Simulating EXPSPACE Turing machines using P systems with active membranes

Artiom Alhazov[1,2] Alberto Leporati[1] Giancarlo Mauri[1]
Antonio E. Porreca[1] Claudio Zandron[1]

[1]Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca

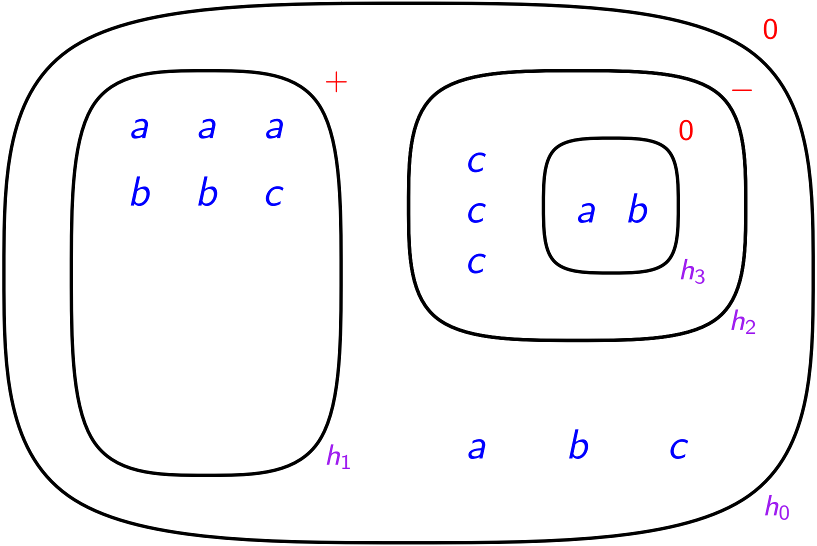
[2]Institute of Mathematics and Computer Science
Academy of Sciences of Moldova

13th Italian Conference on Theoretical Computer Science
19–21 September 2012, Varese, Italy

Outline

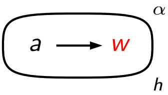
1. P systems
2. Space complexity and the time-space trade-off
3. Simulating Turing machines
4. Conclusions and open problems

P systems (with restricted elementary active membranes)

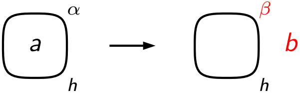


Computation rules

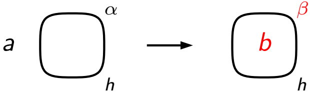
Object evolution



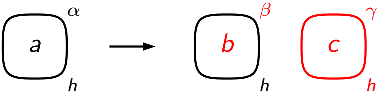
Communication (send-out)



Communication (send-in)



Elementary division

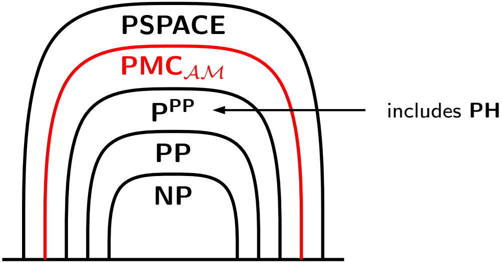


Parallelism and efficiency

- The rules are applied in a maximally parallel way
- There may be nondeterminism, but we require confluence
- Division may create exponentially many processing units in linear time
- So we can solve hard problems in polynomial time by trading space for time

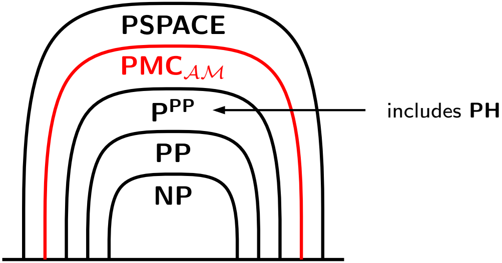
Parallelism and efficiency

- The rules are applied in a maximally parallel way
- There may be nondeterminism, but we require confluence
- Division may create exponentially many processing units in linear time
- So we can solve hard problems in polynomial time by trading space for time



Parallelism and efficiency

- The rules are applied in a maximally parallel way
- There may be nondeterminism, but we require confluence
- Division may create exponentially many processing units in linear time
- So we can solve hard problems in polynomial time by trading space for time



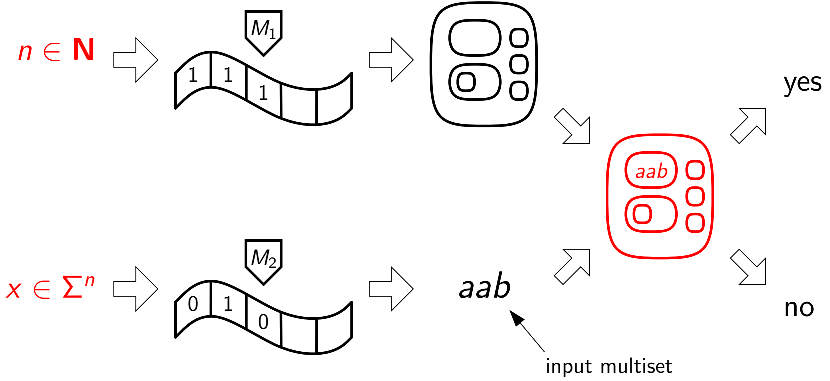
- Membrane division is provably needed

Uniformity

We decide membership in some language L by using a uniform family of P systems

Uniformity

We decide membership in some language L by using a uniform family of P systems



Formalising the time-space trade-off

What's the exact meaning of trading space for time?

