BIOINFORMÁTICA

Asignatura 400CIS016
Doctorado en Ingeniería

Pontificia Universidad
JAVERIANA
Cali

PROFESORA
Dra. Diana Hermith
PhD en Ingenierías y Ciencias de la Información, Universidad de Siena, Italia
Maestría en Ingenierías y Ciencias de la Computación, Pontificia Universidad Javeriana-Cali, Colombia
Pregrado en Biología Molecular, Universidad del Valle-Cali, Colombia

www.linkedin.com/in/dianahermith
twitter.com/dianahermith
dphermith@javerianacali.edu.co

23-Ene-2017-03-Jun-2017 Miércoles 14:00-18:00 LG-0.5
Módulo Modelamiento Computacional y Simulación:6 Semanas; Semana Lun 23 Enero a la Semana Lunes 27 de Febrero

“Computer science is to biology what mathematics is to physics -Harold Morowitz”

INTRODUCCIÓN AL MODELAMIENTO COMPUTACIONAL

Abstraction is a generic technique that allows the scientist or engineer to focus only on certain features of a system while hiding others.

{Understanding biology from a computational/system perspective}

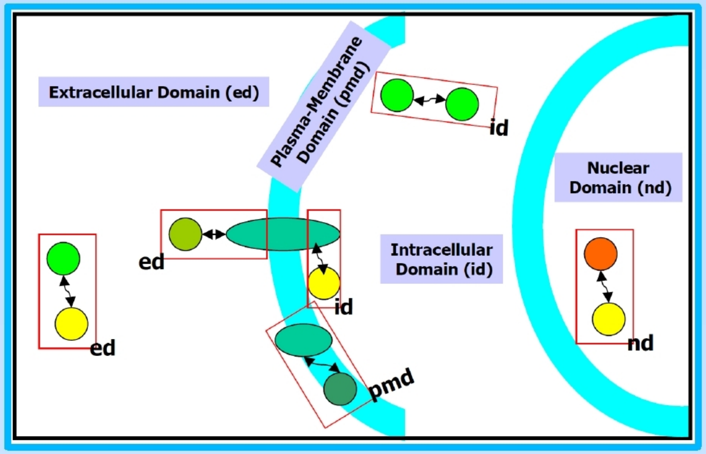
“Computer science is to biology what mathematics is to physics -Harold Morowitz”

Podemos distinguir entre dos objetivos generales y complementarios para la modelización:
reproducir la complejidad mediante las simulaciones computacionales y reducir la complejidad mediante modelos que codifican principios generales.

There has been no comparable effort on formalizing the knowledge known from lab experiments into *biological properties* that can be used to build faithful models. The promise of such a *formal specification* would be to systematically validate and maintain models using automated reasoning tools.

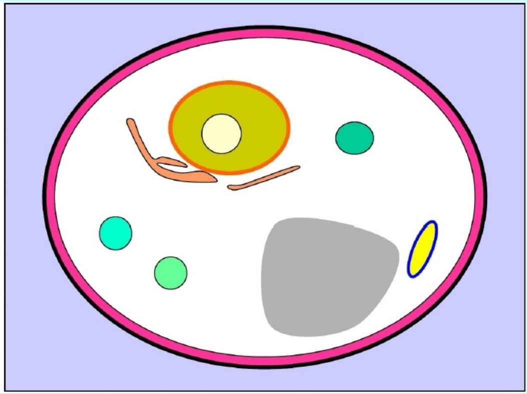
Some examples via a Logic-based approach

Logic-based Modeling


Extracellular Domain (ed)
Plasma-Membrane Domain (pmd)
id
Nuclear Domain (nd)
ed
Intracellular Domain (id)
id
ed
nd
pmd

Basic local interactions and subcellular "locations" in a **signalling pathway**.

