Synchronization 1: Introduction

Peter J. Cameron

LTCC

10-11 June 2010

This is part of an investigation involving, among others, João Araújo, Michael Brough, Ian Gent, Cristy Kazanidis, Tom Kelsey, Peter Neumann, Colva Roney-Dougal, Nik Ruskuc, Jan Saxl, Csaba Schneider, Pablo Spiga, and Ben Steinberg.

Their order here is not intended to reflect the importance of their contributions to the subject!

Much of the communication was done by email. Various presentations of the results have been given, notably a short series by Peter Neumann in Perth last year.

A small example

This example was suggested to me by Olof Sisask.

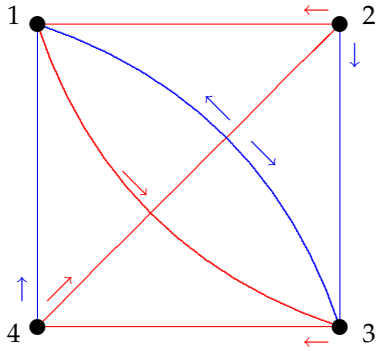
Example 1. A certain calculator has an 'On' button but no 'Off' button. To switch it off, you hold down the 'Shift' key and press the 'On' button. The 'Shift' key has no effect if the calculator is switched off. Assuming that you can't see the screen, how can you ensure that the calculator is switched off?

Obviously, pressing the 'On' button leaves the calculator switched on, no matter what its former state; and then 'Shift-On' will switch it off.

Note that if, instead, there is a single 'On-Off' button which toggles the states, then the problem would have no answer.

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.

Automata and reset words

A (finite, deterministic) *automaton* consists of a finite set Ω of *states* and a finite set of *transitions*, each transition being a function from Ω to itself.

A *reset word* is a sequence of transitions such that the composition of the transitions in the sequence, applied to any starting vertex, brings you to the same state. An automaton which possesses a reset word is called *synchronizing*.

Not every finite automaton has a reset word. For example, if every transition is a permutation, then every word in the transitions is a permutation.

Combinatorially, an automaton is an edge-coloured digraph with one edge of each colour out of each vertex. Vertices are states, colours are transitions.

Transformation monoids

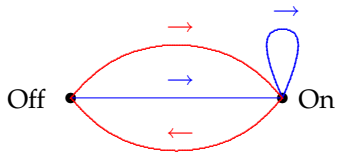
Let $\Omega = \{1, \ldots, n\}$.

The *full transformation monoid* T_n is the set of all functions from Ω to itself. Equipped with the operation of composition, it is a *monoid*: that is, the operation is associative, and there is an identity element (the identity function on Ω).

Algebraically, an automaton is a submonoid of T_n, with a distinguished set of generators. (The "distinguished generators" are the transitions of the automaton. Since we are allowed to compose these arbitrarily, the allowable transitions are all words in the distinguished generators.)

An automaton is synchronizing if and only if (as transformation monoid) it contains a constant function.

Example 2. Consider the example of the calculator. As an edge-coloured directed graph (with blue for 'On' and red for 'Shift-On') it looks like this:



The generators of the monoid are the the function mapping everything to 'On', and the function interchanging 'Off' and 'On'. The remaining elements of the monoid are the function mapping everything to 'Off' (as we saw in the example), and the identity function. Thus, it is the full transformation monoid on the set of states.

Exercise: Calculate the monoid generated by the two transitions in the dungeon example.

The Road-Colouring Conjecture

The underlying digraph of an automaton with n transitions is a digraph with the property that every vertex has exactly n edges leaving it.

Conversely, and trivially, given any digraph with this property, it is clear that it can be edge-coloured so as to represent an automaton.

The resulting automaton may or may not be synchronizing. What are necessary and sufficient conditions for there to be an edge-colouring representing a synchronizing automaton?

We will assume that the automaton can be synchronized in any given state by a suitable reset word. A necessary condition for this is that it is possible to get from any state to any other; in other words, the digraph must be strongly connected. (And, if an automaton is synchronizing and strongly connected, then it can be synchronized at any vertex.)

It is also necessary that the greatest common divisor of the lengths of cycles in the digraph is 1. For suppose the g.c.d. of cycle lengths is d. Choose any vertex v, and let Ω_i be the set of vertices reachable from v in a number of steps congruent to i mod d, for $i = 0, 1, \ldots, d-1$. The sets Ω_i are pairwise disjoint, and so no automaton based on the digraph can be synchronizing.

