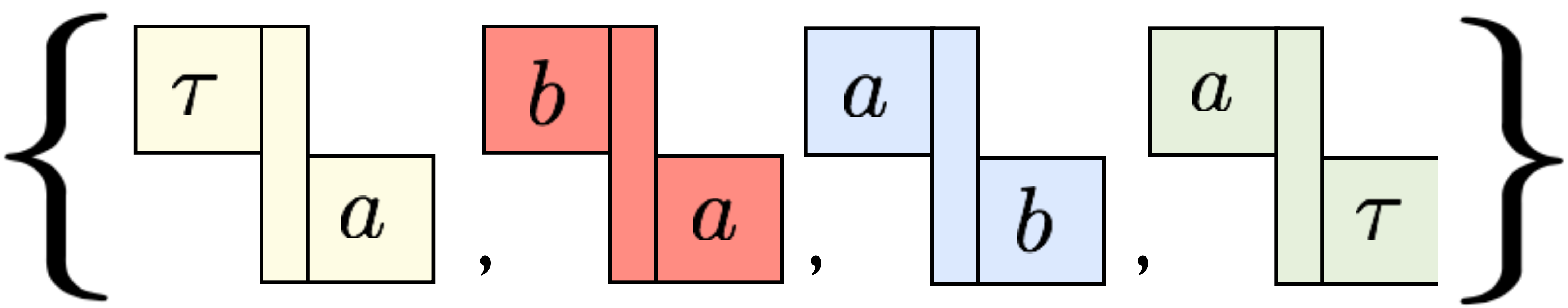


symbolic configuration

Symbolic configurations



link chain



symbolic configuration

Symbolic Configurations

Let L be a multiset of solid links. We define the (symbolic) configuration <L> as the set

$$\langle L\rangle = \{s \in VC \mid \textit{ there exists } s_i \bowtie l_i \textit{ for all } l_i \in L \textit{ s.t. } s = s_1 \bullet s_2 \bullet \cdots \bullet s_n\}$$

We say that $\langle L\rangle$ is a valid configuration if the set above is not empty.

$$L = \{{}^{a}\backslash_{b}, {}^{\tau}\backslash_{a}, {}^{\tau}\backslash_{b}\}$$

not valid configuration

Symbolic Configurations

Let L be a multiset of solid links. We define the (symbolic) configuration <L> as the set

$$\langle L \rangle = \{ s \in VC \mid \textit{ there exists } s_i \bowtie l_i \textit{ for all } l_i \in L \textit{ s.t. } s = s_1 \bullet s_2 \bullet \cdots \bullet s_n \}$$

We say that $\langle L \rangle$ is a valid configuration if the set above is not empty.

$$L = \{ {}^{a}\backslash_{b}, {}^{\tau}\backslash_{a}, {}^{\tau}\backslash_{b} \}$$

not valid configuration

Proposition 1 (Valid Configurations). *Let L be a non-empty multiset of solid links. Then, $\langle L \rangle$ is valid iff τ appears at most once in L as input and at most once as output.*

Symbolic Configurations

Let L be a multiset of solid links. We define the (symbolic) configuration <L> as the set

$$\langle L \rangle = \{ s \in VC \mid \textit{ there exists } s_i \bowtie l_i \textit{ for all } l_i \in L \textit{ s.t. } s = s_1 \bullet s_2 \bullet \cdots \bullet s_n \}$$

We say that $\langle L \rangle$ is a valid configuration if the set above is not empty.

$$L = \{ {}^{a}\backslash_{b}, {}^{\tau}\backslash_{a}, {}^{\tau}\backslash_{b} \}$$

not valid configuration

Proposition 1 (Valid Configurations). *Let L be a non-empty multiset of solid links. Then, $\langle L \rangle$ is valid iff τ appears at most once in L as input and at most once as output.*

Restriction acting on configurations

Let γ be a configuration and $a \in C$. We define the configuration

$$(\nu a)\gamma = \{s \in VC \mid \text{ there exists } s' \in \gamma \text{ and } s = (\nu a)s'\}$$

We say that $(\nu a)\gamma$ is valid if the set above is not empty.

