# A Characterisation of **NL** Using Membrane Systems without Charges and Dissolution

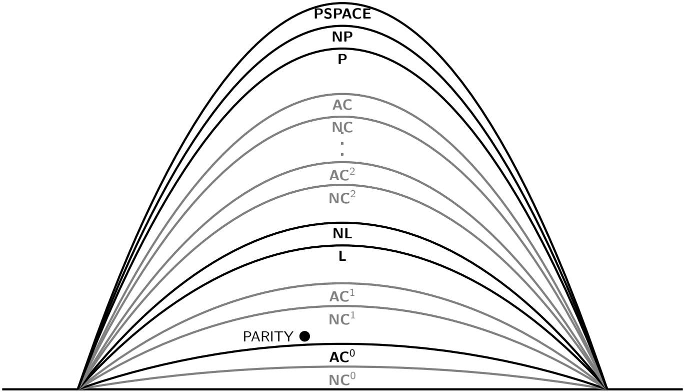
Niall Murphy[1] Damien Woods[2]

[1]Department of Computer Science, NUI Maynooth, Ireland.
Funded by the Irish Research Council for Science Engineering and Technology

[2]Department of Computer Science and Artificial Intelligence, University of Seville, Seville, Spain.
Funded by Junta de Andalucía grant TIC–581.

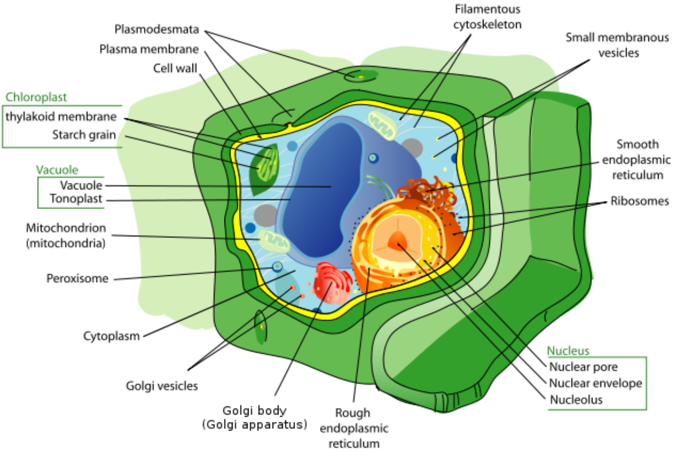
August 26, 2008

# The classes we will be discussing

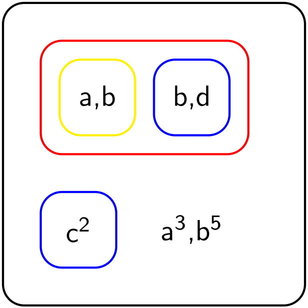



$$\mathbf{AC}^0 \subset \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC}^2 \subseteq \mathbf{P}$$

# Cell Models


Plasmodesmata
Plasma membrane
Cell wall
Chloroplast
thylakoid membrane
Starch grain
Vacuole
Vacuole
Tonoplast
Mitochondrion
(mitochondria)
Peroxisome
Cytoplasm
Golgi vesicles
Golgi body
(Golgi apparatus)
Rough
endoplasmic
reticulum
Filamentous
cytoskeleton
Small membranous
vesicles
Smooth
endoplasmic
reticulum
Ribosomes
Nucleus
Nuclear pore
Nuclear envelope
Nucleolus

# Membrane systems




$$[\ a\ ] \rightarrow [\ cccd\ ]$$
$$[\ a\ ] \rightarrow [\ d\ ][\ d\ ]$$
$$[\ c\ ] \rightarrow \lambda$$
$$[\ a\ ] \rightarrow [\ ]c$$
$$[\ ]d \rightarrow [\ a\ ]$$

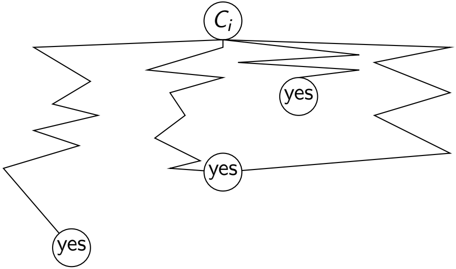
# Evolution rules of active membranes without charges

