

# Deducing Interactions in Partially Unspecified Biological Systems

P. Baldan<sup>1</sup>   A. Bracciali<sup>2</sup>   L. Brodo<sup>3</sup>   R. Bruni<sup>2</sup>

<sup>1</sup>Università di Padova

<sup>2</sup>Università di Pisa

<sup>3</sup>Università di Sassari

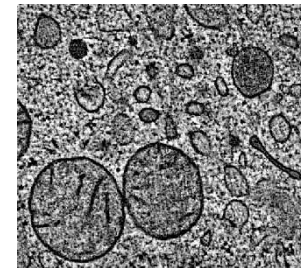
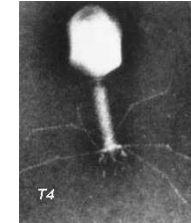
Algebraic Biology - Hagenberg, July 2-4, 2007

# Outline

- 1 Open Biological Systems
- 2 Open distributed and concurrent systems
- 3 Symbolic Transition System
- 4 Biological examples

# biological systems are *open* in different meanings

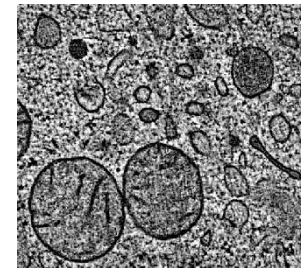
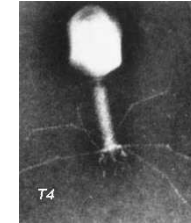
- **perturbable** - any living system expresses an external behaviour
- **incomplete** - our knowledge of biological systems is incomplete
- **biotechnology** - we may always assume to add some external components to a given complex



Partial information imposes that the approach is to predict the behaviours (of missing knowledge).

biological systems are *open* in different meanings

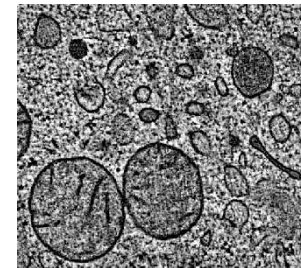
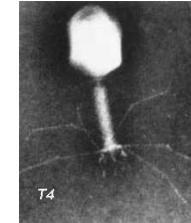
- **perturbable** - any living system expresses an external behaviour
- **incomplete** - our knowledge of biological systems is incomplete
- **biotechnology** - we may always assume to add some external components to a given complex



Partial information imposes that the approach is to predict the behaviours (of missing knowledge).

biological systems are *open* in different meanings

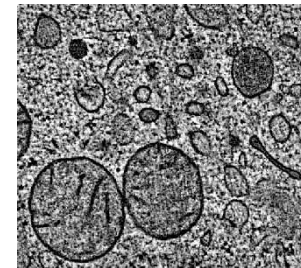
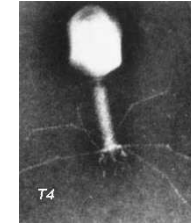
- **perturbable** - any living system expresses an external behaviour
- **incomplete** - our knowledge of biological systems is incomplete
- **biotechnology** - we may always assume to add some external components to a given complex



Partial information imposes that the approach is to predict the behaviours (of missing knowledge).

biological systems are *open* in different meanings

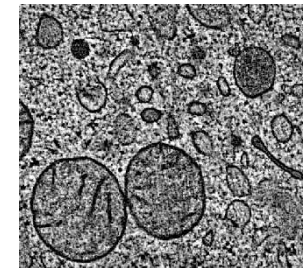
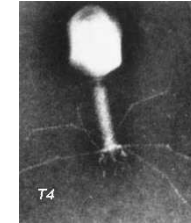
- **perturbable** - any living system expresses an external behaviour
- **incomplete** - our knowledge of biological systems is incomplete
- **biotechnology** - we may always assume to add some external components to a given complex



Partial information imposes that the approach is to predict the behaviours (of missing knowledge).

biological systems are *open* in different meanings

- **perturbable** - any living system expresses an external behaviour
- **incomplete** - our knowledge of biological systems is incomplete
- **biotechnology** - we may always assume to add some external components to a given complex



Partial information imposes that the approach is to predict the behaviours (of missing knowledge).

# partial biological knowledge, an abstraction

## molecule - as - computation

Systems of interacting molecular entities are described and modelled by a system of interacting computational entities.

Regev, Shapiro - *Nature*, September 2002.

- **relevant:** essential properties
- **computable:** computational knowledge
- **understandable:** conceptual framework
- **extensible:** capture other real properties in the same mathematical framework

## partial knowledge - as - openness

Modelling *incomplete systems* by applying a theory for *open systems*.

# partial biological knowledge, an abstraction

## molecule - as - computation

Systems of interacting molecular entities are described and modelled by a system of interacting computational entities.

Regev, Shapiro - *Nature*, September 2002.

- **relevant**: essential properties
- **computable**: computational knowledge
- **understandable**: conceptual framework
- **extensible**: capture other real properties in the same mathematical framework

## partial knowledge - as - openness

Modelling *incomplete systems* by applying a theory for *open systems*.

# partial biological knowledge, an abstraction

## molecule - as - computation

Systems of interacting molecular entities are described and modelled by a system of interacting computational entities.

