7th Workshop on Membrane Computing (WMC7) Leiden, 17-21 July 2006

Identifying P rules from membrane structures with an error-correcting approach

José M. Sempere and Damián López
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

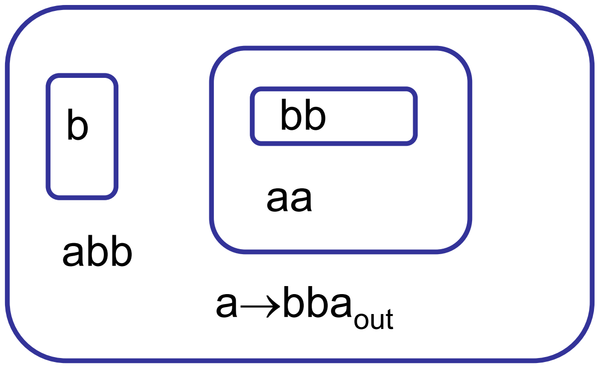
http://www.dsic.upv.es/users/tlcc/tlcc.html

Outline

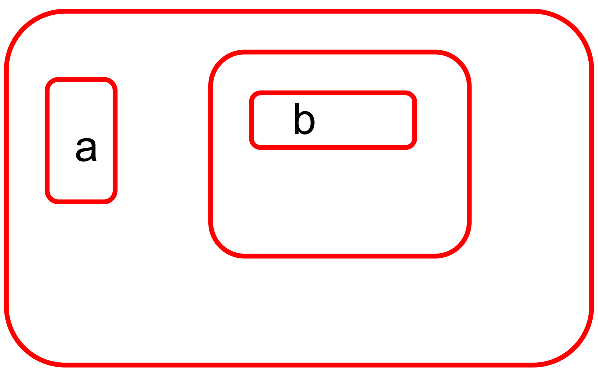
1 Basic concepts: Membrane structures and multiset tree automata
2 Basic concepts: Grammatical Inference
3 Inferring membrane structures with an error-correcting approach
4 Conclusions and future research

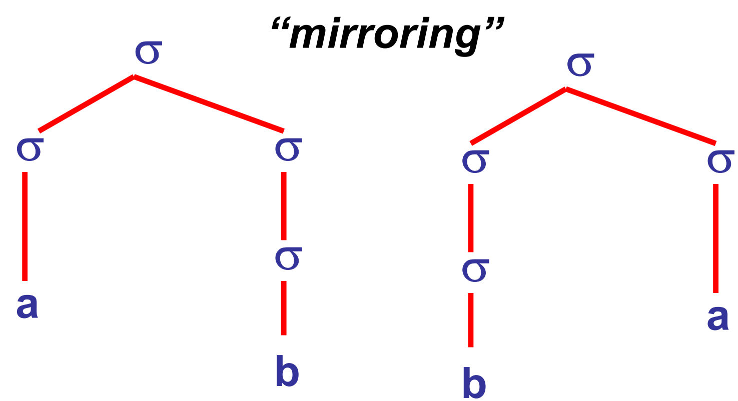
P systems and membrane structures

$\Pi = (V, T, C, \mu, w_1, \ldots, w_m, (R_1,\rho_1), \ldots, (R_m\ \rho_m), i_0)$



$\Pi = (V, T, C, \mu, w_1, \ldots, w_m, (R_1,\rho_1), \ldots, (R_m\ \rho_m), i_0)$





Multiset Tree Automata (MTA)

$\prod = (Q, V, \delta, F)$

V is a *ranked alphabet* $(V' \times N)$

$V_i = \{ a \in V' : (a,i) \in V \}$

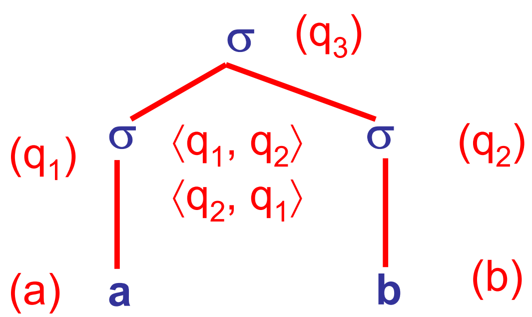
$\delta_i: V_i \times M_i (Q \cup V_0) \to P (M_1(Q))$

