The synchronizing probability function of an automaton

Raphaël Jungers

UCL/MIT

JM2010, Sept. 2010

Joint work with

Michel Goemans

Outline

- Synchronizing automata and Cerny’s conjecture
- Previous approaches

- A game theoretic/probabilistic approach
- Results

- Conclusion
- Discussion

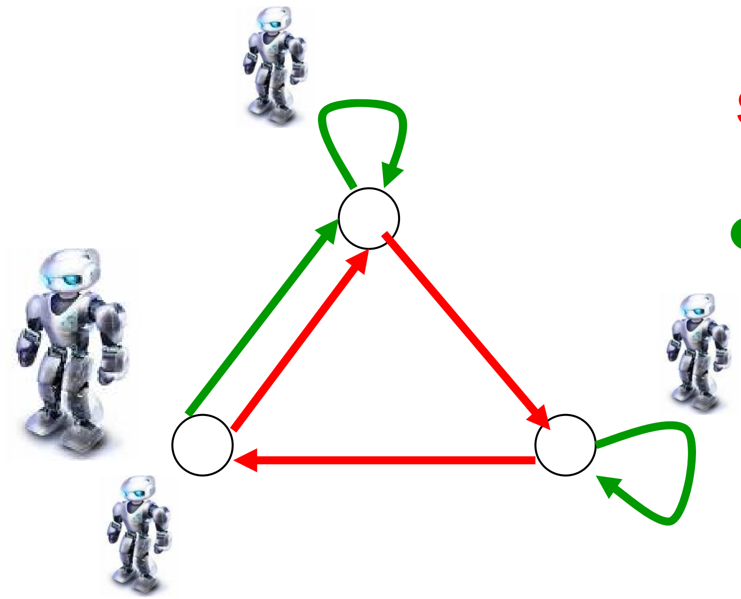
Outline

- Synchronizing automata and Cerny’s conjecture
- Previous approaches

- A game theoretic/probabilistic approach
- Results

- Conclusion
- Discussion

Synchronizing automata

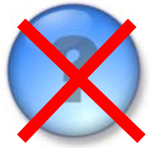


Synchronizing word
(or reset sequence)

Definition: A (complete deterministic) automaton is synchronizing if there is a sequence of colors such that all the paths compatible with this sequence end in the same node.

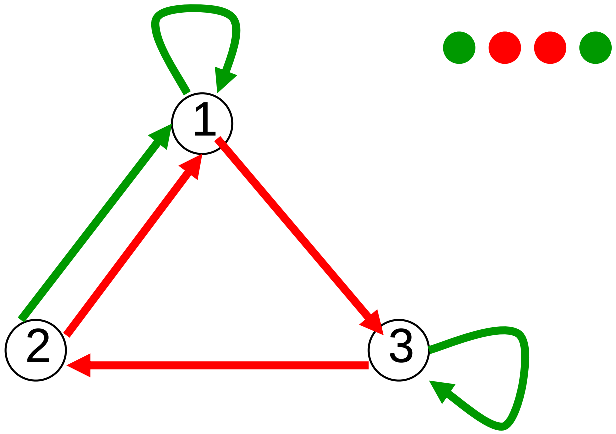
Cerny's conjecture (1964): If a graph is synchronizing, then it admits a synchronizing sequence of length at most $(n-1)^2$.

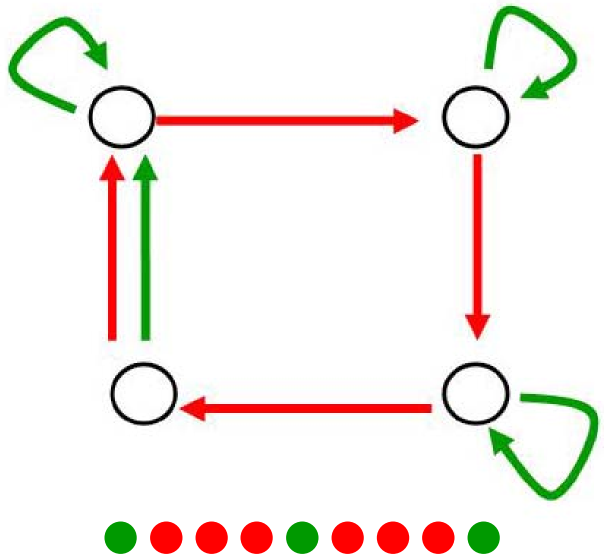
Connected with the Road coloring conjecture [1977 Adler et al.] [2007 Trahtman]