$L = \{{}^{a}\backslash_{a}\}$

$L = \{{}^{\tau}\backslash_{a}, {}^{a}\backslash_{\tau}, {}^{a}\backslash_{b}\}$

not valid configurations $(\nu a)\langle L\rangle$

$L = \{{}^{\tau}\backslash_{a}, {}^{b}\backslash_{\tau}, {}^{a}\backslash_{c}\}$ this is valid $(\nu a)\langle L\rangle$

because of $\tau\backslash_{a}^{a}\backslash_{c}^{\Box}\backslash_{\Box}^{b}\backslash_{\tau}$

Properties of restricted valid configurations

Let $\gamma = (\nu \boldsymbol{x})\langle L \rangle$ *be a valid configuration and*

$a \in fn(\gamma)$. $(\nu a)\gamma$ *is valid iff the three conditions below hold:*

1. **Matched**: *a occurs the same number of times as input and as output in* γ.
2. **Extremes**: *there exist two links* ${}^{\alpha}\backslash_{\beta}, {}^{\alpha'}\backslash_{\beta'}$ *in* γ *where* $\alpha, \beta' \neq a$.
3. **Synchronizations**: *if both* ${}^{\tau}\backslash_{a}$ *and* ${}^{a}\backslash_{\tau}$ *occur in* L, *then either* $names(L) = \{a, \tau\}$ *or there exist two links* ${}^{a}\backslash_{\beta}, {}^{\beta'}\backslash_{a}$ *in* L *s.t.* $\beta, \beta' \notin \{a, \tau\}$.

Merging valid configurations

Let $(\nu a_1, ..., a_n)\langle L\rangle$ *and* $(\nu b_1, ...b_m)\langle L'\rangle$ *be two valid configurations. By alpha conversion, we assume that the names* $a_1, ..., a_n$ *(resp.* $b_1, ..., b_m$*) do not occur in* L' *(resp.* L*). We define*

$$(\nu a_1, ..., a_n)\langle L\rangle \bullet (\nu b_1, ...b_m)\langle L'\rangle = (\nu a_1, ..., a_n, b_1, ..., b_m)\langle L \uplus L'\rangle$$

where $\uplus$ *denotes multiset union.*

Roadmap

- A brief introduction to the link-calculus
- Symbolic link chains
- Definition of the symbolic semantics
- Definition of the symbolic bisimulation
- Conclusion and future work

Symbolic semantics rules

$$\frac{}{\ell.P \xRightarrow{\langle\{\ell\}\rangle} P}\ Act_s$$

$$\frac{P \xRightarrow{\gamma} P'}{(\nu a)P \xRightarrow{(\nu a)\gamma} (\nu a)P'}\ Res_s \qquad \frac{P \xRightarrow{\gamma} P' \quad Q \xRightarrow{\gamma'} Q'}{P \mid Q \xRightarrow{\gamma \bullet \gamma'} P' \mid Q'}\ Com_s$$

$$\frac{P \xRightarrow{\gamma} P'}{P + Q \xRightarrow{\gamma} P'}\ Lsum_s \qquad \frac{P \xRightarrow{\gamma} P'}{P \mid Q \xRightarrow{\gamma} P' \mid Q}\ Lpar_s \qquad \frac{P \xRightarrow{\gamma} P' \quad A \triangleq P}{A \xRightarrow{\gamma} P'}\ Ide_s$$

Symbolic semantics rules

$$\dfrac{}{\ell.P \xRightarrow{\langle\{\ell\}\rangle} P}\ Act_s$$

$$\dfrac{P \xRightarrow{\gamma} P'}{(\nu a)P \xRightarrow{(\nu a)\gamma} (\nu a)P'}\ Res_s \qquad \dfrac{P \xRightarrow{\gamma} P' \quad Q \xRightarrow{\gamma'} Q'}{P \,|\, Q \xRightarrow{\gamma \bullet \gamma'} P' \,|\, Q'}\ Com_s$$

$$\dfrac{P \xRightarrow{\gamma} P'}{P + Q \xRightarrow{\gamma} P'}\ Lsum_s \qquad \dfrac{P \xRightarrow{\gamma} P'}{P \,|\, Q \xRightarrow{\gamma} P' \,|\, Q}\ Lpar_s \qquad \dfrac{P \xRightarrow{\gamma} P' \quad A \triangleq P}{A \xRightarrow{\gamma} P'}\ Ide_s$$