$M_i(\cdot)$ is a *bounded multiset*
(the sum of elements is i)

$\delta_0(a) = M_\Psi(a) \in M_1 (Q \cup V_0)$

$$\delta = \bigcup_{1 \le i \le maxarity(V)} \delta_i$$

Any multiset tree automaton performs a bottom-up parsing over any tree

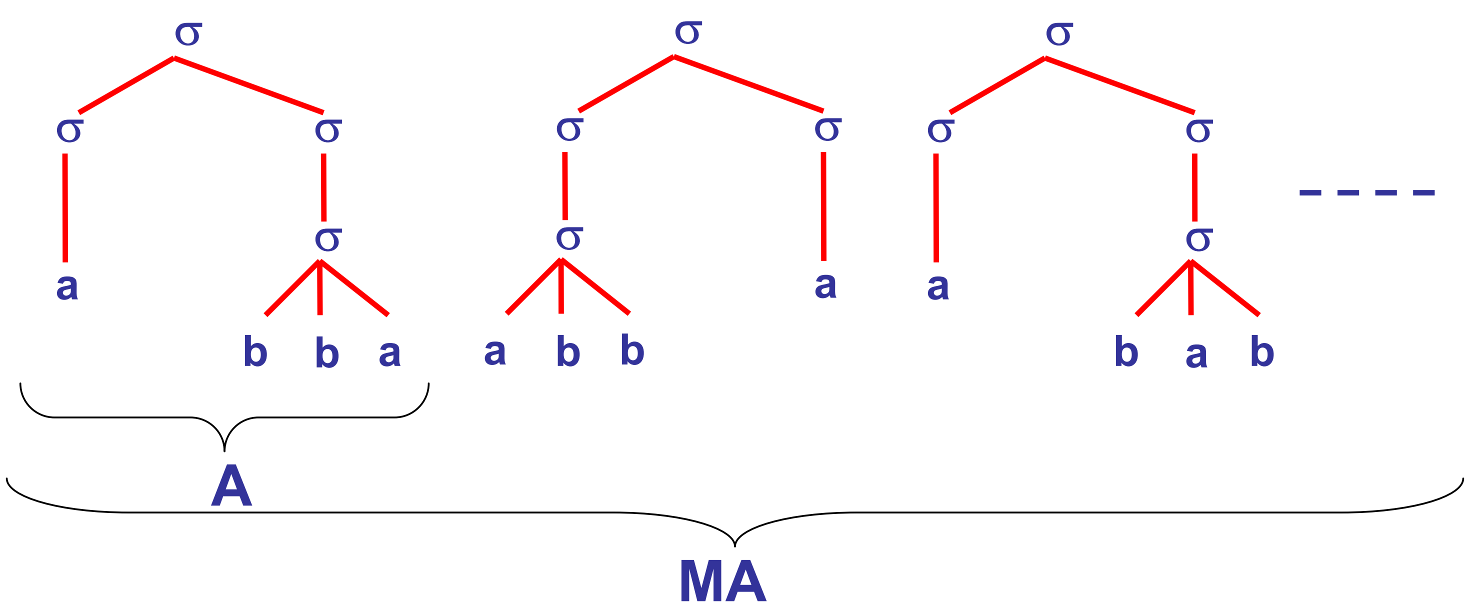


$\langle q_3 \rangle \in \delta_2(\sigma, \langle q_1, q_2 \rangle)$

Multiset Tree Automata (MTA)

Any tree automaton A induces a multiset tree automaton MA such that if t ∈ L(A) then t ∈ L(MA)

For any tree automaton A, its induced multiset tree automaton MA accepts all the trees that A accepts together with all their "*mirrored*" trees



Multiset Tree Automata (MTA)

The membrane structures of any P system can be accepted/regulated by a multiset tree automaton