$$\text{time} = \#\text{computation steps}$$

Formalising the time-space trade-off

What's the exact meaning of trading space for time?

time = #computation steps

space = #membranes + #objects

Known results on space complexity

Theorem

Polynomial-space P systems and polynomial-space Turing machines solve the same class of decision problems, namely **PSPACE**

Known results on space complexity

Theorem

Polynomial-space P systems and polynomial-space Turing machines solve the same class of decision problems, namely **PSPACE**

Proof.

- A family of P systems working in space $f(n)$ can be simulated by a TM working in $O(f(n) \log f(n))$ space

Known results on space complexity

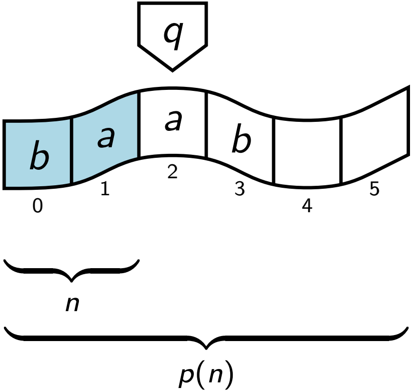
Theorem

Polynomial-space P systems and polynomial-space Turing machines solve the same class of decision problems, namely **PSPACE**

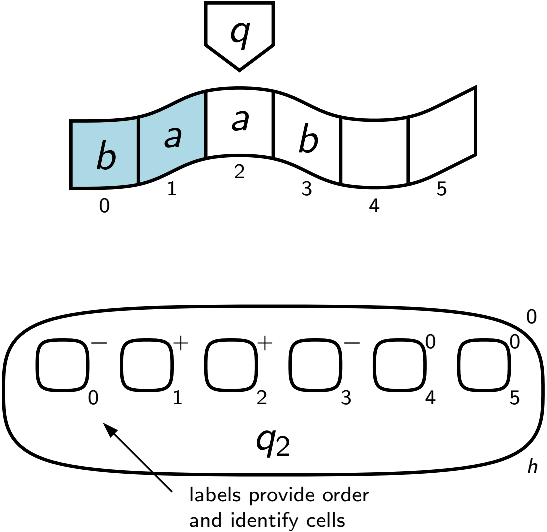
Proof.

- A family of P systems working in space $f(n)$ can be simulated by a TM working in $O(f(n) \log f(n))$ space
- A TM working in space $f(n)$ can be simulated by a family of P systems in space $O(f(n))$ as long as f is a polynomial □

Simulating polynomial-space TMs



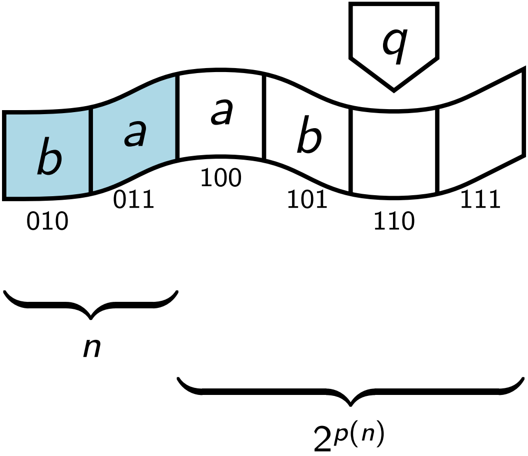
Simulating polynomial-space TMs



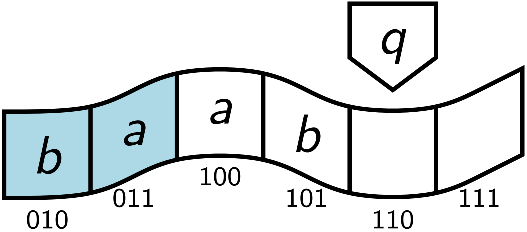
The problem with superpolynomial space bounds

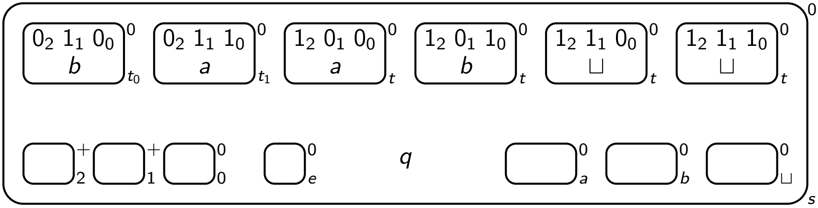
- We cannot use exponentially many labels (polytime uniformity)
- We must create the tape-membranes at runtime via membrane division
- But the new membranes are indistinguishable from the outside (they all have the same label)
- Solution: order and identify membranes according to their contents