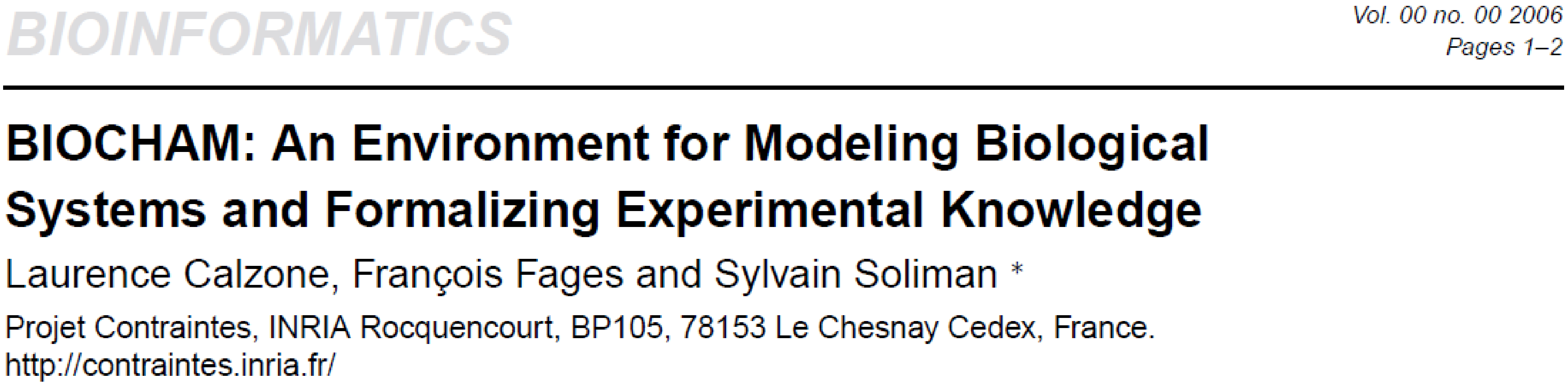
LEGEND					
ed	🟪	zgd	🟩	nld	🟨
pmd	🟥	md	🟨	eed	🟩
cpd	⬛	mod	🟦	ld	🟦
id	⬜	nmd	🟧	erd	🟧
cd	▨	nd	🟨		



Cellular Locations	
ID	**Name**
ed	Extracellular Domain
pmd	Plasma-Membrane Domain
cpd	Cell Projection Domain
id	Intracellular Domain
cd	Cytosolic Domain
zgd	Zymogen-Granule Domain
md	Mitochondrial Domain
mod	Mitochondrial Outer Membrane Domain
nmd	Nuclear Membrane Domain
nd	Nuclear Domain
nld	Nucleolar Domain
eed	Early Endosomal Domain
ld	Lysosomal Domain
erd	Endoplasmic-reticulum Domain

Reactive computation over time: Signaling pathways allow cells to read environmental cues, translate them into intracellular commands, and react with an appropriate response.

Locality: Biochemical interactions take place in fixed locations, which gives order to cell signaling.



BIOINFORMATICS

Vol. 00 no. 00 2006
Pages 1–2

BIOCHAM: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge

Laurence Calzone, François Fages and Sylvain Soliman *

Projet Contraintes, INRIA Rocquencourt, BP105, 78153 Le Chesnay Cedex, France.
http://contraintes.inria.fr/

https://lifeware.inria.fr/biocham/

Some biological properties of interest:

Reachability: whether a particular molecule can be produced from an initial state;

Checkpoints: whether a particular molecule or state is compulsory to reach another state;

Stability: whether the system can (or will) always verify some property.

Computation Tree Logic (CTL) is an extension of classical logic that allows reasoning about an infinite tree of state transitions.

CTL uses operators about branches (non-deterministic choices) and time (state transitions).

Two *path quantifiers* A and E are thus introduced to handle non-determinism:

$A\phi$: meaning that ϕ is true on all branches, and

$E\phi$: meaning that ϕ is true on at least one branch.

Five *time operators* F, G, X, U, and W are thus introduced to handle state transitions:

$F\phi$: meaning that ϕ is eventually true,
$G\phi$: meaning that ϕ is always true,
$X\phi$: meaning that ϕ is true at the next transition,
$\phi U \psi$: meaning that ϕ is always true until ψ becomes true, and
$\phi \mathrm{W} \psi$: meaning that ϕ is either always true or until and when ψ becomes true.

CTL is sufficiently expressive for formalizing *qualitative* biological properties, such as:

– *Reachability:* where reachable(P) stands for EF(P);