Symbolic semantics rules

$$\boxed{\dfrac{}{\ell.P \xRightarrow{\langle\{\ell\}\rangle} P}\ Act_s}$$

$$\dfrac{P \xRightarrow{\gamma} P'}{(\nu a)P \xRightarrow{(\nu a)\gamma} (\nu a)P'}\ Res_s \qquad \boxed{\dfrac{P \xRightarrow{\gamma} P' \quad Q \xRightarrow{\gamma'} Q'}{P \mid Q \xRightarrow{\gamma \bullet \gamma'} P' \mid Q'}\ Com_s}$$

$$\dfrac{P \xRightarrow{\gamma} P'}{P + Q \xRightarrow{\gamma} P'}\ Lsum_s \qquad \dfrac{P \xRightarrow{\gamma} P'}{P \mid Q \xRightarrow{\gamma} P' \mid Q}\ Lpar_s \qquad \dfrac{P \xRightarrow{\gamma} P' \quad A \triangleq P}{A \xRightarrow{\gamma} P'}\ Ide_s$$

Soundness and Completeness

Let P be a process and assume that $P \overset{\gamma}{\Longrightarrow} P'$. *Then,*

for all $s \in \gamma$, $P \xrightarrow{\bar{s}} P'$.

Let P be a process and assume that $P \xrightarrow{s} P'$. *Then, there*

exists γ *s.t.* $P \overset{\gamma}{\Longrightarrow} P'$ *and* $s \in \gamma$.

where γ is a symbolic configuration, possibly restricted.

introducing the extraction operator *ext*(*s*)

$$s = {}^{\tau}\backslash{}^{a}_{a}\backslash{}^{c}_{c}\backslash{}^{b}_{b}\backslash{}_{\tau}$$

introducing the extraction operator *ext*(*s*)

$$s = {}^{\tau}\setminus^{a}_{a}\setminus^{c}_{c}\setminus^{b}_{b}\setminus_{\tau}$$

$$s' = (\nu\ a)(\nu\ b)\,{}^{\tau}\setminus^{a}_{a}\setminus^{c}_{c}\setminus^{b}_{b}\setminus_{\tau} = {}^{\tau}\setminus^{\tau}_{\tau}\setminus^{c}_{c}\setminus^{\tau}_{\tau}\setminus_{\tau}$$

introducing the extraction operator *ext*(*s*)

$$s = {}^{\tau}\backslash^{a}_{a}\backslash^{c}_{c}\backslash^{b}_{b}\backslash_{\tau}$$

$$s' = (\nu\ a)(\nu\ b)\,{}^{\tau}\backslash^{a}_{a}\backslash^{c}_{c}\backslash^{b}_{b}\backslash_{\tau} = {}^{\tau}\backslash^{\tau}_{\tau}\backslash^{c}_{c}\backslash^{\tau}_{\tau}\backslash_{\tau}$$

$$ext(s') = (\nu\ x)\langle {}^{\tau}\backslash_{x}, {}^{x}\backslash_{c}, {}^{c}\backslash_{x}, {}^{x}\backslash_{\tau}\rangle$$

Completeness, refined

Let P *be a process and assume that* $P \xrightarrow{s} P'$. *Then, there exists* $\gamma \subseteq \mathtt{ext}(s)$ *s.t.* $P \xRightarrow{\gamma} P'$.

ext(...) is a super set ...

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \stackrel{\gamma}{\Longrightarrow} P'$.

ext(...) is a super set ...