The conjecture that these two necessary conditions are also sufficient was made in 1970 by Weiss and Adler, and became known as the *Road-Colouring Conjecture*. It was proved by Avraham Trahtman in 2007:

Theorem 3. *Let D be a digraph which is strongly connected and has coonstant out-degree, and suppose that the greatest common divisor of the cycle lengths in D is 1. Then D can be edge-coloured so as to produce a synchronizing automaton.*

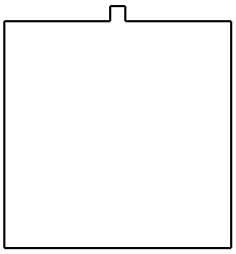
Applications

- Industrial robotics: 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.

- Bioinformatics: If a soup of DNA molecules is to perform some computation, we need the molecules to be all in a known state first. We

can simultaneously apply a reset word to all of them, where the transitions are induced by some chemical or biological process.

Let us consider a simplified example of the robotics application. Suppose that the component is square, with a projection.

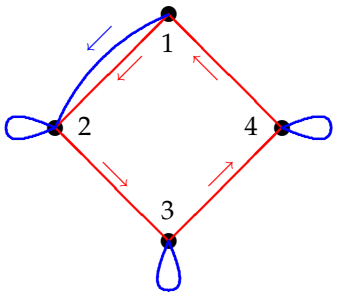


It can sit in a tray on the conveyor belt in any one of four orientations.

The following transitions are easy to implement:

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

Here is a diagram of the automaton. Each state represents the position of the component with the projection on that side.



	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

In 1968, Černý made the following conjecture:

> Suppose that an automaton with n states is synchronizing. Then it has a reset word of length at most $(n-1)^2$.

This conjecture is still open after more than forty years!

Note that the conjecture, if true, is best possible. The example we have just discussed has 4 states, and it can be shown that the reset word of length 9 that we found is best possible. There is an obvious generalisation, where the square is replaced by a regular n-gon, which has n states and shortest reset word of length $(n-1)^2$.

The calculator example with which we began is the case $n = 1$.

An approach via permutation groups

Ben Steinberg and João Araújo suggested an approach to the Černý conjecture, based on permutation groups, which motivates this course. It has not led to a proof of the conjecture, but many interesting probems and conjectures have come up.

Recall the algebraic view of an automaton, a monoid of transformations of a set Ω with a prescribed set of generators. It is synchronizing if it contains a constant function. Now:

- If all the generators are permutations, then the whole monoid consists of permutations; that is, it is a subgroup of the *symmetric group* S_n of all permutations of $\Omega = \{1, \ldots, n\}$.
- In this sense, permutations are the worst transitions for synchronization!
- Moreover, every permutation in the monoid actually lies in the subgroup generated by those transitions which are permutations.

The philosophy is: analyse the group generated by the permutations first!

Let G be a permutation group on Ω, that is, a subgroup of S_n. With an abuse of terminology, we say that G is *synchronizing* if the following is true:

> If f is any function from Ω to itself which is *not* a permutation, then the monoid

$\langle G, f\rangle$ generated by G and f is synchronizing (that is, contains a constant function).

Now the attack on the Černý conjecture goes like this: assume we are in the fortunate position that the transitions which are permutations generate a synchronizing group. Then, given any non-permutation f, there is a reset word in $\langle G, f\rangle$; we need to bound

- the number of occurrences of f in a reset word; and
- the number of generators of G occurring between successive occurrences.

We can suppose that the reset word starts and ends with f; so it has the form $fg_1fg_2f\cdots fg_{r-1}f$ (with r occurrences of f). Our aim might be

- show that we can arrange that each successive application of f reduces the size of the image by at least one (so that $r \leq n-1$);
- show that at most $n-1$ generators of G are required between successive fs (the job of the g_i is to re-position the image of the preceding part of the word so that the next application of f really does shrink it).

Both goals are met in the examples we gave above showing that the Černý bound is sharp.

We will show that the first part of this programme can be achieved by assuming a condition on G which is a strengthening of the synchronizing property just defined.

The second part is more difficult, but group theory gives us some tools for tackling this kind of question.

The basic idea is that we know much more about groups than we do about monoids!

An example

Let us note here that the definition of a synchronizing group requires that adding any non-permutation generates a reset word.

Example 4. In the earlier example of a 4-state automaton with shortest reset word of length 9, the group G is cyclic (generated by the permutation $(1,2,3,4)$). But the cyclic group of order 4 is not a synchronizing group. If we add the function f which maps 1 and 3 to 1, and 2 and 4 to 2, then the image of any element of $\langle G, f\rangle$ outside G consists of two consecutive points of the cycle.

So we should not expect this approach to solve the Černý conjecture completely!

The road ahead

So let us repeat the definition:

A permutation group G on Ω is *synchronizing* if, given any map $f : \Omega \to \Omega$ which is not a permutation, the monoid $\langle G, f\rangle$ generated by G and f is a synchronizing monoid (that is, contains a constant function).

The goal of these lectures is to understand synchronizing permutation groups and related classes of groups. So we turn next to the general theory of permutation groups.