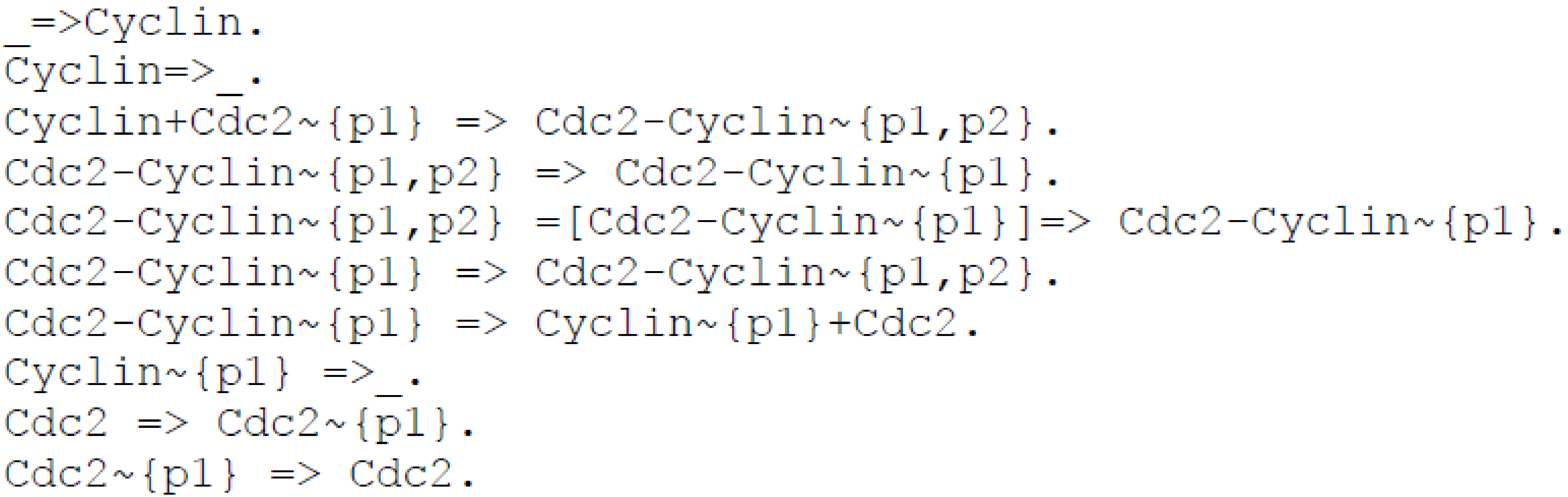
– *Steady states:* where steady(P) stands for EG(P);

– *Stable states:* where stable(P) stands for AG(P);

– *Checkpoints:* where checkpoint(Q,P) stands for ¬E(¬Q U P);

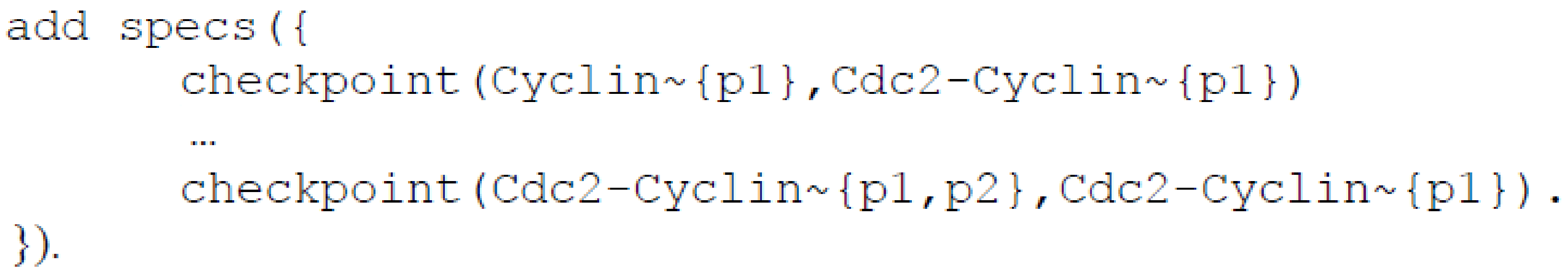
An Example

Define the biochemical reaction rules:



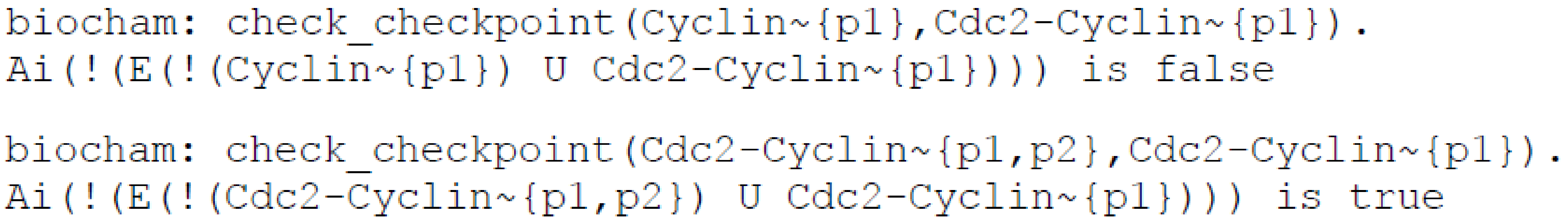
```
_=>Cyclin.
Cyclin=>_.
Cyclin+Cdc2~{p1} => Cdc2-Cyclin~{p1,p2}.
Cdc2-Cyclin~{p1,p2} => Cdc2-Cyclin~{p1}.
Cdc2-Cyclin~{p1,p2} =[Cdc2-Cyclin~{p1}]=> Cdc2-Cyclin~{p1}.
Cdc2-Cyclin~{p1} => Cdc2-Cyclin~{p1,p2}.
Cdc2-Cyclin~{p1} => Cyclin~{p1}+Cdc2.
Cyclin~{p1} =>_.
Cdc2 => Cdc2~{p1}.
Cdc2~{p1} => Cdc2.
```

