

Dynamics of Biological Systems

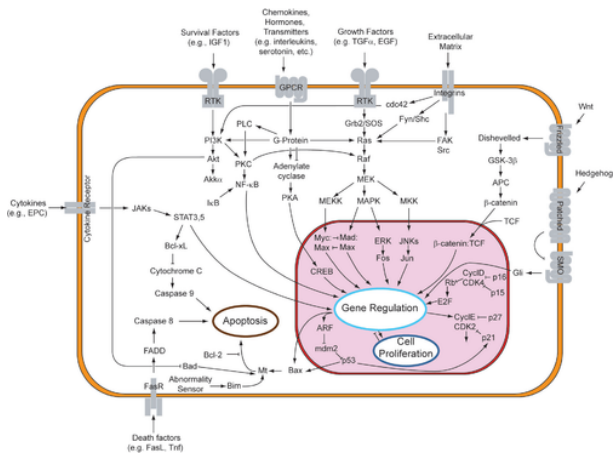
Part III - Formal modelling of biological systems

Introduction

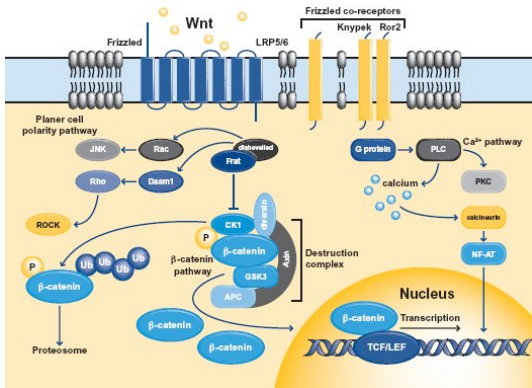
In scientific articles the dynamics of cellular pathways is often described by means of informal/ambiguous diagrams accompanied by a textual description

Let's see some examples....

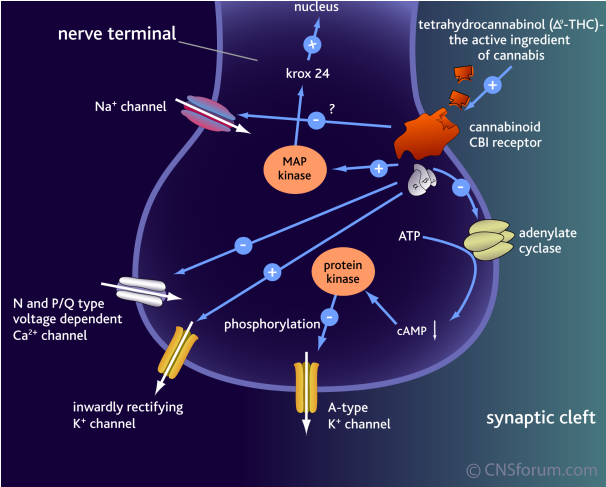
Introduction



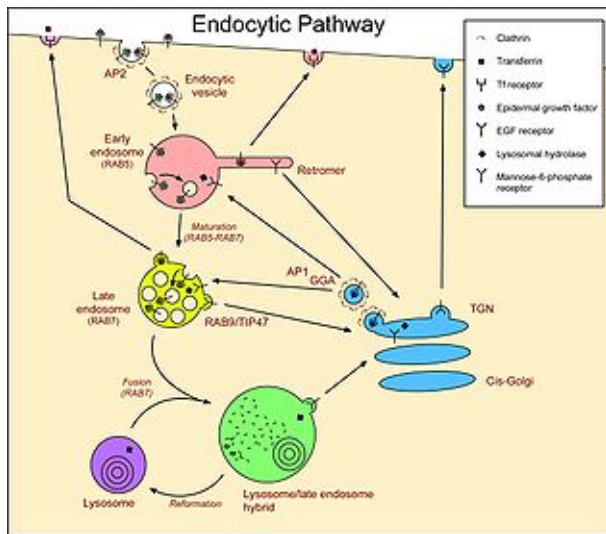
Introduction



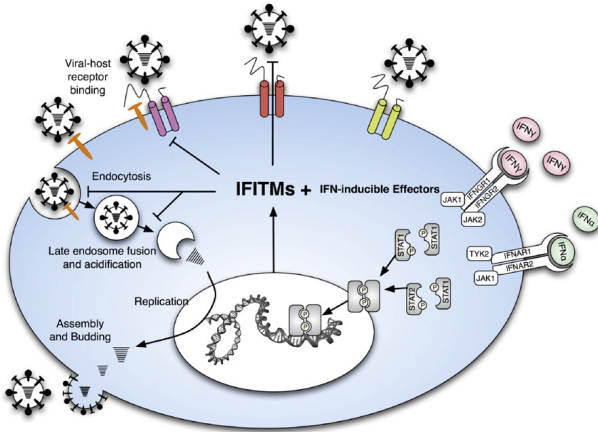
Introduction



Introduction



Introduction



Introduction

This informal way of describing the dynamics of biological systems is not suitable for the application of mathematical and computational analysis means