Regev, Shapiro - *Nature*, September 2002.

- **relevant**: essential properties
- **computable**: computational knowledge
- **understandable**: conceptual framework
- **extensible**: capture other real properties in the same mathematical framework

## partial knowledge - as - openness

Modelling *incomplete systems* by applying a theory for *open systems*.

# partial biological knowledge, an abstraction

## molecule - as - computation

Systems of interacting molecular entities are described and modelled by a system of interacting computational entities.

Regev, Shapiro - *Nature*, September 2002.

- **relevant**: essential properties
- **computable**: computational knowledge
- **understandable**: conceptual framework
- **extensible**: capture other real properties in the same mathematical framework

## partial knowledge - as - openness

Modelling *incomplete systems* by applying a theory for *open systems*.

# Outline

- 1 Open Biological Systems
- 2 Open distributed and concurrent systems
- 3 Symbolic Transition System
- 4 Biological examples

# Modelling open distributed and concurrent systems

## Main ideas:

- open systems can evolve (*no universal closure, higher level of dynamics*);
- constructive methodology (*unification based - the most general unifier is choosed to infer the transition*);
- generality and friendly notation;
- complementary with contextualization techniques (*using contexts as labels to derive LTS, for which bisimilarity is a congruence*);

## The advantage is to import:

- theories
- automatic tools

# Modelling open distributed and concurrent systems

## Main ideas:

- open systems can evolve (*no universal closure, higher level of dynamics*);
- constructive methodology (*unification based - the most general unifier is choosed to infer the transition*);
- generality and friendly notation;
- complementary with contextualization techniques (*using contexts as labels to derive LTS, for which bisimilarity is a congruence*);

## The advantage is to import:

- theories
- automatic tools

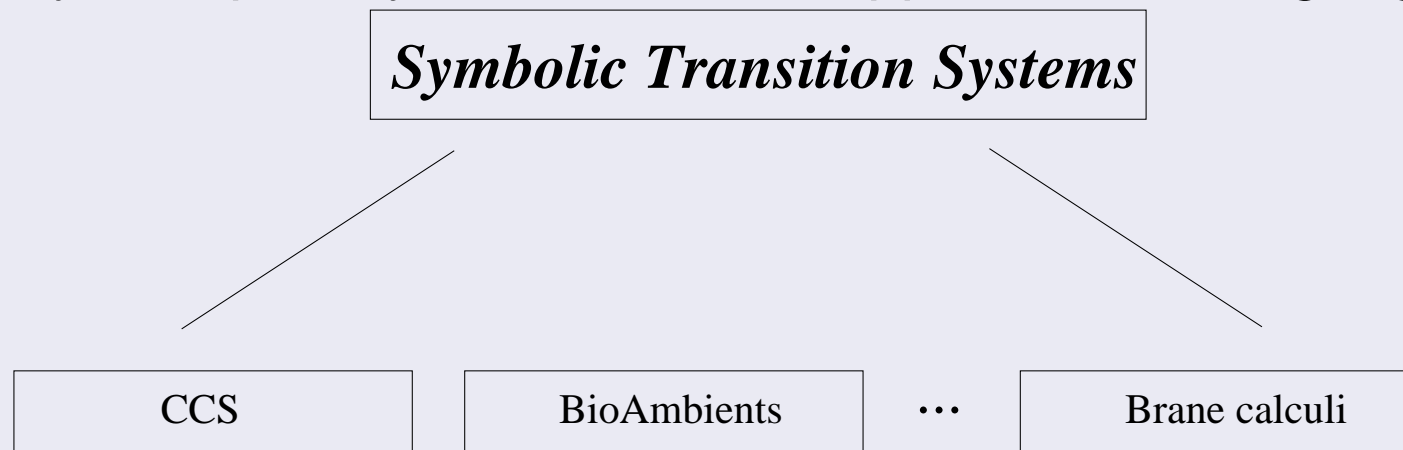
# Modelling language

Process algebras are specification languages defined over a set basic actions and operators to compose them.

- general approach (basic action definition depend on the phenomenon to model)
- easy-to-use formalism

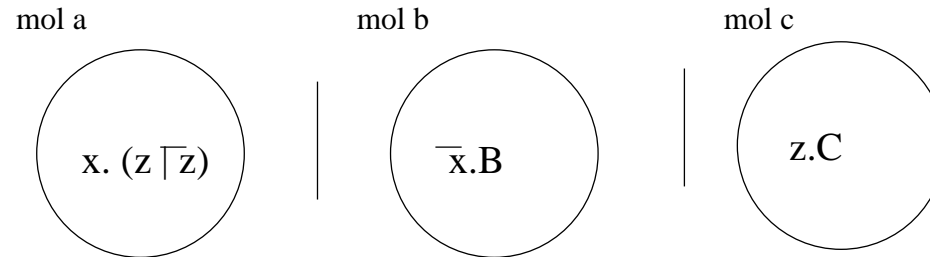
## Not linked to a specific language

The theory for *Open System* we use is applicable to a language family.



# Process calculi

Process algebra terms model interacting entities



## process calculi

CCS	action, co-action
BioAmbients	$[\dots]$ , <i>enter a</i> , <i>accept a</i> , ...
Brane Calculi	..., phagocytosis, exocytosis, ...
...	...