Add a temporal logic specification to the model:



```
add specs({
      checkpoint(Cyclin~{p1},Cdc2-Cyclin~{p1})
      …
      checkpoint(Cdc2-Cyclin~{p1,p2},Cdc2-Cyclin~{p1}).
}).
```

Boolean Analysis-Verification of formal biological properties:



```
biocham: check_checkpoint(Cyclin~{p1},Cdc2-Cyclin~{p1}).
Ai(!(E(!(Cyclin~{p1}) U Cdc2-Cyclin~{p1}))) is false

biocham: check_checkpoint(Cdc2-Cyclin~{p1,p2},Cdc2-Cyclin~{p1}).
Ai(!(E(!(Cdc2-Cyclin~{p1,p2}) U Cdc2-Cyclin~{p1}))) is true
```

Numerical Temporal Properties

Temporal *quantitative* properties about concentrations and their derivatives can be formalized as well in *Linear Time Logic with numerical constraints over the reals*, noted LTL(R).

The version of LTL(R) with arithmetic constraints, considers first-order atomic formulae with equality, inequality and arithmetic operators ranging over real values of concentrations and of their derivatives.

LTL(R) is sufficiently expressive for formalizing *quantitative* biological properties, such as:

–F([A]>10): expresses that the concentration of A eventually gets above the threshold value 10.

-G([A]+[B]<[C]): expresses that the concentration of C is always greater than the sum of the concentrations of A and B.

-Oscillation (oscil(M,K)): defined as a change of sign of the derivative of M at least K times:
F((d[M]/dt > 0) & F((d[M]/dt < 0) & F((d[M]/dt > 0)...)))

Constraints on the periods of oscillations can be expressed with a formula such as *period(A,75)* defined as:

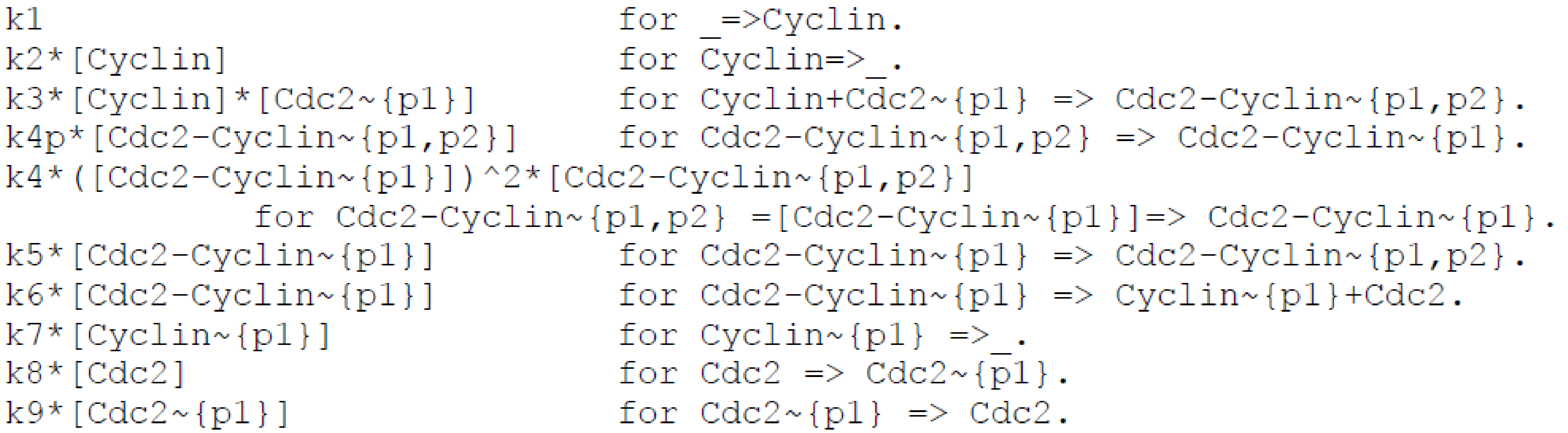
$\exists t \exists \nu\ F(\mathit{Time}=t \ \&\ [A]=\nu \ \&\ d([A])/dt > 0 \ \&\ X(d([A])/dt < 0) \ \&\ F(\mathit{Time}= t +75 \ \&\ [A]=\nu \ \ \&\ d([A])/dt > 0 \ \&\ X(d([A])/dt < 0)))$ where Time is the time variable.

G(([RAF-RAFK] >= [RAF~{p1}]) U (d([RAF])/dt < 0.3)):

expresses that all along the simulation trace, the concentration of the RAF-RAFK complex is greater than that of phosphorylated RAF, until the derivative of the concentration of RAF becomes lower than 0.3.

