# A P-Lingua Programming Environment for Membrane Computing

Daniel Díaz–Pernil, Ignacio Pérez–Hurtado,
Mario J. Pérez–Jiménez, Agustín Riscos–Núñez

Research Group on Natural Computing
Dpt. Computer Science and Artificial Intelligence
University of Sevilla

9th Workshop on Membrane Computing
Edinburgh (UK), July 28–31, 2008

# Outline

# Introduction

- Membrane computing: emerging branch of Natural Computing
- Different models have been defined
- It is necessary to develop software: simulation

# Introduction

## Common elements of simulators for P systems

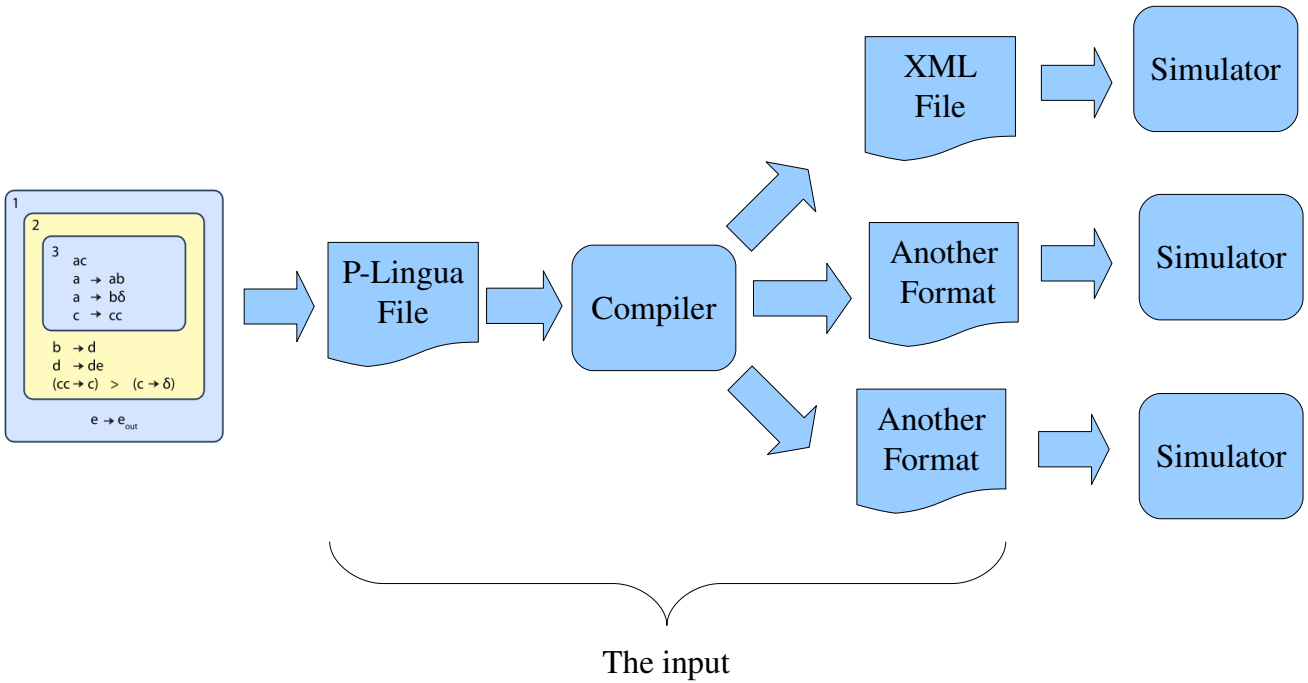
- Input: definition of a P system, and/or the input multisets
- Simulation engine
- Output: information about the simulation/results

- The solutions could be reusable
- The main goal of this proposal is to give a reusable solution to the *input*

# The P-Lingua programming language

- P-Lingua: easy-to-learn programming language to define P systems in a modular and parametric way
- Decoupled from its applications
- By using compiling tools, the P-Lingua programs are translated to other file formats

# The P-Lingua programming language



# Characteristics

- Written in JAVA
- GNU General Public Licensed (GNU GPL)
- Lexical and syntactical analyzers provided by JavaCC (GNU GPL)
- Can be downloaded from the RGNC web page
  `http://www.gcn.us.es`

# A command–line compilation tool

- It translates programs written in P-lingua into XML documents
- Plug-ins can be designed to produce object code with different formats
- It checks possible programming errors (syntactical parser)

# A command–line simulation tool

- A command–line simulator for P systems with active membranes
- It runs one of the possible computations
- It gets the answer that the system outputs to its environment
- It saves a text file with a step-by-step report of the computation

# An integrated development environment

- IDE: application for software development
- It can be updated to accept future versions of the language
- It uses the JEdit framework (GNU GPL), including:
  - A source code editor
  - A compilation tool to generate XML files
  - A simulation tool for debugging
  - A graphical user interface

# Software demonstration

An IDE for P-Lingua

# Conclusions

- Users can define P systems in a modular and parametric way by using an easy-to-learn programming language
- This method is decoupled from its applications
- P-Lingua programs can be translated to other file formats

# Future work

- Expand the language to other types of P systems
- Improve the current software tools and develop new ones
- Design a solution for the "simulation engine": a *distributed* framework
- Develop a plug-in for the Eclipse framework
- Implement heuristics providing "good" computations

# Thanks for your attention

Thank you!