Encoding exponential space configurations



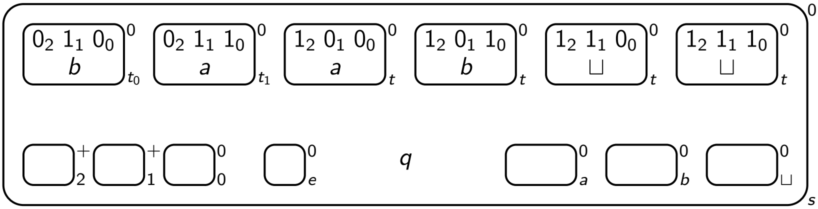
Encoding exponential space configurations





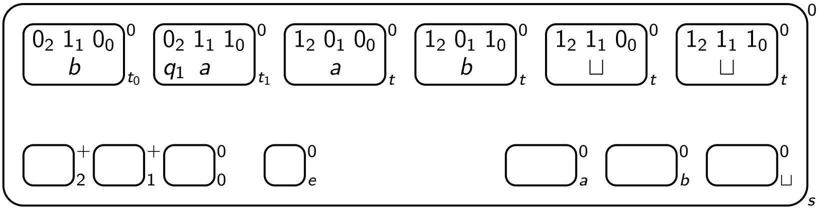
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



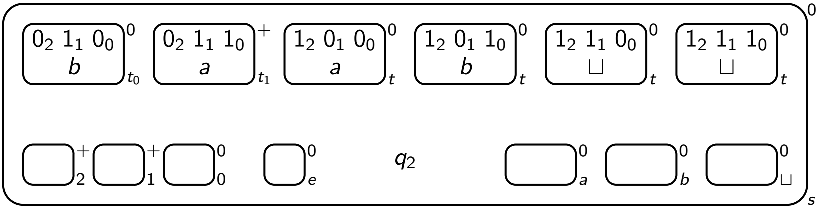
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



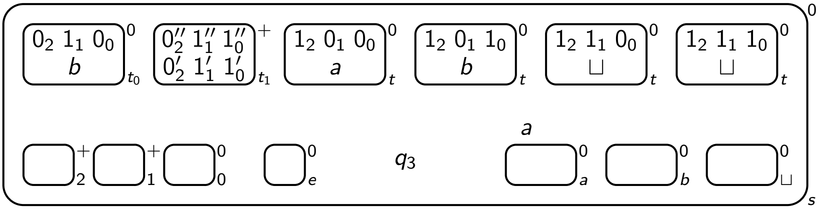
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



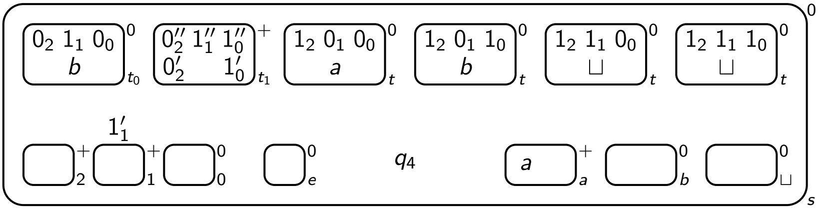
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



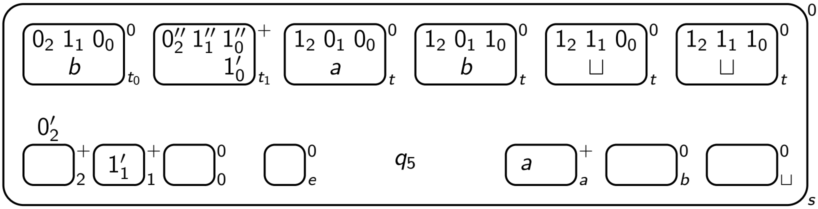
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



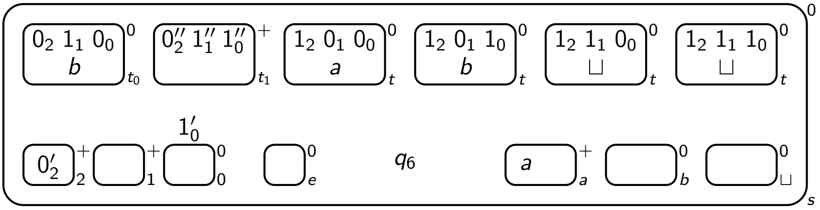
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



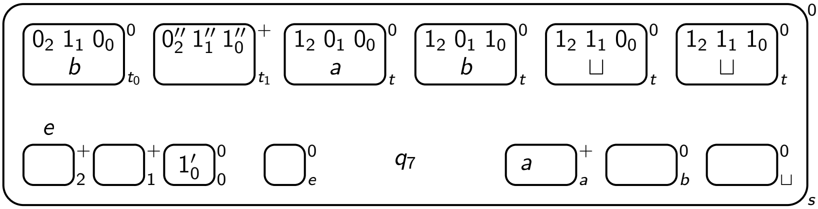
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