MTA + **Editing distances** ⇒ **Membrane structure regulation**

- **Insertion of a state/leaf** → **Membrane creation**

  $a \rightarrow [_h\ b\ ]_h$

- **Deletion of a (sub)tree** → **Membrane dissolving**

  $[_h\ b\ ]_h \rightarrow a$

- **Substitution of a tree by a state** → **Membrane dissolving + creation**

Initial automaton → **Initial configuration**

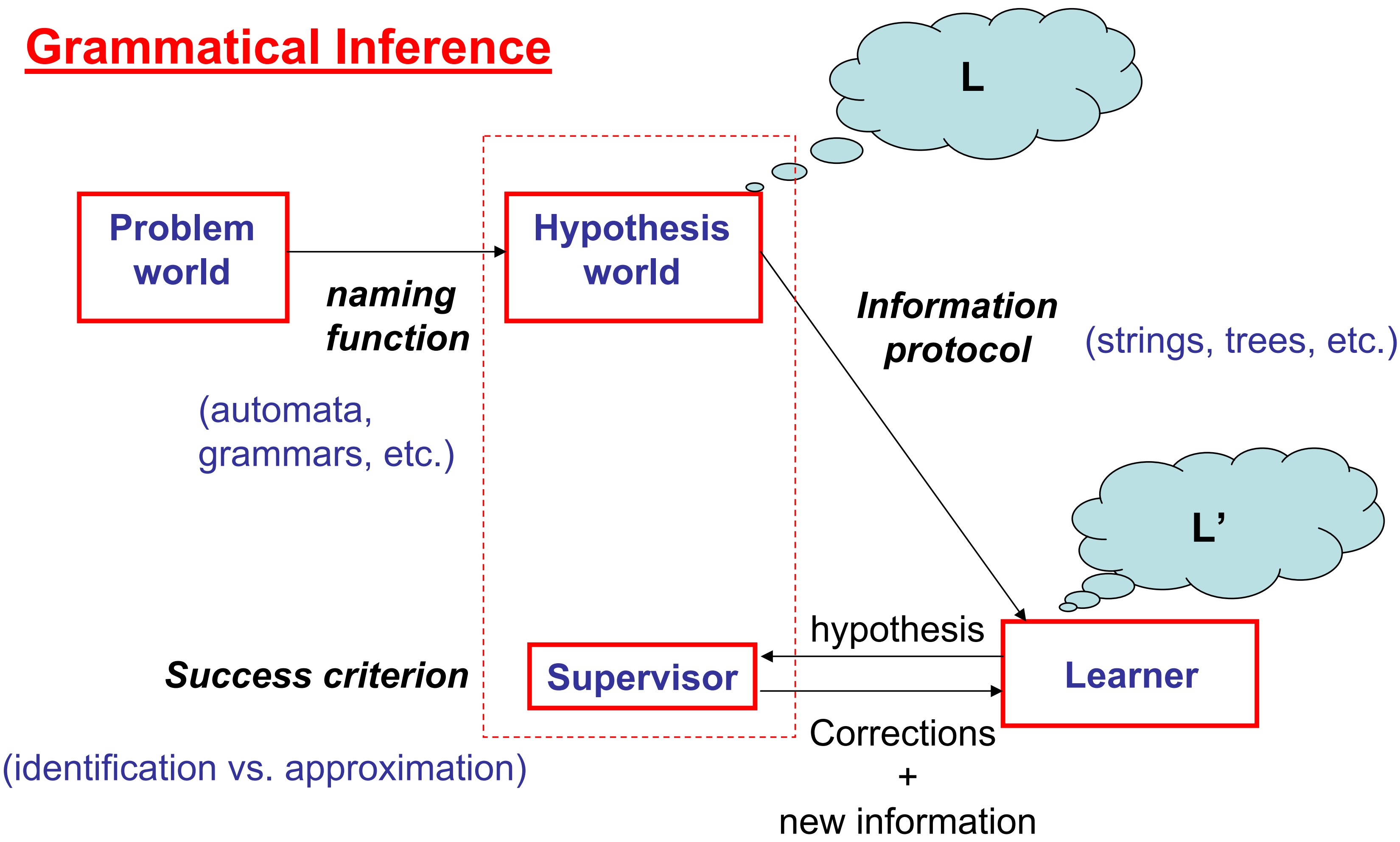
The membrane structure identification problem (MSI)

Let $\prod$ be an arbitrary P system. Let $M=\{\mu_1, \mu_2, \ldots, \mu_n\}$ be a (finite) set of (ordered) configurations of the membrane structure of $\prod$.

