COSC 594

Final Presentation

Membrane Systems/ P Systems

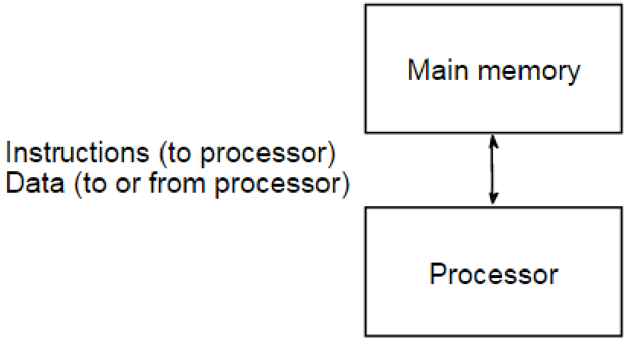
Mesbah Uddin & Gangotree Chakma

November 23, 2016

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Motivation of Unconventional Computing

- Parallel computing-- restricted in conventional computers
- Deterministic conventional computing
- Exhaustive search exponential in classic computing
- Power and scaling--restraining element against better computing performance



THE UNIVERSITY OF TENNESSEE KNOXVILLE

What is Membrane System/P System?

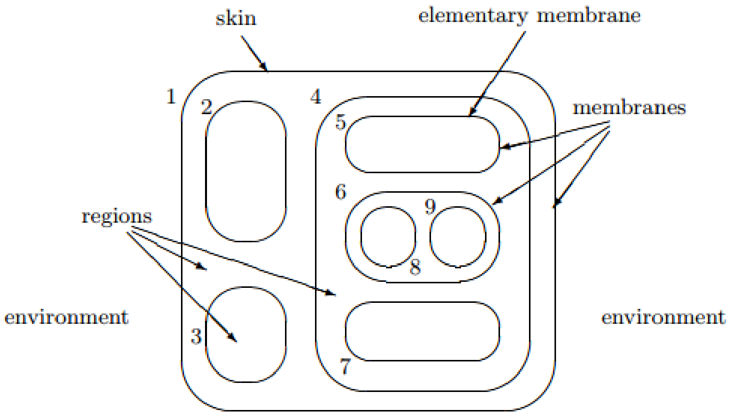
- Bio-inspired natural computing model
- Concept first introduced by Gheorghe Paun in 1998
- Structure based on biological cells
- Capable of parallel programming

History of P Systems

- Mechanically computable-- Turing Machine (1935)
- Neural Computing -- Anderson (1996)
- Genetic algorithms and evolutionary computing/programming -- Koza and Rice (1992)
- DNA computing -- Adleman (1994)

Elements of P System

- Basic Elements
  - Environment
  - Membranes
  - Symbols
  - Catalysts
  - Rules
- The Cell-like Membrane Structure



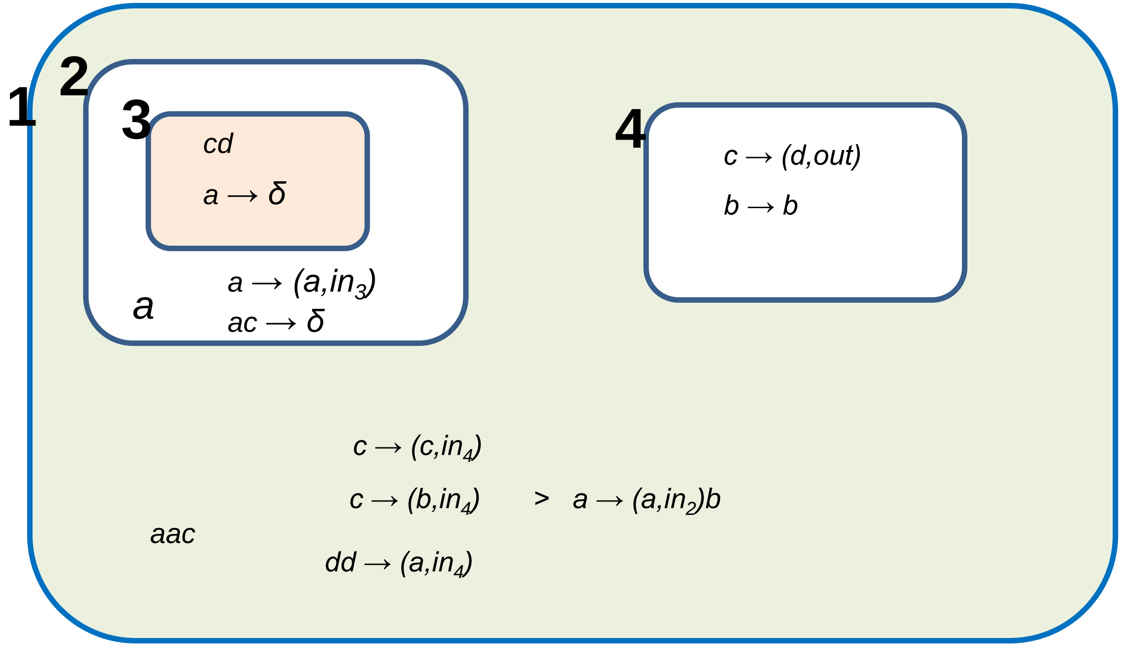
Păun, Gheorghe. "Introduction to membrane computing." *Applications of Membrane Computing*. Springer Berlin Heidelberg, 2006. 1-42.

Representation of a P system

$\Pi = (V, T, C, \mu, M_1,\ldots\ldots,M_m, R_1,\ldots\ldots,R_m\ R_m,i_0\ )$

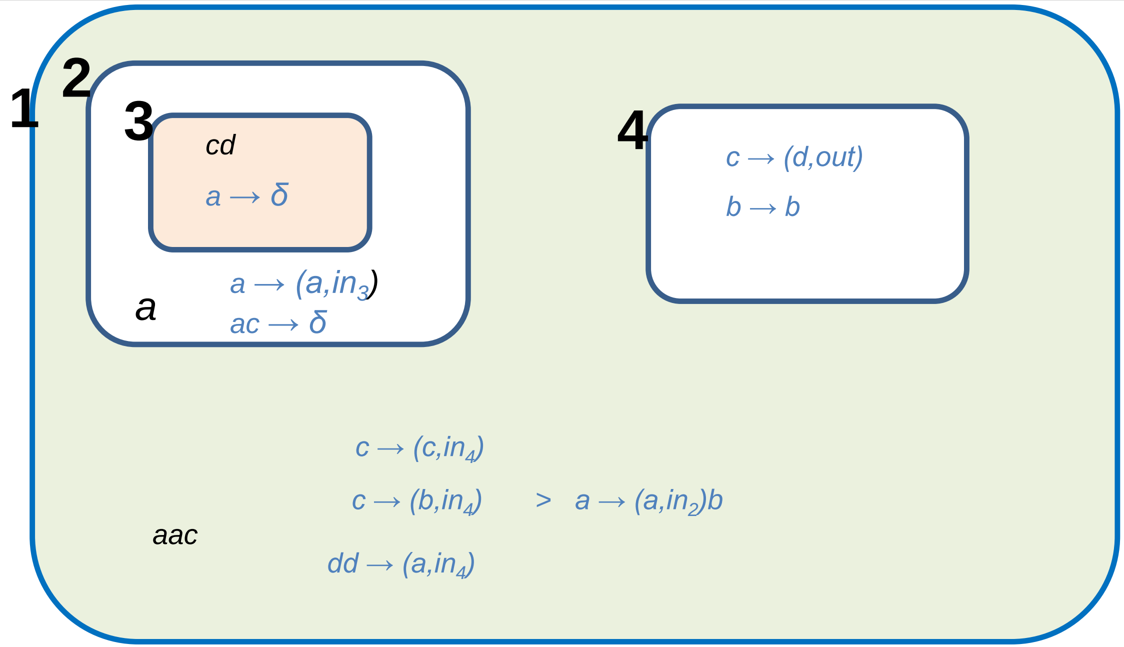
- V : the alphabet of the system
- $T \subseteq V$: the terminal alphabet
- C: set of catalysts
- μ : the membrane structure (of degree m here)
- $M_1,\ldots\ldots,M_m$: finite set of objects (strings/multisets) present in ‘m’ regions of the membrane structure, μ
- $R_1,\ldots\ldots,R_m$: finite rules associated with those ‘m’ regions of μ
- i_0 : Output membrane