# Contexts

Main ingredients: contexts and process variables.

## Contexts: processes with holes

$$P[X_1]||Q[X_2]$$

The evolution of the system depends on the particular components will substitute  $X_1$  and  $X_2$ .

Computational steps, involving external components, require some hypothesis to fix the particular behaviour needed to compute the desired action.

A simulation trace is equipped with the required hypothesis.

# Outline

- 1 Open Biological Systems
- 2 Open distributed and concurrent systems
- 3 Symbolic Transition System**
- 4 Biological examples

# Transitions of open systems

## Required behaviour

$$Q[X] \rightarrow Q'[Y]$$

## Recording

- the hypothesis assumed at each computation step for each unspecified component;
- how the computation is modified by the new assumption taken

by using formulas as labels:

$$\varphi ::= X \mid \diamond a \varphi \mid f(\varphi_1, \dots, \varphi_n)$$

# Formulae as labels

General structure of labels:

$$C[X_1, \dots, X_n] \xrightarrow{(\varphi_1, \dots, \varphi_n)}_a D[Y_1, \dots, Y_m]$$

The formula labelling a transition, from one state to another, expresses the structural and behavioural assumptions on the unspecified components in order to perform the transition.

## formulae

Formula definition depends on the particular process algebra adopted:

- basic actions
- structural operators
- composition operators

# Simulating systems

## Operational semantics

### Inference rules

$$\frac{}{m[in\ n.Q\ | R] \ | n[P] \ \rightarrow_{\tau} \ n[m[Q\ | R] \ | P]} \text{ (in capability)}$$

$$\frac{}{n[P] \ | \ open\ n.Q \ \rightarrow_{\tau} \ P \ | \ Q} \text{ (open capability)}$$

$$\frac{P_1 \ \rightarrow_{\alpha} \ Q_1 \quad P_2 \ \rightarrow_{\alpha^{\perp}} \ Q_2}{P_1 \ | \ P_2 \ \rightarrow_{\tau} \ Q_1 \ | \ Q_2} \text{ (communication)}$$

### specific formula set

$$\varphi ::= X \ | \ \diamond in\ n\ \varphi \ | \ \diamond open\ n\ \varphi \ | \ (\varphi_1 \ | \ \varphi_2)$$

# Simulating systems

## Operational semantics

### Inference rules

$$\frac{}{m[in\ n.Q\ | R] \ | \ n[P] \ \rightarrow_{\tau} \ n[m[Q\ | R] \ | P]} \text{ (in capability)}$$

$$\frac{}{n[P] \ | \ open\ n.Q \ \rightarrow_{\tau} \ P \ | \ Q} \text{ (open capability)}$$

$$\frac{P_1 \ \rightarrow_{\alpha} \ Q_1 \quad P_2 \ \rightarrow_{\alpha^{\perp}} \ Q_2}{P_1 \ | \ P_2 \ \rightarrow_{\tau} \ Q_1 \ | \ Q_2} \text{ (communication)}$$

### specific formula set

$$\varphi ::= X \ | \ \diamond in\ n\ \varphi \ | \ \diamond open\ n\ \varphi \ | \ (\varphi_1 \ | \ \varphi_2)$$

# Symbolic Transition

$$v[ X ] \mid c[open\ v.(prot \mid rna^\perp)]$$

## initial state

If we want the *virus*  $v[\dots]$  to enter the *cell*  $c[\dots]$ , we have to assume that it has the capability to do that via a suitable action.

$$\xrightarrow[\tau]{in\ c.Y|Z}$$

## label

The formula captures the most general shape of the virus capable to enter the cell  $c[\dots]$ !

$$c[v[Y \mid Z] \mid open\ v.(prot \mid rna^\perp)]$$

## residual

In the next state (i.e. the continuation) there is a modification of the configuration of the process variables, due to the assumptions made.

# Symbolic Transition

$$v[ X ] \mid c[open\ v.(prot \mid rna^\perp)]$$

## initial state

If we want the *virus*  $v[...]$  to enter the *cell*  $c[...]$ , we have to assume that it has the capability to do that via a suitable action.

$$\begin{array}{c} in\ c.Y|Z \\ \longrightarrow \tau \end{array}$$

## label

The formula captures the most general shape of the virus capable to enter the cell  $c[...]$ !

$$c[v[Y \mid Z] \mid open\ v.(prot \mid rna^\perp)]$$

## residual

In the next state (i.e. the continuation) there is a modification of the configuration of the process variables, due to the assumptions made.

# Symbolic Transition

$$v[ X ] \mid c[open\ v.(prot \mid rna^\perp)]$$

## initial state

If we want the *virus*  $v[...]$  to enter the *cell*  $c[...]$ , we have to assume that it has the capability to do that via a suitable action.

$$\xrightarrow{\text{in } c.Y|Z} \tau$$

## label

The formula captures the most general shape of the virus capable to enter the cell  $c[...]$ !

$$c[v[Y \mid Z] \mid open\ v.(prot \mid rna^\perp)]$$

## residual

In the next state (i.e. the continuation) there is a modification of the configuration of the process variables, due to the assumptions made.

