Communication gap for Finite Memory Devices

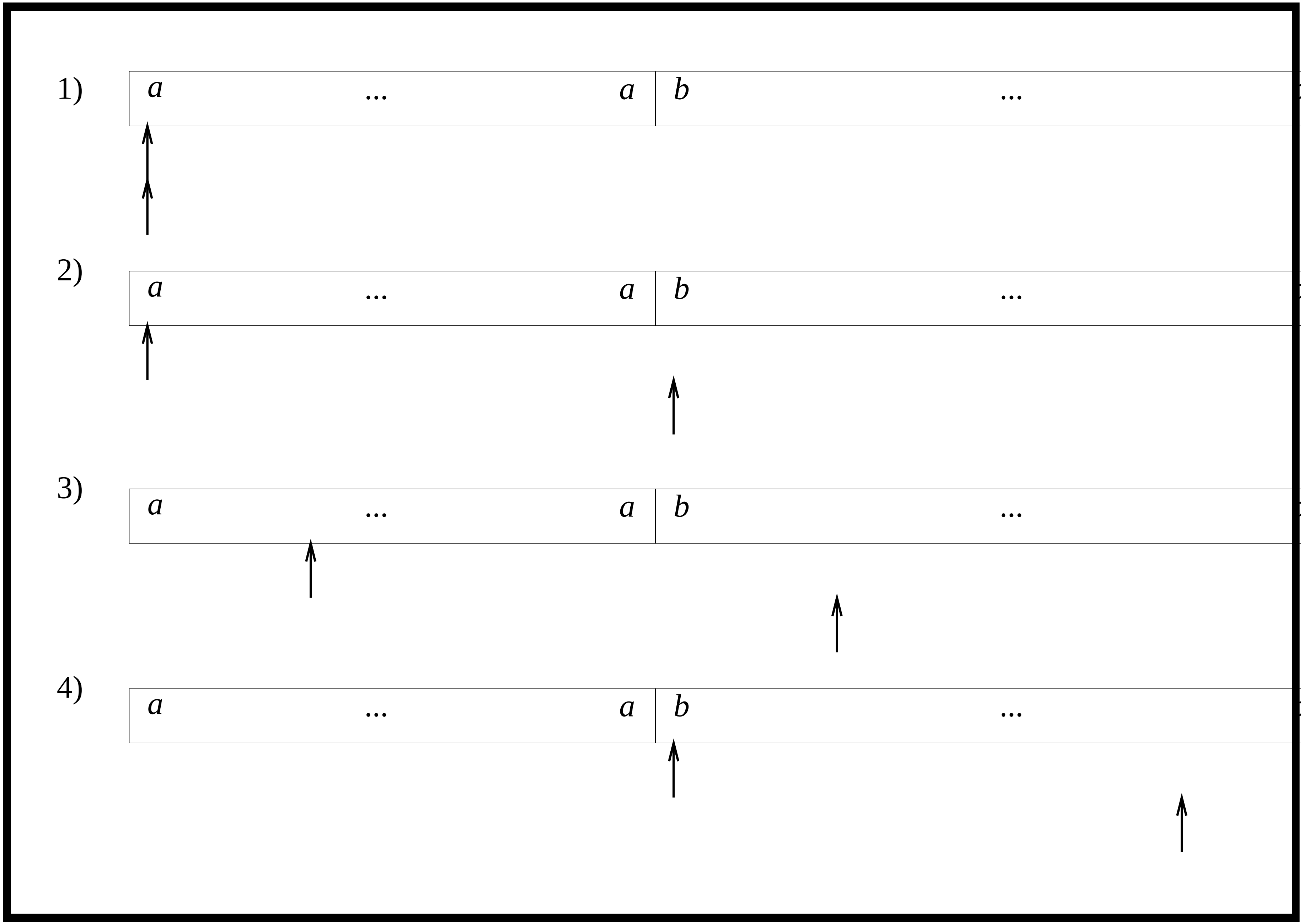
Tomek Jurdziński (Chemnitz and Wrocław)

Mirek Kutyłowski (Wrocław and Poznań)

The model

- shared read-only input string
- finite memory devices (finite automata) reading the input
- automata communicate by messages
- the number of messages crucial, not the length of the computation
- a single step: each automaton behaves according to its transition function

Recognizing language $L_0 = \{a^n b^n : n \in \mathbb{N}\}$:

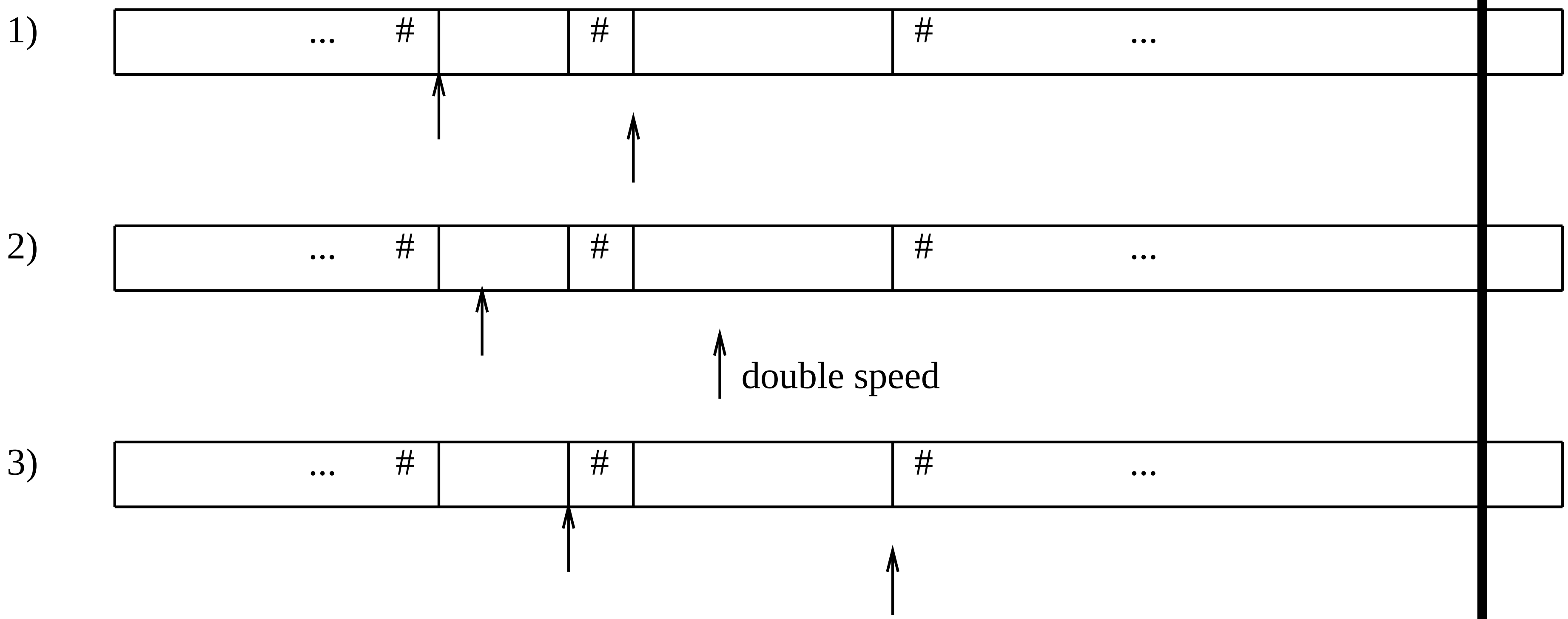


1)
a ... a b ... b
2)
a ... a b ... b
3)
a ... a b ... b
4)
a ... a b ... b

Recognizing language $L_1 = \{1^{2^0}\#1^{2^1}\#1^{2^2}\#\ldots\#1^{2^k} : k \in \mathbb{N}\}$:

to check: each block is twice as long as the previous one

number of messages = number of blocks = $O(\log n)$



Double-logarithmic number of messages

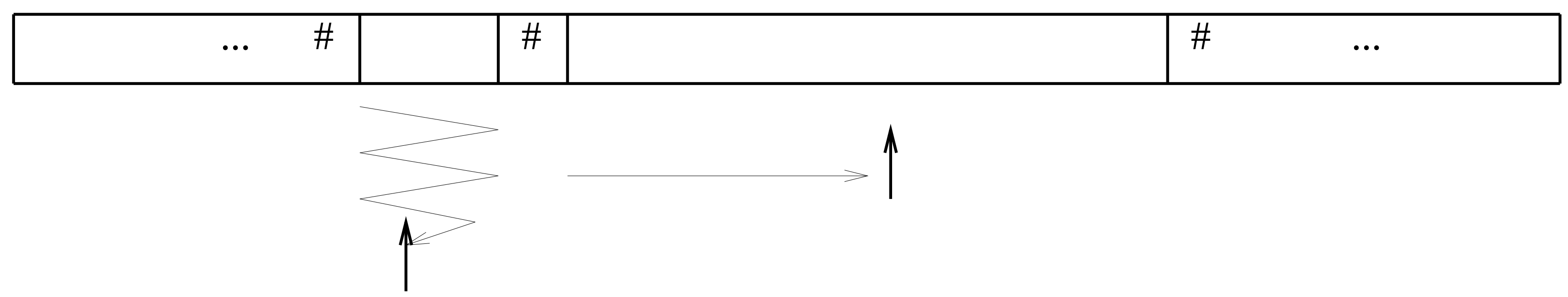
L_2 consists of words of the form

$$1^{f_1}\#1^{f_2}\#\ldots\#1^{f_k}$$

where $f_1 = 2, \quad f_2 = 3$ and

$$f_{i-1}|(f_i - 1), \quad f_{i-2}|(f_i - 1), \quad f_i > 1 \text{ for } i = 3, \ldots, k$$