- Object evolution, type $(a)$, $\boxed{\text{a}} \longrightarrow \boxed{\text{bc}}$
- Communication in, type $(b)$, $\text{a}\boxed{\phantom{\text{a}}} \longrightarrow \boxed{\text{c}}$
- Communication out, type $(c)$, $\boxed{\text{a}} \longrightarrow \boxed{\phantom{\text{a}}}\text{c}$
- Dissolution, type $(d)$, $\boxed{\text{a}} \longrightarrow \text{c}$
- Elementary division, type $(e)$, $\boxed{\text{a}} \longrightarrow \boxed{\text{b}}\ \boxed{\text{c}}$
- Non-elementary division, type $(f)$, $\boxed{\boxed{\text{a}}\ \boxed{\text{b}}\ \boxed{\text{c}}} \longrightarrow \boxed{\boxed{\text{a}}\ \boxed{\text{c}}}\ \boxed{\boxed{\text{b}}\ \boxed{\text{c}}}$

# Recogniser membrane systems

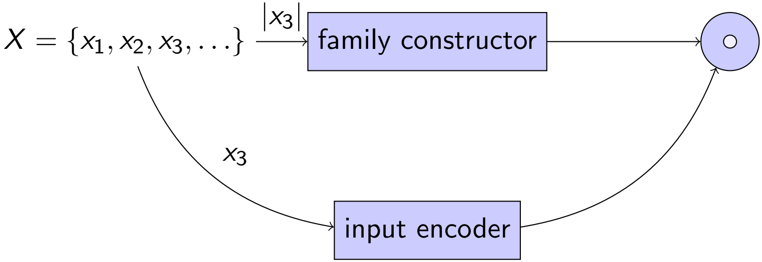
- Decide language problems
  - Have an *input* membrane
  - Produce the correct `yes` or `no` response to the given problem in polynomial time
- Maximally parallel
- Non-deterministic
- Computation is confluent

# Confluence, all computation paths are valid



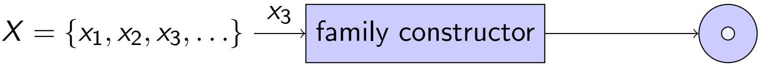
# P uniformity

- A membrane system is constructed for an input size by a polynomial time Turing machine.
- The specific instance input is encoded by a Turing machine in polynomial time.

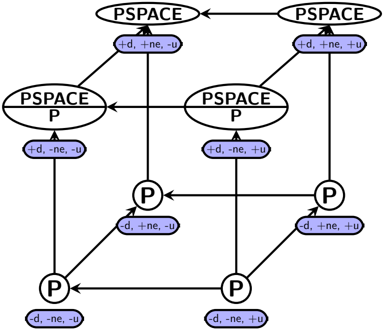


# P semi-uniformity

- A membrane system is constructed for each problem instance by a polynomial time Turing machine.
- Problem instance is hard coded into the membrane system.



# Previously with **P** uniformity

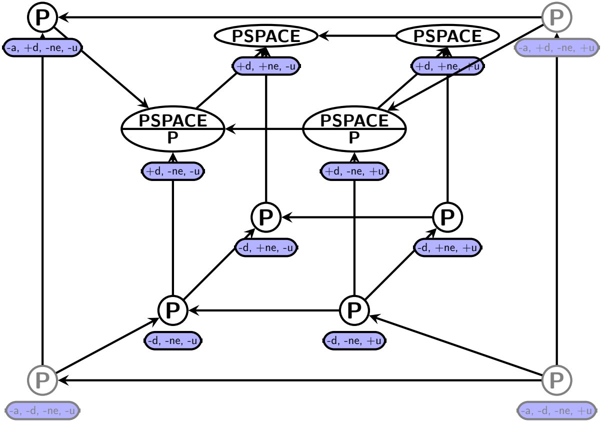



Artiom Alhazov and Mario Pérez-Jiménez. 2006

Petr Sosík and Alfonso Rodríguez-Patón. 2005

Miguel Gutiérrez-Naranjo, Mario Pérez-Jiménez, Agustín Riscos-Núñez, and Francisco Romero-Campero. 2006