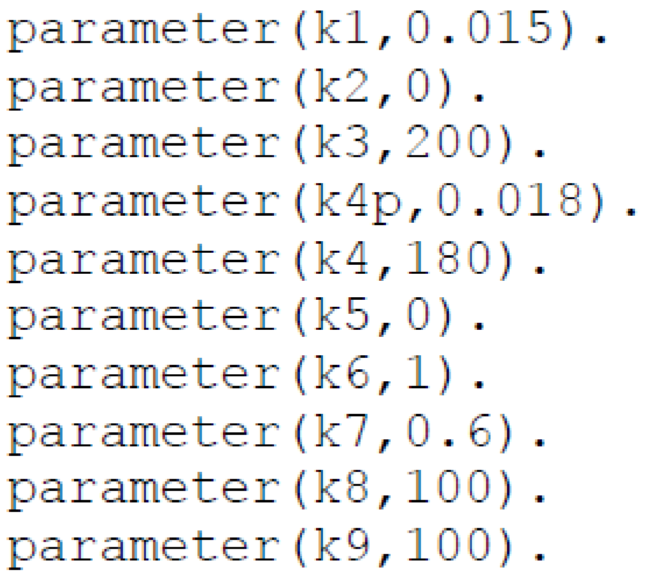
oscil(cycB,3): expresses the fact that, along the trace, the concentration of cycB goes up and down twice.

Assign kinetics to the biochemical reaction rules:



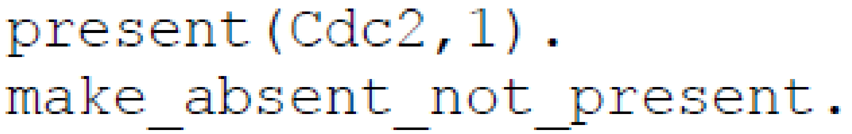
```
k1                            for _=>Cyclin.
k2*[Cyclin]                   for Cyclin=>_.
k3*[Cyclin]*[Cdc2~{p1}]       for Cyclin+Cdc2~{p1} => Cdc2-Cyclin~{p1,p2}.
k4p*[Cdc2-Cyclin~{p1,p2}]     for Cdc2-Cyclin~{p1,p2} => Cdc2-Cyclin~{p1}.
k4*([Cdc2-Cyclin~{p1}])^2*[Cdc2-Cyclin~{p1,p2}]
            for Cdc2-Cyclin~{p1,p2} =[Cdc2-Cyclin~{p1}]=> Cdc2-Cyclin~{p1}.
k5*[Cdc2-Cyclin~{p1}]         for Cdc2-Cyclin~{p1} => Cdc2-Cyclin~{p1,p2}.
k6*[Cdc2-Cyclin~{p1}]         for Cdc2-Cyclin~{p1} => Cyclin~{p1}+Cdc2.
k7*[Cyclin~{p1}]              for Cyclin~{p1} =>_.
k8*[Cdc2]                     for Cdc2 => Cdc2~{p1}.
k9*[Cdc2~{p1}]                for Cdc2~{p1} => Cdc2.
```

Define the parameter values:



```
parameter(k1,0.015).
parameter(k2,0).
parameter(k3,200).
parameter(k4p,0.018).
parameter(k4,180).
parameter(k5,0).
parameter(k6,1).
parameter(k7,0.6).
parameter(k8,100).
parameter(k9,100).
```

Define the initial conditions:



```
present(Cdc2,1).
make_absent_not_present.
```

Numerical Analysis:

-Check the rules:

Similar to the boolean analysis, it is possible to query the numerical simulation.

trace_check(F([A]>0.15)

checks that the concentration of the molecule A reaches a value above 0.15.

-Find parameter values:

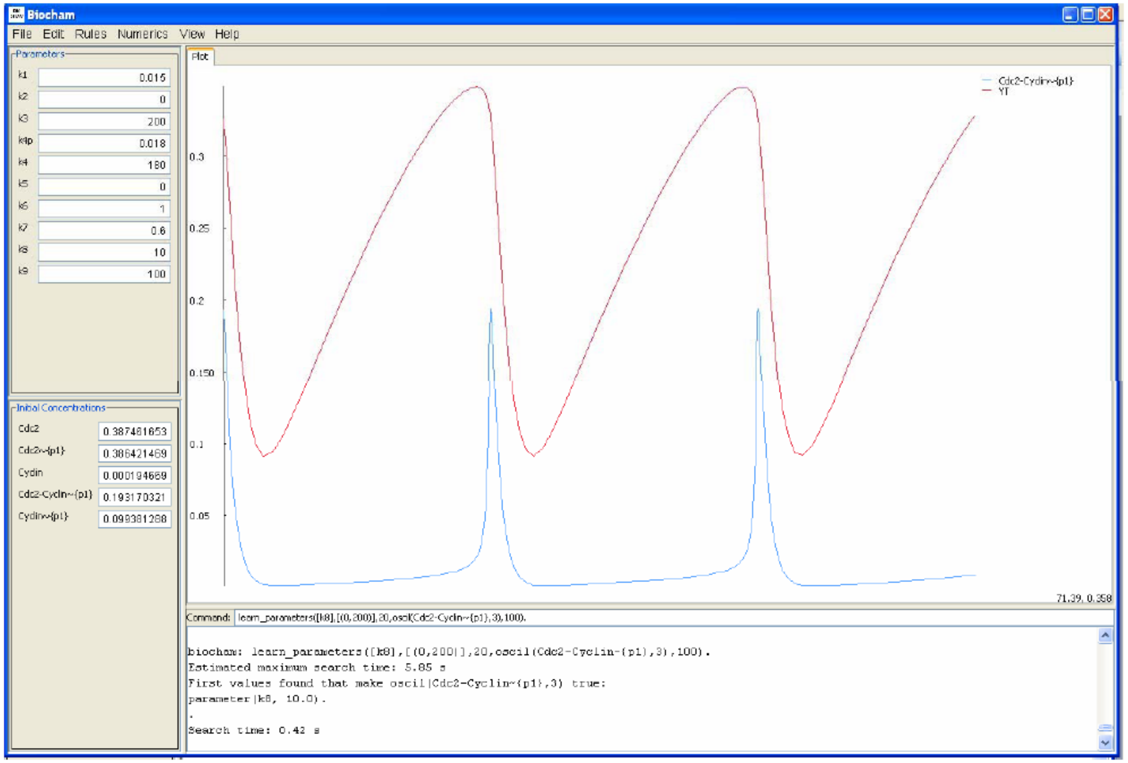
The search for the parameter value is oriented by the LTL specification:

learn_parameters([k1,k2],[(0,10),(0,50)],20,oscil(A,2) & period(A,24),300)

Search for values of k1 and k2 in the respective domains [0,10] and [0,50] with 20 different tries for each value such that the molecule A must oscillate twice with a period of at least 24 in 300 time units.

Choose a parameter to vary in a given interval such that the concentration of the active complex oscillates 3 times in 100 time units:

learn_parameters([k8],[(0,200)],20,oscil(Cdc2-Cyclin~{p1},3),100)



Parameters

k1	0.015
k2	0
k3	200
k4p	0.018
k4	180
k5	0
k6	1
k7	0.6
k8	10
k9	100

Initial Concentrations

Cdc2	0.387481653
Cdc2~{p1}	0.386421469
Cyclin	0.000194669
Cdc2-Cyclin~{p1}	0.193170321
Cyclin~{p1}	0.099381288

Command: learn_parameters([k8],[(0,200)],20,oscil(Cdc2-Cyclin~{p1},3),100).

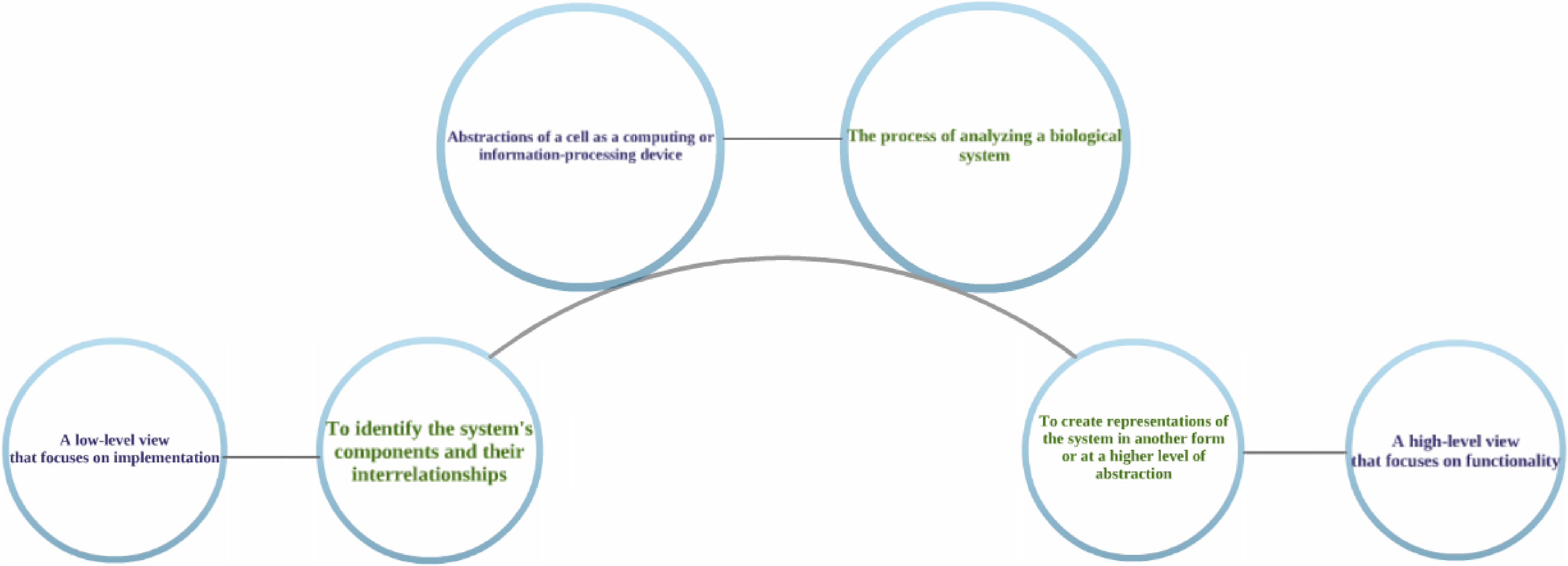
```
biocham: learn_parameters([k8],[(0,200)],20,oscil(Cdc2-Cyclin~{p1},3),100).
Estimated maximum search time: 5.85 s
First values found that make oscil(Cdc2-Cyclin~{p1},3) true:
parameter(k8, 10.0).
.
Search time: 0.42 s
```