- the number of blocks is $O(\log\log n)$ since $f_i \geq f_{i-1} \cdot f_{i-2}$
- checking relations between f_i and f_{i-1} and f_{i-2} requires $O(1)$ messages and two automata



Motivations

- communication complexity for shared data
  (the classical approach: data divided between protocl participants)
- limited memory for processing units
  (finite memory is an oversimplification but most results can be generalized)
- communication should be as small as possible
  (communication channels, power consumption, ...)

Message complexity classes

Language L belongs to MESSAGE$(f(n))$ if

> there is a system of finite automata that uses at most $f(n)$ messages on input x of length n and decides whether $x \in L$

Hierarchy results

Jurdziński, Loryś and myself, COCOON’99:

- there is a dense hierarchy of message complexity classes between

$$\log\log n \text{ and } n$$

- similar result for *one-way automata* for number of messages $\Omega(\log n)$
- there is a dense message complexity hierarchy of functions above n
- for a constant number of messages: even one more message counts!

... & Zatopiański, ’2001

- asynchronous systems require significantly more messages lower bounds that match performance of algorithms obtained by step by step synchronization!

Gap problem

Is the assumption on message complexity

$$f(n) = \Omega(\log\log n) \text{ and } f(n) = \Omega(\log n)$$

due to a weakness of proof techniques

or

this is not a concidence?

Remark: for the classical communication complexity there is no gap theorem

New Results

Theorem 1 *For $f(n)$ such that $f(n) = \omega(1)$ and $f(n) = o(\log n)$, there is no* one-way *system which requires $\Theta(f(n))$ messages.*

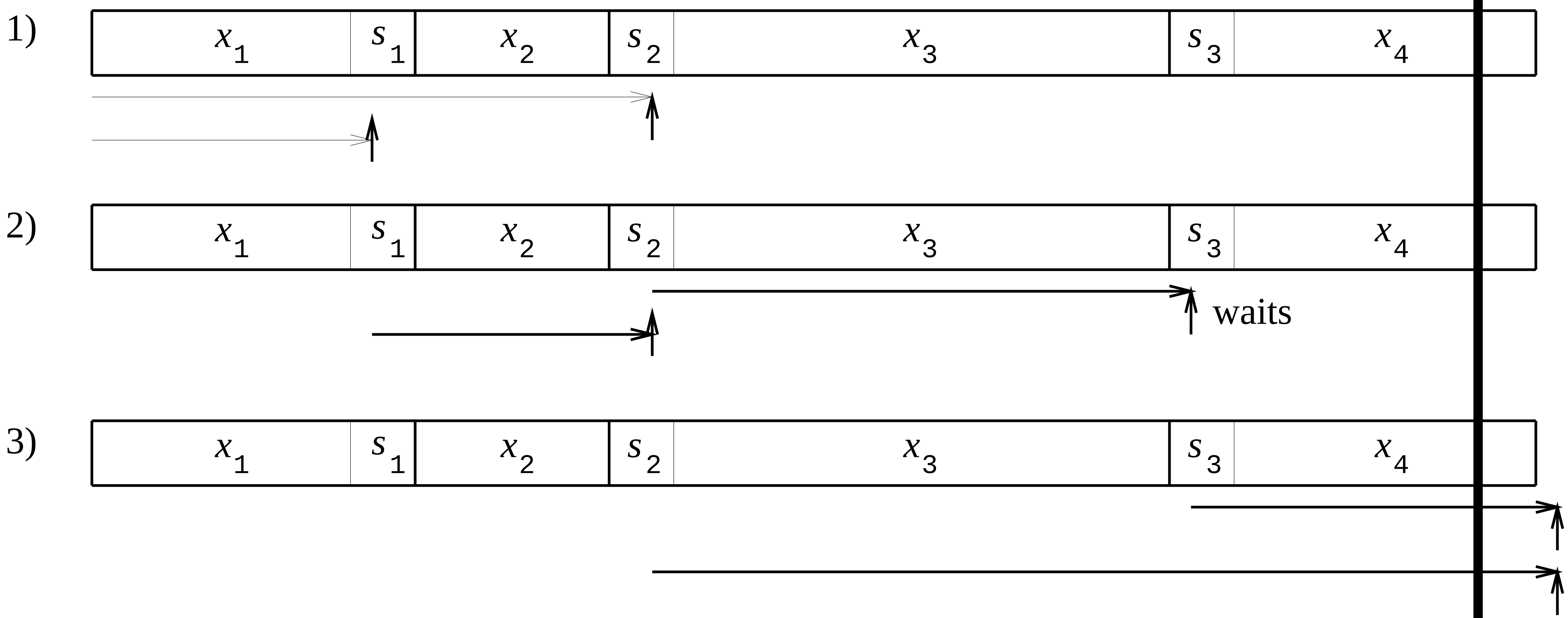
Theorem 2 *There is a constant c such that for $f(n) = \omega(1)$ and $f(n) = o((\log\log\log n)^c)$, there is no* two-way *system which requires $\Theta(f(n))$ messages.*