Synchronizing automata

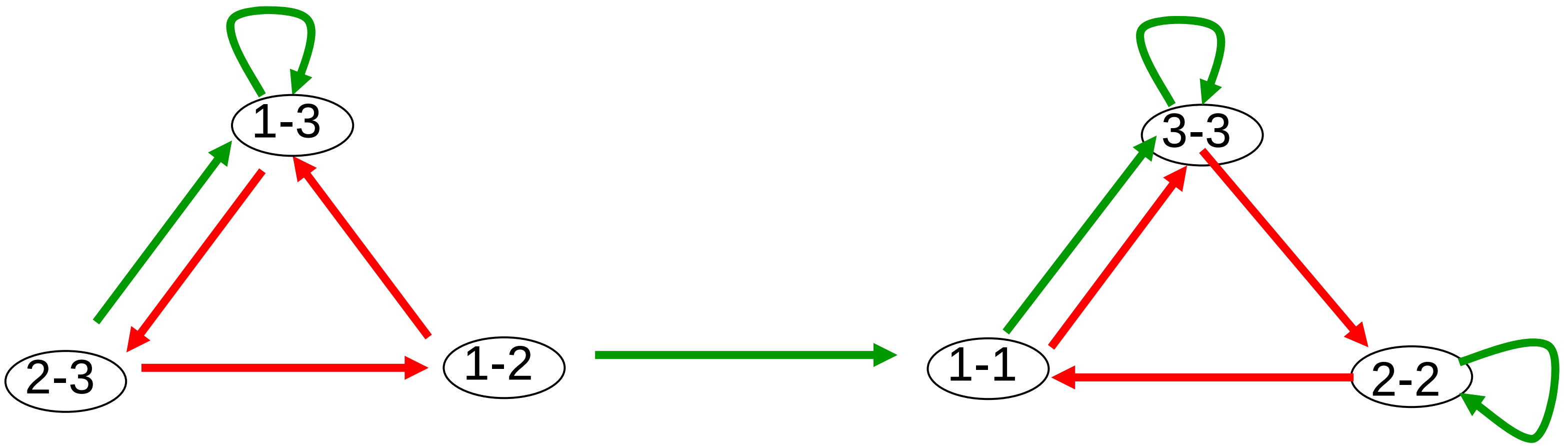
[Cerny, 1960's]





Theorem [1990 Eppstein]**:** Synchronizing graphs are Recognizable in polynomial time.

Length $(n-1)^2$



Outline

- Synchronizing automata and Cerny’s conjecture
- Previous approaches

- A game theoretic/probabilistic approach
- Results

- Conclusion
- Discussion

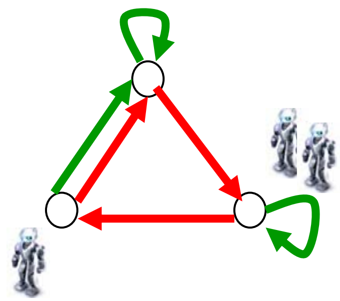
Previous approaches (1)

Cerny's conjecture (1964): If a graph is synchronizing, then it admits a synchronizing sequence of length at most $(n-1)^2$.

Known upper bounds on the shortest synchronizing word:

- [1964 Cerny] 2^n-n
- [1966 Starke] $n^3/2-3/2\ n^2+n+1$
- [1970 Kohavi] $n(n-1)^2/2$
- [1978 Pin] $7/27\ n^3 - 17/18\ n^2 + 17/6\ n - 3$
- [1982 Frankl (Pin)] $(n^3-n)/6$
  - Does not look at the graph, but enumerates subsets
  - Greedy
  - The best so far!

$(n^3 -n)/6$



Previous approaches (2)

Cerny's conjecture (1964): If a graph is synchronizing, then it admits a synchronizing sequence of length at most $(n-1)^2$.

- **Particular cases**

  - [1981 Pin] small rank (log(n)), circular of prime size
  - [1990 Eppstein] monotonic
  - [1998 Dubuc] circular
  - [2001 Kari] Eulerian
  - [2009 Trahtman] aperiodic
  - [2009 Beal Perrin] one-cluster
  - [2009 Carpi d'Alessandro] locally strongly transitive
  - [2009 Volkov] partial order-related
  - [2010 Steinberg] ...