Representation of a P system



- $\Pi = (V, T, C, \mu, M_1, \ldots\ldots, M_m, R_1, \ldots\ldots, R_m, i_0)$

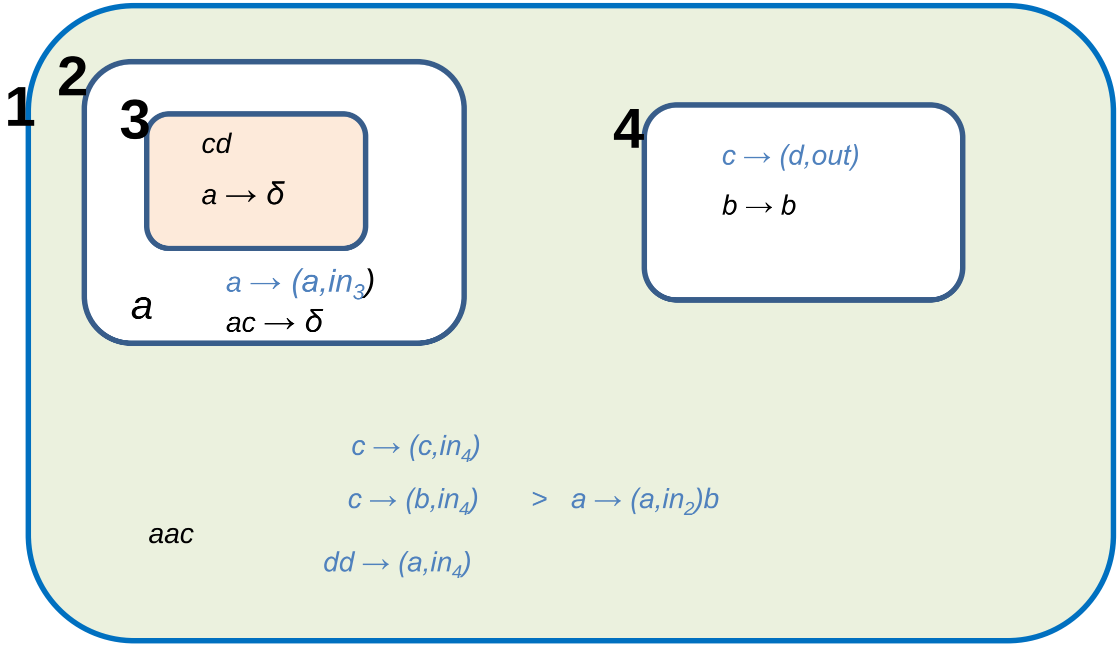
Rules of a P system



- Multiset rewriting rule
- Splicing rule
- Symport and antiport rule

- Rules are analogous to the chemical reaction rules working on objects present in that compartment

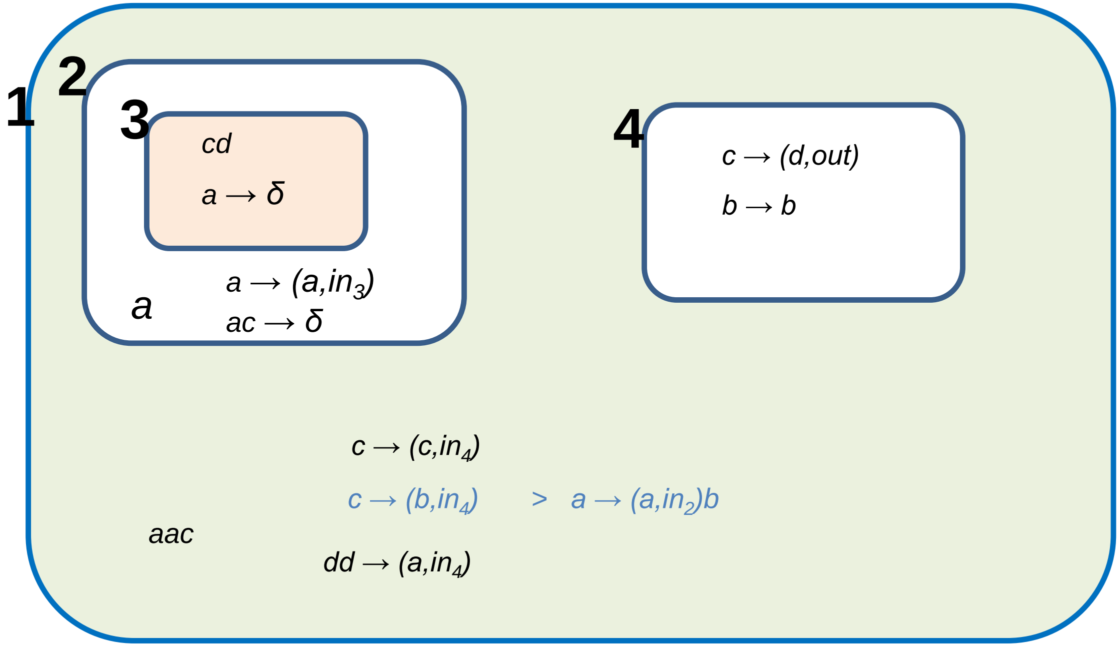
Rules of a P system



- Rules can indicate the flow of objects from one membrane to another membrane.
- Can be of forms like a→(a,in), a→(a,out) or a→(a,here)

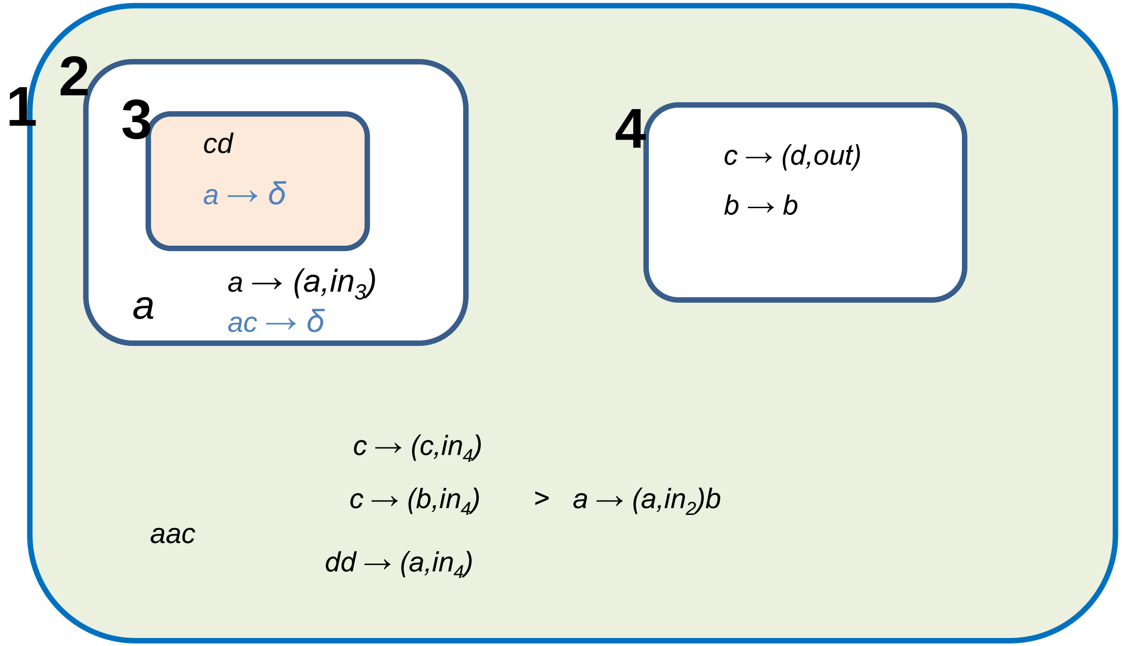
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Rules of a P system



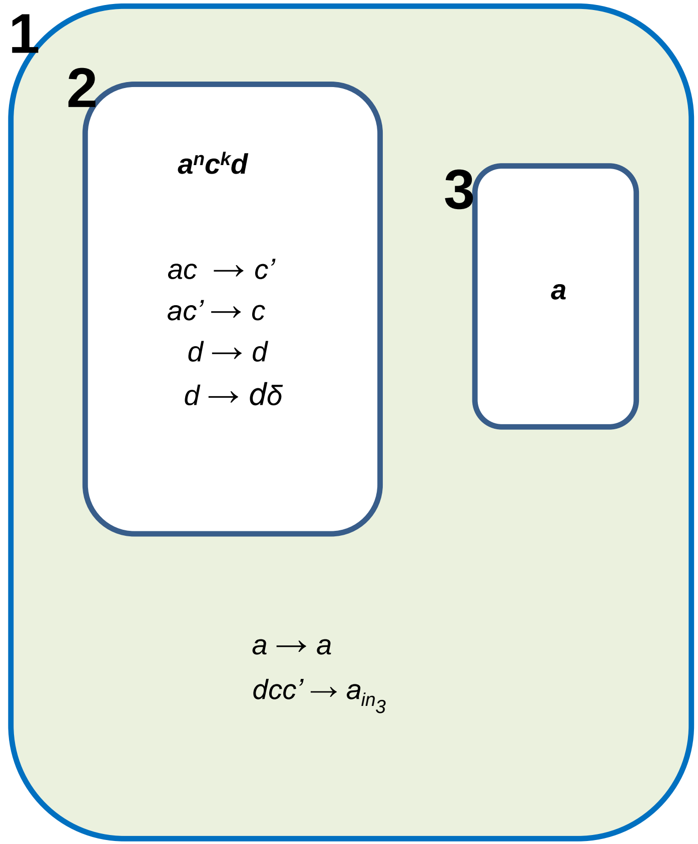
- Can have priority rules
- Can also have multiple sets of parallel rules that can't be applied simultaneously, chosen randomly