On the other hand cellular processes cannot be described only as (bio)chemical reactions. In fact, we have to take into account:

- the structure of the cell as a nesting of compartments
- the dynamics of membranes
- the modularity of some structures (e.g. of the DNA as a sequence of genes)

SBML (an XML dialect) is nowadays almost a standard for the description of cellular processes to be analysed with computational means:

- it allows reactions in a compartmentalized structure to be described
- does not allow membrane dynamics to be described

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level1" level="1" version="2">
  <model name="the_name_of_my_model">
    <listOfUnitDefinitions>
      ...
    </listOfUnitDefinitions>
    <listOfCompartments>
      ...
    </listOfCompartments>
    <listOfSpecies>
      ...
    </listOfSpecies>
    <listOfParameters>
      ...
    </listOfParameters>
    <listOfRules>
      ...
    </listOfRules>
    <listOfReactions>
      ...
    </listOfReactions>
  </model>
</sbml>
```

```
<listOfUnitDefinitions>
  <unitDefinition name="mmls">
    <listOfUnits>
      <unit kind="mole" scale="-3"/>
      <unit kind="liter" exponent="-1"/>
      <unit kind="second" exponent="-1"/>
    </listOfUnits>
  </unitDefinition>
</listOfUnitDefinitions>
```

```
<model>
  ...
  <listOfUnitDefinitions>
    <unitDefinition name="volume">
      <listOfUnits>
        <unit kind="liters" scale="-3"/>
      </listOfUnits>
    </unitDefinition>
  </listOfUnitDefinitions>
  ...
</model>
```

```
<model>
  ...
  <listOfCompartments>
    <compartment name="cytosol" volume="2.5"/>
    <compartment name="mitochondria" volume="0.3"/>
  </listOfCompartments>
  ...
</model>
```

```
<model>
  ...
  <listOfSpecies>
    <species name="Glucose" compartment="cell" initialAmount="4"/>
    <species name="Glucose_6_P" compartment="cell" initialAmount="0.75"/>
  </listOfSpecies>
  ...
</model>
```

```

<model>
  ...
  <listOfSpecies>
    ...
  </listOfSpecies>
  <listOfParameters>
    <parameter name="Km1" value="2.3" units="second"/>
    <parameter name="Km2" value="10.7" units="second"/>
  </listOfParameters>
  <listOfReactions>
    ...
  </listOfReactions>
  ...
</model>

```

```

<model>
  ...
  <listOfRules>
    <parameterRule name="k" formula="k3/k2"/>
    <speciesConcentrationRule species="s2" formula="k * z/(1 + k)"/>
    <compartmentVolumeRule compartment="A" formula="0.10 * k4"/>
  </listOfRules>
  ...
</model>

```