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \xrightarrow{s \bullet s'} \mathbf{0}$$

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \overset{\gamma}{\Longrightarrow} P'$.

ext(...) is a super set ...

$$(\nu a)({}^{b}\backslash_{a} | {}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c} | {}^{c}\backslash_{d}) \xrightarrow{s \bullet s'} \mathbf{0}$$

$$w = s \bullet s' = {}^{d}\backslash^{\tau}_{\tau} \backslash^{\square}_{d} \backslash^{b}_{\square} \backslash^{\tau}_{\tau} \backslash_{b}$$

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \stackrel{\gamma}{\Longrightarrow} P'$.

ext(...) is a super set ...

$$(\nu a)({}^b\backslash_a|{}^a\backslash_b) \mid (\nu c)({}^d\backslash_c|{}^c\backslash_d) \xrightarrow{s \bullet s'} \mathbf{0}$$

$$w = s \bullet s' = {}^d\backslash^{\tau}_{\tau} \backslash^{\square}_{d} \backslash^{b}_{\square} \backslash^{\tau}_{\tau} \backslash_b$$

$$(\nu a)({}^b\backslash_a|{}^a\backslash_b) \mid (\nu c)({}^d\backslash_c|{}^c\backslash_d) \stackrel{\gamma \bullet \gamma'}{\Longrightarrow} \mathbf{0}$$

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \overset{\gamma}{\Longrightarrow} P'$.

ext(...) is a super set ...

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \xrightarrow{s \bullet s'} \mathbf{0}$$

$$w = s \bullet s' = {}^{d}\backslash^{\tau}_{\tau} \backslash^{\square}_{d} \backslash^{b}_{\square} \backslash^{\tau}_{\tau} \backslash_{b}$$

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \overset{\gamma \bullet \gamma'}{\Longrightarrow} \mathbf{0}$$

$$\psi = \gamma \bullet \gamma' = (\nu\, a, c)\langle {}^{b}\backslash_{a}, {}^{a}\backslash_{b}, {}^{d}\backslash_{c}, {}^{c}\backslash_{d} \rangle$$

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \xRightarrow{\gamma} P'$.

ext(...) is a super set ...

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \xrightarrow{s \bullet s'} \mathbf{0}$$

$$w = s \bullet s' = {}^{d}\backslash^{\tau}_{\tau} \backslash^{\square}_{d} \backslash^{b}_{\square} \backslash^{\tau}_{\tau} \backslash_{b}$$

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \xRightarrow{\gamma \bullet \gamma'} \mathbf{0}$$

$$\psi = \gamma \bullet \gamma' = (\nu\, a, c)\langle {}^{b}\backslash_{a}, {}^{a}\backslash_{b}, {}^{d}\backslash_{c}, {}^{c}\backslash_{d} \rangle$$

$$\mathtt{ext}(w) = (\nu x)\langle {}^{b}\backslash_{x}, {}^{x}\backslash_{b}, {}^{d}\backslash_{x}, {}^{x}\backslash_{d} \rangle$$

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there

exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \stackrel{\gamma}{\Longrightarrow} P'$.

ext(...) is a super set ...

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \xrightarrow{s \bullet s'} \mathbf{0}$$

$$w = s \bullet s' = {}^{d}\backslash^{\tau}_{\tau}\backslash^{\square}_{d}\backslash^{b}_{\square}\backslash^{\tau}_{\tau}\backslash_{b}$$

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \stackrel{\gamma \bullet \gamma'}{\Longrightarrow} \mathbf{0}$$

$$\psi = \gamma \bullet \gamma' = (\nu\, a, c)\langle {}^{b}\backslash_{a}, {}^{a}\backslash_{b}, {}^{d}\backslash_{c}, {}^{c}\backslash_{d}\rangle$$

$$\mathtt{ext}(w) = (\nu x)\langle {}^{b}\backslash_{x}, {}^{x}\backslash_{b}, {}^{d}\backslash_{x}, {}^{x}\backslash_{d}\rangle$$

$$w' = {}^{b}\backslash^{\tau}_{\tau}\backslash^{d}_{d}\backslash^{\tau}_{\tau}\backslash_{b} \in \mathtt{ext}(w)$$

Completeness, refined

Let P be a process and assume that $P \xrightarrow{s} P'$. Then, there exists $\gamma \subseteq \mathtt{ext}(s)$ s.t. $P \stackrel{\gamma}{\Longrightarrow} P'$.

ext(...) is a super set ...