Proof techniques

1. establishing connection between behavior of systems of finite automata and systems of diophantine equations

2. minimal solutions for systems of diophantine equations $\Rightarrow$ short inputs with a given number of messages

Toy example



$t_{i,k}$ = time required by automaton k to read x_i

$$t_{11} + t_{21} = t_{12}, \quad t_{22} \geq t_{31}, \quad t_{41} = t_{32} + t_{42}.$$

Description of computation - diophantine systems

Idea:

- *silent blocks* with no communication and *communication positions*
- a computation may find relations between lengths of silent blocks
- variables denoting time spent by automata on a given silent block
  (depending on the initila state)

- computation $\Rightarrow$ integer solution for these variables

Technical Problems

Problems:

- behaviour of automaton inside a block depends on the block contents. The speed may vary!!

- on two-way systems: a block may be scanned many times before the second automaton decides to send a message system recognizing language L_2
  $\Rightarrow$ linear diophantine systems do not suffice to describe computations

- existance of integer solutions for systems of equations of degree 2 is already undecidable!

Graph characterization of one-way computation

nodes: states of automaton

edges: labelled by input symbols and time interval (how many steps of automaton are required to move to the left one position)

computation within a silent block: a path through the graph

time to traverse a silent block: sum of time labels along the path

Remark: to be done simultaneously for all automata at the same time

Analysis of the paths on the graph

- the graph is of a finite size,
- a path *loops in a certain sense*
- number of loops of each kind in a silent block determines traversal time

  $\Rightarrow$ variables denoting the number of loops of each kind describe the block in a sufficient way

... it is not that easy

Solution

- induction on the number of nodes in the graph
- combining the descriptions of the subpaths not going through a node s, entering and leaving s
  $\Rightarrow$ new, more complex diophantine systems

Two-way systems

- additional feature: looping over (many) silent blocks before another automaton send a message
- variables denoting the number of such loops
- divisibility relations necessary to describe where one automaton is when the second one sends a message
- $\Rightarrow$ systems of linear equations, inegualities and divisibility relations

Representation of computation via diophantine systems

- the number of variables and equations, inequalities, divisibilities is $O(g)$, where g is the number of messages
- each computation corresponds to an integer solution of the system
- **each integer solution of the system corresponds to an input and an computation on it**
- a *small* integer solution $\Rightarrow$ an input with the given number of messages, where time spent on each silent block is *small*

$\Rightarrow$ silent blocks are short

$\Rightarrow$ input is short

the number of messages is large with respect to the input length

Minimal solutions for linear diophantine systems

Theorem 3 *(von zur Gathen, Sieveking, 1978)*
Let A, b, C, d be respectively $m \times n$, $m \times 1$, $p \times n$, $p \times 1$ matrices with integer coefficients with absolute values bounded by a constant f. If there exists an integer solution x for $Ax = b$ and $Cx \geq d$, then there is a solution x' with absolute values of coefficients bounded by $2^{c_f n}$.

$\Rightarrow$ minimal solutions for our systems describing one-way systems of automata are exponential in the number of messages
$\Rightarrow$ existance of inputs for which the number of messages is log-

arithmic in the input length!

Diophantine systems with divisibilities

- not the general case of diophantine systems of degree 2
- **Theorem 4** *(Lipshitz, 1978)*
  Diophantine systems with divisibilities are decidable.
- proof direction of Lipshitz:
  show that an integer solution exists iff there is a solution in modular arithmetic for some large (but bounded) modulus
- our job: check how large is the integer solution constructed by the method of Lipshitz
  $\Rightarrow$ lower bound on two-way systems

Conclusions and open problems:

- a gap between the lower bound $\omega((\log\log\log n)^c)$ and the upper bound $O(\log\log n)$
- better estimations for minimal solutions of diphantine divisibilities?
- asynchronous systems: the gap might be larger!