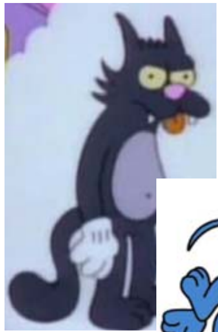
- **Complexity** issues
  - NP-hard [1990 Eppstein]
  - Apx-hard [2010 Berlinkov]

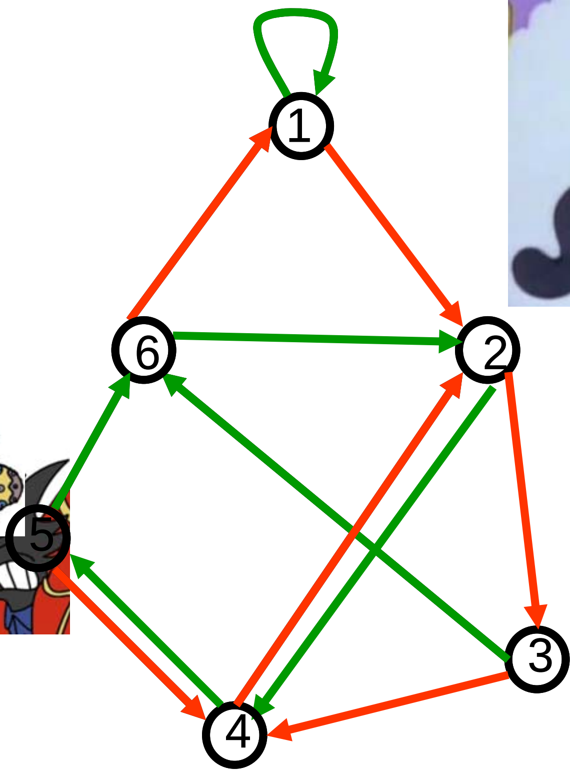
Outline

- Synchronizing automata and Cerny's conjecture
- Previous approaches
- A game theoretic/probabilistic approach
- Results
- Conclusion
- Discussion

A simple game

- Two players playing on a graph:
  the « mouse » and the « cat »
- A (here, t

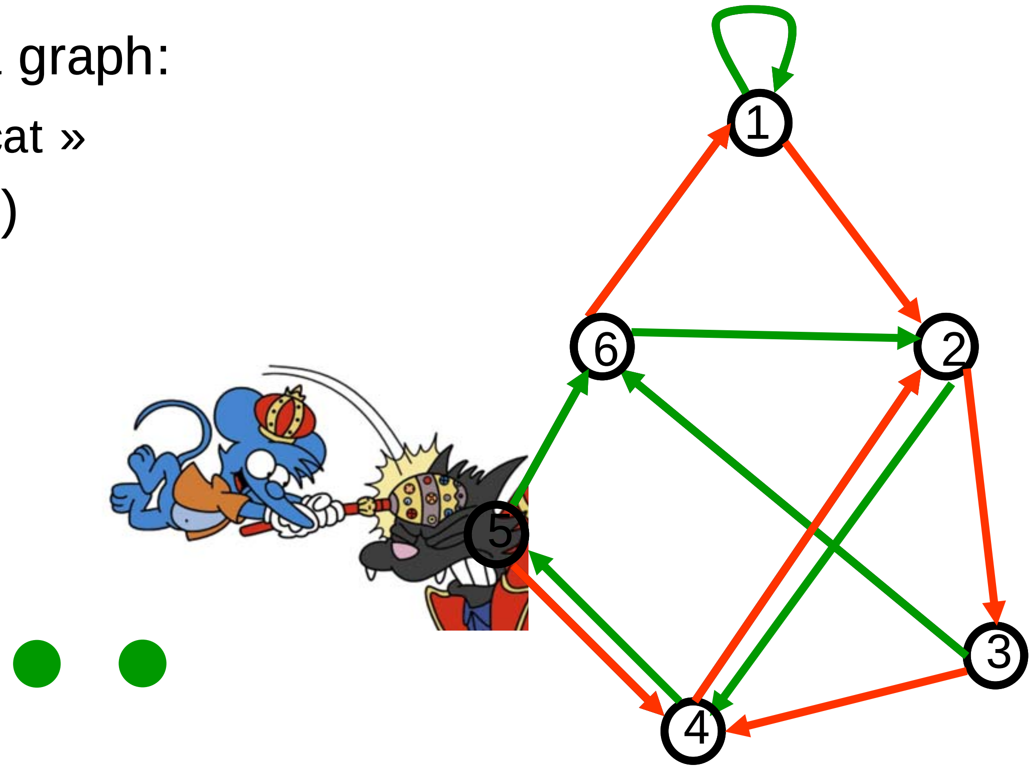




- The cat is hidden somewhere on a colored graph, and the mouse must pick up a node where to take him
- Before to do that, the mouse may impose the cat to follow a particular sequence of colors of length t
- The cat wants to minimize the probability to get caught