# Symbolic Transition

$$v[ X ] \mid c[open\ v.(prot \mid rna^\perp)]$$

## initial state

If we want the *virus*  $v[ \dots ]$  to enter the *cell*  $c[ \dots ]$ , we have to assume that it has the capability to do that via a suitable action.

$$\begin{array}{c} in\ c.Y|Z \\ \longrightarrow \tau \end{array}$$

## label

The formula captures the most general shape of the virus capable to enter the cell  $c[ \dots ]$ !

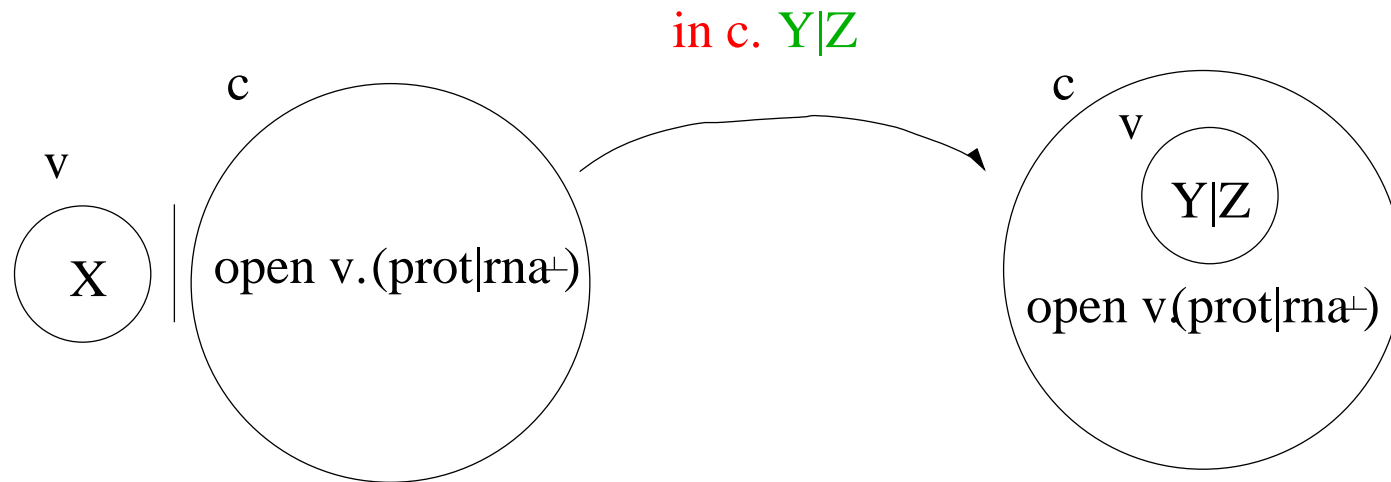
$$c[v[Y \mid Z] \mid open\ v.(prot \mid rna^\perp)]$$

## residual

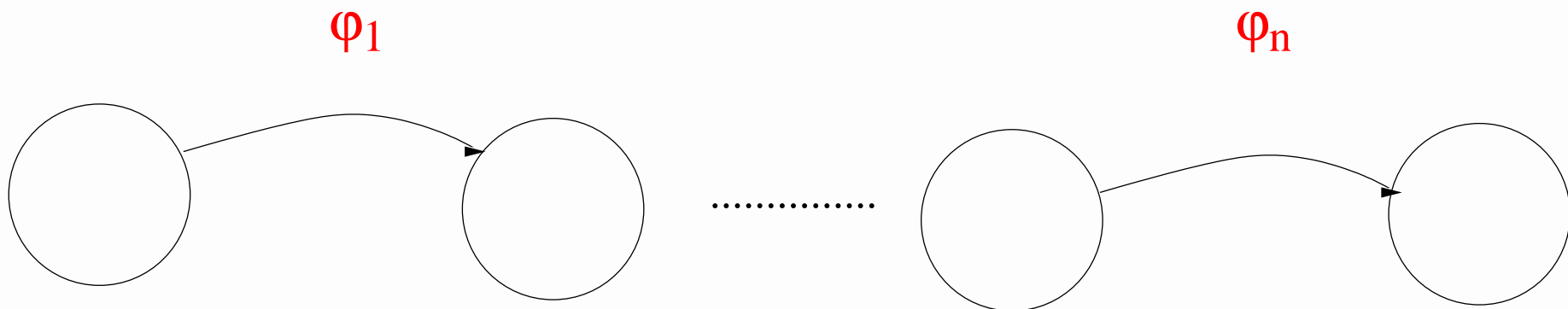
In the next state (i.e. the continuation) there is a modification of the configuration of the process variables, due to the assumptions made.