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \xrightarrow{s \bullet s'} \mathbf{0}$$

$$w = s \bullet s' = {}^{d}\backslash^{\tau}_{\tau}\backslash^{\Box}_{d}\backslash^{b}_{\Box}\backslash^{\tau}_{\tau}\backslash_{b}$$

$$(\nu a)({}^{b}\backslash_{a}|{}^{a}\backslash_{b}) \mid (\nu c)({}^{d}\backslash_{c}|{}^{c}\backslash_{d}) \stackrel{\gamma \bullet \gamma'}{\Longrightarrow} \mathbf{0}$$

$$\psi = \gamma \bullet \gamma' = (\nu\, a, c)\langle {}^{b}\backslash_{a}, {}^{a}\backslash_{b}, {}^{d}\backslash_{c}, {}^{c}\backslash_{d} \rangle$$

$$\mathtt{ext}(w) = (\nu x)\langle {}^{b}\backslash_{x}, {}^{x}\backslash_{b}, {}^{d}\backslash_{x}, {}^{x}\backslash_{d} \rangle$$

$$w' = {}^{b}\backslash^{\tau}_{\tau}\backslash^{d}_{d}\backslash^{\tau}_{\tau}\backslash_{b} \in \mathtt{ext}(w) \qquad w' \notin \psi$$

Roadmap

- A brief introduction to the link-calculus
- Symbolic link chains
- Definition of the symbolic semantics
- Definition of the symbolic bisimulation
- Conclusion and future work

Bisimulation

recall

A network bisimulation $\mathbf{R}$ *is a binary relation over* `link` *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* $P' \mathbf{R} Q'$
- *and viceversa*

We let $\sim_n$ *denote the largest network bisimulation and we say that* P *is network bisimilar to* Q *if* $P \sim_n Q$.

then, we introduce

Let $\bowtie$ *be the least symmetric relation on valid configurations s.t.* $\gamma \bowtie \gamma'$

iff for all $s \in \gamma$ *there exists* $s' \in \gamma'$ *s.t.* $s' \bowtie s$.

Symbolic bisimulation

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

- *If* $P \xRightarrow{\gamma} P'$, *then, there exists* $\gamma' \bowtie \gamma$ *s.t.* $Q \xRightarrow{\gamma'} Q'$ *and* $P'\mathbf{R}Q'$.
- *If* $Q \xRightarrow{\gamma} Q'$, *then, there exists* $\gamma' \bowtie \gamma$ *s.t.* $P \xRightarrow{\gamma'} P'$ *and* $Q'\mathbf{R}P'$.

We let $\sim_s$ *be the largest symbolic network bisimulation and we say that* P *and* Q *are bisimilar if* $P \sim_s Q$.

Symbolic bisimulation 2

we introduce the concept of capability...

given $\gamma = (\nu \boldsymbol{x})\langle L \rangle$

we say that $[a{\cdot}b] \in cap(\gamma)$

$if\ {}^{a}\backslash_{b} \in L$

or it is possible to create a link chain as the following

${}^{a}\backslash^{x_1}_{x_1}\backslash_{x_2} \cdots {}^{x_{n-1}}\backslash^{x_n}_{x_n}\backslash_{b}$ where $x_1, \ldots, x_n \in \boldsymbol{x}$

Symbolic bisimulation 3

Let $s \in \gamma$*. For all solid link* ${}^{a}\backslash_{b}$, ${}^{a}\backslash_{b}$ appears in s

$$\textit{iff } [a \cdot b] \in \mathtt{cap}(\gamma)$$

Moreover, let γ, γ' *be valid configurations.*

$$\textit{Then, } \gamma \bowtie \gamma' \textit{ iff } \mathtt{cap}(\gamma) = \mathtt{cap}(\gamma')$$

Therefore, checking $\gamma \bowtie \gamma'$ can be done in polynomial time

Symbolic bisimulation, last

Let P and Q be processes. Then, $P \sim_n Q$ iff $P \sim_s Q$.