Rules of a P system



- Membrane dissolving rules: $a \to \delta$

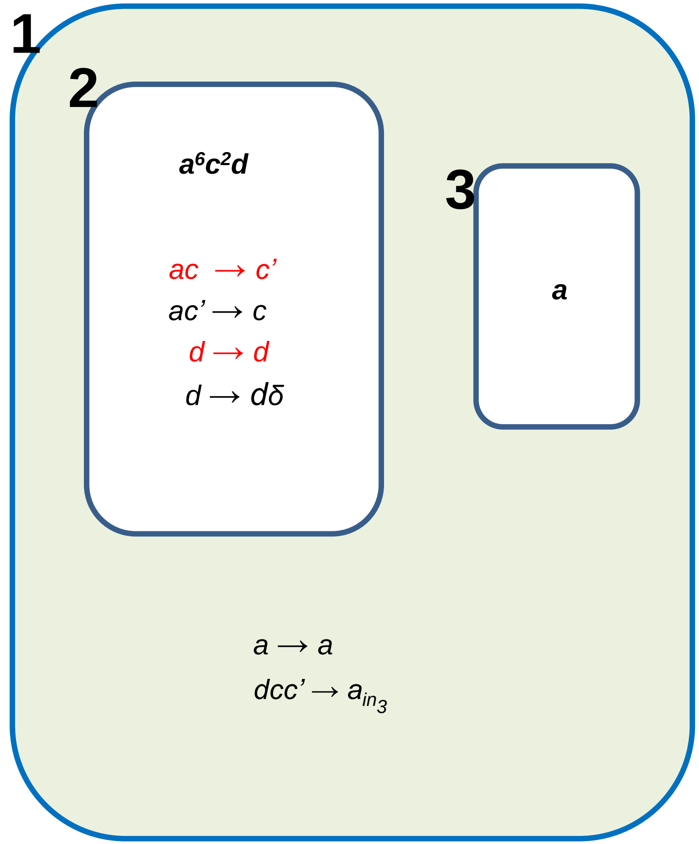
Find if n is divisible by k or not



- Find if n is divisible by k or not
- Membrane 3 is the resultant chamber

Find if n is divisible by k or not

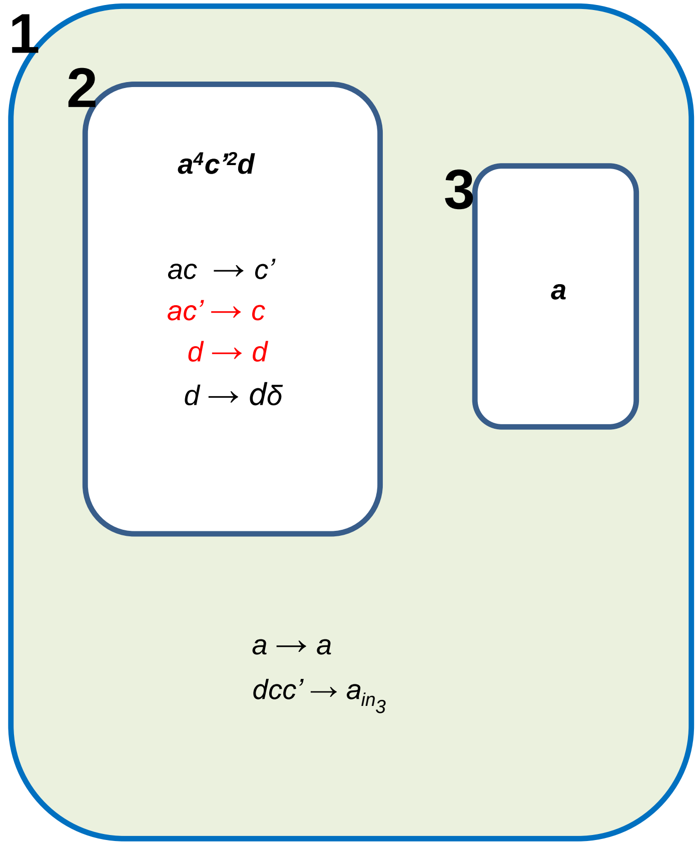
Step 1



Example:
n = 6
K = 2

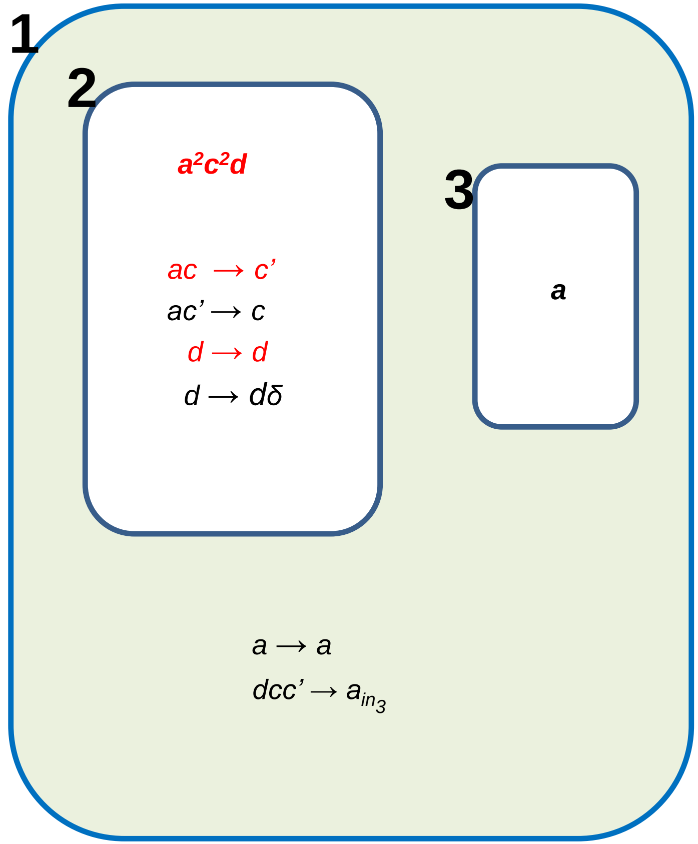
Find if n is divisible by k or not

Step 2



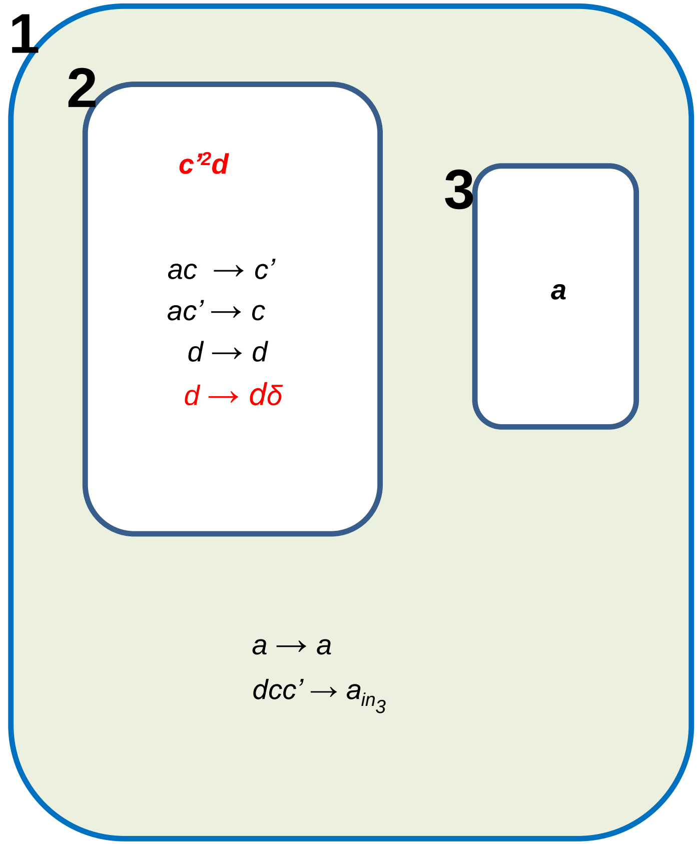
Find if n is divisible by k or not

Step 3