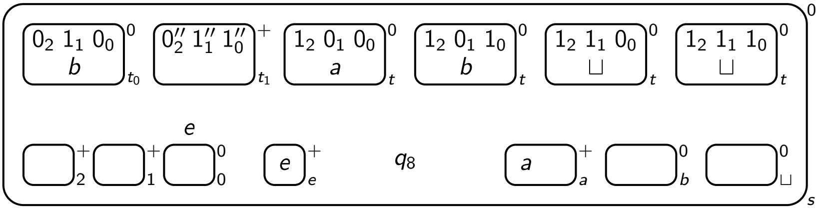
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



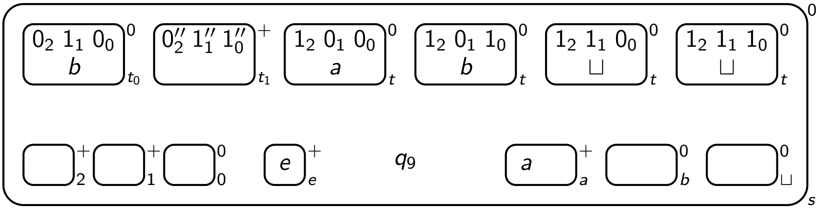
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



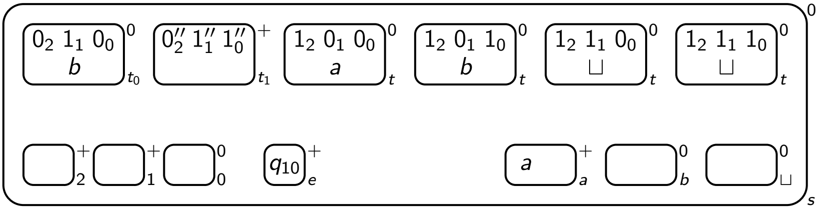
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



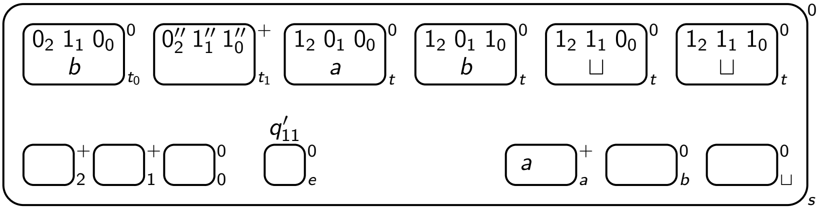
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



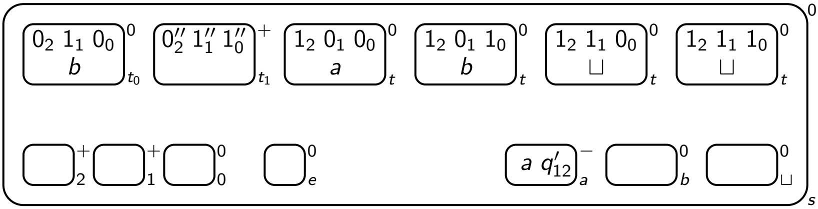
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



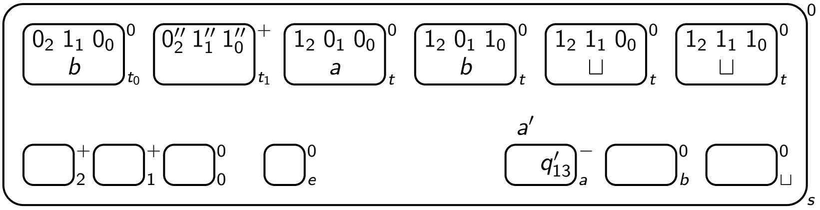
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



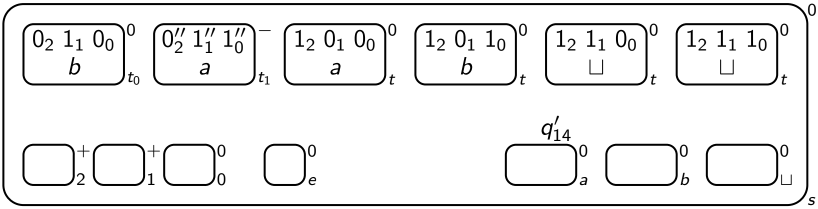
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



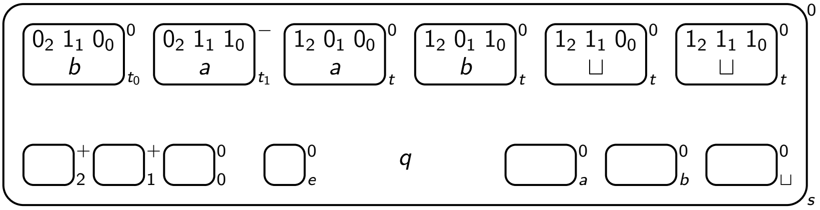
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



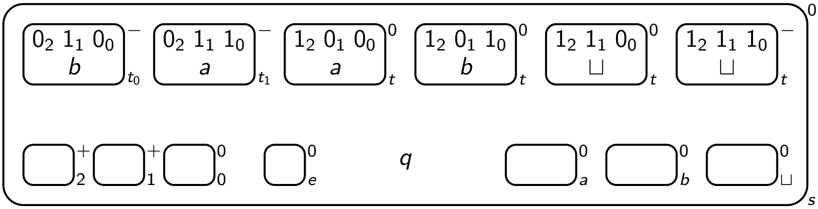
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



