Endomorphisms and synchronization, 1: Synchronization

Peter J. Cameron

BIRS, November 2014

Thanks . . .

Many people have been involved in the synchronization project. For the material I am talking about here, thanks specially to João Araújo, Wolfram Bentz, James Mitchell Peter Neumann, Gordon Royle, Artur Schaefer.

Thanks ...

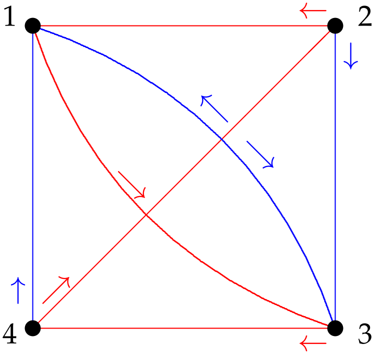
Many people have been involved in the synchronization project. For the material I am talking about here, thanks specially to João Araújo, Wolfram Bentz, James Mitchell Peter Neumann, Gordon Royle, Artur Schaefer.
Thanks also to the developers of the software (GAP and its graphs and semigroups share packages for algebraic computing, MINION for constraint satisfaction problems).

The dungeon

You are in a dungeon consisting of a number of rooms. Passages are marked with coloured arrows. Each room contains a special door; in one room, the door leads to freedom, but in all the others, to instant death. You have a schematic map of the dungeon, but you do not know where you are.

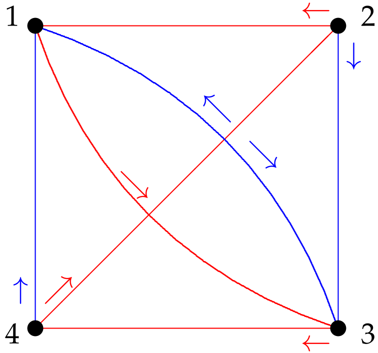
The dungeon

You are in a dungeon consisting of a number of rooms. Passages are marked with coloured arrows. Each room contains a special door; in one room, the door leads to freedom, but in all the others, to instant death. You have a schematic map of the dungeon, but you do not know where you are.



The dungeon

You are in a dungeon consisting of a number of rooms. Passages are marked with coloured arrows. Each room contains a special door; in one room, the door leads to freedom, but in all the others, to instant death. You have a schematic map of the dungeon, but you do not know where you are.



You can check that (Blue, Red, Blue) takes you to room 1, no matter where you start.

Definitions

A (finite deterministic) automaton consists of a finite set Ω of states, with a finite set S of transitions, maps from Ω to Ω. (This is the simplest kind of automaton: it does not write symbols, and it does not have an accepting state, so there is no associated language.)

Definitions

A (finite deterministic) automaton consists of a finite set Ω of states, with a finite set S of transitions, maps from Ω to Ω. (This is the simplest kind of automaton: it does not write symbols, and it does not have an accepting state, so there is no associated language.)

The automaton is synchronizing if there is a word in the transitions which evaluates to a map of rank 1.

Definitions

A (finite deterministic) automaton consists of a finite set Ω of states, with a finite set S of transitions, maps from Ω to Ω. (This is the simplest kind of automaton: it does not write symbols, and it does not have an accepting state, so there is no associated language.)

The automaton is synchronizing if there is a word in the transitions which evaluates to a map of rank 1.

Combinatorially, an automaton is an edge-coloured directed graph on Ω such that every vertex is the source of a unique arc of each colour. The elements of Ω are the states, and the colours index the transitions.

Definitions

A (finite deterministic) automaton consists of a finite set Ω of states, with a finite set S of transitions, maps from Ω to Ω. (This is the simplest kind of automaton: it does not write symbols, and it does not have an accepting state, so there is no associated language.)

The automaton is synchronizing if there is a word in the transitions which evaluates to a map of rank 1.

Combinatorially, an automaton is an edge-coloured directed graph on Ω such that every vertex is the source of a unique arc of each colour. The elements of Ω are the states, and the colours index the transitions.

The digraph is strongly connected if and only if, for any pair of states, there is a sequence of transitions which carries the first to the second.

Transformation monoids

We are interested in composing the transitions. So algebraically, an automaton is a set of transformations of Ω (maps from Ω to itself) which is closed under composition and contains the identity map (corresponding to the empty word).