```
<model>
  ...
  <listOfReactions>
    <reaction name="J1">
      <listOfReactants>
        <speciesReference species="X0" stoichiometry="1"/>
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="S1" stoichiometry="1"/>
      </listOfProducts>
      <kineticLaw formula="k1*X0">
        <listOfParameters>
          <parameter name="k1" value="0"/>
        </listOfParameters>
      </kineticLaw>
    </reaction>
  </listOfReactions>
  ...
</model>
```

Formal modelling of cellular processes

In the last few years formalisms proposed by computer scientists to describe systems of interactive components have been applied to the description of biological systems.

- The π -calculus
- Petri Nets
- concurrent automata
- ...

This allows:

- unambiguous description of phenomena
- development of simulators
- application of formal analysis tools unknown to biologists (model checkers, behavioural equivalences)

A partial view...

In Pisa, we have developed a simple formalism for the modelling of biological systems.

At the beginning of our work our aim was to try to apply formal methods to models of biological systems

We were looking for a formalism

- based on term rewriting
- with a simple semantics
- very general

As a consequence, we defined the Calculus of Looping Sequences (CLS)...

Outline of the talk

1 Introduction

2 The Calculus of Looping Sequences (CLS)

- Definition of CLS
- The lac operon in CLS

3 Bisimulations in CLS

- A labeled semantics for CLS
- Bisimulations in CLS
- Bisimulations applied to the CLS model of the lac operon

4 Stochastic CLS

The Calculus of Looping Sequences (CLS)

We assume an alphabet \mathcal{E} . **Terms** T and **Sequences** S of CLS are given by the following grammar:

$$\begin{aligned} T &::= S \mid (S)^L \mid T \mid T \\ S &::= \epsilon \mid a \mid S \cdot S \end{aligned}$$

where a is a generic element of \mathcal{E} , and ϵ is the empty sequence.

The operators are:

$S \cdot S$: Sequencing

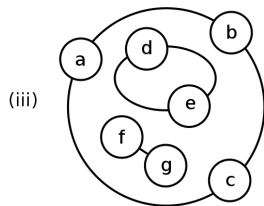
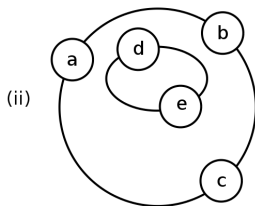
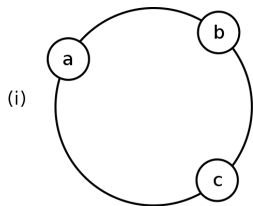
$(S)^L$: Looping (S is closed and it can rotate)

$T_1 \mid T_2$: Containment (T_1 contains T_2)

$T \mid T$: Parallel composition (juxtaposition)

Actually, looping and containment form a single binary operator $(S)^L \mid T$.

Examples of Terms



$$(i) \quad (a \cdot b \cdot c)^L \rfloor \epsilon$$

$$(ii) \quad (a \cdot b \cdot c)^L \rfloor (d \cdot e)^L \rfloor \epsilon$$

$$(iii) \quad (a \cdot b \cdot c)^L \rfloor (f \cdot g \mid (d \cdot e)^L \rfloor \epsilon)$$

Structural Congruence

The **Structural Congruence** relations \equiv_S and \equiv_T are the least congruence relations on sequences and on terms, respectively, satisfying the following rules:

$$S_1 \cdot (S_2 \cdot S_3) \equiv_S (S_1 \cdot S_2) \cdot S_3 \quad S \cdot \epsilon \equiv_S \epsilon \cdot S \equiv_S S$$

$$T_1 \mid T_2 \equiv_T T_2 \mid T_1 \quad T_1 \mid (T_2 \mid T_3) \equiv_T (T_1 \mid T_2) \mid T_3$$