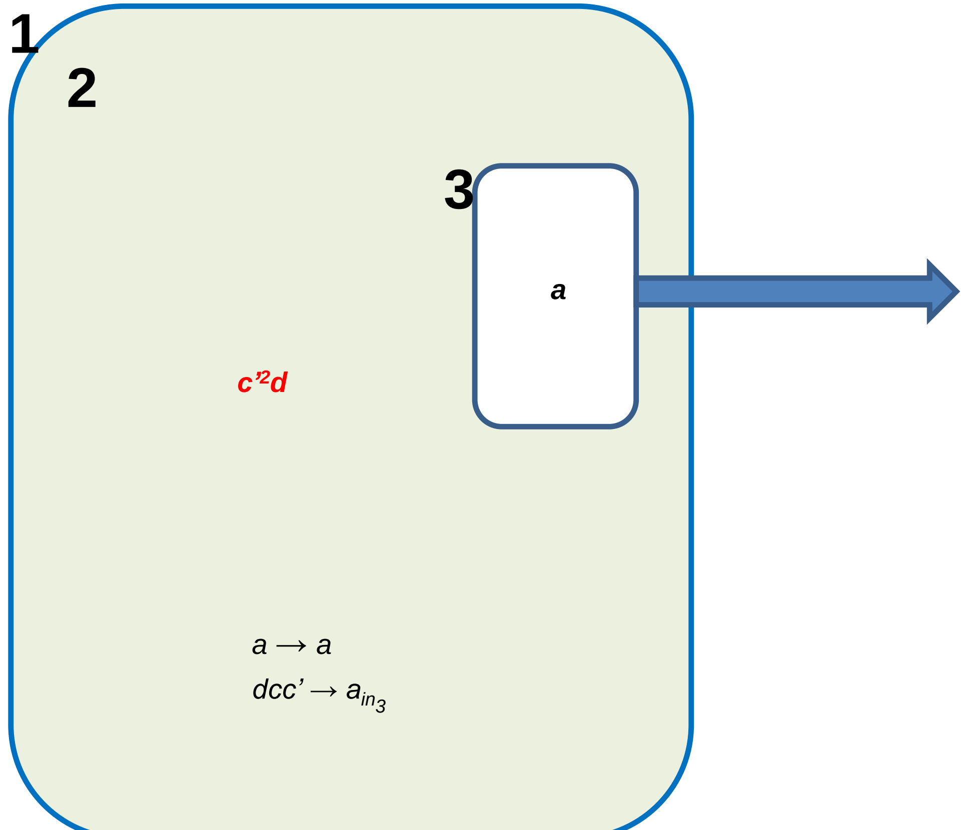
Find if n is divisible by k or not

Step 4



Find if n is divisible by k or not

Step 5

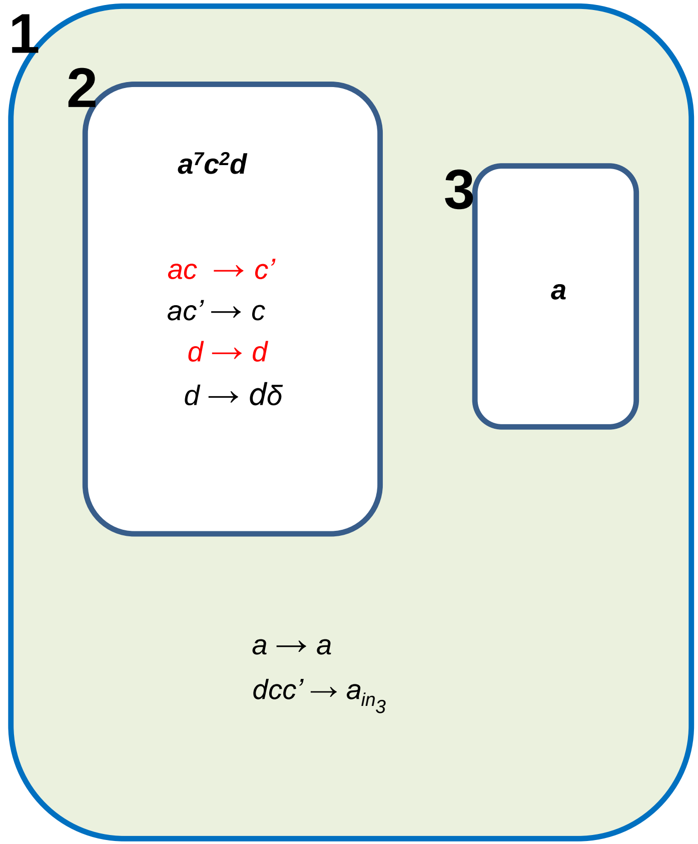


One copy of ‘a’, so in this example n is divisible by k

Computation halts

Find if n is divisible by k or not

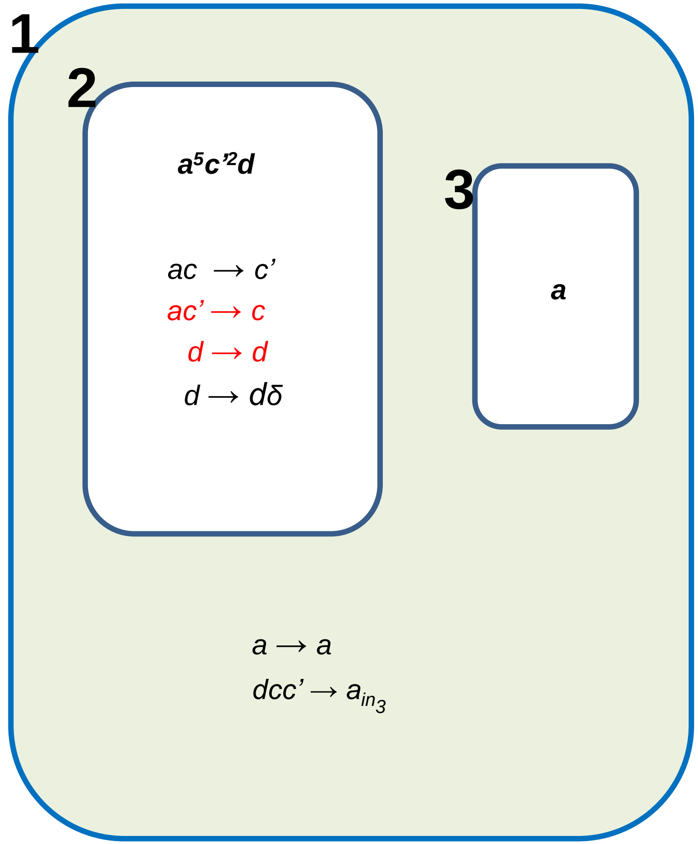
Step 1



Another example:
n = 7
k = 2

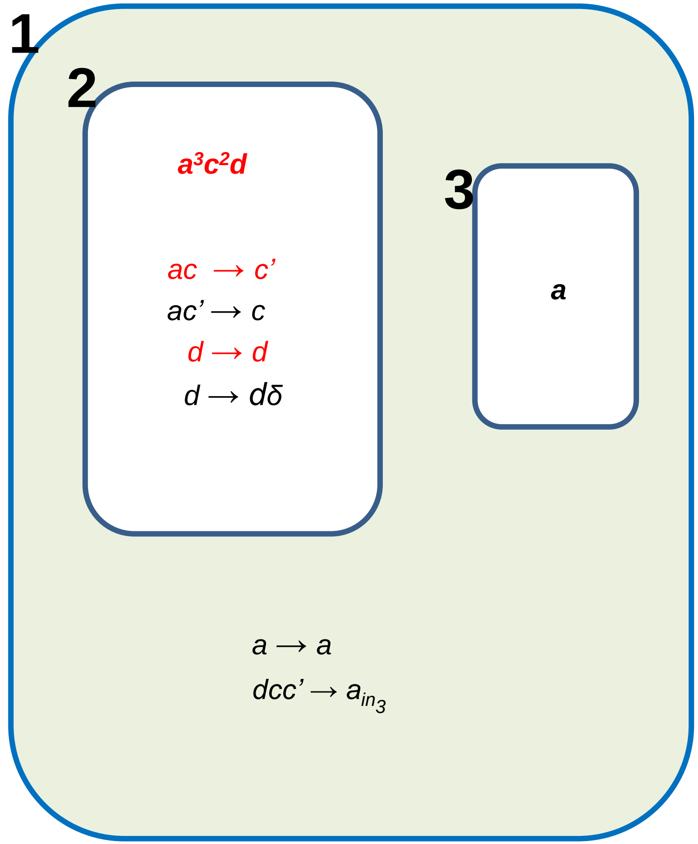
Find if n is divisible by k or not

Step 2



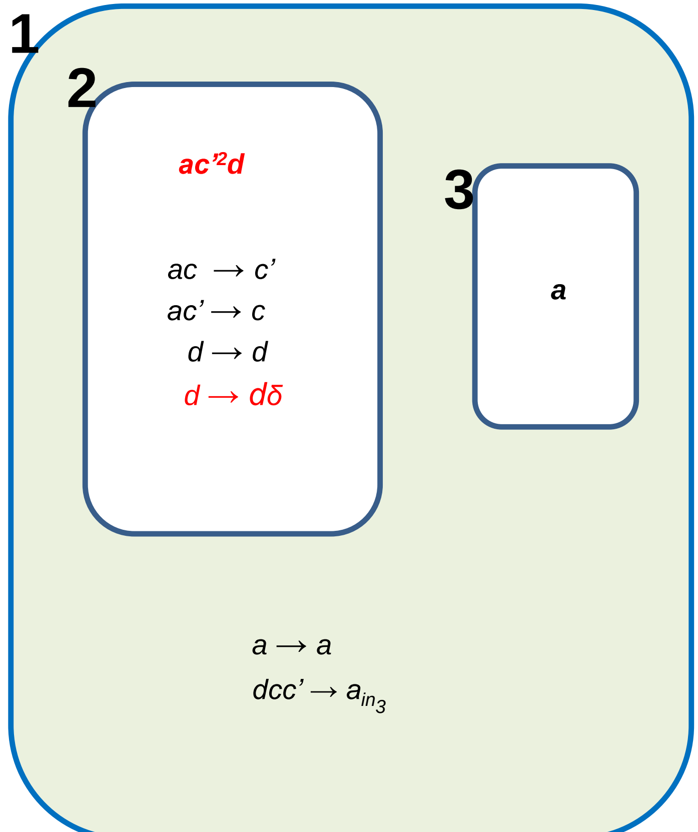
Find if n is divisible by k or not