# Symbolic Transition

At each step of the computation, new hypotheses are required

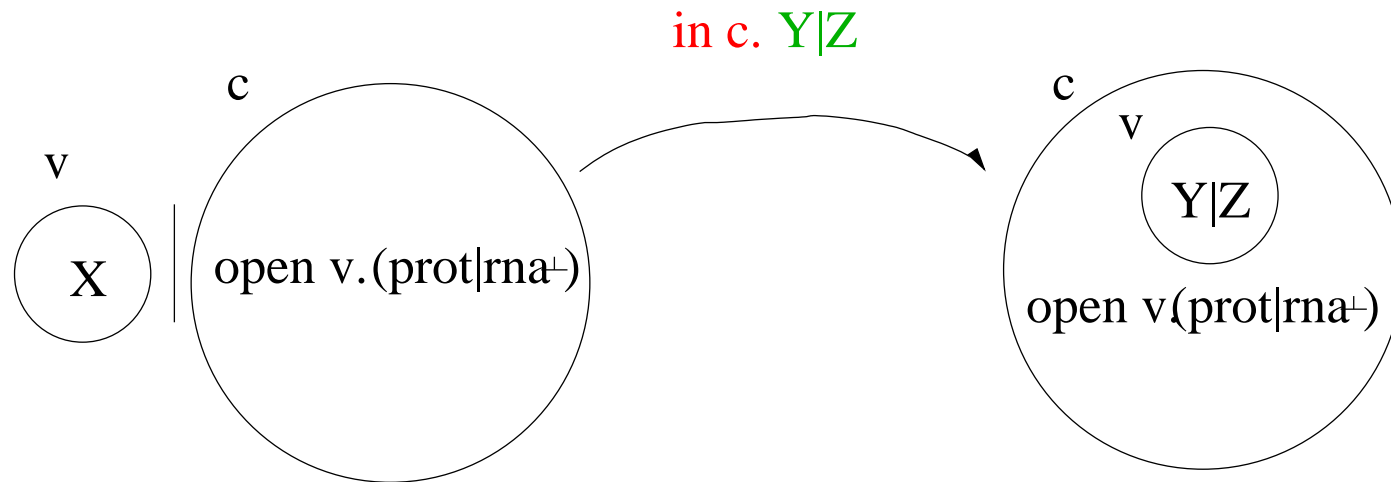


how to build a trace ?

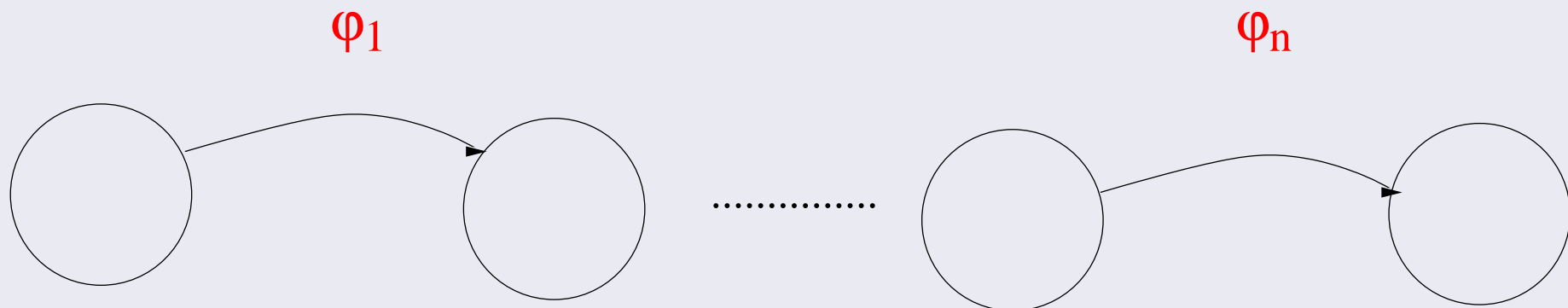


# Symbolic Transition

At each step of the computation, new hypotheses are required



how to build a trace ?



# Symbolic Trace

$$E[X] = v[X] \mid c[\text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

first transition

$$E[X] \xrightarrow{\text{in } c.Y|Z} \tau c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

second transition

$$c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)] \xrightarrow{Y,Z} \tau c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp]$$

third transition

$$c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp] \xrightarrow{\diamond \text{rna} W,Z} c[W \mid Z \mid \text{prot}]$$

label composition

$$\text{in } c.(\diamond \text{rna } W) \mid Y$$

# Symbolic Trace

$$E[X] = v[X] \mid c[\text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

## fi rst transition

$$E[X] \xrightarrow{\text{in } c.Y|Z} \tau c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

## second transition

$$c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)] \xrightarrow{Y,Z} \tau c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp]$$

## third transition

$$c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp] \xrightarrow{\diamond \text{rna}W,Z} c[W \mid Z \mid \text{prot}]$$

## label composition

$$\text{in } c.(\diamond \text{rna } W) \mid Y$$

# Symbolic Trace

$$E[X] = v[X] \mid c[\text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

## first transition

$$E[X] \xrightarrow{\text{in } c.Y|Z} \tau c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

## second transition

$$c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)] \xrightarrow{Y,Z} \tau c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp]$$

## third transition

$$c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp] \xrightarrow{\diamond \text{rna } W,Z} c[W \mid Z \mid \text{prot}]$$

## label composition

$$\text{in } c.(\diamond \text{rna } W) \mid Y$$

# Symbolic Trace

$$E[X] = v[X] \mid c[\text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

## first transition

$$E[X] \xrightarrow{\text{in } c.Y|Z} \tau c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)]$$