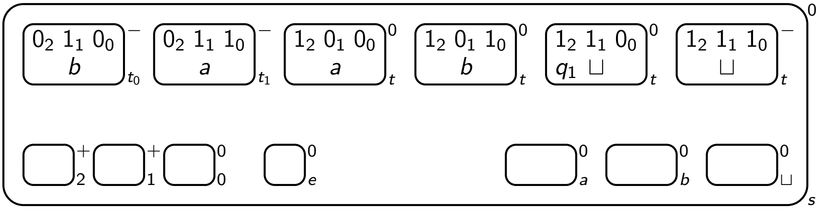
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



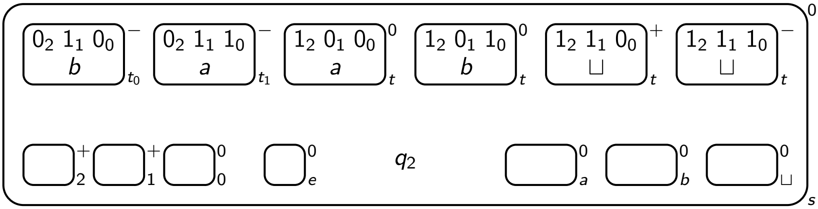
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



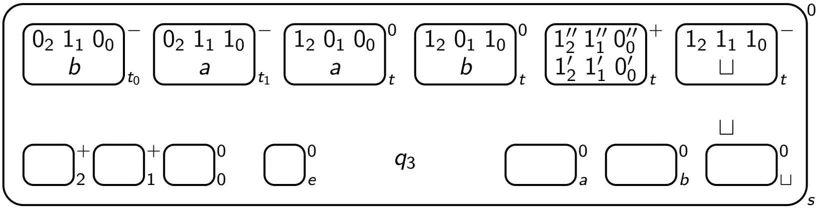
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



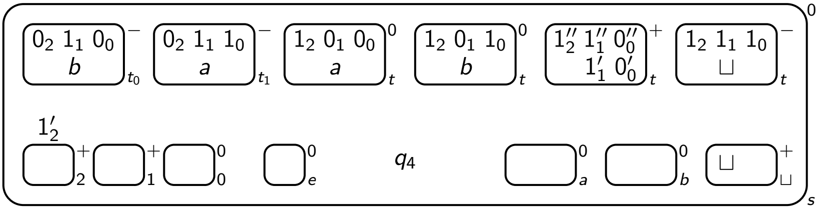
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



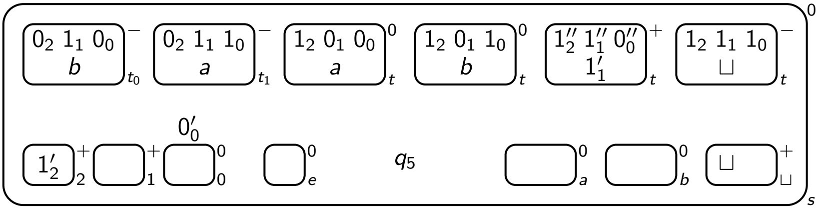
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



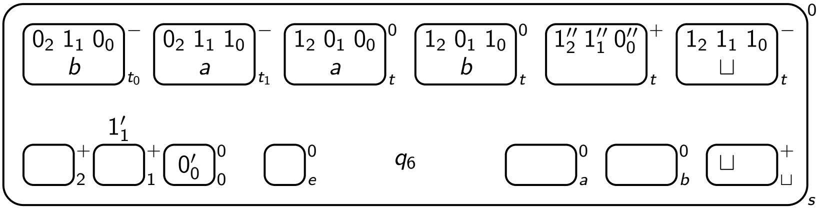
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



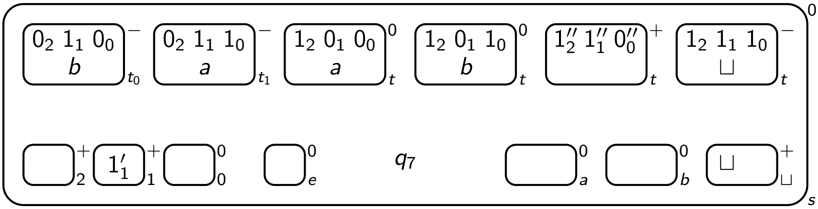
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



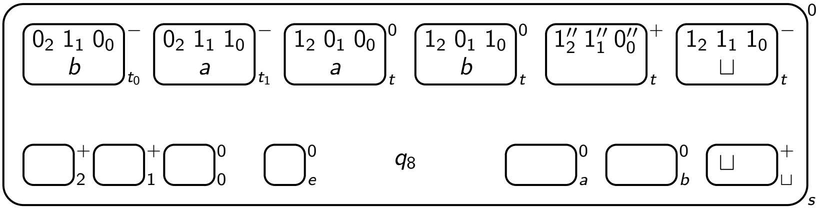
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