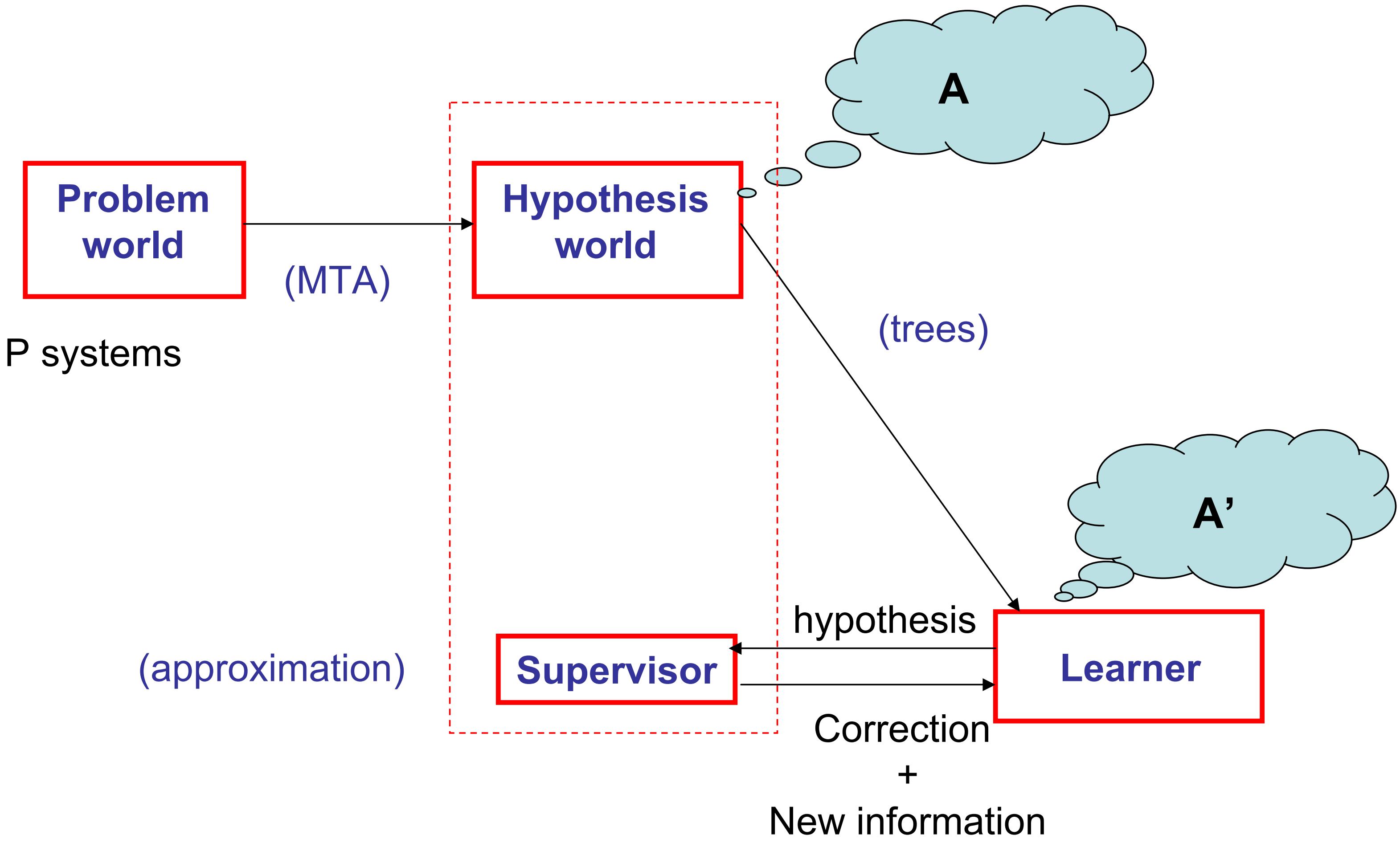
Q: What is the (minimum) set of rules for $\prod$ to generate the membrane structures of M ?