Step 3



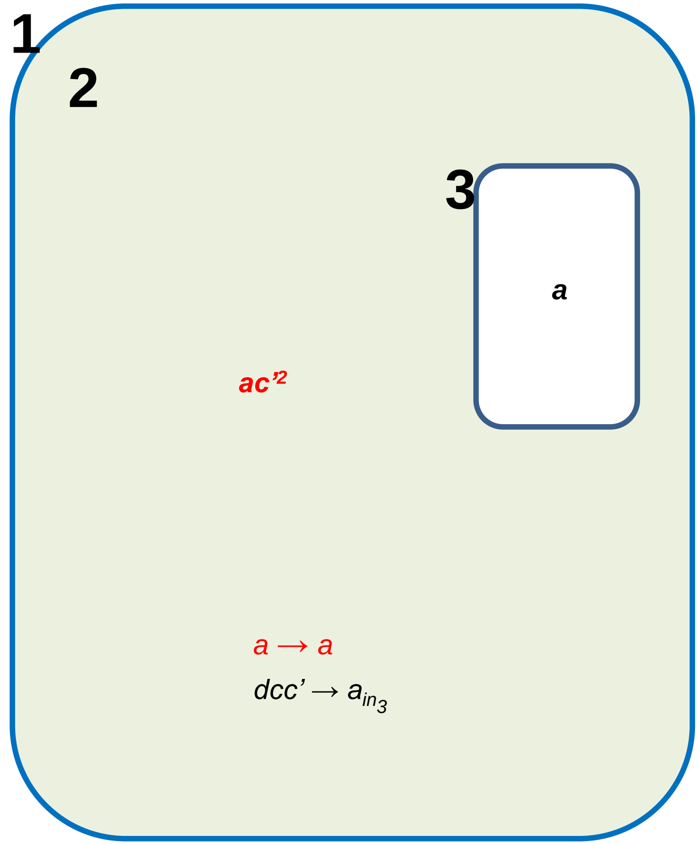
Find if n is divisible by k or not

Step 4



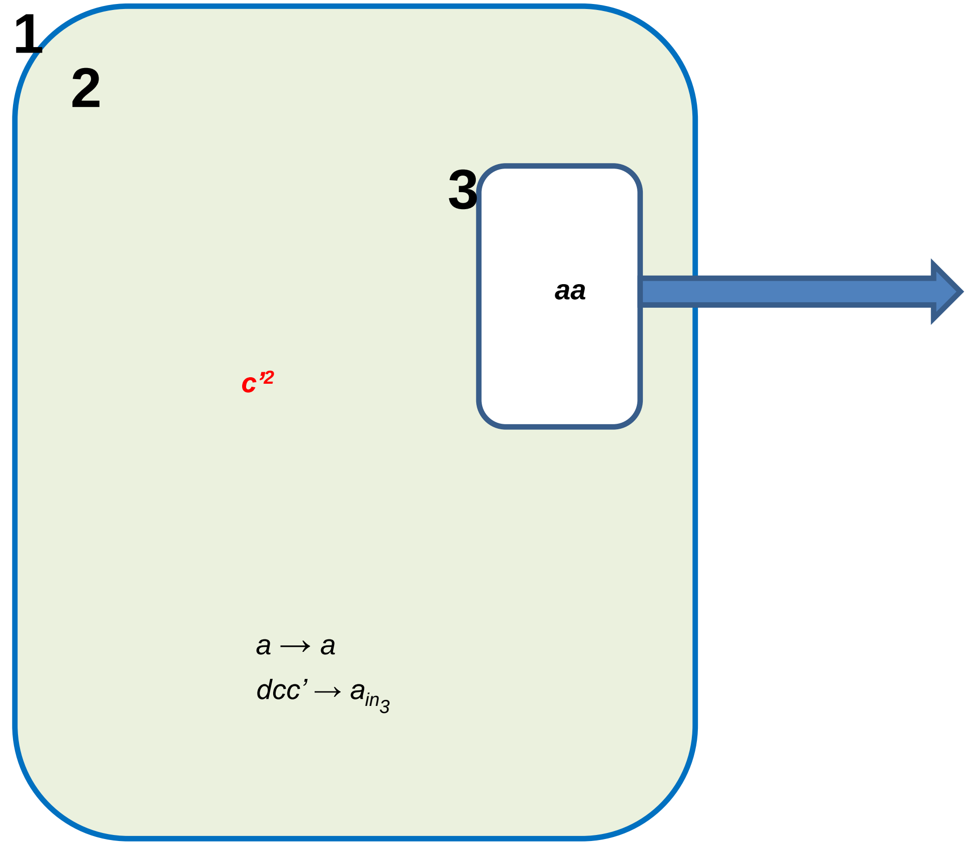
Find if n is divisible by k or not

Step 5



Find if n is divisible by k or not

Step 6



Computation halts

Two copies of ‘a’, so in this example n is not divisible by k

Another example: SAT(Boolean Satisfiability Problem)

NP-complete problem

Representation:

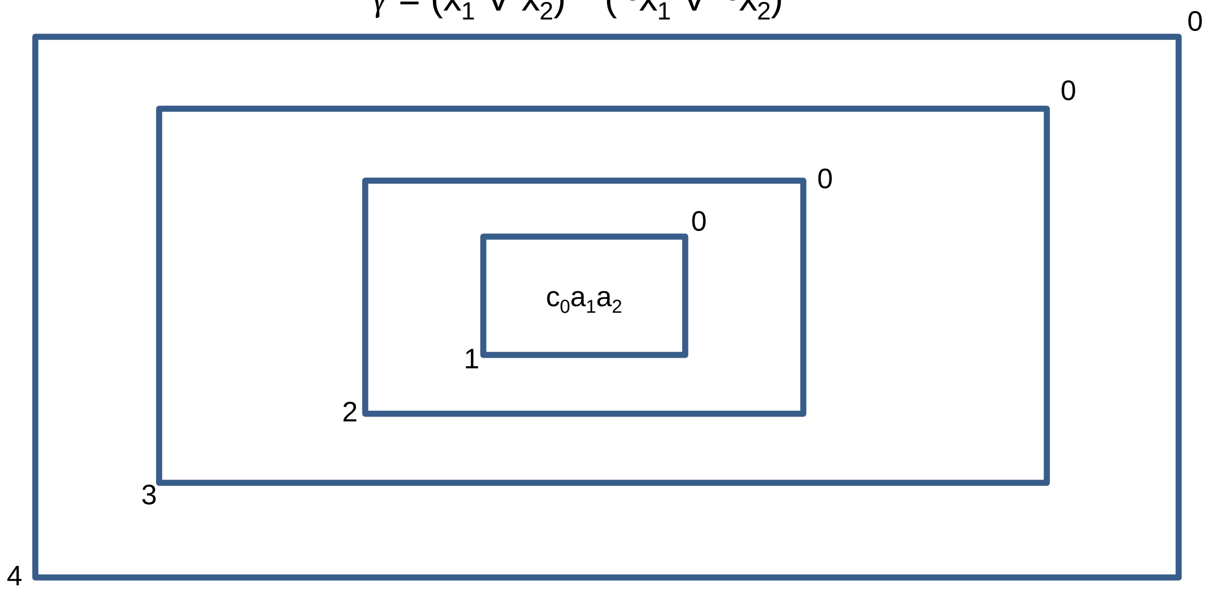
$\gamma = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ with