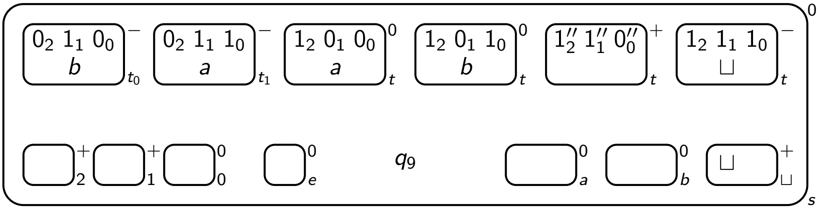
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



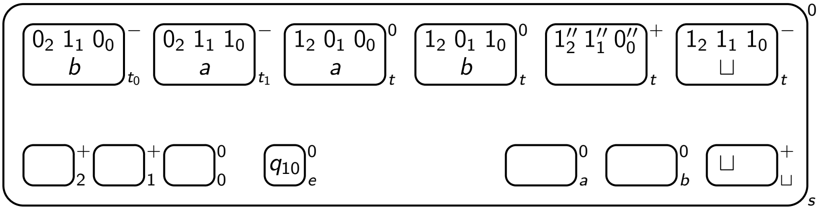
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



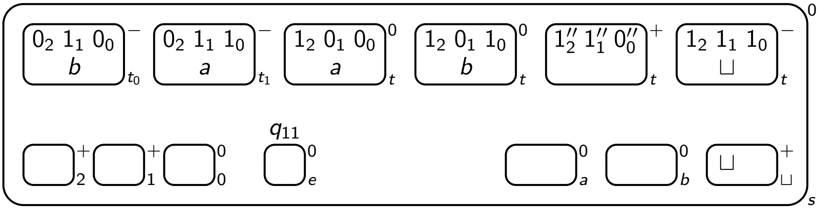
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



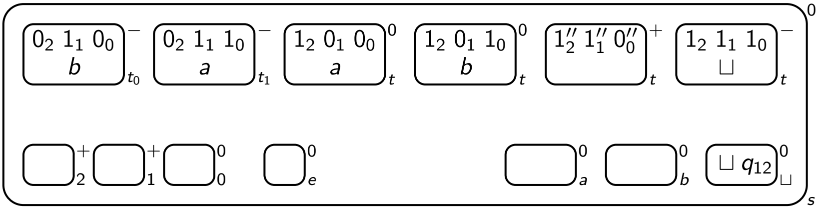
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



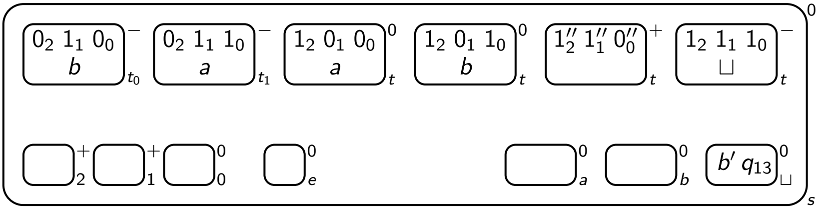
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



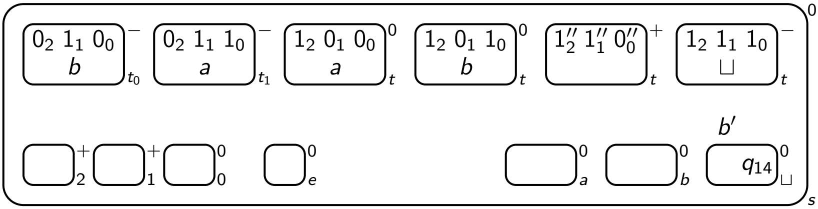
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



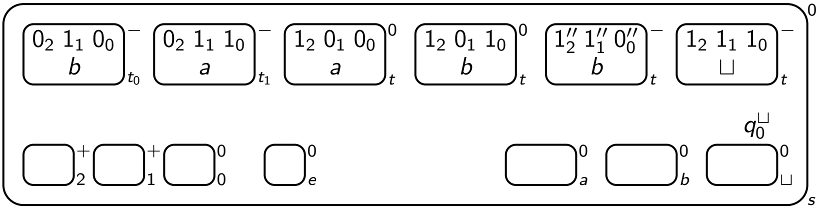
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



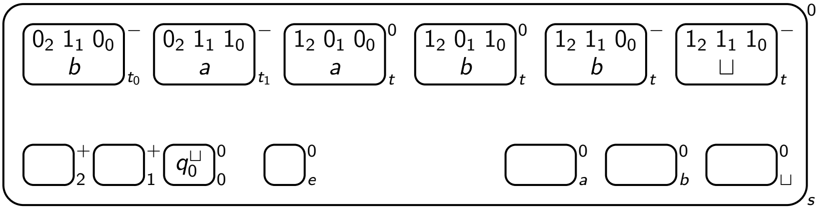
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