$\boxed{\sim_s \textit{ is a congruence.}}$

The tool

- We have implemented the symbolic semantics in Maude (http://maude.cs.illinois.edu)
- it is available at http://subsell.logic.at/links

Conclusions

A symbolic semantics and bisimulation for an open and multiparty interaction process calculus

An efficient procedures to check the validity of a symbolic configuration

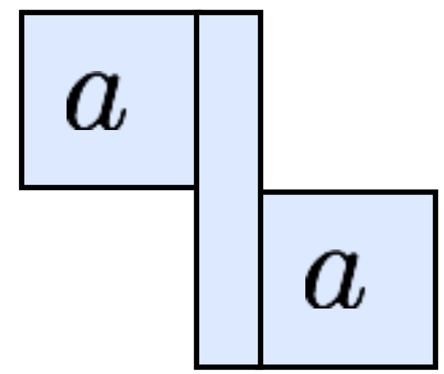
Our semantics is adequate wrt the operational semantics

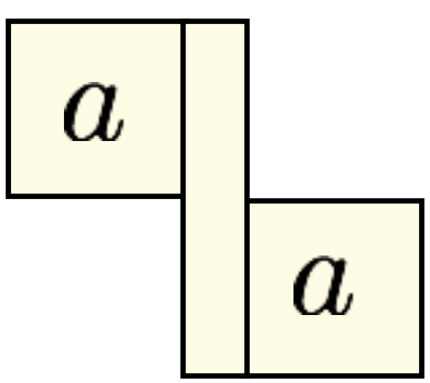
Future work

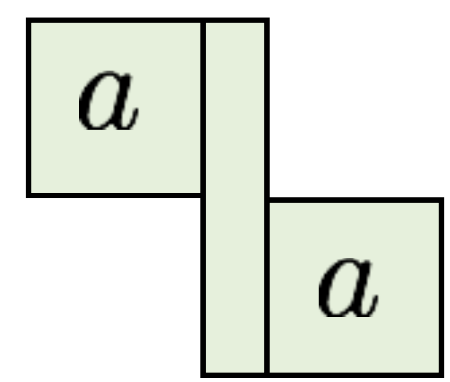
- Implementing a procedure to check (symbolic) bisimulation in the link-calculus.
- Use the extraction procedure (ext(s)), that over approximates the semantics, as basis for abstract debugging and analysis of link-calculus specifications
- Symbolic semantics for the link-calculus with data


Thanks for the attention!

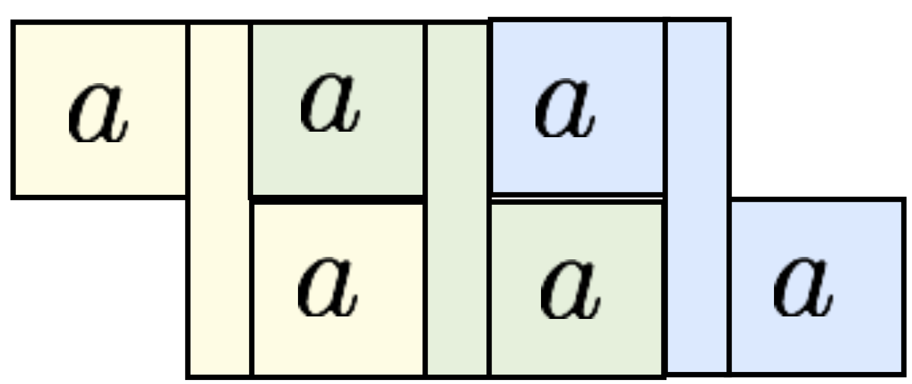
Examples: CSP







Examples: CSP



The dining philosophers

$$
\begin{aligned}
Phil_i &\triangleq {}^{\tau}\backslash_{think_i}.Phil_i \; + {}^{up_i}\backslash_{up_{(i+1)\,\mathbf{mod}\,n}}.PhilEat_i \\
PhilEat_i &\triangleq {}^{\tau}\backslash_{eat_i}.{}^{dw_i}\backslash_{dw_{(i+1)\,\mathbf{mod}\,n}}.Phil_i \\
Fork_i &\triangleq {}^{\tau}\backslash_{up_i}.{}^{\tau}\backslash_{dw_i}.Fork_i \; + \; {}^{up_i}\backslash_{\tau}.{}^{dw_i}\backslash_{\tau}.Fork_i
\end{aligned}
$$

We do not have the deadlock problem anymore,
we can focus on ... starvation.