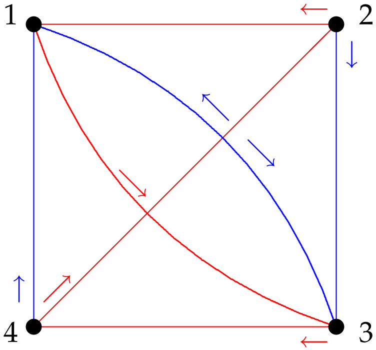
Algorithmic aspects of synchronization

Peter J. Cameron

Queen Mary Algorithms Day, February 2013

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, Blue) takes you to room 3 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.

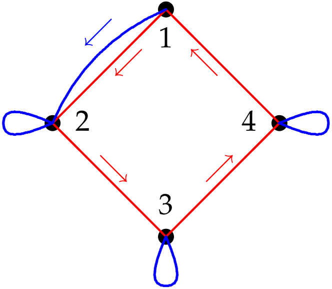
The automaton is synchronizing if there is a word in the transitions which evaluates to a map of rank 1.

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.

Algebraically, since we are interested in composing maps, an automaton is a transformation monoid on Ω (a set of transformations closed under composition and containing the identity map) with a prescribed set S of generators.

Another example

This example arises in industrial robotics.



	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.

Problems

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 show that, if true, it is best possible.

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

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

Testing synchronization

Proposition

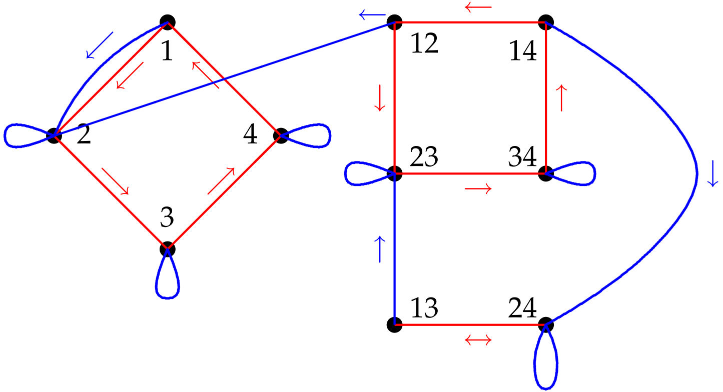
An automaton (Ω, S) is synchronizing if and only if, for any two states $a, b \in \Omega$, there is a word $w_{a,b}$ in the elements of S which maps a and b to the same place.

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.



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.
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 a 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.

Graph endomorphisms

There is a test for synchronization which is not computationally efficient but of very great theoretical value.
An endomorphism of a graph is a map on the vertex set of the graph which maps edges to edges; we do not care what it does to non-edges.

Theorem

A transformation semigroup is not synchronizing if and only if it is contained in the endomorphism monoid of a non-null simple graph with clique number equal to chromatic number.

To prove sufficiency note that an endomorphism of a non-null graph cannot map an edge to a single vertex.

For necessity, suppose we are given a transformation semigroup S on Ω. let the graph X be defined by the rule that vertices v and w of Ω are adjacent if there is no map $s \in S$ which maps v and w to the same place.

A short argument shows that every element of S is an endomorphism of X. If S is not synchronizing then not every pair of points can be collapsed, so the graph is non-null. If s is an element of minimal rank in S, then the image of S is a clique and the map s is a colouring.

Permutation groups

Let G be a permutation group on Ω, a subgroup of the symmetric group on Ω.
The group G is

- transitive if any element of Ω can be mapped by any other by some element of G, that is, there is no non-trivial G-invariant subset of Ω;
- primitive if there is no non-trivial equivalence relation on Ω;
- 2-transitive if it acts transitively on the set of pairs of distinct elements of Ω, that is, there is no non-trivial binary relation on Ω.

A set or relation is trivial if it is invariant under the symmetric group.

Synchronizing groups

There is a lot of interest in semigroups generated by a permutation group together with one non-permutation.
By abuse of language, we call the group G synchronizing if, for any non-permutation s, the semigroup $\langle G, s\rangle$ contains an element of rank 1.
Now the main question is:

Question

Which permutation groups are synchronizing?

Synchronizing groups, 2

Theorem

A permutation group G on Ω is non-synchronizing if and only if there is a non-trivial G-invariant graph on the vertex set Ω with clique number equal to chromatic number.

The forward implication is immediate from the preceding theorem. Conversely, if X is a G-invariant graph with clique number and chromatic number $r > 1$, and s is an r-colouring of X with values in an r-clique, then s is an endomorphism of X, and so $\langle G, s \rangle \leq \mathrm{End}(X)$.

Synchronizing groups, 3

It follows immediately from the theorem that a 2-transitive group is synchronizing (since it preserves no non-trivial graph at all), and a synchronizing group is primitive (since an imprimitive group preserves a complete multipartite graph). Neither implication reverses.

Using the Classification of Finite Simple Groups, we know quite a lot about primitive groups. Can we use this knowledge to find efficient algorithms for testing synchronization?

An algorithm

Given a permutation group G on Ω, is it synchronizing?
Both primitivity and 2-transitivity can be tested in polynomial time, so we may assume that G is primitive but not 2-transitive.

- Compute the non-trivial G-invariant graphs. There are $2^r - 2$ of these, where r is the number of G-orbits on 2-sets. This is potentially exponentially large, but for many interesting groups r is much smaller than n.
- For each such graph, check whether clique number is equal to chromatic number. If we find one, G is non-synchronizing; otherwise it is synchronizing. Of course, clique number and chromatic number are hard in general, but we have highly symmetric graphs here, which shortens the calculation.

Chromatic number of symmetric graphs

Existing software such as `Grape` will find the clique number of a vertex-transitive graph quite fast. Ideally the speed will be improved by a factor which is almost the order of the automorphism group (with a small overhead for managing the group).

Question

How can we best exploit symmetry of a graph to find its chromatic number more efficiently?

An improvement

In a vertex-transitive graph on n vertices, the product of the clique number and the independence number is at most n.

Thus, if clique number equals chromatic number, then equality holds in the above bound, and all colour classes are independent sets of maximum size.

So we can modify the preceding algorithm as follows: first test one of each complementary pair of graphs to see whether the product of clique number and independence number is n. If not, then clique number and chromatic number cannot be equal.

Only for graphs failing this test do we need to compute chromatic number.

Are primitive groups close to synchronizing?

We saw that synchronizing groups are primitive, but not conversely. Of course, if there were only a few primitive non-synchronizing groups, and we could recognise them quickly, we would have a good test for synchronization.

There are a couple of other directions in which it seems that the two properties are close.

João Araújo has conjectured that, if G is primitive, and s is a map which is not uniform (that is, not all its kernel classes have the same size), then $\langle G, s\rangle$ contains an element of rank 1.

Non-synchronizing ranks

Define a *non-synchronizing rank* of G to be a number r for which there is a map of rank r not synchronized by G. Let $NS(G)$ be the set of non-synchronizing ranks.
It is known that, if G is imprimitive, then $NS(G)$ is large (at least $(\frac{3}{4} - o(1)))$, and conjectured that, if G is primitive, then $NS(G)$ is small (maybe only $O(\log n)$. There is a strengthening of "primitive" known as "basic", and it is conjectured that the size of $NS(G)$ for basic groups G is even smaller.