$C_i = y_{i1} \vee y_{i2} \vee \ldots \vee y_{ip}$

C: clause, y: literal

Example SAT problem:

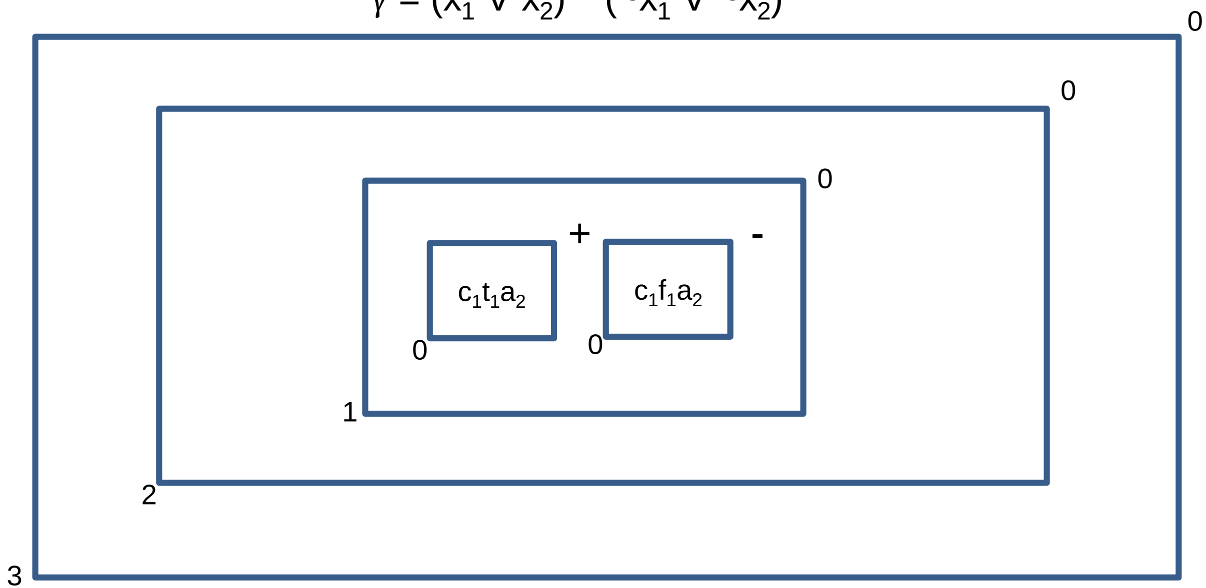
$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



- Two variables a_1 and a_2
- C_0 is a object which increments in each time step

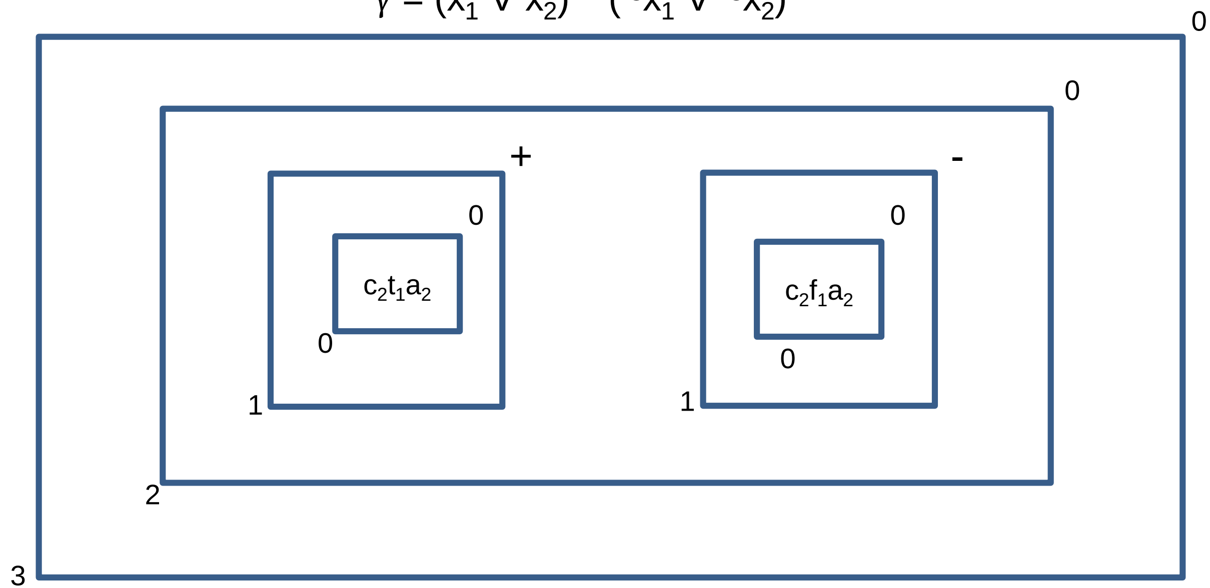
Solving SAT problem using P system:

$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



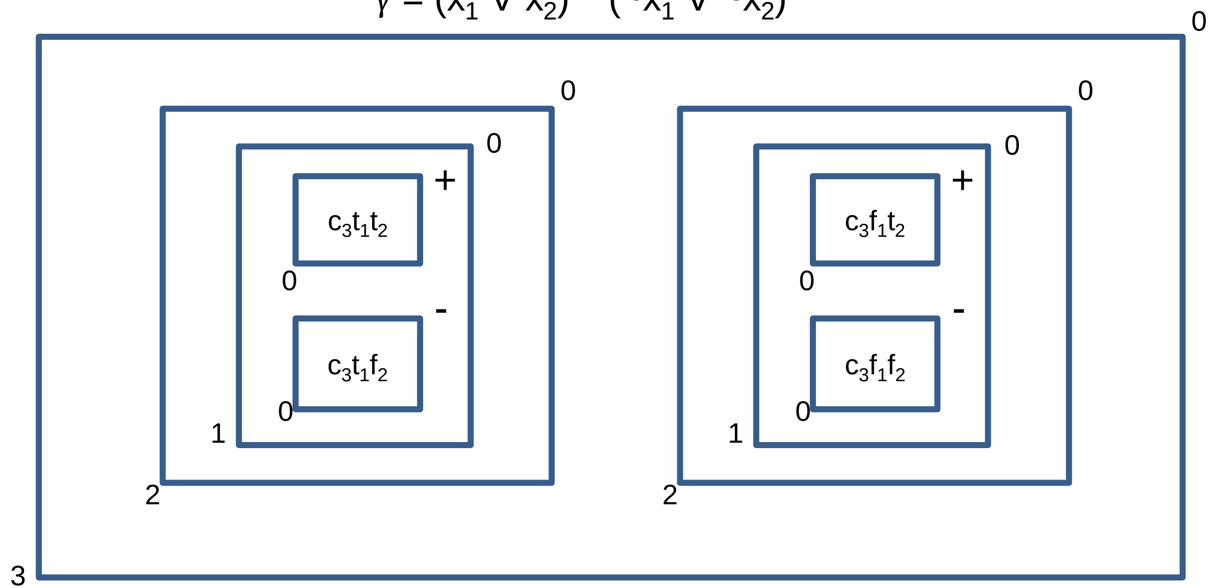
Solving SAT problem using P system:

$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



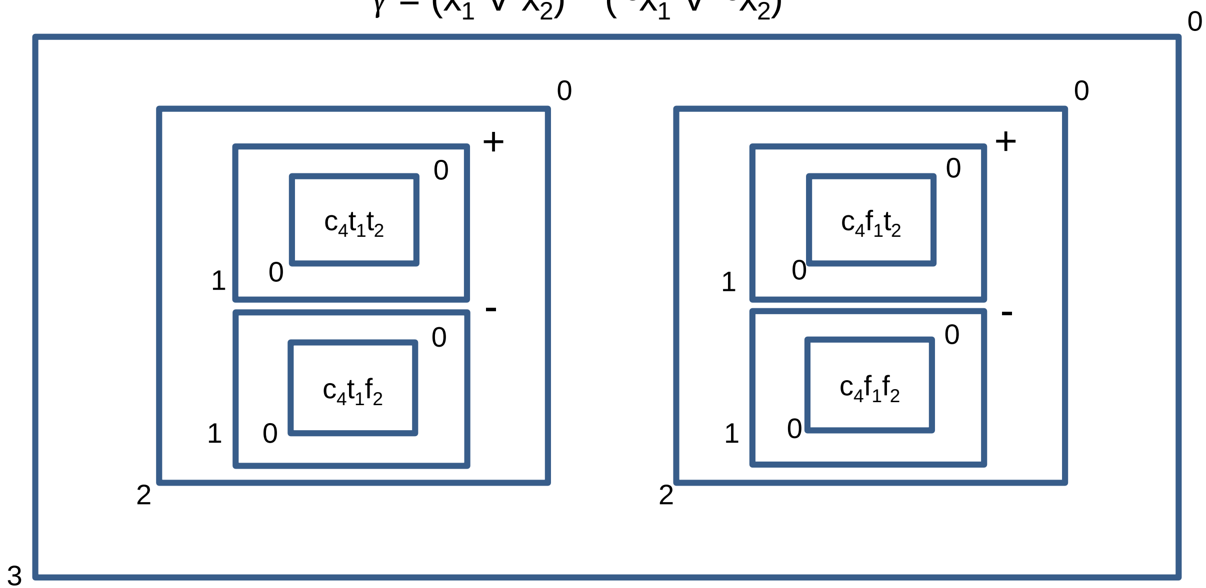
Solving SAT problem using P system:

$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



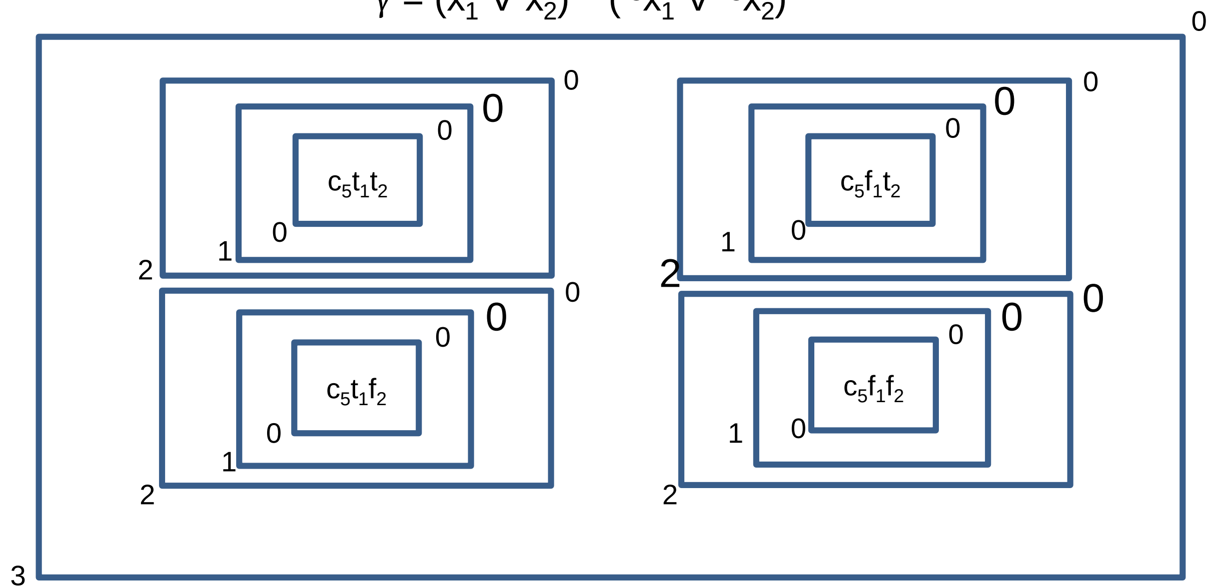
Solving SAT problem using P system:

$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



Solving SAT problem using P system:

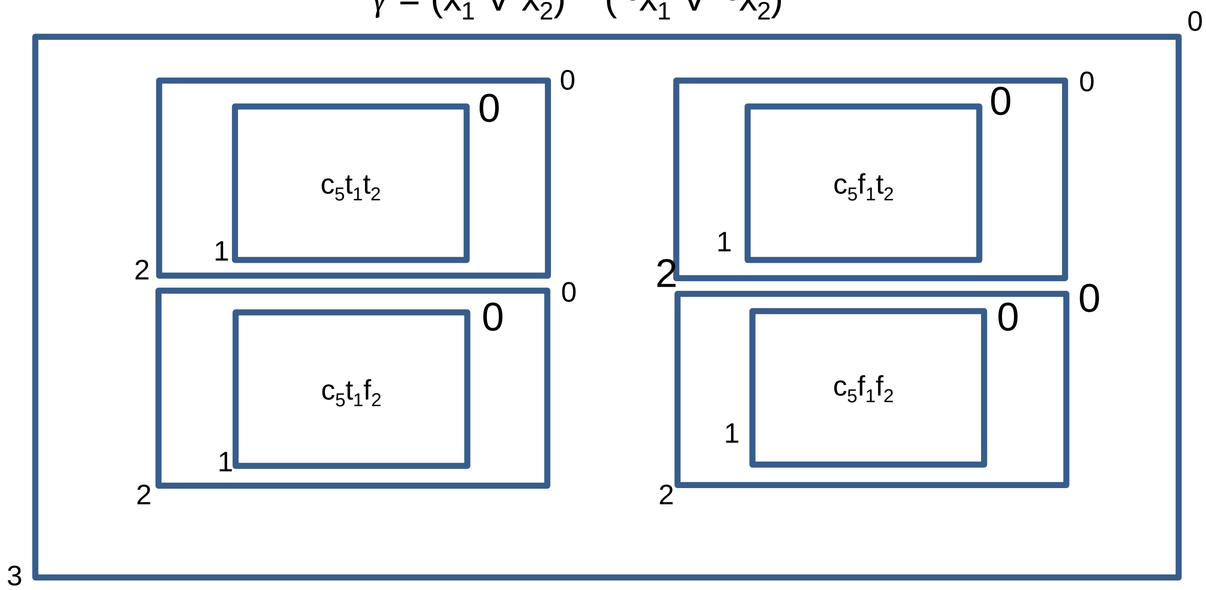
$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Solving SAT problem using P system:

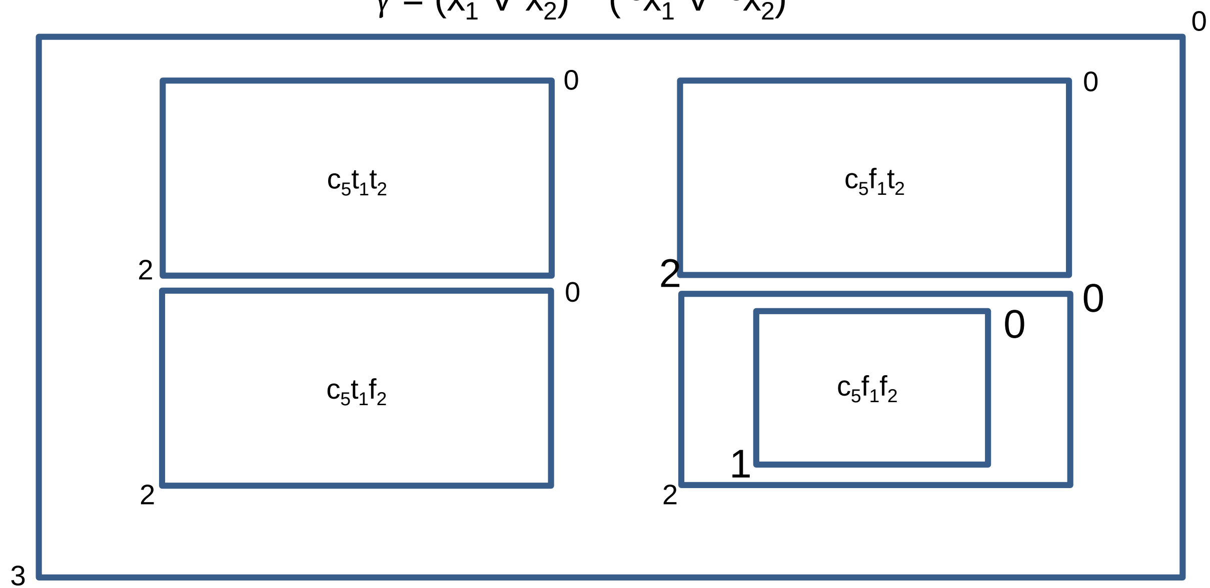
$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