Transformation monoids

We are interested in composing the transitions. So algebraically, an automaton is a set of transformations of Ω (maps from Ω to itself) which is closed under composition and contains the identity map (corresponding to the empty word).
In other words, an automaton is a transformation monoid on Ω, with a prescribed set of generators.

Transformation monoids

We are interested in composing the transitions. So algebraically, an automaton is a set of transformations of Ω (maps from Ω to itself) which is closed under composition and contains the identity map (corresponding to the empty word).

In other words, an automaton is a transformation monoid on Ω, with a prescribed set of generators.

If all the transformations are permutations, then we have instead a permutation group.

Synchronizing monoids

An automaton is synchronizing if and only if the corresponding transformation monoid contains an element of rank 1, that is, a transformation whose image has cardinality 1.

Synchronizing monoids

An automaton is synchronizing if and only if the corresponding transformation monoid contains an element of rank 1, that is, a transformation whose image has cardinality 1. We will call such a transformation monoid synchronizing.

Synchronizing monoids

An automaton is synchronizing if and only if the corresponding transformation monoid contains an element of rank 1, that is, a transformation whose image has cardinality 1. We will call such a transformation monoid synchronizing. Note that a permutation group is never synchronizing (unless $|\Omega| = 1$). Later, we will re-define synchronization for a permutation group.

Industrial robotics

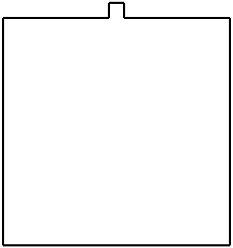
Here is an application of synchronization.

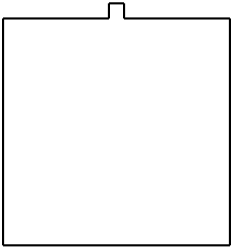
Industrial robotics

Here is an application of synchronization.
Pieces arrive to be assembled by a robot. The orientation is critical. You could equip the robot with vision sensors and manipulators so that it can rotate the pieces into the correct orientation. But it is much cheaper and less error-prone to regard the possible orientations of the pieces as states of an automaton on which transitions can be performed by simple machinery, and apply a reset word before the pieces arrive at the robot.

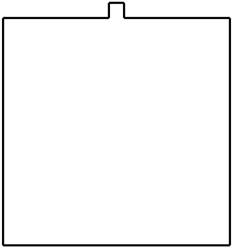
Industrial robotics

Here is an application of synchronization.
Pieces arrive to be assembled by a robot. The orientation is critical. You could equip the robot with vision sensors and manipulators so that it can rotate the pieces into the correct orientation. But it is much cheaper and less error-prone to regard the possible orientations of the pieces as states of an automaton on which transitions can be performed by simple machinery, and apply a reset word before the pieces arrive at the robot.
For a simple example, consider a square plate with a projection on one side, as shown on the next slide.



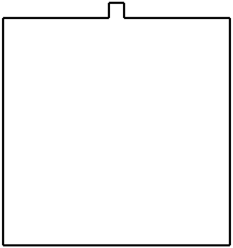


It can sit in a tray on the conveyor belt in any one of four orientations.



It can sit in a tray on the conveyor belt in any one of four orientations.

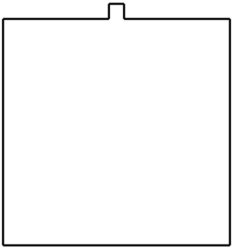
The following transititions are easy to implement:



It can sit in a tray on the conveyor belt in any one of four orientations.

The following transititions are easy to implement:

- R: rotate through 90° in the positive direction;



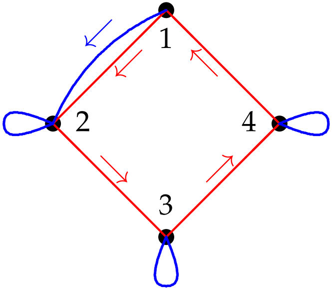
It can sit in a tray on the conveyor belt in any one of four orientations.

The following transititions are easy to implement:

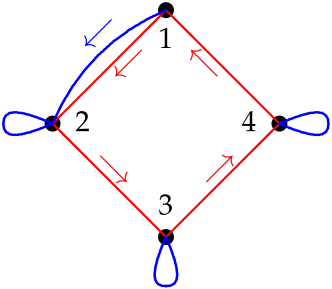
- R: rotate through 90° in the positive direction;
- B: rotate through 90° if the projection points up, otherwise do nothing.

