The Calculus of Looping Sequences

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

We defined the Calculus of Looping Sequences (CLS): a formalism to describe biochemical systems in cells

In this talk:

1. we recall the definition of CLS
2. we present bisimulation relations for CLS
3. we show 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:

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

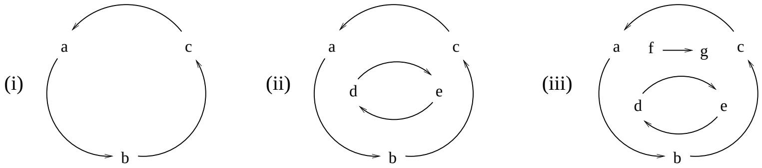
where a is a generic element of $\mathcal{E}$, and ϵ is the empty sequence.

The operators are:

$S \cdot S$	:	Sequencing
$(S)^L$	:	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 $(S)^L \rfloor T$.

Example of Terms



$$(i) \qquad \left(a \cdot b \cdot c\right)^L \rfloor \epsilon$$

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

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

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:

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

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

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

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

Dinamics of the Calculus (1)

Let $\mathcal{T}_\mathcal{V}$ be the set of terms which may contain variables of three kinds:

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

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

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

A rule $T \mapsto T'$ can be applied to all terms $T\sigma$.

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

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 (1)

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

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

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:

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

where $T \in \mathcal{T}$.

$C[T]$ 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:

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

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

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:

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

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

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 $\left(d\right)^L \rfloor a \not\xrightarrow{\square \mid b}$.

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:

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

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

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 $(a)^L \rfloor (b \mid d)$

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)

Consider the following set of rewrite rules:

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

We have that $a \sim d$, because

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

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

and $f \approx d$, because

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

On the other hand, $f \not\sim e$ 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 $(R_2, T_2') R (\mathcal{R}_1, T_1')$.

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.

Bisimulations in CLS (4)

Consider the following sets of rewrite rules

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

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

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

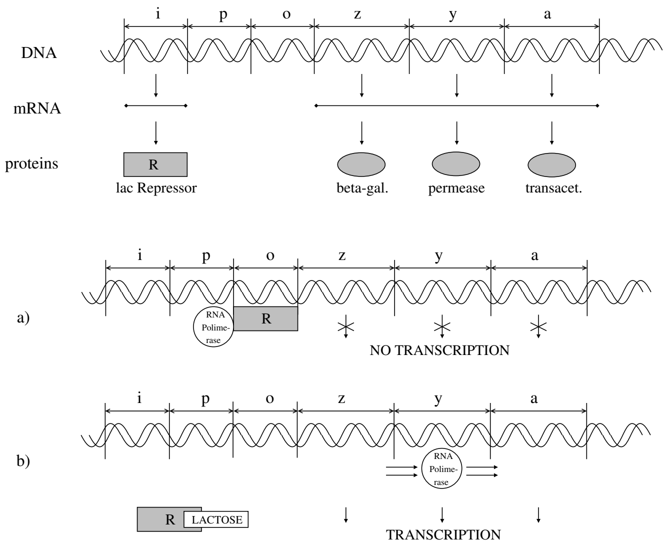
and $\langle b, \mathcal{R}_1 \rangle \approx \langle d, \mathcal{R}_2 \rangle$, because

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

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

$$\mathcal{R}_1 : a \mid b \xrightarrow{\square} c \qquad \mathcal{R}_2 : c \mid d \stackrel{\square}{\not\rightarrow}$$

The Lactose Operon in E.coli (1)



The Lactose Operon in E.coli (2)

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

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)

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

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:

$$
\begin{aligned}
repr \mid \widetilde{x} \cdot lacO \cdot \widetilde{y} &\longrightarrow \widetilde{x} \cdot RO \cdot \widetilde{y} && \text{(R8)}\\
LACT \mid \big(m \cdot \widetilde{x}\big)^{L} \rfloor X &\longrightarrow \big(m \cdot \widetilde{x}\big)^{L} \rfloor (X \mid LACT) && \text{(R9)}\\
\widetilde{x} \cdot RO \cdot \widetilde{y} \mid LACT &\longrightarrow \widetilde{x} \cdot lacO \cdot \widetilde{y} \mid RLACT && \text{(R10)}\\
\big(\widetilde{x}\big)^{L} \rfloor (perm \mid X) &\longrightarrow \big(perm \cdot \widetilde{x}\big)^{L} \rfloor X && \text{(R11)}\\
LACT \mid \big(perm \cdot \widetilde{x}\big)^{L} \rfloor X &\longrightarrow \big(perm \cdot \widetilde{x}\big)^{L} \rfloor (LACT \mid X) && \text{(R12)}\\
betagal \mid LACT &\longrightarrow betagal \mid GLU \mid GAL && \text{(R13)}
\end{aligned}
$$

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:

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

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

The Calculus of Looping Sequences can be used to describe biological systems

The bisimulation relations we have defined can be used

- to find equivalent reduced models
- to verify properties

If we consider models in which the same set of rewrite rules is used, strong and weak bisimulations are congruences.

We used bisimulations on a model of a real biological phenomenon:

- to find an equivalent reduced model
- to verify a causality property