A simple game

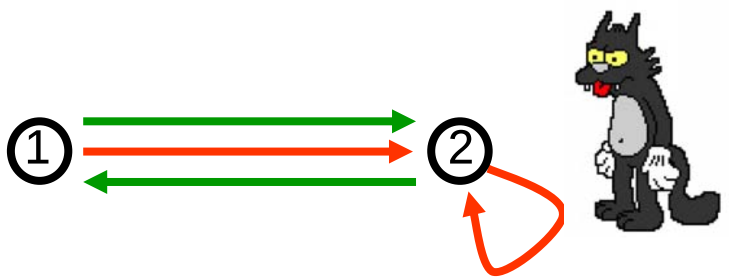
- Two players playing on a graph:
  the « mouse » and the « cat »
- A parameter t (here, t=2)



- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught,
  whatever strategy (of length t) the mouse chooses

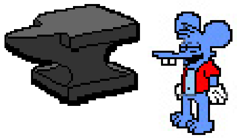
A simple game

- The cat's strategy must be probabilistic
  (i.e. a probability function on the nodes)

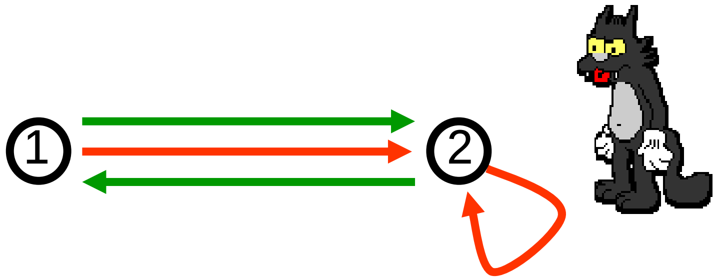


- Definition: The synchronizing probability function k(t)
  of the automaton is the smallest probability the cat can ensure to get caught,
  whatever strategy (of length t) the mouse chooses

A simple game



- The cat's strategy must be probabilistic
  (i.e. a probability function on the nodes)



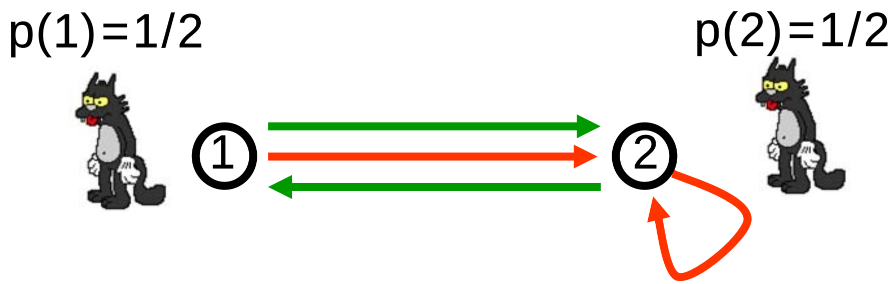
- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught,
  whatever strategy (of length t) the mouse chooses

A simple game

- The cat's strategy must be probabilistic
  (i.e. a probability function on the nodes)

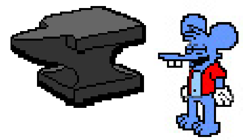
k(0)=1/2

k(1)=1



- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught,
  whatever strategy (of length t) the mouse chooses

A simple game



- The cat's strategy must be probabilistic
  (i.e. a probability function on the nodes)

k(0)=1/2

k(1)=1

- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught,
  whatever strategy (of length t) the mouse chooses

A simple game

- The cat's strategy must be probabilistic
  (i.e. a probability function on the nodes)

k(0)=1/2

k(1)=1

- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught,
  whatever strategy (of length t) the mouse chooses