The effect of these transformations is shown in the following diagram.

The effect of these transformations is shown in the following diagram.

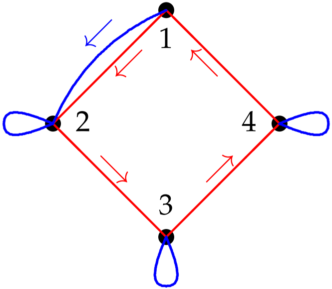


The effect of these transformations is shown in the following diagram.



	B	R	R	R	B	R	R	R	B
1	2	3	4	1	2	3	4	1	2
2	2	3	4	1	2	3	4	1	2
3	3	4	1	2	2	3	4	1	2
4	4	1	2	3	3	4	1	2	2

The effect of these transformations is shown in the following diagram.



	B	R	R	R	B	R	R	R	B
1	2	3	4	1	2	3	4	1	2
2	2	3	4	1	2	3	4	1	2
3	3	4	1	2	2	3	4	1	2
4	4	1	2	3	3	4	1	2	2

So BRRRBRRRB is a reset word.

The Černý conjecture

Problem (The Černý conjecture)

If an n-state automaton is synchronizing, then it has a reset word of length at most $(n-1)^2$.

The Černý conjecture

Problem (The Černý conjecture)

If an n-state automaton is synchronizing, then it has a reset word of length at most $(n-1)^2$.

This problem is still open after nearly fifty years. The example on the previous slide and the obvious generalisation (replacing a square by a regular n-gon) show that, if true, it is best possible.

The Černý conjecture

Problem (The Černý conjecture)

If an n-state automaton is synchronizing, then it has a reset word of length at most $(n-1)^2$.

This problem is still open after nearly fifty years. The example on the previous slide and the obvious generalisation (replacing a square by a regular n-gon) show that, if true, it is best possible.

The best reference on the Černý conjecture is the paper by Mikhail V. Volkov, "Synchronizing automata and the Černý conjecture", *Language and Automata Theory and Applications* Lecture Notes in Computer Science **5196** (2008), 11–27.

Some history

Volkov discusses, among other things, the history of the problem. He points out:

- The first reference to synchronization was ten years earlier than Černý's paper, in Ross Ashby's book *An Introduction to Cybernetics* (Chapman and Hall, 1956). I give Ashby's example on the next slide.

Some history

Volkov discusses, among other things, the history of the problem. He points out:

- The first reference to synchronization was ten years earlier than Černý's paper, in Ross Ashby's book *An Introduction to Cybernetics* (Chapman and Hall, 1956). I give Ashby's example on the next slide.
- In his first paper, Černý gave upper and lower bounds for the length of a reset word in the worst case, but didn't formulate his conjecture until a talk in Bratislava in 1969 (published in 1971).

"Graveside"
Wits End
Haunts.

Dear Friend,
Some time ago I bought this old house, but found it to be haunted by two ghostly noises—a ribald Singing and a sardonic Laughter. As a result it is hardly habitable. There is hope, however, for by actual testing I have found that their behaviour is subject to certain laws, obscure but infallible, and that they can be affected by my playing the organ or burning incense.
In each minute, each noise is either sounding or silent—they show no degrees. What each will do during the ensuing minute depends, in the following exact way, on what has been happening during the preceding minute: The Singing, in the succeeding minute, will go on as it was during the preceding minute (sounding or silent) unless there was organ-playing with no Laughter, in which case it will change to the opposite (sounding to silent, or vice versa). As for the Laughter, if there was incense burning, then it will sound or not according as the Singing was sounding or not (so that the Laughter copies the singing a minute later). If however there was no incense burning, the Laughter will do the opposite of what the Singing did.
At this minute of writing, the Laughter and Singing are both sounding. Please tell me what manipulations of incense and organ I should make to get the house quiet, and keep it so.

Results on the Černý conjecture

The best known bound in general is cubic (of order $\frac{1}{6}n^3$). However, better results are known in some special cases, for example:

Results on the Černý conjecture

The best known bound in general is cubic (of order $\frac{1}{6}n^3$). However, better results are known in some special cases, for example:

- Trahtman showed in 2007 that the conjecture is true for aperiodic automata (indeed, in this case he gave a bound $n(n-1)/2$).

