Synchronizing Automata and Černý’s Conjecture

Narad Rampersad

Department of Mathematics
University of Liège
A finite automaton
Here a finite automaton is a directed multigraph where:

- every vertex has constant out-degree $k$, and
- the outgoing arcs of each vertex are labeled by distinct elements of a fixed $k$-element set.
Terminology

- We call the vertices **states** and denote the set of states by $Q$.
- We call the arcs **transitions**.
- Arcs are labeled by **letters**.
- A sequence of letters is called a **word**.
The transition function \( \delta(p, a) = q \) denotes a transition from \( p \) to \( q \) labeled by \( a \).

If \( w = w_1w_2 \cdots w_n \) is a word then \( \delta(q, w) \) is the state reached by starting at \( q \) and following the sequence of arcs labeled \( w_1, w_2, \ldots, w_n \).

If \( A \subseteq Q \) then

\[
\delta(A, w) = \bigcup_{q \in A} \delta(q, w).
\]
Synchronizing automata

- A word $w$ such that $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$ is a reset word.
- An automaton with a reset word is synchronizing.
- Equivalently, there exists a state $p$ and a word $w$ such that $\delta(Q, w) = \{p\}$.
- Given an automaton, can we decide if it is synchronizing?
- If so, can we find the shortest reset word?
A synchronizing automaton

Reset word: *abbbabbbba*.
Applications

- Moore’s Gedanken-experiments (1950’s):
  - Imagine a satellite orbiting the moon.
  - Its behaviour while on the dark side of the moon cannot be observed.
  - When control is reestablished, we wish to reset the system to a particular configuration.
Applications

- Robotics (Natarajan 1980’s):
  - Imagine parts arriving on an assembly line with arbitrary orientations.
  - The parts must be manipulated into a fixed orientation before proceeding with assembly.
Černý’s Conjecture

Černý’s Conjecture (1964)

The shortest reset word of any synchronizing automaton with $n$ states has length at most $(n - 1)^2$. 
Černý’s construction

Reset word: \((ab^{n-1})^{n-2}a\) (length \((n - 1)^2\)).
Partial results

- E.g., Kari (2003) verified the conjecture for synchronizing automata whose underlying digraphs are Eulerian.
- Conjecture verified for several other classes of synchronizing automata.
- Steinberg (preprint) unified and simplified many of these proofs.
Best known upper bound

- $M$ is a synchronizing automaton:

- There are sets $Q = P_1, P_2, \ldots, P_t$, and words $w_1, w_2, \ldots, w_{t-1}$, such that
  - $\delta(P_i, w_i) = P_{i+1}$, for $i = 1, \ldots, t - 1$;
  - $|P_i| > |P_{i+1}|$, for $i = 1, \ldots, t - 1$;
  - $|P_t| = 1$.

- $w = w_1w_2\cdots w_{t-1}$ is a reset word for $M$. 
An example

Reset word: $a \ b b a \ b b a$. 

Diagram: 

- States: 1, 2, 3, 4
- Edges: 1→2 (a,b), 2→4 (b), 4→3 (b), 3→2 (a), 2→1 (a,b)
An example

Reset word: \textit{a bbba bbba}.
An example

Reset word: \textit{a bbba bbba}.
An example

Reset word: \(a \ bbba \ bbba\).
An example

Reset word: $a b b b a b b b a$. 

![Diagram](image-url)
An example

Reset word: $a b b b a b b b a$. 

\begin{center}
\begin{tikzpicture}
\node[shape=circle,draw=black,fill=white] (T1) at (1,1) {1};
\node[shape=circle,draw=black,fill=white] (T2) at (3,1) {2};
\node[shape=circle,draw=black,fill=white] (T3) at (3,-1) {3};
\node[shape=circle,draw=black,fill=white] (T4) at (1,-1) {4};
\path[->]
(T1) edge node {a,b} (T2);
(T2) edge node {b} (T3);
(T3) edge node {b} (T4);
(T4) edge node {a} (T1);
(1.5,-1) edge [out=45,in=135,loop] node {a} (1.5,-1);
(3,0) edge [out=-45,in=-135,loop] node {a} (3,0);
\end{tikzpicture}
\end{center}
An example

Reset word: \( a \text{ bbba bbba}. \)
An example

Reset word: \( a \ bbba \ bbba. \)
An example

Reset word: $a \textbf{bbba} b\textbf{bbba}$. 
An example

Reset word: $a \ b b \ b a \ b b a$.
An example

Reset word: $a \ b b b a \ b b b a$.
The greedy algorithm

Algorithm to find reset word $w$

Set $P_1 = Q$ and $t = 1$.

While $|P_t| > 1$:

Find a smallest word $w_t$ such that $|\delta(P_t, w_t)| < |P_t|$.

Set $P_{t+1} = \delta(P_t, w_t)$ and increment $t$.

Return $w = w_1 w_2 \cdots w_{t-1}$. 
Length of the reset word found

- What is the maximum length of $w$ found by the greedy algorithm?
- In the worst case, $|P_i| - |P_{i+1}| = 1$, so that $t = n$.
- Consider a generic step $k$: i.e., $P_k$ and $w_k$ such that $|\delta(P_k, w_k)| < |P_k|$.
- What is the longest that $w_k$ can be?
Let \( w_k = a_1 a_2 \cdots a_{m+1} \) (the \( a \)'s letters).