The MSI problem can be solved as a Grammatical Inference Problem

Grammatical Inference



Grammatical Inference of membrane structures



Error-Correcting Grammatical Inference for the MSI problem

Input: A finite sample $S = \{t_1, t_2, \ldots, t_n\}$ (membrane representations)

Output: A multiset tree automaton (MTA) consistent with the sample (this can be converted into a "dummy" P system)

Method:

Construct a MTA A for t_1
Remove the example from S
repeat
    A = *Expand*(A, t_i)
    Remove t_i from S
until $S=\emptyset$

Expand(A,t_i)

Input: A MTA A with a tree t_i

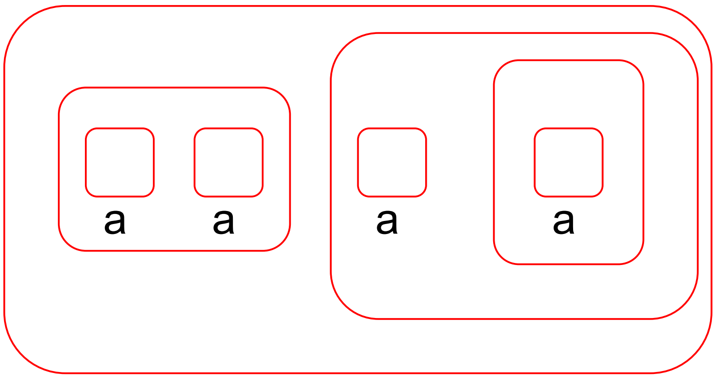
Output: A MTA A' consistent with L(A) $\cup$ { t_i }

Method:

- Make an error-correcting analysis for t_i to A (*see Ref.*)
- Obtain the minimum cost path for the edition of the tree, Δt_i
- Add the transitions which make A consistent with Δt_i and obtain A'

Ref. D. López, J.M. Sempere; Editing distances between membrane structures
Proc. of the 6th International Workshop on Membrane Computing (WMC6)
R. Freund, Gh. Paun, G. Rozenberg, A. Salomaa (eds.)
LNCS 3850, pp 326-341. Springer. 2006.

A short example (I)



$t_1 = \sigma(\sigma(a,a),\sigma(a,\sigma(a)))$

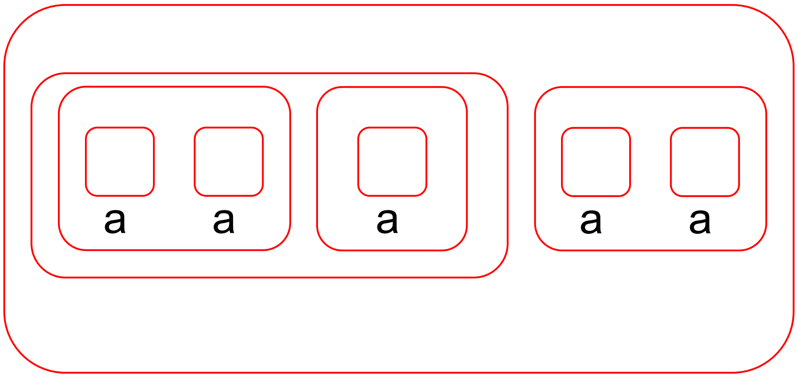
A:

$\delta(\sigma,aa)=q_1$
$\delta(\sigma,a)=q_2$
$\delta(\sigma,aq_2)=q_3$
$\delta(\sigma,q_1q_3)=q_4 \in F$

A short example (II)

A:

$\delta(\sigma,aa)=q_1$
$\delta(\sigma,a)=q_2$
$\delta(\sigma,aq_2)=q_3$
$\delta(\sigma,q_1q_3)=q_4 \in F$