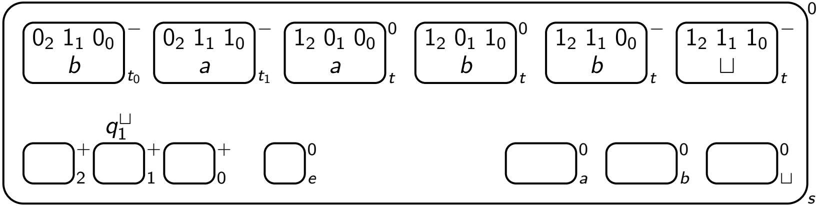
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



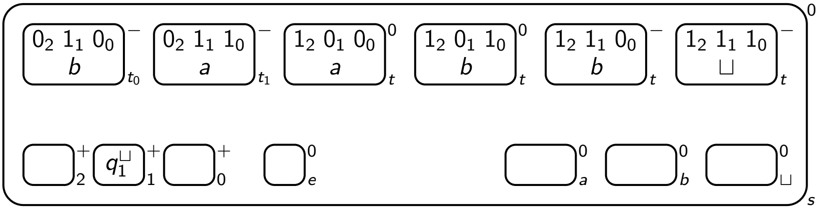
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



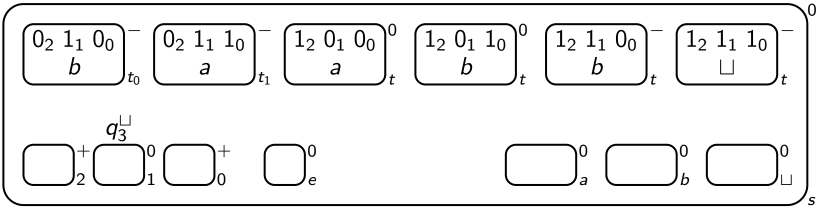
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



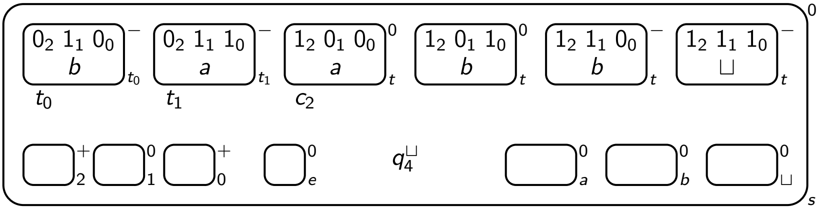
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



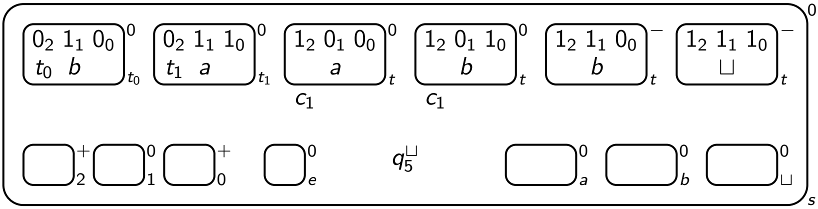
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



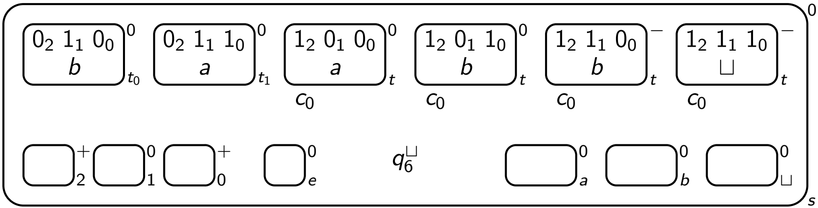
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \lhd)$$



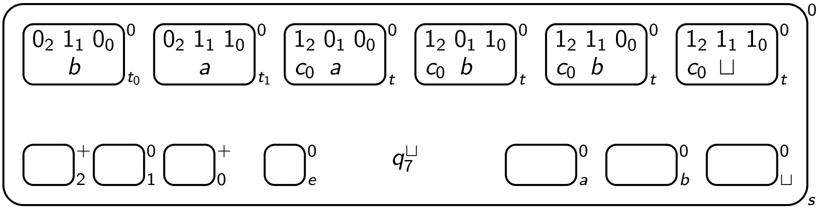
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



