A Rewrite–based Formalism for Describing Biological Systems

Roberto Barbuti   Andrea Maggiolo Schettini
Paolo Milazzo   Angelo Troina

Dipartimento di Informatica, Università di Pisa, Italy

Como – October 18, 2006

Introduction

Formal models for systems of interactive components can be easily used or adapted for the modelling of biological phenomena

- Examples: Petri Nets, π–calculus, Mobile Ambients

The modelling of biological systems allows:

1. the development of simulators
2. the verification of properties

In this talk we present:

1. the Calculus of Looping Sequences (CLS): a formalism to describe biochemical systems in cells
2. bisimulation relations for CLS
3. the CLS model of a gene regulation process in E. Coli

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:

$$
\begin{array}{rcl}
T & ::= & S \quad \big| \quad \left(S\right)^L \rfloor T \quad \big| \quad T \mid T \\
S & ::= & \epsilon \quad \big| \quad a \quad \big| \quad S \cdot S
\end{array}
$$

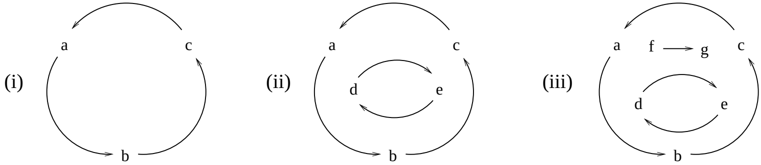
where a is a generic element of $\mathcal{E}$, and ϵ is the empty sequence.

The operators are:

$S \cdot S$	:	Sequencing
$\left(S\right)^L$	:	Looping (S is closed and it can rotate)
$T_1 \rfloor T_2$	:	Containment (T_1 contains T_2)
$T \mid T$	:	Parallel composition (juxtaposition)

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

Example of Terms



$$(i) \qquad (a \cdot b \cdot c)^L$$

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

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

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:

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

$$S_1 \equiv_S S_2 \text{ implies } S_1 \equiv_T S_2 \text{ and } \left(S_1\right)^L \rfloor\, T \equiv_T \left(S_2\right)^L \rfloor\, T$$
$$T_1 \mid T_2 \equiv_T T_2 \mid T_1 \qquad T_1 \mid (T_2 \mid T_3) \equiv_T (T_1 \mid T_2) \mid T_3$$
$$T \mid \epsilon \equiv_T T \quad \left(\epsilon\right)^L \rfloor\, \epsilon \equiv_T \epsilon \quad \left(S_1 \cdot S_2\right)^L \rfloor\, T \equiv_T \left(S_2 \cdot S_1\right)^L \rfloor\, T$$

We write $\equiv$ for $\equiv_T$.

Dinamics of the Calculus (1)

Let TV be a set of term variables $(X, Y, Z, \ldots)$, SV be a set of sequence variables $(\widetilde{x}, \widetilde{y}, \widetilde{z}, \ldots)$, EV be a set of element variables $(x, y, z, \ldots)$, and $\mathcal{V} = TV \cup SV \cup EV$. Let $\mathcal{T}_\mathcal{V}$ be the set of terms which may contain variables.

- An *istantiation* is a function $\sigma : V \rightarrow \mathcal{T}$ such that $\sigma(X)$ is a term, $\sigma(\widetilde{x})$ is a sequence and $\sigma(x)$ is a single element.
- $T\sigma$ denotes the term obtained by replacing any variable in T with the corrseponding term, sequence or element.
- Σ denotes the set of all possible instantiations.

A **Rewrite Rule** is a pair (T, T'), denoted $T \mapsto T'$, where:

- $T, T' \in \mathcal{T}_\mathcal{V}$
- variables in T' are a subset of those in T

Dinamics of the Calculus (2)

A rule $T \mapsto T'$ can be applied to all terms $T\sigma$ s.t. $\sigma \in \Sigma$

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 $a \cdot c \cdot c \cdot a$

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

$$\frac{T \mapsto T' \in \mathcal{R} \qquad T\sigma \not\equiv \epsilon \qquad \sigma \in \Sigma}{T\sigma \rightarrow T'\sigma}$$

and closed under structural congruence and all the operators

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 contexts in which rules can be applied.