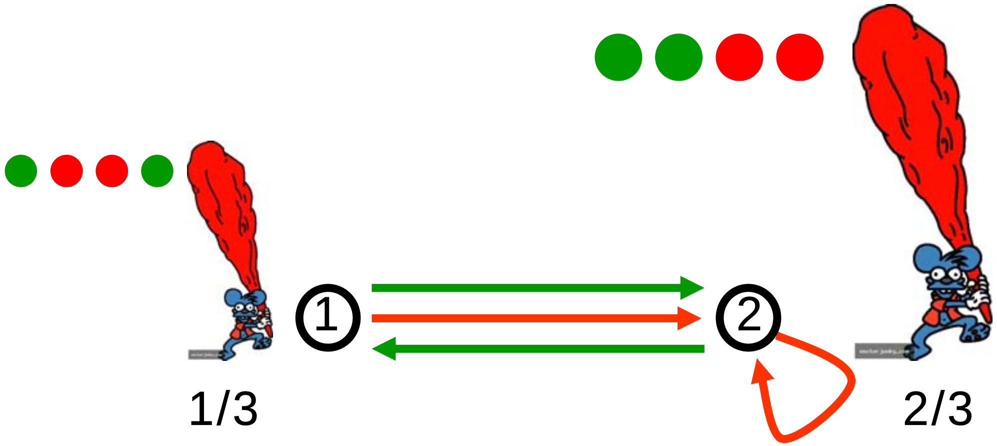
A simple game

- Proposition: The automaton has a synchronizing word of length t if and only if k(t)=1

- Thus Cerny's conjecture is:

$$k((n-1)^2)=1$$

- Note that in general, the mouse's policy might be probabilistic as well



A few equations…

- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught, whatever strategy (of length t) the mouse chooses

$$
\begin{aligned}
\min_{p} \quad & k \\
s.t. \quad & Bp \leq k\mathbf{e} \quad \forall B \in \Sigma^{\leq t} \\
& \mathbf{e}^T p = 1 \\
& p \geq 0.
\end{aligned}
$$

- The problem he has to solve is an LP (Linear Program)!

A few equations…

- Definition: The synchronizing probability function k(t) of the automaton is the smallest probability the cat can ensure to get caught,

  whatever strategy (of length t) the mouse chooses

$$
\begin{aligned}
\min_p \quad & k \\
s.t. \quad & Bp \leq k\mathbf{e} \quad \forall B \in \Sigma^{\leq t} \\
& \mathbf{e}^T p = 1 \\
& p \geq 0.
\end{aligned}
\qquad\qquad
\begin{aligned}
\max_q \quad & k \\
s.t. \quad & q^T A \geq k\mathbf{e} \\
& \mathbf{e}^T q = 1 \\
& q \geq 0.
\end{aligned}
$$

- The problem he has to solve is an LP (Linear Program)!

Outline

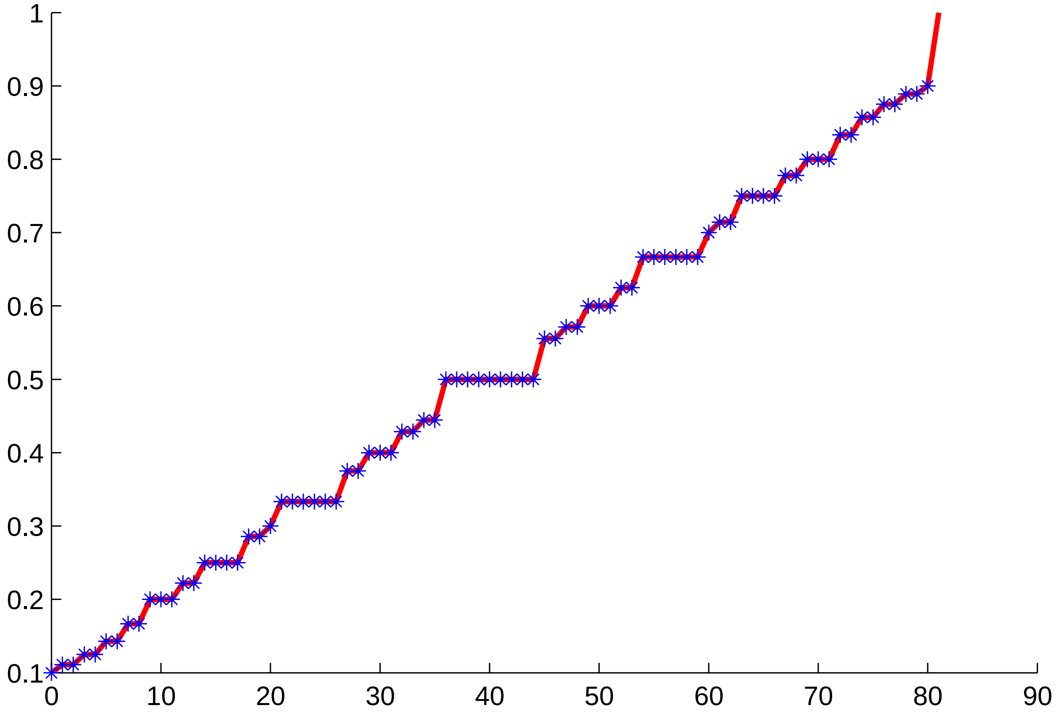
- Synchronizing automata and Cerny’s conjecture
- Previous approaches