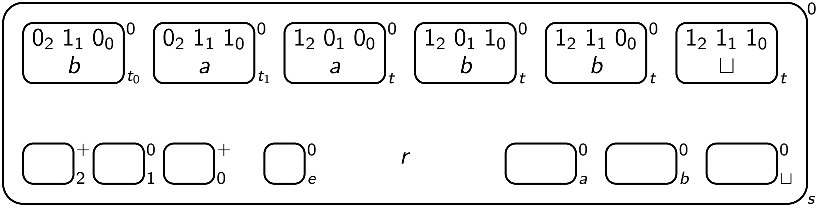
Simulating a computation step of the TM

$$\delta(q, \sqcup) = (r, b, \triangleleft)$$

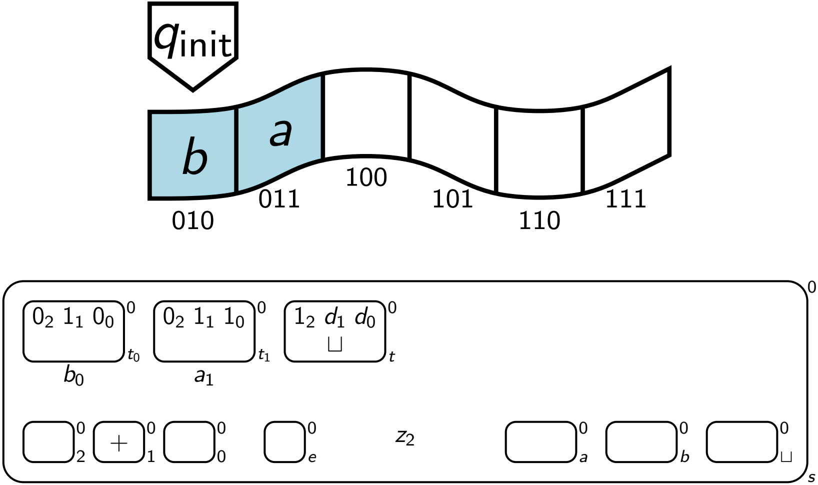


Simulating a computation step of the TM

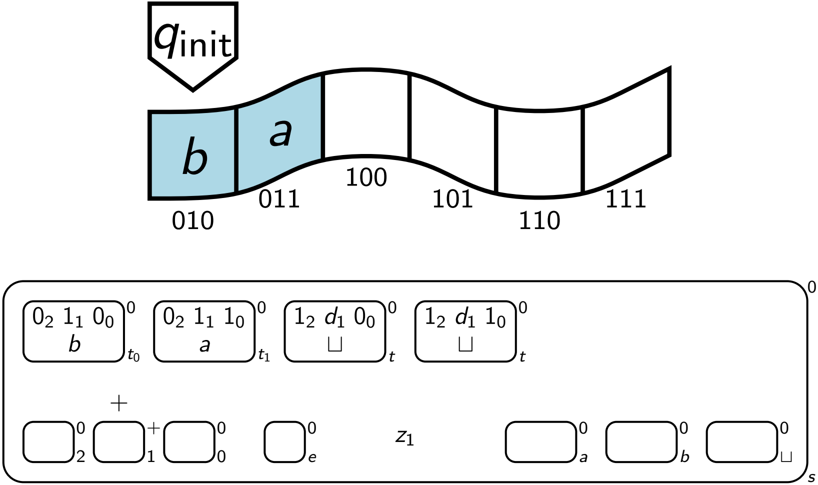
$$\delta(q, \sqcup) = (r, b, \triangleleft)$$



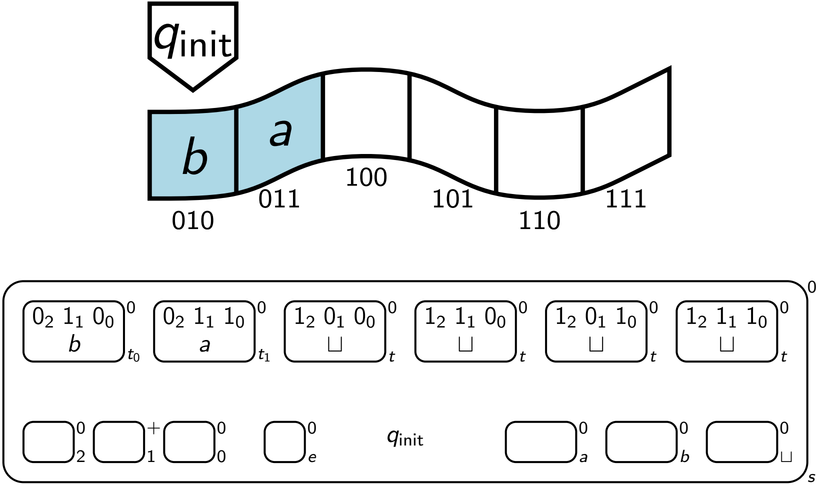
Initialising the system



Initialising the system



Initialising the system



Main result

Theorem

Let M be a single-tape deterministic Turing machine working in time $t(n)$ and space $s(n)$, where $s(n) \leq n + 2^{p(n)}$ for some polynomial p. Then there exists a uniform family of confluent P systems with restricted elementary active membranes $\mathbf{\Pi}$ operating in time $O\big(t(n)s(n)\log s(n)\big)$ and space $O\big(s(n)\log s(n)\big)$ such that $L(\mathbf{\Pi}) = L(M)$ □

Corollary

Both exponential-space TMs and exponential-space P systems solve exactly the problems in **EXPSPACE** □

Conclusions and open problems

- P systems and TMs have identical computing power both in polynomial and in exponential space
- Preliminary results for sub-polynomial space
  - A weaker uniformity condition is needed
- Is there a general equivalence in power wrt all space bounds?
  - Probably, though the simulation must be improved
  - Possible solution: unary encoding of tape cell numbers

Grazie per l'attenzione!

Thanks for your attention!