$t_2 = \sigma(\sigma(\sigma(a),\sigma(a,a)),\sigma(a,a))$

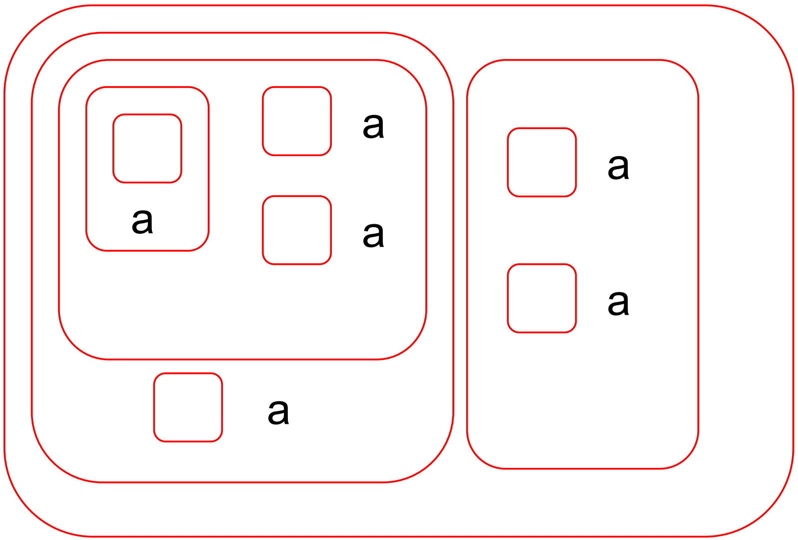
A':

$\delta(\sigma,aa)=q_1$
$\delta(\sigma,a)=q_2$
$\delta(\sigma,aq_2)=q_3$
$\delta(\sigma,q_1q_3)=q_4 \in F$
$\delta(\sigma,aa)=q_5$
$\delta(\sigma,q_2q_5)=q_6$
$\delta(\sigma,q_6q_1)=q_4 \in F$

	q_1	q_2	q_3	q_4
s_1	1	**0**	2	8
s_2	**0**	1	3	9
s_3	7	6	**4**	2
s_3	**0**	1	3	7
s_5	--	--	--	**4**

distance matrix

A short example (and III)



A:

$\delta(\sigma,aa)=q_1$
$\delta(\sigma,a)=q_2$
$\delta(\sigma,aq_2)=q_3$
$\delta(\sigma,q_1q_3)=q_4 \in F$
$\delta(\sigma,aa)=q_5$
$\delta(\sigma,q_2q_5)=q_6$
$\delta(\sigma,q_6q_1)=q_4 \in F$

$t_2 = \sigma(\sigma(\sigma(a,\sigma(a),a),a),\sigma(a,a))$

A':

$\delta(\sigma,aa)=q_1$
$\delta(\sigma,a)=q_2$
$\delta(\sigma,aq_2)=q_3$
$\delta(\sigma,q_1q_3)=q_4 \in F$
$\delta(\sigma,aa)=q_5$
$\delta(\sigma,q_2q_5)=q_6$
$\delta(\sigma,q_6q_1)=q_4 \in F$

$\delta(\sigma,a)=q_7$
$\delta(\sigma,aq_7a)=q_8$
$\delta(\sigma,aq_8)=q_3$

	q_1	q_2	q_3	q_4	q_5	q_6
s_1	1	**0**	2	8	1	6
s_2	2	**3**	1	7	2	5
s_3	6	5	**4**	6	6	6
s_3	**0**	1	3	9	0	7
s_5	--	--	--	--	--	**4**

distance matrix

Some remarks to ECGI

- ECGI performs an error-correcting inference approach so it is not a characterizable method (… what is the set of trees that ECGI can infer ?)

- ECGI has a running time which is polynomial with the size of the input sample. (it has a good performance for real/practical problems !!!)

- ECGI is order dependent (same input sample with different orderings produces different output)

Conclusions

- Rules for membrane modifications can be inferred from only observation of the membrane configurations.
- ECGI is an error-correcting technique, so it infers rules at the minimum editing cost in an efficient manner.

Future research

- Different methods to infer (multiset) tree automata have been proposed

reversible tree languages
testable tree languages
piecewise testable tree languages, etc.

We can adapt these methods to work with membrane structures in order to produce Multiset Tree Automata (+ editing operations)

What is the set of P systems whose membrane structures are defined by reversible/piecewise testable/testable/etc. Multiset Tree Automata ?

(A new characterization of P systems from the membrane structures)