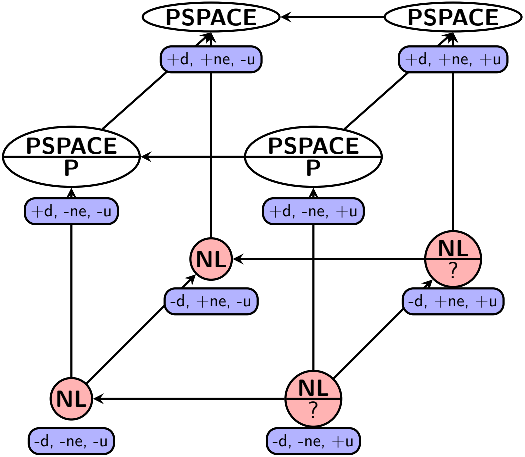
# Previously with **P** uniformity




Artiom Alhazov and Mario Pérez-Jiménez. 2006

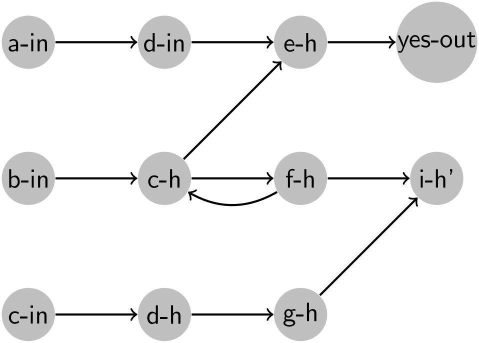
Petr Sosík and Alfonso Rodríguez-Patón. 2005

# Our result


PSPACE
PSPACE
+d, +ne, -u
+d, +ne, +u
PSPACE
P
PSPACE
P
+d, -ne, -u
+d, -ne, +u
NL
NL
?
-d, +ne, -u
-d, +ne, +u
NL
NL
?
-d, -ne, -u
-d, -ne, +u

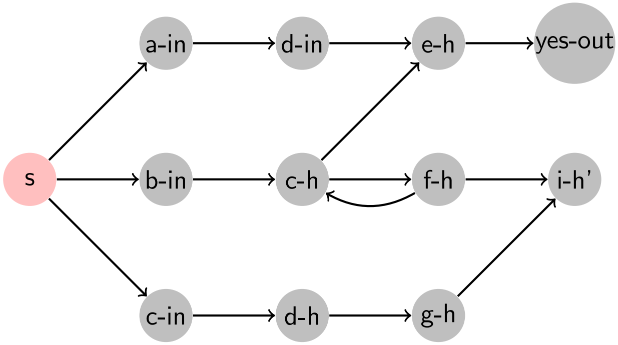
# An **NL** upper bound

- Reduced computation to PATH which is in **P**.
- We observe that PATH is more accurately **NL**-complete.
- This necessitates a tighter uniformity condition than **P**.



# An **NL** upper bound

- Reduced computation to PATH which is in **P**.
- We observe that PATH is more accurately **NL**-complete.
- This necessitates a tighter uniformity condition than **P**.

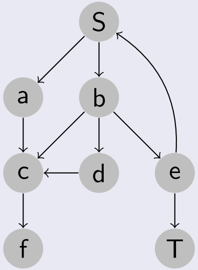


# Logspace uniformity

- We need a uniformity machine that can solve less than **NL**.
- $\mathbf{AC}^0 \subset \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC}^2 \subseteq \mathbf{P}$ .
- **Logspace** is the set of problems solved by a deterministic Turing machine with a constant number of binary counters.

# An **NL** lower bound for semi-uniform systems

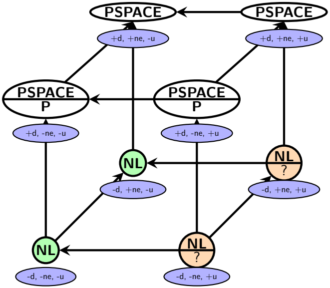
## An instance of PATH

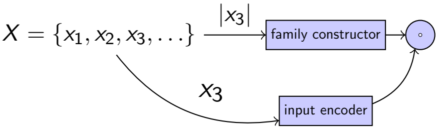



## Rules to recognise this PATH instance

- $[S \rightarrow a, b]$
- $[a \rightarrow c]$
- $[b \rightarrow c, d, e]$
- $[c \rightarrow f]$
- $[d \rightarrow c]$
- $[e \rightarrow T]$
- $[e \rightarrow S]$
- $[T \rightarrow yes]$

# Now about semi-uniform systems



# An attempt at a lower-bound for uniform families

- Same set of rules for all problem instances of the same size.
- Lets try to solve PARITY.

# An attempt at a lower-bound for uniform families

- Same set of rules for all problem instances of the same size.
- Lets try to solve PARITY.

- One symbol must encode whole input string.
- There are $2^n$ strings input strings of length $n$.
- Only $n^k$ possible objects...
- Some sort of preprocessing (encoding) can help?