- A game theoretic/probabilistic approach
- Results

- Conclusion
- Discussion

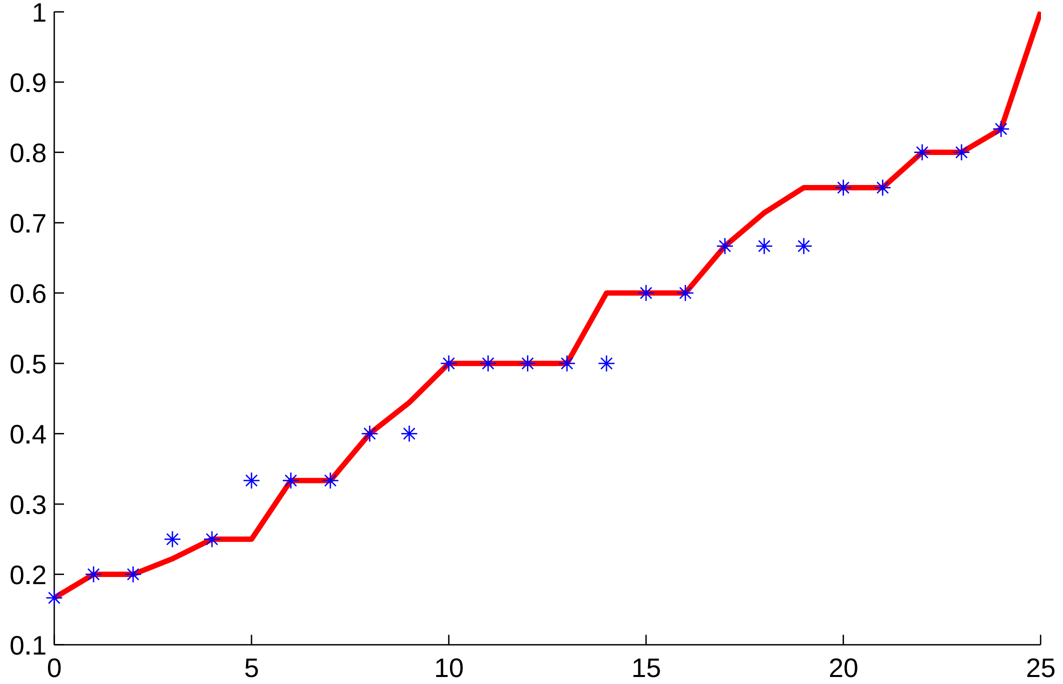
The synchronizing function on practical examples

- Cerny’s automaton



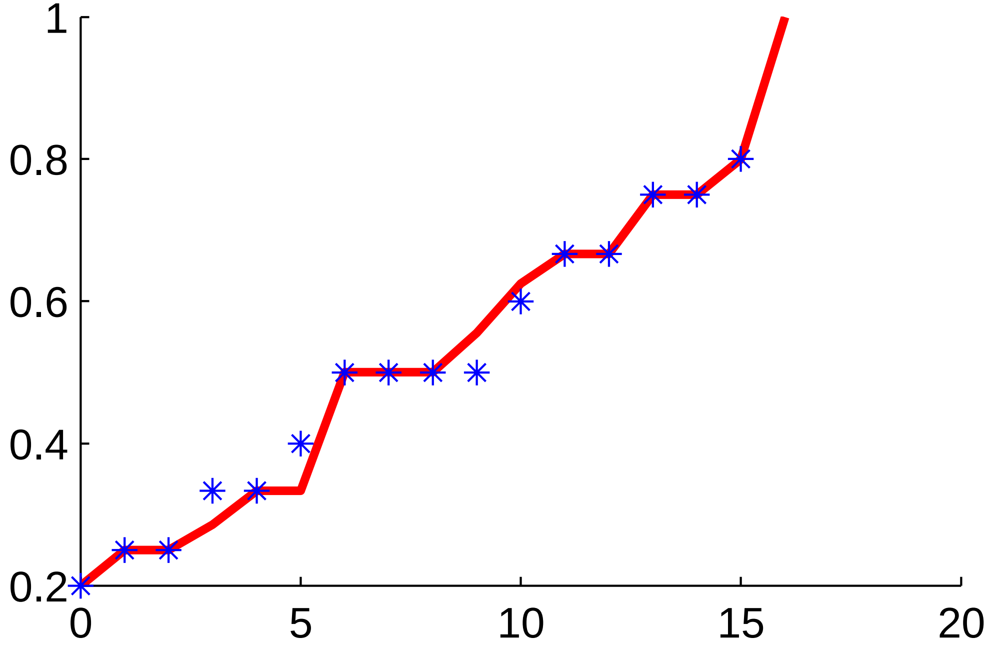
The synchronizing function on practical examples