$$T \mid \epsilon \equiv_T T \quad (S_1 \cdot S_2)^L \rfloor T \equiv_T (S_2 \cdot S_1)^L \rfloor T$$

We write \equiv for \equiv_T .

CLS Patterns

Let us consider variables of three kinds:

- term variables (X, Y, Z, \dots)
- sequence variables ($\tilde{x}, \tilde{y}, \tilde{z}, \dots$)
- element variables (x, y, z, \dots)

Patterns P and **Sequence Patterns** SP of CLS extend CLS terms and sequences with variables:

$$\begin{aligned} P & ::= SP \mid (SP)^L \mid P \mid P \mid X \\ SP & ::= \epsilon \mid a \mid SP \cdot SP \mid x \mid \tilde{x} \end{aligned}$$

where a is a generic element of \mathcal{E} , ϵ is the empty sequence, and x, \tilde{x} and X are generic element, sequence and term variables

The structural congruence relation \equiv extends trivially to patterns

Rewrite Rules

A **Rewrite Rule** is a pair (P, P') , denoted $P \mapsto P'$, where:

- P, P' are patterns
- variables in P' are a subset of those in P

A rule $P \mapsto P'$ can be applied to all terms that are instantiations of P .

Example: $a \cdot x \cdot a \mapsto b \cdot x \cdot b$

- can be applied to $a \cdot c \cdot a$ (producing $b \cdot c \cdot b$)
- cannot be applied to $a \cdot c \cdot c \cdot a$

Example: $(a \cdot \tilde{x})^L \rfloor (b \mid X) \mapsto (c \cdot \tilde{x})^L \rfloor X$

- can be applied to $(a \cdot a \cdot a)^L \rfloor (b \mid b \mid (a)^L \rfloor b)$
- the result is either $(c \cdot a \cdot a)^L \rfloor (b \mid (a)^L \rfloor b)$ or $(a \cdot a \cdot a)^L \rfloor (b \mid b \mid (c)^L \rfloor \epsilon)$

Formal Semantics

$P\sigma$ denotes the term obtained by replacing any variable in T with the corresponding term, sequence or element.

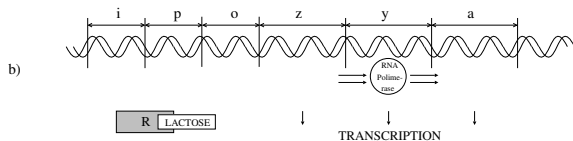
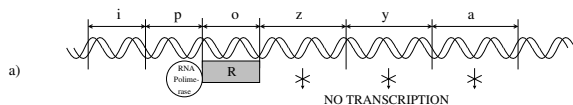
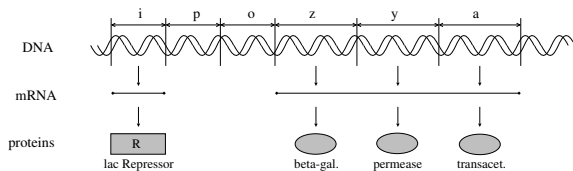
Σ is the set of all possible instantiations σ

Given a set of rewrite rules \mathcal{R} , evolution of terms is described by the transition system given by the least relation \rightarrow satisfying

$$\frac{P \mapsto P' \in \mathcal{R} \quad P\sigma \neq \epsilon \quad \sigma \in \Sigma}{P\sigma \rightarrow P'\sigma}$$
$$\frac{T \rightarrow T'}{T \mid T'' \rightarrow T' \mid T''} \quad \frac{T \rightarrow T'}{(S)^L \mid T \rightarrow (S)^L \mid T'}$$

and closed under structural congruence \equiv .

CLS modeling examples: the *lac* operon (1)



CLS modeling examples: the *lac* operon (2)

$$Ecoli ::= (m)^L \mid (lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym)$$

Rules for DNA transcription/translation:

$$lacI \cdot \tilde{x} \mapsto lacI \cdot \tilde{x} \mid repr \quad (R1)$$

$$polym \mid \tilde{x} \cdot lacP \cdot \tilde{y} \mapsto \tilde{x} \cdot PP \cdot \tilde{y} \quad (R2)$$

$$\tilde{x} \cdot PP \cdot lacO \cdot \tilde{y} \mapsto \tilde{x} \cdot lacP \cdot PO \cdot \tilde{y} \quad (R3)$$

$$\tilde{x} \cdot PO \cdot lacZ \cdot \tilde{y} \mapsto \tilde{x} \cdot lacO \cdot PZ \cdot \tilde{y} \quad (R4)$$

$$\tilde{x} \cdot PZ \cdot lacY \cdot \tilde{y} \mapsto \tilde{x} \cdot lacZ \cdot PY \cdot \tilde{y} \mid betagal \quad (R5)$$