# Uniform families, PARITY lower bound

## Rules to solve all **sorted** instances of PARITY with input length 4

- $[even_{0000} \rightarrow yes]$
- $[odd_{0000} \rightarrow no]$
- $[even_{0001} \rightarrow odd_{0000}]$
- $[odd_{0001} \rightarrow even_{0000}]$
- $[even_{0011} \rightarrow odd_{0001}]$
- $[odd_{0011} \rightarrow even_{0001}]$
- $[even_{0111} \rightarrow odd_{0011}]$
- $[odd_{0111} \rightarrow even_{0011}]$
- $[even_{1111} \rightarrow odd_{0111}]$
- $[odd_{1111} \rightarrow even_{0111}]$

## Huzzah!

- Great, but. . .

# Huzzah!

- Great, but. . .
- . . . such a sorting is in $\mathbf{NC}^1$.
- PARITY $\in \mathbf{NC}^1$.

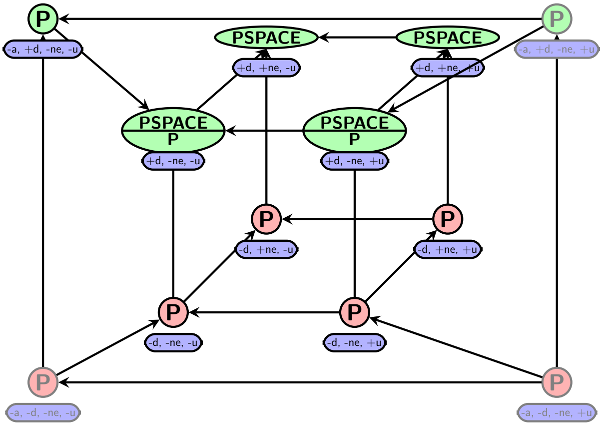
# $\mathbf{AC}^0$ uniformity

- It is known that PARITY $\notin \mathbf{AC}^0$ so is a good choice for uniformity conditions.
- $\mathbf{AC}^0 \subset \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC}^2 \subseteq \mathbf{P}$
- $\mathbf{AC}^0$ circuits are DLOGTIME uniform polynomial sized (in input length $n$), constant depth, circuits with AND, OR, and NOT gates, and unbounded fan in.
- $\mathbf{AC}^0$ is equivalent to constant time PRAM with polynomial processors, and First Order Logic.

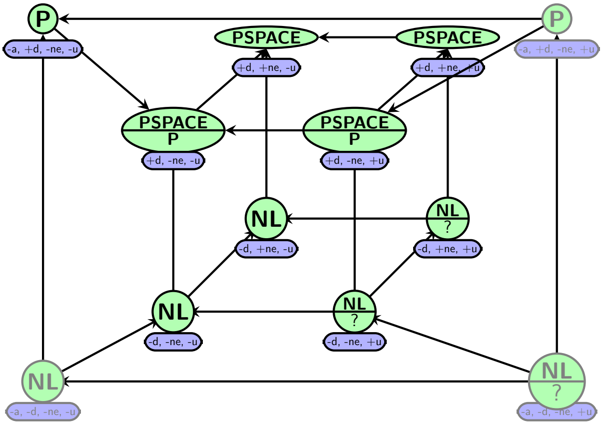
# Some results still hold under $\mathbf{AC}^0$ uniformity

- **PSPACE** characterisations for non-elementary division still hold with $\mathbf{AC}^0$-uniformity.
- The symmetric **P** characterisation still holds.
- The asymmetric **P** lower bound still holds.

# Some results don't hold with $\mathbf{AC}^0$ uniformity



# Some results don't hold with **AC**$^0$ uniformity



# In Summary

- We introduce $\mathbf{AC}^0$ and **L** uniformity conditions.
- We have tightened the upper bound of $\mathbf{PMC}^{S}{}_{\mathcal{AM}^0(-d,\pm ne)}$, from **P** to **NL**.
- **P** uniformity was too strong for a **P** characterisation.
- When we use a weaker uniformity condition the true power of the system is found, **NL**.
- Some **PSPACE** and **P** characterisations are still valid.

# Future work

- What is happening with uniform families?
- What is the power of dissolution?
- What other classes can we characterise? $\mathbf{AC}^k$, **NP**, **PH**?
- What is the relation between membrane systems and graph and circuit problems?
- Can we tighten other membrane systems results by using weaker uniformity conditions? e.g. Spiking Neural systems.