- Kari’s automaton



The synchronizing function on practical examples

- Roman's automaton



A few first results

- Theorem: The players can communicate their policies

- A procedure allowing to compute the function pretty fast in practice

- Proposition: It doesn't help the mouse to allow her to take shorter products

- Proposition: there is always an optimal policy for the mouse with at most n-1 different rows (n is the number of nodes)

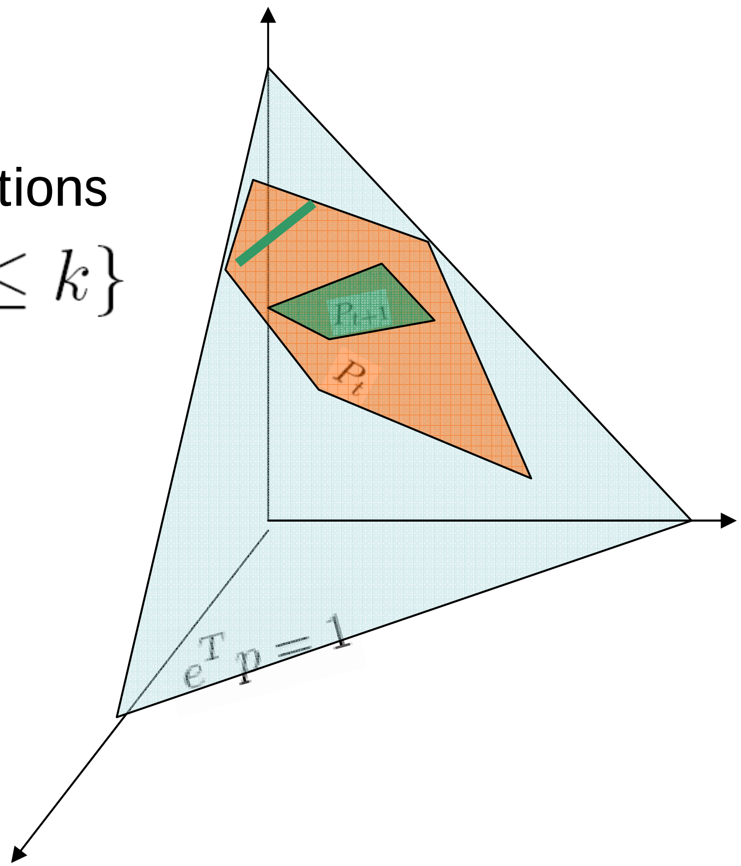
- Theorem: If $k(t)<1$, then $k(t+(n-1))>k(t)$

- Means « k(t) cannot stagnate too long »

Proof of the theorem

Theorem: If k(t)<1, then k(t+(n-1))>k(t)

Proof: suppose k(t)=k(t+1)



