

# Introduction to P-Lingua

Ignacio Pérez Hurtado

Research Group on Natural Computing  
Department of Computer Science and Artificial Intelligence  
University of Seville

September 5, 2011. Osuna, Sevilla, Spain



- 1 A short presentation of me...
- 2 Introduction
- 3 A software framework for Membrane Computing
- 4 P-Lingua syntax by examples
- 5 P-Lingua tools and libraries
- 6 Current state of the art and open problems...



- 1 A short presentation of me...
- 2 Introduction
- 3 A software framework for Membrane Computing
- 4 P-Lingua syntax by examples
- 5 P-Lingua tools and libraries
- 6 Current state of the art and open problems...



# A short presentation of me...

Ignacio Pérez-Hurtado (perezh@us.es)

- PhD on Computer Science
  - Desarrollo y Aplicaciones de un Entorno de Programación para Computación Celular: P-Lingua
- Assistant Professor
- Postdoctoral researcher
  - Proyecto de Excelencia de la Junta de Andalucía con Investigador de Reconocida Valía: P08-TIC-04200
- Research Group on Natural Computing
- Dpt. Computer Science and Artificial Intelligence
- University of Sevilla



# A short presentation of me...

## Research interests

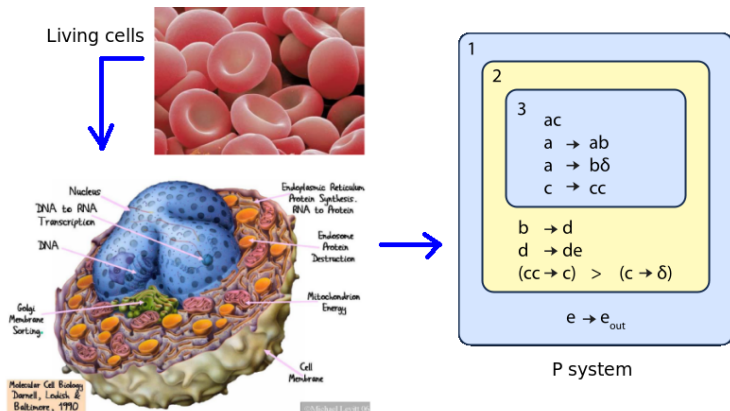
- Unconventional Models of Computation
- Membrane Computing
- Bioinformatics
- Systems & Synthetic Biology
- Software Development
- Artificial Intelligence



- 1 A short presentation of me...
- 2 Introduction**
- 3 A software framework for Membrane Computing
- 4 P-Lingua syntax by examples
- 5 P-Lingua tools and libraries
- 6 Current state of the art and open problems...



# Membrane Computing



# Simulators for P systems

## Motivation

### Simulation vs Implementation

- P systems have not been implemented yet
- It is necessary software/hardware to simulate P system computations

### Applications of simulators

- Informative and educational tools
- Support researching in Membrane Computing
- Simulation, validation and virtual experimentation over models of real-life phenomena



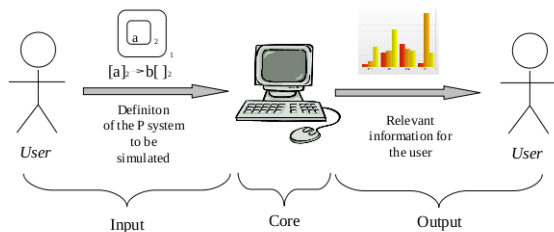


# Simulators for P systems

## General structure

- Many developed simulators <sup>a</sup>
- Similar structure

<sup>a</sup>Software for P systems. D Díaz Pernil et al. The Oxford Handbook of Membrane Computing. 2010, pp. 437–454



- 1 A short presentation of me...
- 2 Introduction
- 3 A software framework for Membrane Computing**
- 4 P-Lingua syntax by examples
- 5 P-Lingua tools and libraries
- 6 Current state of the art and open problems...