Results on the Černý conjecture

The best known bound in general is cubic (of order $\frac{1}{6}n^3$). However, better results are known in some special cases, for example:

- Trahtman showed in 2007 that the conjecture is true for aperiodic automata (indeed, in this case he gave a bound $n(n-1)/2$).
- Dubuc showed in 1998 that the conjecture is true if one of the transitions acts as a cyclic permutation.

Results on the Černý conjecture

The best known bound in general is cubic (of order $\frac{1}{6}n^3$). However, better results are known in some special cases, for example:

- Trahtman showed in 2007 that the conjecture is true for aperiodic automata (indeed, in this case he gave a bound $n(n-1)/2$).
- Dubuc showed in 1998 that the conjecture is true if one of the transitions acts as a cyclic permutation.
- Kari showed in 2001 that it is true if the underlying digraph of the automaton is Eulerian.

Related questions

Two related computational problems. Given an automaton (Ω, S),

- Decide whether it is synchronizing.

Related questions

Two related computational problems. Given an automaton (Ω, S),

- Decide whether it is synchronizing.
- If so, find the shortest reset word.

Related questions

Two related computational problems. Given an automaton (Ω, S),

- Decide whether it is synchronizing.
- If so, find the shortest reset word.

What is the computational complexity of these problems?

Related questions

Two related computational problems. Given an automaton (Ω, S),

- Decide whether it is synchronizing.
- If so, find the shortest reset word.

What is the computational complexity of these problems?
We'll see that the first is easy but the second is hard.

Testing synchronization

Proposition

An automaton (Ω, S) is synchronizing if and only if, for any two states $a, b \in \Omega$, there is a word $w_{a,b}$ in the elements of S which maps a and b to the same place.

Testing synchronization

Proposition

An automaton (Ω, S) is synchronizing if and only if, for any two states $a, b \in \Omega$, there is a word $w_{a,b}$ in the elements of S which maps a and b to the same place.

Proof.

"Only if" is clear, so suppose that the condition holds. Let f be an element of $\langle S \rangle$ of smallest possible rank. If the rank of S is greater than 1, then choose two points a, b in the image; then fw_{ab} has smaller rank than f. So f has rank 1, and the automaton is synchronizing. □

Testing synchronization

Proposition

An automaton (Ω, S) is synchronizing if and only if, for any two states $a, b \in \Omega$, there is a word $w_{a,b}$ in the elements of S which maps a and b to the same place.

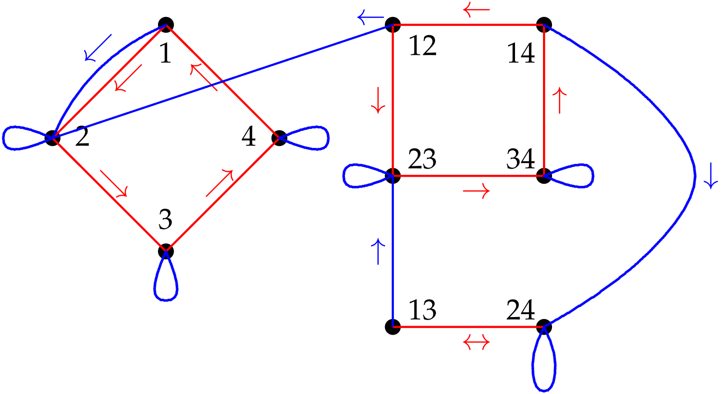
Proof.

"Only if" is clear, so suppose that the condition holds. Let f be an element of $\langle S \rangle$ of smallest possible rank. If the rank of S is greater than 1, then choose two points a, b in the image; then fw_{ab} has smaller rank than f. So f has rank 1, and the automaton is synchronizing. □

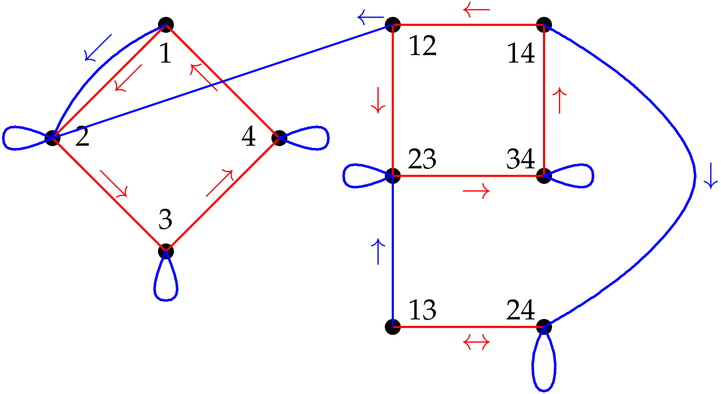
So we only have to consider all pairs of states.