## second transition

$$c[v[Y \mid Z] \mid \text{open } v.(\text{prot} \mid \text{rna}^\perp)] \xrightarrow{Y,Z} \tau c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp]$$

## third transition

$$c[Y \mid Z \mid \text{prot} \mid \text{rna}^\perp] \xrightarrow{\diamond \text{rna}W,Z} c[W \mid Z \mid \text{prot}]$$

## label composition

$$\text{in } c.(\diamond \text{rna } W) \mid Y$$

- Correctness and completeness of Symbolic Transition Systems
  - **correctness** every concrete behaviour (of a full specified system) has an abstract representation (the corresponding in STS)
  - **completeness** every instance of an abstract behaviour (STS) correspond to a concrete behaviour
- symbolic bisimulation for open systems (different from universal closure)
- Symbolic Transition System is built by unification (Prolog)

## bibliography

- Paolo Baldan, Andrea Bracciali, Roberto Bruni. **Bisimulation by unification**. AMAST 2002, LNCS vol.2422: 254-270.

- Correctness and completeness of Symbolic Transition Systems
  - **correctness** every concrete behaviour (of a full specified system) has an abstract representation (the corresponding in STS)
  - **completeness** every instance of an abstract behaviour (STS) correspond to a concrete behaviour
- symbolic bisimulation for open systems (different from universal closure)
- Symbolic Transition System is built by unification (Prolog)

## bibliography

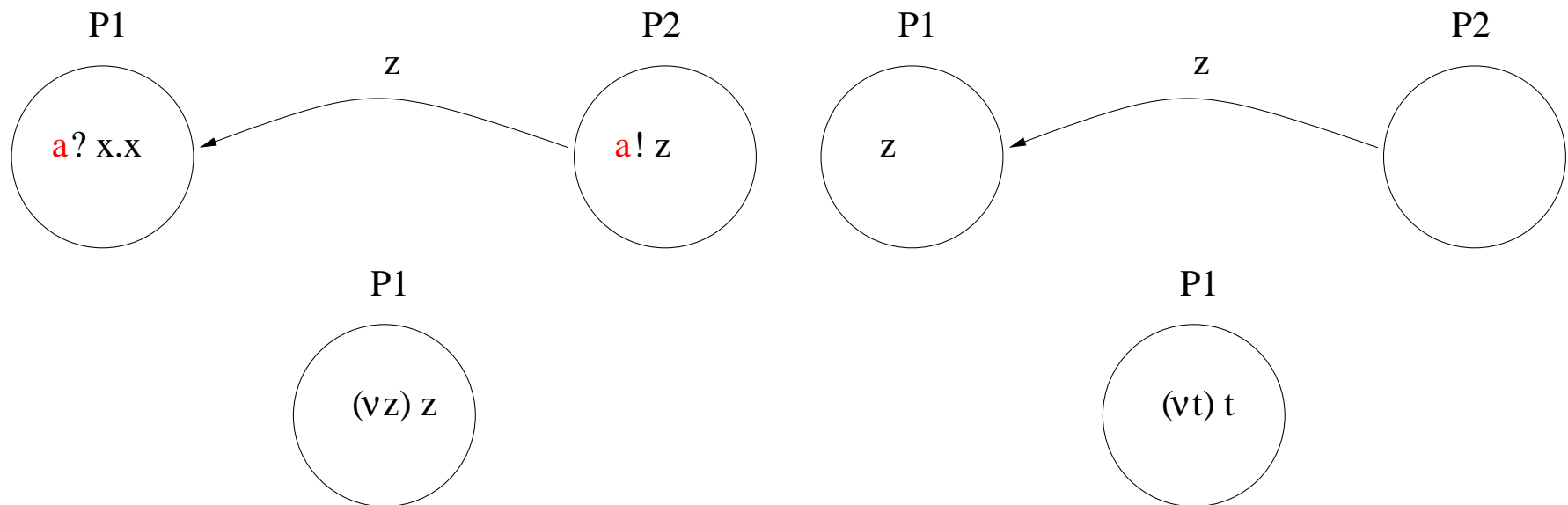
- Paolo Baldan, Andrea Bracciali, Roberto Bruni. **Bisimulation by unification**. AMAST 2002, LNCS vol.2422: 254-270.

# On going work

STS works with a subset of the modeling languages: not able to handle names, **yet!**