THE UNIVERSITY OF TENNESSEE KNOXVILLE

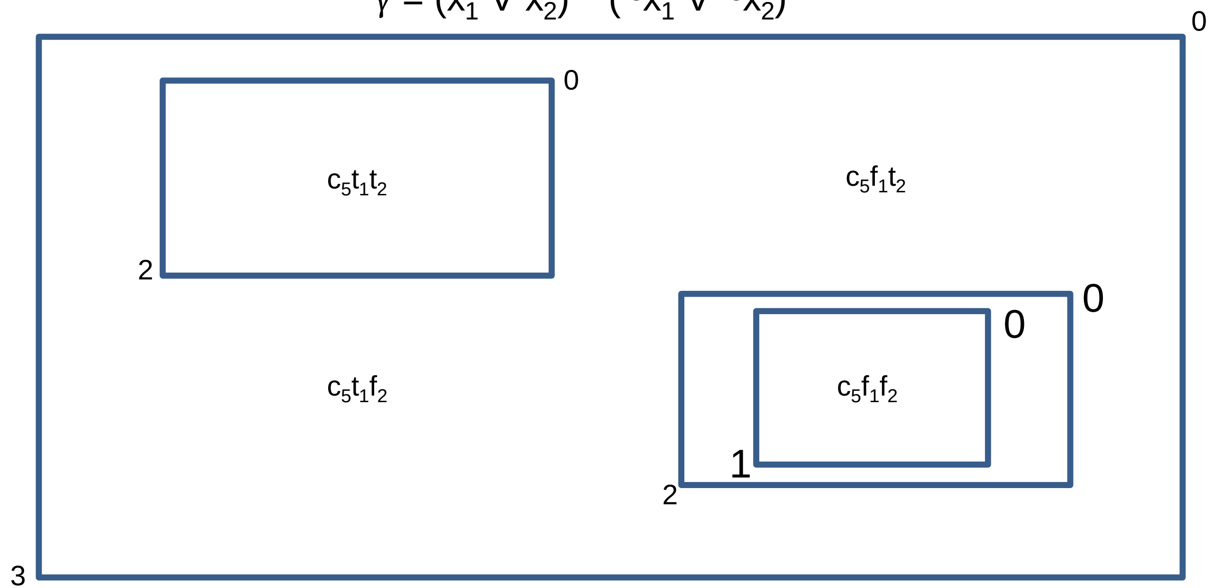
Solving SAT problem using P system:

$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$

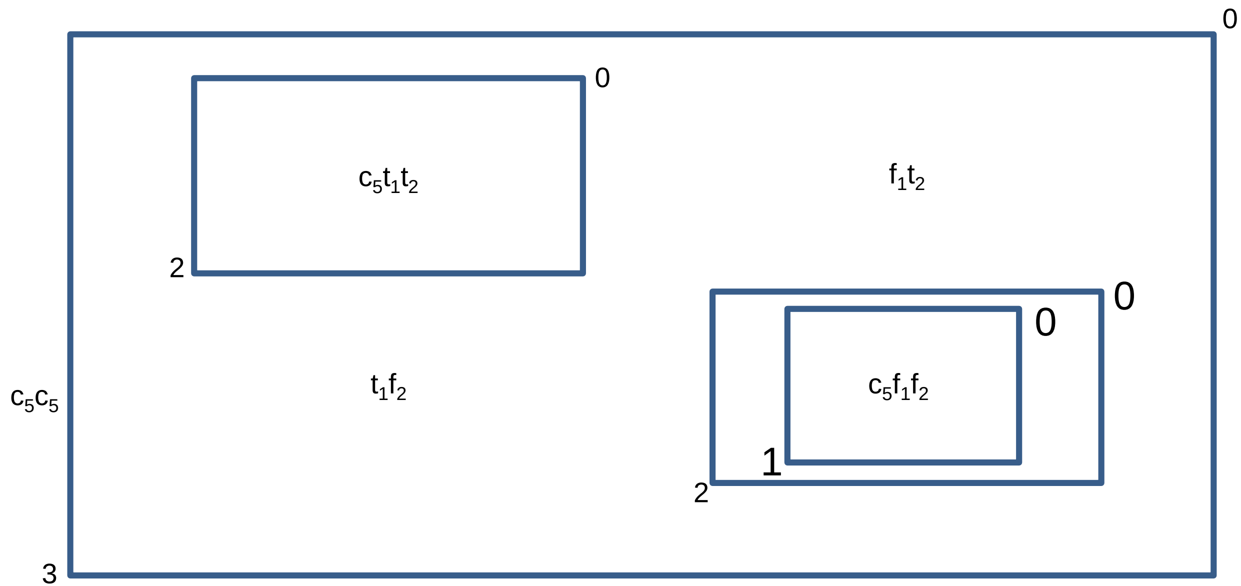


Solving SAT problem using P system:

$\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



Solving SAT problem using P system:



- Membranes which satisfy the conditions dissolve
- NP-complete problem thus can be solved in linear time using P system

Types of P Systems

- Three main types of P system
  - Cell-like P systems
  - Tissue-like P systems
  - Neural-like P systems

Types of P Systems

- Cell-like P systems
  - Imitates the cell and basic membrane structure
  - Objects described by symbols or strings and multisets of objects places in compartments
  - Rules maintained-- Rewriting rules, transport rules, transition rule and string processing rule

Types of P Systems

- Tissue-like P systems
  - One membrane cells evolving in a common environment
  - Both cells and environment contain objects
  - Cells communicate directly or through the environment
  - Channels are given in advance or dynamically established (*population P-systems*)

Types of P Systems

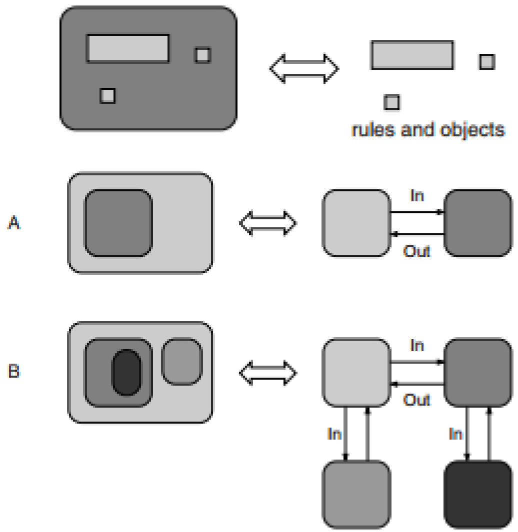
- Neural-like P systems
  - Basically two types.
    - Tissue like neural P systems
    - Spiking neural P systems
  - Tissue like neural P systems inspired by neurons and have a state which controls evolution
  - Spiking neural P system uses only one object, *“spike”* and main information to work with is distance between consecutive spikes

Efficiency of P System Computation

- P systems are powerful (most classes are Turing complete) and efficient (contains enhanced parallelism)
- Speed-up obtained by trading space for time
- Exponential workspace--Membrane creation, separation and string replication
- Investigations with complexity of time and space

Research and Future of P Systems

- Hardware Implementation:
  - Petreska and Teuscher's hardware implementation
    - Fundamental features of P systems
    - Reaction rules applied sequentially in region
    - One level of parallelism

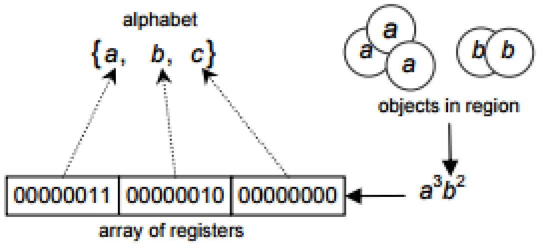


"it is important to underline the fact that "implementing" a membrane system on an existing electronic computer cannot be a real implementation, it is merely a simulation. As long as we do not have genuinely parallel hardware on which the parallelism [...] of membrane systems could be realized, what we obtain cannot be more than simulations, thus losing the main, good features of membrane systems"

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Research and Future of P Systems

- Hardware Implementation:
  - Reconfigurable Hardware (Reconfig-P)
    - V. Nguyen *et al*. implemented on ASIC design
    - Better performance than software based microprocessors
    - Source code generator and an FPGA



Major Application of a P System

- Biology
  - Modeling of cells, tissues, neurons
  - Modeling biological dynamics
- Computer science
  - As another computing system
  - Cryptography, Computer graphics, Optimization problem etc.

Concluding Thoughts

- Compartmentalization
- Non-deterministic and maximally parallel application
- Polynomial or linear solutions to NP-complete problems