The picture shows the previous example, extended to pairs of states.

The picture shows the previous example, extended to pairs of states.



The picture shows the previous example, extended to pairs of states.



Now it suffices to check that there is a path from any vertex on the right to some vertex on the left; this can clearly be done in polynomial time.

Shortest reset word

In order to find the shortest reset word by this method, we would have to extend the diagram to all possible sets of states, and then find the shortest path from Ω to a singleton; the size of the resulting digraph would be exponentially large.

Shortest reset word

In order to find the shortest reset word by this method, we would have to extend the diagram to all possible sets of states, and then find the shortest path from Ω to a singleton; the size of the resulting digraph would be exponentially large.
In fact:

Shortest reset word

In order to find the shortest reset word by this method, we would have to extend the diagram to all possible sets of states, and then find the shortest path from Ω to a singleton; the size of the resulting digraph would be exponentially large.
In fact:

Theorem

Deciding whether an automaton is synchronizing is in P*, but finding the length of the shortest reset word is* NP*-hard.*

Shortest reset word

In order to find the shortest reset word by this method, we would have to extend the diagram to all possible sets of states, and then find the shortest path from Ω to a singleton; the size of the resulting digraph would be exponentially large.
In fact:

Theorem

Deciding whether an automaton is synchronizing is in P*, but finding the length of the shortest reset word is* NP*-hard.*

The above argument gives us a cubic upper bound for the length of a reset word. For we can collapse any given pair of states in at most $\binom{n}{2}$ steps, and we only need to do this $n-1$ times to reset the automaton.

Synchronization in the infinite case

I will not say much about the infinite.

Synchronization in the infinite case

I will not say much about the infinite.
One attempt to define synchronization for transformation monoids on infinite sets would be to say that M is synchronizing if for any two points v, w of the domain, there is an element $f \in M$ with $vf = wf$.

Synchronization in the infinite case

I will not say much about the infinite.
One attempt to define synchronization for transformation monoids on infinite sets would be to say that M is synchronizing if for any two points v, w of the domain, there is an element $f \in M$ with $vf = wf$.
This is OK for maps of finite rank but doesn't do what we want for infinite rank (consider the monoid of all surjective maps, for example).

Synchronization in the infinite case

I will not say much about the infinite.
One attempt to define synchronization for transformation monoids on infinite sets would be to say that M is synchronizing if for any two points v, w of the domain, there is an element $f \in M$ with $vf = wf$.
This is OK for maps of finite rank but doesn't do what we want for infinite rank (consider the monoid of all surjective maps, for example).
We could require M to be closed, in the topology of pointwise convergence. This potentially allows maps of infinite rank to "generate" a constant, but is still not really satisfactory.

Synchronization in the infinite case

I will not say much about the infinite.
One attempt to define synchronization for transformation monoids on infinite sets would be to say that M is synchronizing if for any two points v, w of the domain, there is an element $f \in M$ with $vf = wf$.
This is OK for maps of finite rank but doesn't do what we want for infinite rank (consider the monoid of all surjective maps, for example).
We could require M to be closed, in the topology of pointwise convergence. This potentially allows maps of infinite rank to "generate" a constant, but is still not really satisfactory.

A problem

Problem

Let n and k be given positive integers with $k < n$. Find (in terms of n and k) the smallest m such that the following is true: Given a permutation group $G = \langle S \rangle$ of degree n, and two k-sets A and B lying in the same G-orbit, there is a semigroup word of length at most m which maps A to B.

A problem

Problem

Let n and k be given positive integers with $k < n$. Find (in terms of n and k) the smallest m such that the following is true: Given a permutation group $G = \langle S \rangle$ of degree n, and two k-sets A and B lying in the same G-orbit, there is a semigroup word of length at most m which maps A to B.