## $\pi$ -calculus

$a ! b$       output  
 $a ? x$       input  
 $(\nu n)P$    private names



# Outline

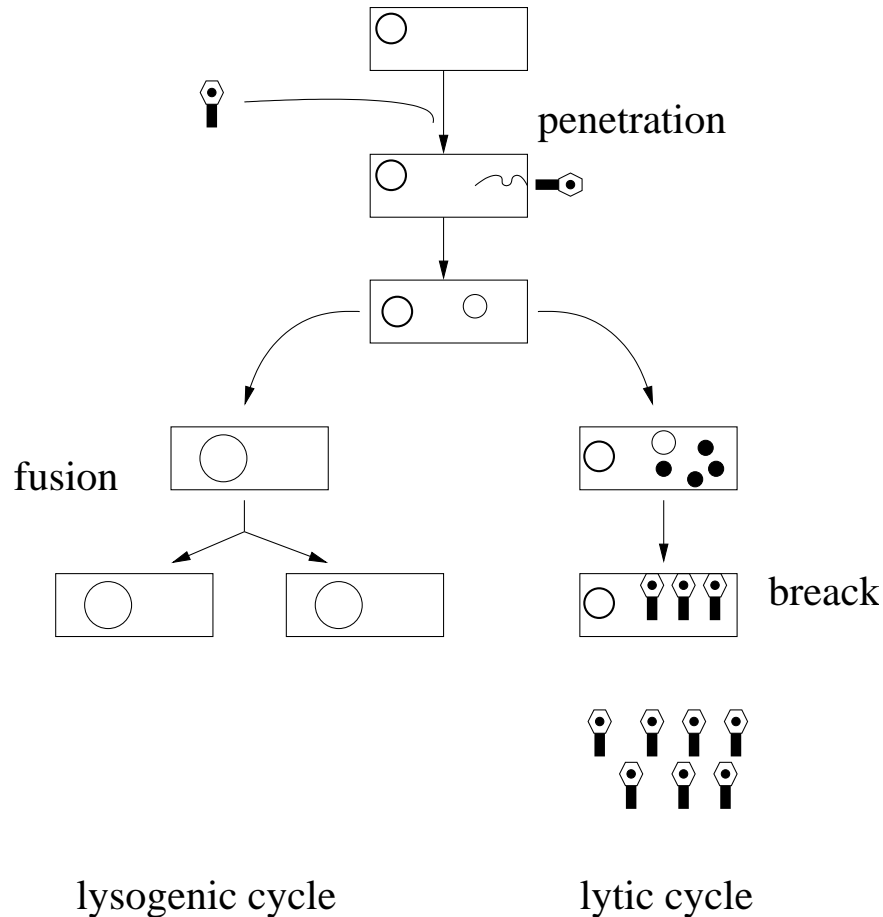
- 1 Open Biological Systems
- 2 Open distributed and concurrent systems
- 3 Symbolic Transition System
- 4 Biological examples**

# BioAmbients

Introduced for the modelling of biological membranes

$[ \dots ]$	empty membrane
$[ a?x \mid a!z ]$	with internal behaviour
$[ \dots \mid a!z ] \mid [ a?x \mid \dots ]$	different membranes with internal behaviour
$[ \dots \mid \textit{sibling } a!z ] \mid [ \textit{sibling } a?x \mid \dots ]$	trespassing membranes
$[ \dots \mid \textit{merge}^+ a ] \mid [ \textit{merge}^- a \mid \dots ]$	merging membranes
$[ \dots \mid \textit{enter } a ] \mid [ \textit{accept } a \mid \dots ]$	nesting membranes
$[ [ \dots \mid \textit{exit } a ] \mid \textit{allow } a \mid \dots ]$	nesting membranes

# Toy example



- each protein is an ambient
- activation as communication
- inhibition as encapsulation

## very simplified hypotheses

- 1 - high concentration of **CI** determines lysogeny
- 2 - absence of **CI** determines lysis
- 3 - **CII** promotes the production of **CI**
- 4 - **CIII** can inhibit **HFL**
- 5 - low concentration of **CRO** stimulates **CI** production
- 6 - high concentration of **CRO** inhibits **CI** production

# BioAmbient code

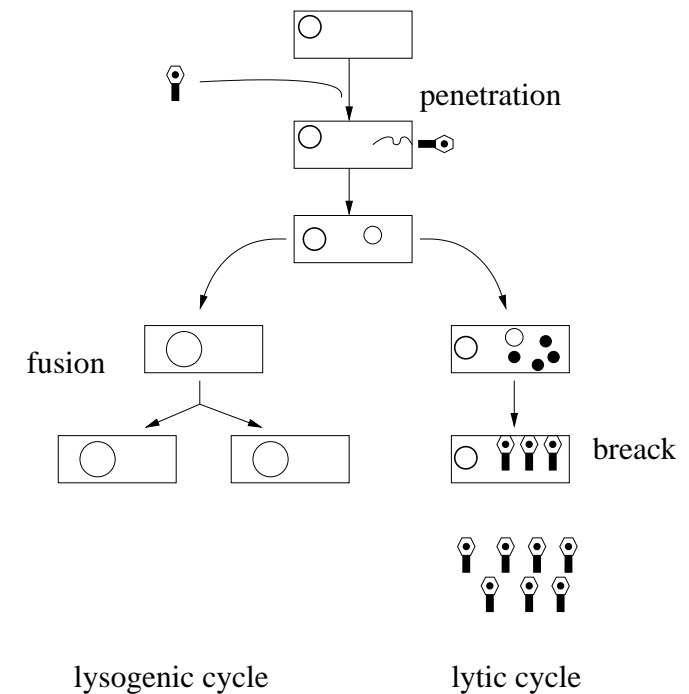
[VIRUS] = [merge<sup>+</sup> virus.([C3] | [C2] | [C1] | [CRO]) | [DNA $\lambda$  ]]  
DNA $\lambda$  = (lyso?.enter dnae.0)