- Look at the polytope P_t of optimal solutions

$$P_t = \{p : A_0 A_1 A_2 \ldots A_{t-1} p \leq k\}$$

- Lemma: P'_{t+1} is in P'_t

- Lemma: P'_{t+1} is different from P'_t

  Because if not, then $P'_{t+2} = P'_{t+1}$

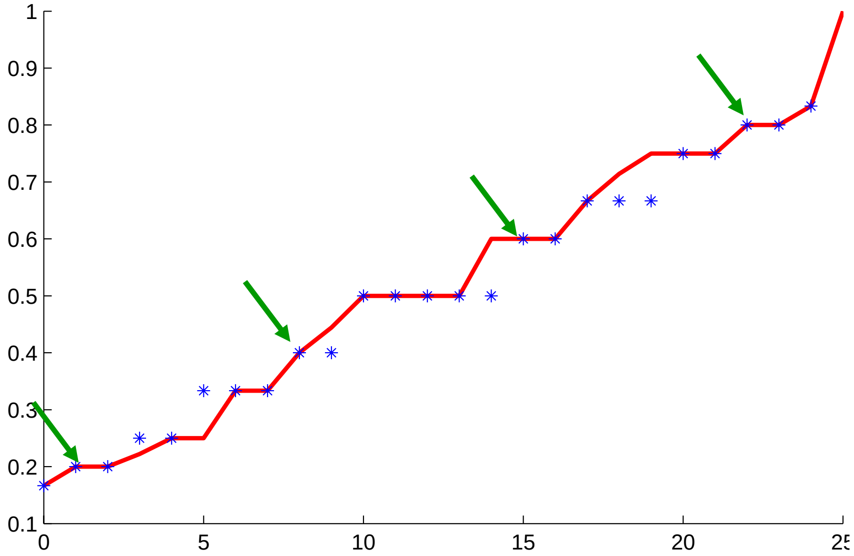
- This implies that dim P'_{t+1} <dim P'_t

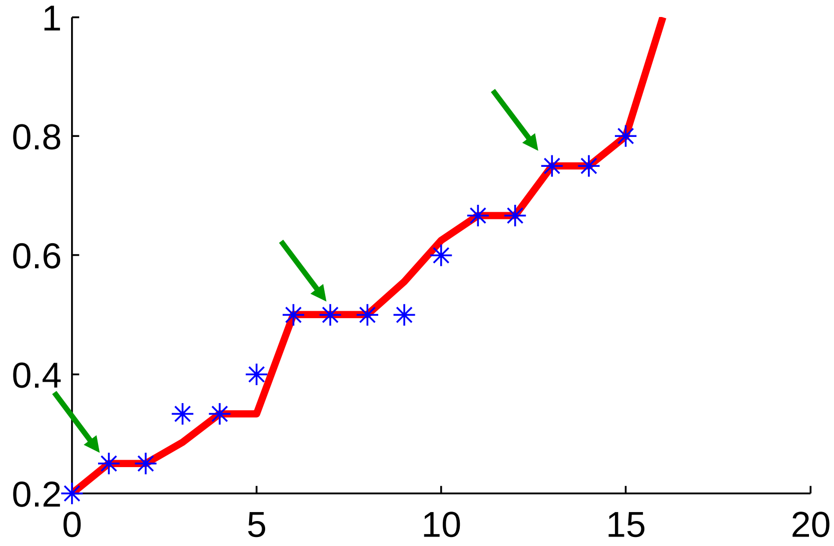
- Since dim P_t <n-1, after at most n-1 steps it cannot drop anymore

A conjecture?

- Observation: At some fixed times, the value of the function is always higher (or equal) than Cerny's automaton

t=1+ (n+1) i





- Conjecture: It is always the case
- Corollary-conjecture: There is always a product of length n+2 with a row of weight three (i.e., a word of length n+2, and a node with a preimage of size three)

An easier conjecture?

- Corollary-conjecture: there always exists a word of length n+2 such that one node has a preimage of size three

- What is the best bound for the apparition of a row of weight 3?
  - N^2 easy: there are at most $n^2/2$ lines
  - $N^2/4$ with classical methods
  - ?

- Our method can make $n^2/(6.4)$

Outline

- Synchronizing automata and Cerny's conjecture
- Previous approaches

- A game theoretic/probabilistic approach
- Results

- Conclusion
- Discussion

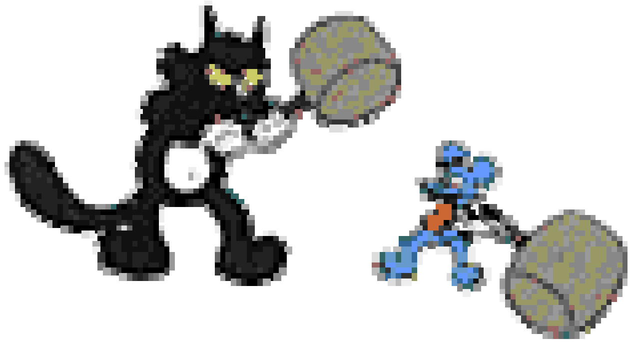
Conclusion and future work

- Future work: Plenty of things!
  - What with other automata: non-synchronizing automata, Non-deterministic…
  - Particular cases
  - Weight three
  - Use our concepts to generate slowly synchronizing automata

- Applications!

- Our approach tried to connect this longstanding problem with other fields of mathematics.

  The connection seems meaningful and suggests new questions. Is it useful?



Thanks!

Questions ?

More on: http://www.inma.ucl.ac.be/~jungers/