- P-Lingua: A programming language to define P systems
- Tools for compilation and simulation
- pLinguaCore: Library implementing simulation algorithms

# P-Lingua: A language to define P systems

- Language close to scientific notation
- Standar, modular and parametric
- Desacoupled from its applications
- Several supported variants of P systems *cell-like* and *tissue-like*
- Extensible
- Website: <http://www.p-lingua.org>



# P-Lingua: A language to define P systems

Página web: <http://www.p-lingua.org>

The screenshot shows a Mozilla Firefox browser window titled "Main Page - The P-Lingua Website". The address bar shows the URL [http://www.p-lingua.org/wiki/index.php/Main\\_Page](http://www.p-lingua.org/wiki/index.php/Main_Page). The page content includes a navigation menu with "page", "discussion", "view source", and "history" tabs. The main heading is "Main Page". Below it, a notice states "This website is under construction". The text describes P-Lingua as a programming language for membrane computing, developed by the Research Group on Natural Computing at the University of Sevilla, Spain. It mentions the pLinguaCore Java library and its capabilities for simulating P systems. A search box is located on the left side of the page. At the bottom, there is a footer with GNU FDL license information, a date of last modification (21 August 2009), and a "Terminado" status.

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

[http://www.p-lingua.org/wiki/index.php/Main\\_Page](#)

Más visitados ▾ Marcadores inteli... ▾ Getting Started Latest Headlines ▾

page discussion view source history

## Main Page

**This website is under construction**

P-Lingua is a programming language for [membrane computing](#) which aims to be a standard to define [P systems](#). It and its [associated tools](#) have been developed by members of the [Research Group on Natural Computing](#), at the [University of Sevilla](#), Spain.

We provide P-Lingua and its associated tools as a free and reusable package for the development of software/hardware applications capable of simulate P system computations.

In order to implement this idea, a [Java](#) library called [pLinguaCore](#) has been produced as a software framework for cell-like P system simulators. It is able to handle input files (either in [XML format](#) or in [P-Lingua format](#)) defining P systems from a number of different [supported models](#). Moreover, the library includes several [built-in simulators](#) for each model. For the sake of [software portability](#), [pLinguaCore](#) can export a P system definition to any convenient [output format](#) (currently [XML format](#) and [binary format](#) are available). [P.LinguaCore](#) is not a closed product, but it can be extended to accept new input or output formats and also new models or simulators.

There are several [applications](#) in development using P-Lingua. This website is available to [download](#) the libraries and applications, as well as provides [technical information](#). In addition, this site aims to be a meeting point for users and developers through the use of [forums](#).

Please, [contact us](#) for any suggestion or comment.

search

Go Search

toolbox

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link

GNU FDL  
FREE DOCUMENTATION  
LICENSE

This page was last modified on 21 August 2009, at 16:29. This page has been accessed 91 times. Content is available under [GNU Free Documentation License 1.2](#).

[Privacy policy](#) [About The P-Lingua Website](#) [Disclaimers](#)

Powered by MediaWiki

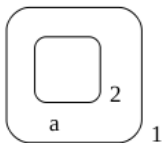
Terminado



- 1 A short presentation of me...
- 2 Introduction
- 3 A software framework for Membrane Computing
- 4 P-Lingua syntax by examples**
- 5 P-Lingua tools and libraries
- 6 Current state of the art and open problems...



# Example 1: Active membranes with division rules



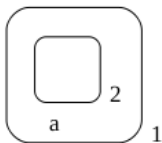
$[a \rightarrow a, b]_1$

$b[]_2 \rightarrow [c]_2^+$

$[c]_2^+ \rightarrow [d]_2 [e]_2^-$

```
@model<membrane_division>
def main()
{
  @mu = [[]'2]'1;
  @ms(1) = a;
  [a --> a,b]'1;
  b[]'2 --> +[c]'2;
  +[c]'2 --> [d]'2 -[e]'2;
}
```

## Example 2: Active membranes with creation rules



$[a \rightarrow a, b]_1$

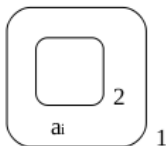
$b[]_2 \rightarrow [c]_2^+$

$[c]_2^+ \rightarrow [[d^2]_3^-]_2$

```
@model<membrane_creation>
def main()
{
  @mu = [[]'2]'1;
  @ms(1) = a;
  [a --> a,b]'1;
  b[]'2 --> +[c]'2;
  +[c]'2 --> [-[d*2]'3]'2;
}
```



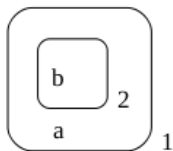
# Example 3: Transition P systems



$$[a_i \ [ \ ]_2]_1 \longrightarrow [a_{i+1} \ [b_i]_2]_1 \quad 1 \leq i \leq 10$$

```
@model<transition>
def main()
{
  @mu = [[]'2]'1;
  @ms(1) = a{1};
  [a{i} \ []'2]'1 --> [a{i+1} \ [b{i}]'2]'1 : 1<=i<=10;
}
```