The minimum value for k8 to get 3 oscillations in 100 time units is 10

It is possible to verify if the global properties of the system, expressed by temporal logic formulae, are conserved.

It is possible to automatically search for parameter values that reproduce the specified behavior of the system in different conditions.

Take-Home Message



Abstractions of a cell as a computing or information-processing device

The process of analyzing a biological system

A low-level view that focuses on implementation

To identify the system's components and their interrelationships

To create representations of the system in another form or at a higher level of abstraction

A high-level view that focuses on functionality

Computational Approaches to Biological Questions

La secuencia y la homología estructural (o similitud) entre las moléculas se pueden utilizar para inferir la similitud estructural y funcional.

Computational Approaches to Biological Questions

To build a simplified representation that captures all the features is complicated.

A **model** is an **abstract** way of describing a complex system.

Computational Approaches to Biological Questions

Some examples:

- **Accessing 3D molecules through a 1D representation:** strings of single letters.
- **Abstractions for modeling protein structure:** a protein is treated as a series of beads (representing the individual amino acids) on a string (representing the backbone).
- **Mathematical modeling of biochemical systems:** *metabolic control analysis*: describe a biochemical process in terms of the concentrations of chemical species involved in a pathway, and the reactions and fluxes that affect those concentrations.

Computational Approaches to Biological Questions

Theoretical and computational modeling in biology provides testable hypotheses, not definitive answers:

- It becomes easier to preselect targets for experimentation in molecular biology and biochemistry.
- The discovery of general rules and properties in data.
- Developing database structures and query tools.