$$\tilde{x} \cdot PY \cdot lacA \mapsto \tilde{x} \cdot lacY \cdot PA \mid perm \quad (R6)$$

$$\tilde{x} \cdot PA \mapsto \tilde{x} \cdot lacA \mid transac \mid polym \quad (R7)$$

CLS modeling examples: the *lac* operon (3)

$$Ecoli ::= (m)^L \rfloor (lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym)$$

Rules to describe the binding of the lac Repressor to gene o, and what happens when lactose is present in the environment of the bacterium:

$$repr \mid \tilde{x} \cdot lacO \cdot \tilde{y} \mapsto \tilde{x} \cdot RO \cdot \tilde{y} \quad (R8)$$

$$LACT \mid (m \cdot \tilde{x})^L \rfloor X \mapsto (m \cdot \tilde{x})^L \rfloor (X \mid LACT) \quad (R9)$$

$$\tilde{x} \cdot RO \cdot \tilde{y} \mid LACT \mapsto \tilde{x} \cdot lacO \cdot \tilde{y} \mid RLACT \quad (R10)$$

$$(\tilde{x})^L \rfloor (perm \mid X) \mapsto (perm \cdot \tilde{x})^L \rfloor X \quad (R11)$$

$$LACT \mid (perm \cdot \tilde{x})^L \rfloor X \mapsto (perm \cdot \tilde{x})^L \rfloor (LACT \mid X) \quad (R12)$$

$$betagal \mid LACT \mapsto betagal \mid GLU \mid GAL \quad (R13)$$

CLS modeling examples: the *lac* operon (4)

$$Ecoli ::= (m)^L \rfloor (lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym)$$

Example:

$$Ecoli \mid LACT \mid LACT$$

$$\rightarrow^* (m)^L \rfloor (lacI' \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym \mid repr) \mid LACT \mid LACT$$

$$\rightarrow^* (m)^L \rfloor (lacI' \cdot lacP \cdot RO \cdot lacZ \cdot lacY \cdot lacA \mid polym) \mid LACT \mid LACT$$

$$\rightarrow^* (m)^L \rfloor (lacI' \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym \mid RLACT) \mid LACT$$

$$\rightarrow^* (perm \cdot m)^L \rfloor (lacI' - A \mid betagal \mid transac \mid polym \mid RLACT) \mid LACT$$

$$\rightarrow^* (perm \cdot m)^L \rfloor (lacI' - A \mid betagal \mid transac \mid polym \mid RLACT \mid GLU \mid GAL)$$

Outline of the talk

1 Introduction

2 The Calculus of Looping Sequences (CLS)

- Definition of CLS
- The lac operon in CLS

3 Bisimulations in CLS

- A labeled semantics for CLS
- Bisimulations in CLS
- Bisimulations applied to the CLS model of the lac operon

4 Stochastic CLS

Bisimulations

Bisimilarity is widely accepted as the finest extensional behavioral equivalence one may impose on systems.

- Two systems are bisimilar if they can perform step by step the same interactions with the environment.
- Properties of a system can be verified by assessing the bisimilarity with a system known to enjoy them.

Bisimilarities need semantics based on labeled transition relations capturing the potential interactions with the environment.

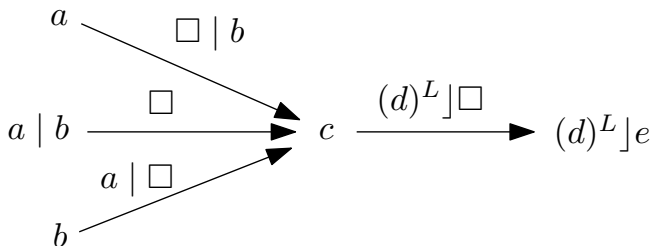
- In process calculi, transitions are usually labeled with actions.
- In CLS labels are contexts in which rules can be applied.

Labeled semantics

The idea: There is a (labeled) transition between terms T and T' if there exists a context C such that a rewrite rule can be applied to $C[T]$ with T' as result.

- C is used as transition label