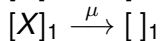
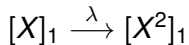
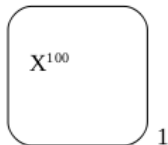
# Example 4: Symport/antiport P systems



$a[b]_2 \rightarrow b[a]_2$

```
@model<symport_antiport>
def main()
{
  @mu = [[ ]' 2]' 1;
  @ms(1) = a;
  @ms(2) = b;
  a[b]' 2 --> b[a]' 2;
}
```

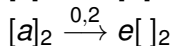
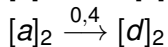
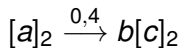
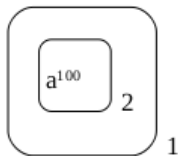
# Example 5: Stochastic P systems



$$\lambda = 1 \quad \mu = 1, 1$$

```
@variant<stochastic>
def BirthDeath(lambda,mu,nX)
{
  @mu = []'1;
  @ms(1) = X*nX;
  [X]'1 --> [X*2]'1::lambda;
  [X]'1 --> [#]'1::mu;
}
def main() {
  call BirthDeath(1,1.1,100);
}
```

# Example 6: Probabilistic P systems



```
@model<probabilistic>
def main()
{
  @mu = [[]'2]'1;
  @ms(2) = a*100;
  [a]'2 --> b[c]'2 :: 0.4;
  [a]'2 --> [d]'2 :: 0.4;
  [a]'2 --> e[]'2 :: 0.2;
}
```

# Example 7: tissue P systems

```
@model<tissue_psystems>
def main()
{
  @mu = [[]'1 []'2]'0;
  @ms(0) = a;
  @ms(1) = b*5;
  @ms(2) = c*10,d;
  [b]'1 <--> [c*2]'2;
  [c]'1 <--> [a]'0;
  [d]'2 --> [e]'2 [f]'2;
}
```

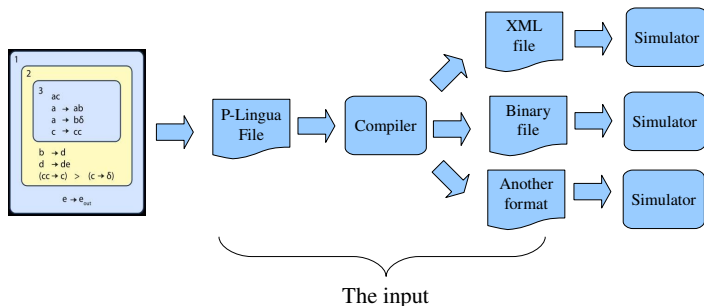


- 1 A short presentation of me...
- 2 Introduction
- 3 A software framework for Membrane Computing
- 4 P-Lingua syntax by examples
- 5 P-Lingua tools and libraries**
- 6 Current state of the art and open problems...



# Command-line compilation tool

## Interoperability



- It checks possible programming errors
- It locates errors on the files

## Example: A division rule in “membrane creation”

```
Semantics error: The rule doesn't match  
the "membrane_creation" specification  
in line 38 : 2--28  
Division rules are not allowed
```



- Java library to implement simulators
- Free software (GNU GPL license)
- It reads P-Lingua files
- It implements several simulation algorithms
- It exports to other file formats
- Text interface
- It can be used in other Java applications
- It can be extended
- Web page: <http://www.p-lingua.org>



```
FileInputStream stream = new FileInputStream("sat.pli");
AbstractParserFactory pf = new InputParserFactory();
InputParser parser =
    (InputParser) pf.createParser("P-Lingua");
parser.setVerbosityLevel(5);
Psystem ps = parser.parse(stream);
ISimulator sim =
    ps.createSimulator(false, false, "active_membranes");
sim.setTimed(true);
sim.setVerbosity(1);
sim.run();
```



- 1 A short presentation of me...
- 2 Introduction
- 3 A software framework for Membrane Computing
- 4 P-Lingua syntax by examples
- 5 P-Lingua tools and libraries
- 6 Current state of the art and open problems...



# Current state of the art

- Several P system models supported
- The last one: Spiking Neural P systems (CMC12)
- Several algorithms implementations
- The last one: A new algorithm for Probabilistic P systems
- Two end-degree projects in course
  - A compiler based on AntLR (September)
  - A Web Interface (December)



# Open problems

- A more generical syntax
  - Can we define P systems by ingredients?
- Is it possible a generical simulator?
- More powerful compilers
  - Can we generate optimized source code (I.e. in C/C++)?
- More efficient simulators: HPC, GPUs, FPGAs...



# Thank you!