This question is clearly related to questions about the diameter of a permutation group with respect to a given set of generators, with a couple of significant differences:

A problem

Problem

Let n and k be given positive integers with $k < n$. Find (in terms of n and k) the smallest m such that the following is true: Given a permutation group $G = \langle S \rangle$ of degree n, and two k-sets A and B lying in the same G-orbit, there is a semigroup word of length at most m which maps A to B.

This question is clearly related to questions about the diameter of a permutation group with respect to a given set of generators, with a couple of significant differences:

- we use semigroup words, rather than group words (that is, we are not allowed to use inverses);

A problem

Problem

Let n and k be given positive integers with $k < n$. Find (in terms of n and k) the smallest m such that the following is true: Given a permutation group $G = \langle S \rangle$ of degree n, and two k-sets A and B lying in the same G-orbit, there is a semigroup word of length at most m which maps A to B.

This question is clearly related to questions about the diameter of a permutation group with respect to a given set of generators, with a couple of significant differences:

- we use semigroup words, rather than group words (that is, we are not allowed to use inverses);
- we do not need to express an arbitrary group element in terms of generators, but only some word in an arbitrary coset of a subset stabiliser.

Discussion

For $k = 1$, the answer is clearly $n - 1$. For, if $A = \{a\}$ and $B = \{b\}$ where a and b are in the same orbit, there is a path from a to b, and so the shortest path has length at most $n - 1$. If S consists of a single cyclic permutation s and $b = s^{-1}$, then $n - 1$ steps are required.

Discussion

For $k = 1$, the answer is clearly $n - 1$. For, if $A = \{a\}$ and $B = \{b\}$ where a and b are in the same orbit, there is a path from a to b, and so the shortest path has length at most $n - 1$. If S consists of a single cyclic permutation s and $b = s^{-1}$, then $n - 1$ steps are required.

Using this, we get a bound of $\binom{n}{k} - 1$ in general, which is probably much too large.

Discussion

For $k = 1$, the answer is clearly $n - 1$. For, if $A = \{a\}$ and $B = \{b\}$ where a and b are in the same orbit, there is a path from a to b, and so the shortest path has length at most $n - 1$. If S consists of a single cyclic permutation s and $b = s^{-1}$, then $n - 1$ steps are required.

Using this, we get a bound of $\binom{n}{k} - 1$ in general, which is probably much too large.

For $k = 2$, if $S = \{s\}$ where s has two cycles of coprime lengths close to $n/2$, the number of steps required is about $n^2/4$. For transitive groups G, it appears to be *much* smaller, maybe linear in n.

Discussion

For $k = 1$, the answer is clearly $n - 1$. For, if $A = \{a\}$ and $B = \{b\}$ where a and b are in the same orbit, there is a path from a to b, and so the shortest path has length at most $n - 1$. If S consists of a single cyclic permutation s and $b = s^{-1}$, then $n - 1$ steps are required.

Using this, we get a bound of $\binom{n}{k} - 1$ in general, which is probably much too large.

For $k = 2$, if $S = \{s\}$ where s has two cycles of coprime lengths close to $n/2$, the number of steps required is about $n^2/4$. For transitive groups G, it appears to be *much* smaller, maybe linear in n.

The case $k = 2$ is specially relevant to synchronization …

A quadratic bound?

Recent work by João Araújo gives some hope that, for a synchronizing automaton in which all but one of the generators are permutations which generate a transitive group, a quadratic bound for the length of a reset word can be found.

A quadratic bound?

Recent work by João Araújo gives some hope that, for a synchronizing automaton in which all but one of the generators are permutations which generate a transitive group, a quadratic bound for the length of a reset word can be found. In the worst case, the non-permutation f has rank $n - 1$. We will require at least $n - 1$ applications of f to obtain a synchronizing word, since each only reduces the size of the image by 1.

A quadratic bound?

Recent work by João Araújo gives some hope that, for a synchronizing automaton in which all but one of the generators are permutations which generate a transitive group, a quadratic bound for the length of a reset word can be found.

In the worst case, the non-permutation f has rank $n-1$. We will require at least $n-1$ applications of f to obtain a synchronizing word, since each only reduces the size of the image by 1.

Araújo has shown that we can always find a reset word with no more than this minimum number of occurrences of f. In other words, we use f for reducing the rank, not for "moving things around".