- C must not provide the whole left hand side of the applied rewrite rule

An example: Let $\mathcal{R} = \{ a \mid b \mapsto c, (d)^L \rfloor c \mapsto (d)^L \rfloor e \}$



Labeled semantics

Contexts \mathcal{C} are given by the following grammar:

$$\mathcal{C} ::= \square \mid \mathcal{C} \mid T \mid T \mid \mathcal{C} \mid (S)^L \mid \mathcal{C}$$

where $T \in \mathcal{T}$ and $S \in \mathcal{S}$. Context \square is called the *empty context*.

Given a set of rewrite rules $\mathcal{R} \subseteq \mathfrak{R}$, the **labeled semantics** of CLS is the labeled transition system given by the following inference rules:

$$\begin{array}{c} \text{(rule_appl)} \quad \frac{P \mapsto P' \in \mathcal{R} \quad \mathcal{C}[T''] \equiv P\sigma \quad T'' \neq \epsilon \quad \sigma \in \Sigma \quad \mathcal{C} \in \mathcal{C}}{T'' \xrightarrow{\mathcal{C}} P'\sigma} \\ \\ \text{(cont)} \quad \frac{T \xrightarrow{\square} T'}{(S)^L \mid T \xrightarrow{\square} (S)^L \mid T'} \qquad \text{(par)} \quad \frac{T \xrightarrow{\mathcal{C}} T' \quad \mathcal{C} \in \mathcal{C}_P}{T \mid T'' \xrightarrow{\mathcal{C}} T' \mid T''} \end{array}$$

where \mathcal{C}_P are contexts that do not include $(S)^L \mid \mathcal{C}$ and the dual version of the *(par)* rule is omitted.

Bisimulations in CLS (1)

A binary relation R on terms is a **strong bisimulation** if, given T_1, T_2 such that $T_1 R T_2$, the two following conditions hold:

- $T_1 \xrightarrow{C} T'_1 \implies \exists T'_2$ s.t. $T_2 \xrightarrow{C} T'_2$ and $T'_1 R T'_2$
- $T_2 \xrightarrow{C} T'_2 \implies \exists T'_1$ s.t. $T_1 \xrightarrow{C} T'_1$ and $T'_2 R T'_1$.

The *strong bisimilarity* \sim is the largest of such relations.

A binary relation R on terms is a **weak bisimulation** if, given T_1, T_2 such that $T_1 R T_2$, the two following conditions hold:

- $T_1 \xrightarrow{C} T'_1 \implies \exists T'_2$ s.t. $T_2 \xRightarrow{C} T'_2$ and $T'_1 R T'_2$
- $T_2 \xrightarrow{C} T'_2 \implies \exists T'_1$ s.t. $T_1 \xRightarrow{C} T'_1$ and $T'_2 R T'_1$.

The *weak bisimilarity* \approx is the largest of such relations.

Theorem: Strong and weak bisimilarities are congruences.

Bisimulations in CLS (2)

Consider the following set of rewrite rules:

$$\mathcal{R} = \{ a \mid b \mapsto c, \quad d \mid b \mapsto e, \quad e \mapsto e, \quad c \mapsto e, \quad f \mapsto a \}$$

We have that $a \sim d$, because

$$a \xrightarrow{\square \mid b} c \xrightarrow{\square} e \xrightarrow{\square} e \xrightarrow{\square} \dots$$

$$d \xrightarrow{\square \mid b} e \xrightarrow{\square} e \xrightarrow{\square} \dots$$

and $f \approx d$, because

$$f \xrightarrow{\square} a \xrightarrow{\square \mid b} c \xrightarrow{\square} e \xrightarrow{\square} e \xrightarrow{\square} \dots$$

On the other hand, $f \not\approx e$ and $f \not\approx e$.

$$e \xrightarrow{\square} e \xrightarrow{\square} e \xrightarrow{\square} \dots$$

Bisimulations in CLS (3)

Let us consider systems (T, \mathcal{R}) ...