Labeled Semantics (1)

Assume T' and T'' as follows:

$$T' \equiv T_1 \mid \ldots \mid T_n \mid T'_1 \mid \ldots \mid T'_m$$
$$T'' \equiv T_1 \mid \ldots \mid T_n \mid T''_1 \mid \ldots \mid T''_o$$

with T_i, T'_j, T''_k non–parallel and non–empty. We define

$$T' \sqcap T'' \equiv T_1 \mid \ldots \mid T_n$$

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

$$\mathcal{C} ::= \Box \quad \mid \quad \mathcal{C} \mid T \quad \mid \quad T \mid \mathcal{C} \quad \mid \quad (S)^L \rfloor \mathcal{C}$$

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

Parallel Contexts $\mathcal{C}_P$ are given by the following grammar (where $T \in \mathcal{T}$):

$$\mathcal{C}_P ::= \Box \quad \mid \quad \mathcal{C}_P \mid T \quad \mid \quad T \mid \mathcal{C}_P.$$

$C[T]$ is context application and $C[C']$ is context composition.

Labeled Semantics (2)

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

$$\text{(rule\_appl)}\ \frac{P_1 \mapsto P_2 \in \mathcal{R} \quad C[T''] \equiv P_1\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} P_2\sigma}$$

$$\text{(cont)}\ \frac{T \xrightarrow{\Box} T'}{(S)^L \rfloor T \xrightarrow{\Box} (S)^L \rfloor T'} \qquad \text{(par)}\ \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \sqcap T'' \equiv \epsilon}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

where the dual version of the *(par)* rule is omitted.

Rule (rule_appl) describes the (potential) application of a rule.

- $T'' \not\equiv \epsilon$ in the premise implies that C cannot provide completely the left hand side of the rewrite rule.
- Example: let $R = a \mid b \mapsto c$, we have $a \xrightarrow{\Box \mid b} c$, but $\epsilon \stackrel{a|b}{\not\longrightarrow}$.

Labeled Semantics (3)

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

$$\text{(rule\_appl)}\ \frac{P_1 \mapsto P_2 \in \mathcal{R} \quad C[T''] \equiv P_1\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} P_2\sigma}$$

$$\text{(cont)}\ \frac{T \xrightarrow{\square} T'}{\left(S\right)^L \rfloor T \xrightarrow{\square} \left(S\right)^L \rfloor T'} \qquad \text{(par)}\ \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \sqcap T'' \equiv \epsilon}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

where the dual version of the *(par)* rule is omitted.

Rule (cont) propagates $\square$–labeled transitions from the inside to the outside of a looping sequence.

- Transition labeled with a non–empty context cannot be propagated.
- Example: let $R = a \mid b \mapsto c$, we have $a \xrightarrow{\square \mid b} c$, but $\left(d\right)^L \rfloor a \overset{\square|b}{\not\longrightarrow}$.

Labeled Semantics (4)

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

$$\text{(rule\_appl)}\ \frac{P_1 \mapsto P_2 \in \mathcal{R} \quad C[T''] \equiv P_1\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} P_2\sigma}$$

$$\text{(cont)}\ \frac{T \xrightarrow{\square} T'}{(S)^L \rfloor T \xrightarrow{\square} (S)^L \rfloor T'} \qquad \text{(par)}\ \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \sqcap T'' \equiv \epsilon}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

where the dual version of the *(par)* rule is omitted.

Rule (par) propagates transitions labeled with parallel contexts in parallel components.

- Example: let $R = (a)^L \rfloor b \mapsto c$, we have $b \xrightarrow{(a)^L \rfloor \square} c$, but $b \mid d \not\xrightarrow{(a)^L \rfloor \square}$ because R cannot be applied $(a)^L \rfloor (b \mid d)$
- $C[\epsilon] \sqcap T'' \equiv \epsilon$ ensures that C is the least necessary to apply the rule.