+

lysi?.( $\lambda$ [exit newph.VIRUS] | expel newph)

[ECOLI] = [merge<sup>-</sup> virus | Dna<sub>e</sub>[accept dnae] | [HFL]]

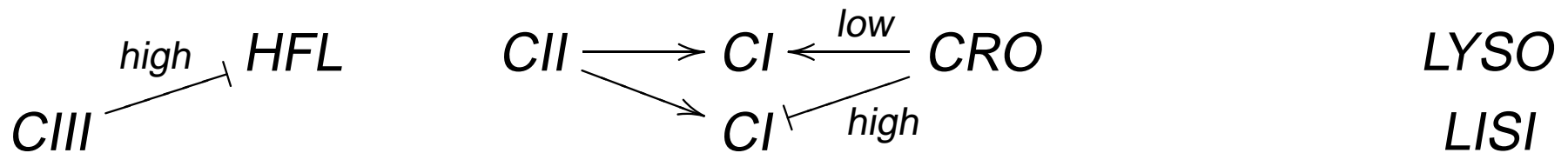
HFL = enter h\_c3.0 + X



# Bioambients - logical formulae

$$\varphi ::= X \mid \diamond a \varphi \mid \varphi_1 + \varphi_2 \mid \varphi_1 \mid \varphi_n \mid a.\varphi$$
$$a ::= n? \mid n! \mid \text{enter } n \mid \text{accept } n \mid \dots$$

# Assuming partial knowledge



$C_3 = l_{c3!.0} + \text{accept } h_{c3}. \text{pro}_{c2!.0}$

$C_2 = \text{pro}_{c2?}. \text{pro}_{c1!.0} + \text{enter } c2.0$

$C_1 = \text{pro}_{c1?}.( h_{cro?}. \text{lysi!.0} + l_{cro?}. \text{lyso!.0})$

$CRO = l_{cro!.0} + h_{cro}.0$

# $\lambda$ phago: lytic trace

$$\begin{aligned}
 & Ecoli \left( ([C3] \mid [C2] \mid [C1] \mid [CRO]) \mid [DNA \lambda] \mid_{Dna_e} [accept dnae] \mid [enter h\_c3.0 + X] \right) \\
 & \quad \xrightarrow{(I\_c3?.Y_1 + Y_2) \mid Y_3} \\
 & Ecoli \left( CIII[0] \mid [C2] \mid [C1] \mid [CRO] \right) \mid [DNA \lambda] \mid_{Dna_e} [accept dnae] \mid_{hfl} [Y_1 \mid Y_3] \\
 & \quad \xrightarrow{(Y_4 + accept c2.Y_5 \mid Y_6), Y_3} \\
 & Ecoli \left( (CIII[0] \mid [C1] \mid [CRO]) \mid [DNA \lambda] \mid_{Dna_e} [accept dnae] \mid_{hfl} [CII[0] \mid (Y_5 \mid Y_6) \mid Y_3] \right) \\
 & \quad \xrightarrow{(Y_7 + lysi!.Y_8), Y_6, Y_3} \\
 & Ecoli \left( (CIII[0] \mid [C1] \mid [CRO]) \mid [\lambda [exit newph.VIRUS] \mid expel newph] \mid_{Dna_e} [accept dnae] \mid_{hfl} [CII[0] \mid (Y_8 \mid Y_6) \mid Y_3] \right) \\
 & \quad \xrightarrow{Y_8, Y_6, Y_3} \\
 & Ecoli \left( (CIII[0] \mid [C1] \mid [CRO]) \mid \lambda [VIRUS] \mid_{Dna_e} [accept dnae] \mid_{hfl} [CII[0] \mid (Y_8 \mid Y_6) \mid Y_3] \right)
 \end{aligned}$$

# The deduced symbolic trace

$$(I_{c3?}.(Y_4 + \text{accept } c2(Y_7 + \text{lysi!}.Y_8) | Y_6) + Y_2) | Y_3$$

The formula encodes the minimum requirements that a system must satisfy in order to simulate the lytic path.

## A valid theory

By correctness of the STS, the trace computed can be obtained in a fully specified system, satisfying the above biological hypotheses.

# The deduced symbolic trace

$$(I_{c3?}.(Y_4 + \text{accept } c2(Y_7 + \text{lysi!}.Y_8) | Y_6) + Y_2) | Y_3$$

The formula encodes the minimum requirements that a system must satisfy in order to simulate the lytic path.

## A valid theory

By correctness of the STS, the trace computed can be obtained in a fully specified system, satisfying the above biological hypotheses.

# Possible applications

How to use our approach:

## Before in silico experiments

Preliminary study of the systems.

## Partial data

When is not possible to design a complete model.

## Extension of existing models

To build new complexes by studying the requirements the new components should have in order to interact in the desired way

# Future work

- **names**: extending the approach to make the STS also works with process algebras with name restriction and name passing;
- **automatic tool** build automatic simulator working with any process algebra of interest (based on unification);
- **quantitative analysis** adding values for calculating quantitative values (probability/rate/concentrations) involved in transitions;

# The end

Thanks for your attention !

# Future work II

- building a *significant* trace requires to follow a criterium to discriminate between infinite moves.
- ... quantitative values may help —