A binary relation R is a **strong bisimulation on systems** if, given (T_1, \mathcal{R}_1) and (T_2, \mathcal{R}_2) such that $(T_1, \mathcal{R}_1)R(T_2, \mathcal{R}_2)$:

- $\mathcal{R}_1 : T_1 \xrightarrow{C} T'_1 \implies \exists T'_2$ s.t. $\mathcal{R}_2 : T_2 \xrightarrow{C} T'_2$ and $(T'_1, \mathcal{R}_1)R(T'_2, \mathcal{R}_2)$
- $\mathcal{R}_2 : T_2 \xrightarrow{C} T'_2 \implies \exists T'_1$ s.t. $\mathcal{R}_1 : T_1 \xrightarrow{C} T'_1$ and $(T_2, \mathcal{R}_2)R(T'_1, \mathcal{R}_1)$.

The *strong bisimilarity on systems* \sim is the largest of such relations.

A binary relation R is a **weak bisimulation on systems** if, given (T_1, \mathcal{R}_1) and (T_2, \mathcal{R}_2) such that $(T_1, \mathcal{R}_1)R(T_2, \mathcal{R}_2)$:

- $\mathcal{R}_1 : T_1 \xrightarrow{C} T'_1 \implies \exists T'_2$ s.t. $\mathcal{R}_2 : T_2 \xRightarrow{C} T'_2$ and $(T'_1, \mathcal{R}_1)R(T'_2, \mathcal{R}_2)$
- $\mathcal{R}_2 : T_2 \xrightarrow{C} T'_2 \implies \exists T'_1$ s.t. $\mathcal{R}_1 : T_1 \xRightarrow{C} T'_1$ and $(T'_2, \mathcal{R}_2)R(T'_1, \mathcal{R}_1)$

The *weak bisimilarity on systems* \approx is the largest of such relations.

Strong and weak bisimilarities on systems are NOT congruences.

Bisimulations in CLS (4)

Consider the following sets of rewrite rules

$$\mathcal{R}_1 = \{a \mid b \mapsto c\} \quad \mathcal{R}_2 = \{a \mid d \mapsto c, b \mid e \mapsto c\}$$

We have that $\langle a, \mathcal{R}_1 \rangle \approx \langle e, \mathcal{R}_2 \rangle$ because

$$\mathcal{R}_1 : a \xrightarrow{\square|b} c \quad \mathcal{R}_2 : e \xrightarrow{\square|b} c$$

and $\langle b, \mathcal{R}_1 \rangle \approx \langle d, \mathcal{R}_2 \rangle$, because

$$\mathcal{R}_1 : b \xrightarrow{\square|a} c \quad \mathcal{R}_2 : d \xrightarrow{\square|a} c$$

but $\langle a \mid b, \mathcal{R}_1 \rangle \not\approx \langle e \mid d, \mathcal{R}_2 \rangle$, because

$$\mathcal{R}_1 : a \mid b \xrightarrow{\square} c \quad \mathcal{R}_2 : e \mid d \not\xrightarrow{\square}$$

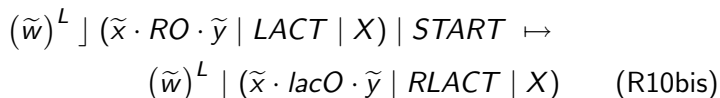
Applying bisimulations to the *lac* operon (1)

By using the weak bisimilarity on systems we can prove that from the state in which the repressor is bound to the DNA we can reach a state in which the enzymes are synthesized only if lactose appears in the environment.

We replace rule



with



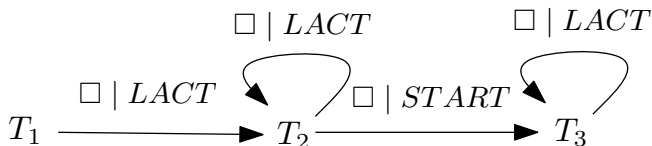
Applying bisimulations to the *lac* operon (2)

The obtained model is weakly bisimilar to (T_1, \mathcal{R}) where \mathcal{R} is

$$T_1 \mid LACT \mapsto T_2 \quad (\text{R1}') \qquad T_2 \mid START \mapsto T_3 \quad (\text{R3}')$$

$$T_2 \mid LACT \mapsto T_2 \quad (\text{R2}') \qquad T_3 \mid LACT \mapsto T_3 \quad (\text{R4}')$$

that is a system satisfying the wanted property.



Outline of the talk

1 Introduction

2 The Calculus of Looping Sequences (CLS)

- Definition of CLS
- The lac operon in CLS

3 Bisimulations in CLS

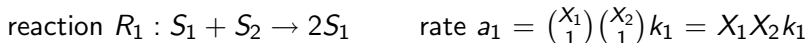
- A labeled semantics for CLS
- Bisimulations in CLS
- Bisimulations applied to the CLS model of the lac operon