Computational Methods

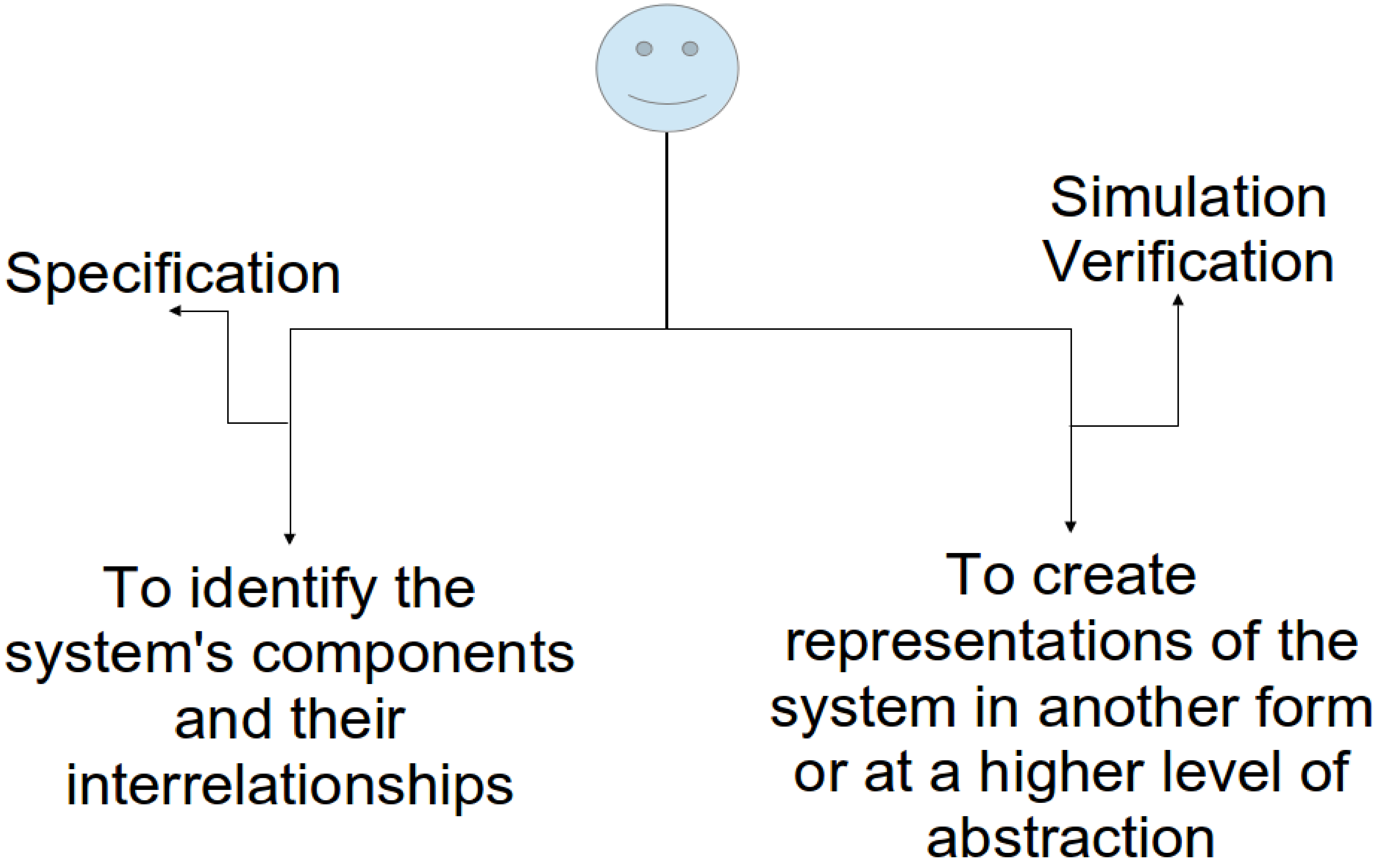
- Using public databases and data formats,
- Sequence alignment and sequence searching,
- Gene prediction,
- Multiple sequence alignment,
- Phylogenetic analysis,
- Extraction of patterns and profiles from sequence data,
- Protein sequence analysis,
- Protein structure prediction,

Computational Methods

- Protein structure property analysis,
- Protein structure alignment and comparison,
- Biochemical simulation,
- Whole genome analysis,
- Primer design,
- DNA microarray analysis,
- Proteomics analysis,
- ...

A Computational Biology Experiment

1. **Identifying the problem:** a scientific experiment always begins with a question.
2. **Separating the problem into simpler components:** a modular approach by breaking down the problem into distinct modules.
3. **Evaluating your needs:** available data and starting points for modeling.
4. **Selecting an appropriate data set.**
5. **Critical evaluation of results:** to establishing the usefulness of a computer modeling in biology.



Specification

Simulation
Verification

To identify the system's components and their interrelationships

To create representations of the system in another form or at a higher level of abstraction

{Understanding biology from a computational/system perspective}

“Computer science is to biology what mathematics is to physics -Harold Morowitz”

CONTENIDO-EVALUACIÓN

SEMANA	TEMA
1 Miércoles 25 Enero-17, 2-6PM, Lago 0.5	PRESENTACIÓN DEL MÓDULO INTRODUCTION TO COMPUTATIONAL BIOLOGY. Ejercicio en clase 10%
2 Miércoles 01 Febrero-17, 2-6PM, Lago 0.5	INTRODUCTION TO COMPUTATIONAL BIOLOGY: THE COMPLEXITY OF CELL-BIOLOGICAL SYSTEMS. Ensayo escrito 15%
3 Miércoles 08 Febrero-17, 2-6PM, Lago 0.5	INTRODUCTION TO COMPUTATIONAL BIOLOGY: COMPUTATIONAL APPROACHES TO BIOLOGICAL QUESTIONS. Ensayo escrito-Presentación oral 15%
4 Miércoles 15 Febrero-17, 2-6PM, Palmas 4.0	TUTORIAL 1: COMPUTATIONAL TOOLS AND APPLICATIONS: The Biochemical Abstract Machine (Biocham): a software and a modeling environment for computational and systems biology. Ejercicio en clase 15%
5 Miércoles 22 Febrero-17,2-6PM, Pendiente realizar reserva	TUTORIAL 2: COMPUTATIONAL TOOLS AND APPLICATIONS: BioNetGen: a multiscale software and a modeling environment for biological systems. Ejercicio en clase 15%
6 Miércoles 01 Marzo-17, 2-6PM, Lago 0.5	Proyecto 30%. Reporte de nota final del módulo.