Now between successive occurrences of f, we have a semigroup word in the remaining generators which carries a pair of points in the image of the last application of f to a pair of points in the same kernel class of f, so that the next occurrence of f will reduce the rank.

Now between successive occurrences of f, we have a semigroup word in the remaining generators which carries a pair of points in the image of the last application of f to a pair of points in the same kernel class of f, so that the next occurrence of f will reduce the rank.
A positive solution to the problem above would show that this can be done with a linear number of generators. This would give a quadratic bound for the length of the reset word.

Image and kernel

Let $f : \Omega \to \Omega$ be a map.

Image and kernel

Let $f : \Omega \to \Omega$ be a map.
The image of f consists of the points a for which $xf = a$ for some $x \in \Omega$.

Image and kernel

Let $f : \Omega \to \Omega$ be a map.
The image of f consists of the points a for which $xf = a$ for some $x \in \Omega$.
The kernel of f is the equivalence relation $\equiv$ on Ω where $x \equiv y$ if and only if $xf = yf$.

Image and kernel

Let $f : \Omega \to \Omega$ be a map.
The image of f consists of the points a for which $xf = a$ for some $x \in \Omega$.
The kernel of f is the equivalence relation $\equiv$ on Ω where $x \equiv y$ if and only if $xf = yf$.
The cardinality of the image is equal to the number of equivalence classes of the kernel.

Image and kernel

Let $f : \Omega \to \Omega$ be a map.
The image of f consists of the points a for which $xf = a$ for some $x \in \Omega$.
The kernel of f is the equivalence relation $\equiv$ on Ω where $x \equiv y$ if and only if $xf = yf$.
The cardinality of the image is equal to the number of equivalence classes of the kernel.
We say that f is uniform if all kernel classes have the same cardinality, non-uniform otherwise.

Permutation groups and synchronization

I will now introduce the ideas which will be the focus of the second lecture.

Permutation groups and synchronization

I will now introduce the ideas which will be the focus of the second lecture.

Let G be a permutation group on Ω. We say that G synchronizes the map $f : \Omega \to \Omega$ if the monoid $\langle G, f \rangle$ is synchronizing in our earlier sense, that is, contains a map of rank 1.

Permutation groups and synchronization

I will now introduce the ideas which will be the focus of the second lecture.

Let G be a permutation group on Ω. We say that G synchronizes the map $f : \Omega \to \Omega$ if the monoid $\langle G, f \rangle$ is synchronizing in our earlier sense, that is, contains a map of rank 1.

The preceding discussion is an approach to proving the Černý conjecture for monoids of the form $\langle G, f \rangle$. (Note that the extremal known examples for the Černý conjecture have this form.)

Permutation groups and synchronization

I will now introduce the ideas which will be the focus of the second lecture.

Let G be a permutation group on Ω. We say that G synchronizes the map $f : \Omega \to \Omega$ if the monoid $\langle G, f \rangle$ is synchronizing in our earlier sense, that is, contains a map of rank 1.

The preceding discussion is an approach to proving the Černý conjecture for monoids of the form $\langle G, f \rangle$. (Note that the extremal known examples for the Černý conjecture have this form.)

We say that a permutation group G is synchronizing if it synchronizes every non-permutation on Ω, and almost synchronizing if it synchronizes every non-uniform map on Ω.

Forward look

The questions that will concern us in the second lecture will be:

Forward look

The questions that will concern us in the second lecture will be:

- How do we characterise synchronizing monoids?

Forward look

The questions that will concern us in the second lecture will be:

- How do we characterise synchronizing monoids?
- Which permutation groups are synchronizing?

Forward look

The questions that will concern us in the second lecture will be:

- How do we characterise synchronizing monoids?
- Which permutation groups are synchronizing?
- Which permutation groups are almost synchronizing?

Forward look

The questions that will concern us in the second lecture will be:

- How do we characterise synchronizing monoids?
- Which permutation groups are synchronizing?
- Which permutation groups are almost synchronizing?
- How do these properties relate to more familiar permutation group properties?

Forward look

The questions that will concern us in the second lecture will be:

- How do we characterise synchronizing monoids?
- Which permutation groups are synchronizing?
- Which permutation groups are almost synchronizing?
- How do these properties relate to more familiar permutation group properties?

I will not be saying any more about the Černý conjecture!