4 Stochastic CLS

Background: Gillespie's simulation algorithm

- represents a chemical solution as a multiset of molecules
- each chemical reaction is associated with a kinetic constant
- computes the reaction rate a_μ by multiplying the kinetic constant by the number of possible combinations of reactants

Example: chemical solution with X_1 molecules S_1 and X_2 molecules S_2



Given a set of reactions $\{R_1, \dots, R_M\}$ and a current time t

- The time $t + \tau$ at which the next reaction will occur is randomly chosen with τ exponentially distributed with parameter $\sum_{\nu=1}^M a_\nu$;
- The reaction R_μ that has to occur at time $t + \tau$ is randomly chosen with probability $\frac{a_\mu}{\sum_{\nu=1}^M a_\nu}$.

At each step t is incremented by τ and the chemical solution is updated.

Stochastic CLS

Stochastic CLS incorporates Gillespie's stochastic framework into the semantics of CLS

- Rewrite rules are enriched with kinetic constants

What is a reactant in Stochastic CLS?

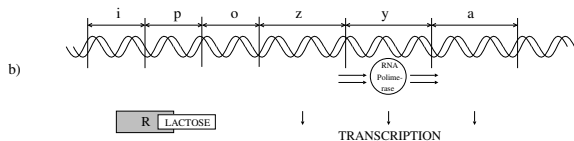
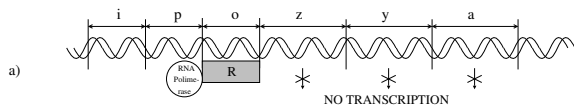
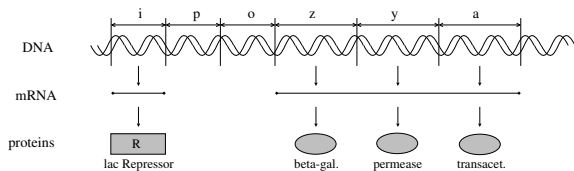
- A *reactant combination* is an occurrence (up to \equiv) of a left hand side of a rewrite rule

Example: The application rate of $a \mid b \xrightarrow{k} c$ to $a \mid a \mid b \mid b$ is $6k$

Example: The application rate of $(a \cdot \tilde{x})^L \mid (b \mid X) \xrightarrow{k} (c \cdot \tilde{x})^L \mid X$ to $(a \cdot a \cdot a)^L \mid (b \mid b) \mid (a \cdot a)^L \mid b$ is

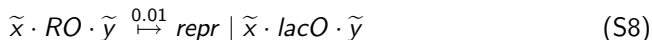
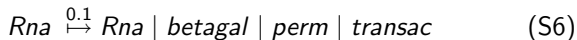
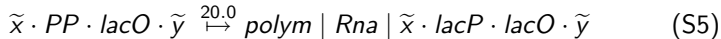
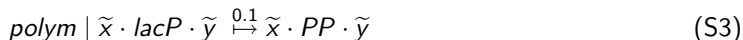
- $6k$, with $(c \cdot a \cdot a)^L \mid b \mid (a \cdot a)^L \mid b$ as result
- $+ 2k$, with $(a \cdot a \cdot a)^L \mid (b \mid b) \mid (c \cdot a)^L \mid \epsilon$ as result
- $= 8k$

A Stochastic CLS model of the *lac* operon (1)



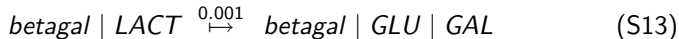
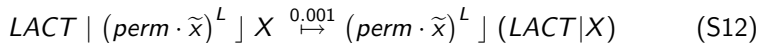
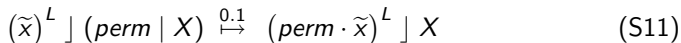
A Stochastic CLS model of the *lac* operon (2)

Transcription of DNA, binding of lac Repressor to gene *o*, and interaction between lactose and lac Repressor:

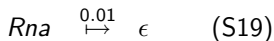
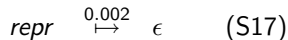
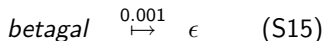


A Stochastic CLS model of the *lac* operon (3)

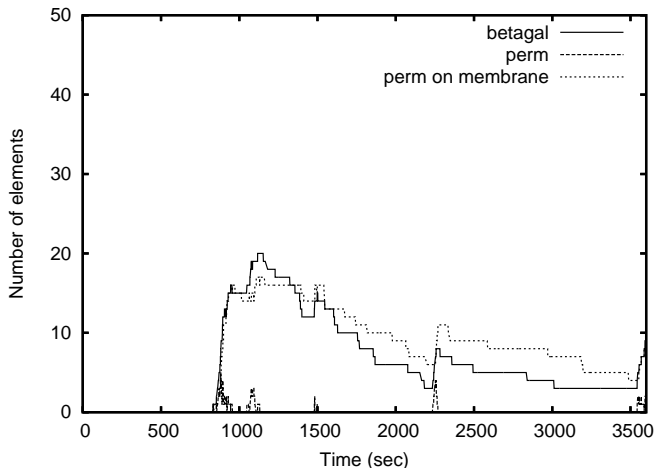
The behaviour of the three enzymes for lactose degradation:



Degradation of all the proteins and mRNA involved in the process:

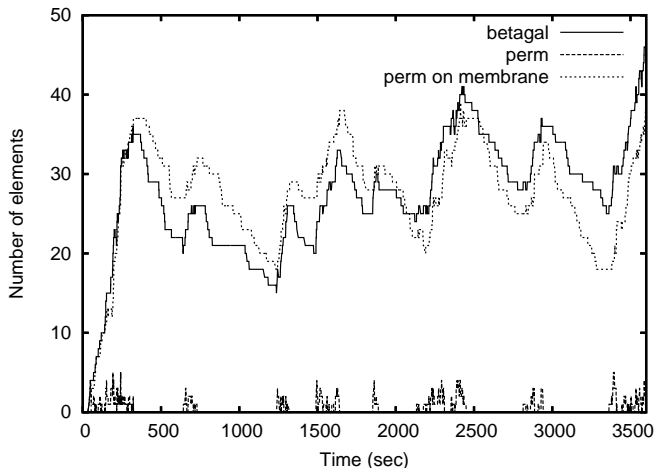


Simulation results (1)



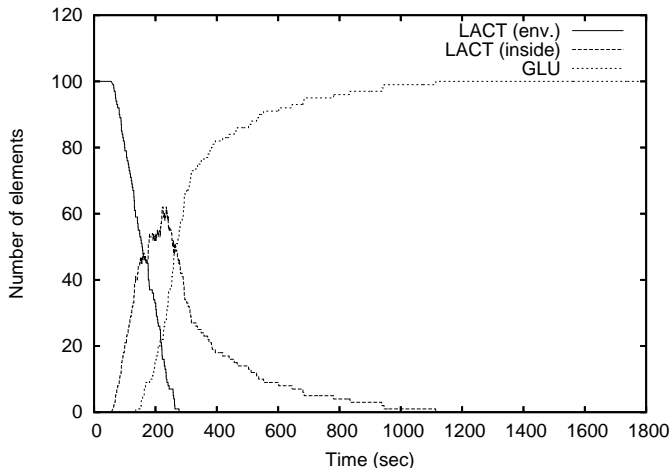
Production of enzymes in the absence of lactose
 $(m)^L \rfloor (lacl - A \mid 30 \times polym \mid 100 \times repr)$

Simulation results (2)



Production of enzymes in the presence of lactose
 $100 \times LACT \mid (m)^L \mid (lacI - A \mid 30 \times polym \mid 100 \times repr)$

Simulation results (3)



Degradation of lactose into glucose