There are sets \( P_k = A_1, A_2, \ldots, A_{m+2} \) such that

- \( \delta(A_i, a_1) = A_{i+1} \) for \( i = 1, \ldots, m + 1 \);
- \( |A_i| = |A_{i+1}| \) for \( i = 1, \ldots, m \);
- \( |A_{m+1}| > |A_{m+2}| \).
Length of the reset word found

- For $i = 1, \ldots, m + 1$,

$$|\delta(A_i, a_i \cdots a_{m+1})| < |A_i|.$$ 

- Thus there exists $q_i, q'_i \in A_i$ such that

$$\delta(q_i, a_i \cdots a_{m+1}) = \delta(q'_i, a_i \cdots a_{m+1}).$$

- To each $A_i$, associate the set $B_i = \{q_i, q'_i\}$. 

Length of the reset word found

- Note that \( B_i \subseteq A_i \).
- Furthermore, for \( i < j \), \( B_j \nsubseteq A_i \).
- Otherwise, we would have a shorter word
  \[
  w_k' = a_1 \cdots a_{i-1} a_j \cdots a_{m+1}
  \]
such that \( |\delta(P_k, w_k')| < |P_k| \).
Let $\overline{A_i}$ denote the complement of $A_i$, i.e., the set $Q \setminus A_i$.

We thus have

- $B_i \cap \overline{A_i} = \emptyset$ for $i = 1, \ldots, m$;
- $B_j \cap \overline{A_i} \neq \emptyset$ for $i < j$.

What is the largest that $m$ can be subject to these constraints?
Theorem (Frankl 1982)

Let $A_1, \ldots, A_m$ be sets of size $r$ and let $B_1, \ldots, B_m$ be sets of size $s$ such that

(a) $A_i \cap B_i = \emptyset$ for $i = 1, \ldots, m$;

(b) $A_i \cap B_j \neq \emptyset$ if $i < j$.

Then $m \leq \binom{r+s}{s}$. 
A bound on the length of the reset word

Let $|Q| = n$. Then $|A_i| = n - k$ (since $|A_i| = k$) and $|B_i| = 2$ for $i = 1, \ldots, m$.

By Frankl’s result, $m \leq \binom{n-k+2}{2}$.

Total length of the reset word at most

$$\sum_{k=2}^{n} \binom{n-k+2}{2} = \frac{n^3 - n}{6}.$$
Running time of the algorithm

- Originally conjectured by Fischler and Tannenbaum (1970) and (independently) by Pin (1981).
- After hearing Pin’s 1981 talk, Frankl proved the necessary combinatorial result (independently rediscovered by Klyachko, Rystsov, and Spivak (1987)).
- Eppstein (1990) showed how to implement the greedy algorithm in $O(n^3 + kn^2)$ time.
- Greedy algorithm does not find a shortest reset word.
Finding a reset word of a given length

**SYNCWORD**

Given an automaton \( A \) and a positive integer \( k \), does \( A \) have a reset word of length at most \( k \)?

- Clearly in NP since it suffices to “guess” a reset word of length at most \( \min\{(n^3 - n)/6, k\} \).
- Eppstein showed it is NP-complete.
Finding a shortest reset word

MIN-SYNCRECORD

Given an automaton $A$ and a positive integer $k$, does $A$ have a shortest reset word of length $k$?

- Olschewski and Ummels (preprint) showed it is DP-complete.
The class DP

- DP consists of all languages $L$ such that $L = L_1 \setminus L_2$ for some languages $L_1, L_2$ in NP.
- A DP-complete problem is both NP-hard and coNP-hard.
- The canonical DP-complete problem is:

<table>
<thead>
<tr>
<th>SAT-UNSAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given CNF formulae $\varphi$ and $\psi$, is $\varphi$ satisfiable and $\psi$ unsatisfiable?</td>
</tr>
</tbody>
</table>
DP-completeness

- **MIN-SYNCDWORD** clearly in DP, since it is the difference of **SYNCDWORD** and

  \[
  \{(A, k) : k > 0 \text{ and } (A, k - 1) \in \text{ SYNCDWORD}\}.
  \]

- To show DP-hardness, reduce from **SAT-UNSAT**.
Approximating the shortest reset word

**Theorem (Berlinkov (preprint))**

Unless $P = NP$, there is no polynomial-time algorithm to approximate the minimum length of a reset word for a given automaton within a constant factor.
Synchronizing colouring

Start with a *strongly connected* directed multigraph $G$ where every vertex has constant out-degree $k$.

Is it possible to assign labels to the arcs so that $G$ becomes synchronizing?

If so, then $G$ has a *synchronizing colouring*. 
The road colouring problem

- Can graphs with synchronizing colourings be characterized?

- A graph is **aperiodic** if the gcd of the lengths of all of its cycles is 1.

- It is not hard to show that aperiodicity is a necessary condition.

- Adler and Weiss (1970) conjectured that it is also a sufficient condition.
The resolution of the problem

Theorem (Trahtman 2007)

Let $G$ be a strongly connected directed multigraph where every vertex has constant out-degree $k$. Then $G$ has a synchronizing coloring if and only if the gcd of the lengths of all of its cycles is 1.
The literature on synchronizing automata is huge. For more information, see:

- Volkov's 2008 survey:
  

- Jean-Eric Pin's webpage:
  
  http://www.liafa.jussieu.fr/~jep/Problemes/Cerny.html

- Avraham Trahtman's webpage:
  
  http://u.cs.biu.ac.il/~trakht/syn.html
The End