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. $T_2 \stackrel{C}{\Longrightarrow} T_2'$ and $T_1' R T_2'$
- $T_2 \xrightarrow{C} T_2' \implies \exists T_1'$ s.t. $T_1 \stackrel{C}{\Longrightarrow} T_1'$ 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)

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 $(R_2, T_2') R (\mathcal{R}_1, T_1')$.

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

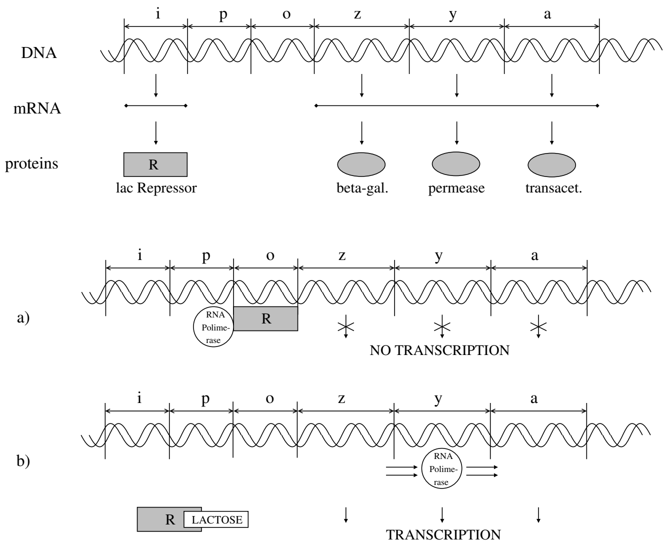
A binary relation R is a **weak bisimulation on systems** if, given $(T_1, \mathcal{R}_1)$ and $(T_2, \mathcal{R}_2)$ such that $(T_1, \mathcal{R}_1) R (T_2, \mathcal{R}_2)$:

- $\mathcal{R}_1 : T_1 \xrightarrow{C} T_1' \implies \exists T_2'$ s.t. $\mathcal{R}_2 : T_2 \stackrel{C}{\Longrightarrow} T_2'$ 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. $\mathcal{R}_1 : T_1 \stackrel{C}{\Longrightarrow} T_1'$ 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.

The Lactose Operon in E.coli (1)



The Lactose Operon in E.coli (2)

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

Rules for DNA transcription/translation:

$$\begin{aligned}
lacI \cdot \widetilde{x} &\longrightarrow lacI' \cdot \widetilde{x} \mid repr && \text{(R1)} \\
polym \mid \widetilde{x} \cdot lacP \cdot \widetilde{y} &\longrightarrow \widetilde{x} \cdot PP \cdot \widetilde{y} && \text{(R2)} \\
\widetilde{x} \cdot PP \cdot lacO \cdot \widetilde{y} &\longrightarrow \widetilde{x} \cdot lacP \cdot PO \cdot \widetilde{y} && \text{(R3)} \\
\widetilde{x} \cdot PO \cdot lacZ \cdot \widetilde{y} &\longrightarrow \widetilde{x} \cdot lacO \cdot PZ \cdot \widetilde{y} && \text{(R4)} \\
\widetilde{x} \cdot PZ \cdot lacY \cdot \widetilde{y} &\longrightarrow \widetilde{x} \cdot lacZ \cdot PY \cdot \widetilde{y} \mid betagal && \text{(R5)} \\
\widetilde{x} \cdot PY \cdot lacA &\longrightarrow \widetilde{x} \cdot lacY \cdot PA \mid perm && \text{(R6)} \\
\widetilde{x} \cdot PA &\longrightarrow \widetilde{x} \cdot lacA \mid transac \mid polym && \text{(R7)}
\end{aligned}$$

The Lactose Operon in E.coli (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:

$$repr \mid \widetilde{x} \cdot lacO \cdot \widetilde{y} \longrightarrow \widetilde{x} \cdot RO \cdot \widetilde{y} \quad \text{(R8)}$$

$$LACT \mid (m \cdot \widetilde{x})^L \rfloor X \longrightarrow (m \cdot \widetilde{x})^L \rfloor (X \mid LACT) \quad \text{(R9)}$$

$$\widetilde{x} \cdot RO \cdot \widetilde{y} \mid LACT \longrightarrow \widetilde{x} \cdot lacO \cdot \widetilde{y} \mid RLACT \quad \text{(R10)}$$

$$(\widetilde{x})^L \rfloor (perm \mid X) \longrightarrow (perm \cdot \widetilde{x})^L \rfloor X \quad \text{(R11)}$$

$$LACT \mid (perm \cdot \widetilde{x})^L \rfloor X \longrightarrow (perm \cdot \widetilde{x})^L \rfloor (LACT \mid X) \quad \text{(R12)}$$

$$betagal \mid LACT \longrightarrow betagal \mid GLU \mid GAL \quad \text{(R13)}$$

The Lactose Operon in E.coli (4)

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

Example:

$$
\begin{aligned}
& Ecoli|LACT|LACT \\
\rightarrow^* & (m)^L \rfloor (lacI' \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym \mid repr)|LACT|LACT \\
\rightarrow^* & (m)^L \rfloor (lacI' \cdot lacP \cdot RO \cdot lacZ \cdot lacY \cdot lacA \mid polym)|LACT|LACT \\
\rightarrow^* & (m)^L \rfloor (lacI' \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA|polym|RLACT)|LACT \\
\rightarrow^* & (perm \cdot m)^L \rfloor (lacI'\text{–}A|betagal|transac|polym|RLACT)|LACT \\
\rightarrow^* & (perm \cdot m)^L \rfloor (lacI'\text{–}A|betagal|transac|polym|RLACT|GLU|GAL)
\end{aligned}
$$

Applying Bisimulations (1)

It can be easily proved that

$$lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA$$
$$\approx$$
$$lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid repr$$

and since weak bisimularity is a congruence the former can be replaced by the latter in the model.

Applying Bisimulations (2)

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

$$\widetilde{x} \cdot RO \cdot \widetilde{y} \mid LACT \longrightarrow \widetilde{x} \cdot lacO \cdot \widetilde{y} \mid RLACT \qquad \text{(R10)}$$

with

$$(\widetilde{w})^L \rfloor (\widetilde{x} \cdot RO \cdot \widetilde{y} \mid LACT \mid X) \mid START \longrightarrow$$
$$(\widetilde{w})^L \rfloor (\widetilde{x} \cdot lacO \cdot \widetilde{y} \mid RLACT \mid X) \qquad \text{(R10bis)}$$

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

$$T_1 \mid LACT \longrightarrow T_2 \quad \text{(R1')} \qquad T_2 \mid START \longrightarrow T_3 \quad \text{(R3')}$$
$$T_2 \mid LACT \longrightarrow T_2 \quad \text{(R2')} \qquad T_3 \mid LACT \longrightarrow T_3 \quad \text{(R4')}$$

that is a system satisfying the property.

Conclusions

- We have introduced the Calculus of Looping Sequences (CLS): a formalism which can be used for modelling systems of Cell Biology.
- We have defined a labeled semantics and bisimulation relations for CLS
- We have modeled an example of gene regulation process in E.Coli and used bisimulations on the model

Further developments:

- The simplest of Cardelli's Brane Calculi has been encoded into CLS. The encoding preserves bisimulations.
- Quantitative extension: Stochastic CLS (Angelo's talk next Friday)
- Spatial extension: Cellular CLS (under development)

References

R. Barbuti, A. Maggiolo Schettini, P. Milazzo and A. Troina.
"A Calculus of Looping Sequences for Modelling Microbiological Systems"
Fundamenta Informaticae, volume 72, number 1–3, pages 21–35, 2006.

R. Barbuti, A. Maggiolo Schettini, P. Milazzo and A. Troina.
"Bisimulation Congruences in the Calculus of Looping Sequences".
International Colloquium on Theoretical Aspects of Computing (ICTAC'06),
LNCS 4281, pages 93–107, Springer, 2006.