Automatic sequences, their kernels and automata

How to cite:

© 2022 Gandhar Joshi

https://creativecommons.org/licenses/by/4.0/

oro.open.ac.uk
Imagine a universe of ducks. On Day 0, an adult \( \exists \) exists. On Day 1, through immaculate conception?! the duck lays an \( \exists \). On Day 2, the \( \exists \) hatches, and becomes a baby \( \exists \). And on Day 3, you guessed it! \( \exists \) grows into a \( \exists \). But simultaneously, every day, the duck we started with is laying eggs religiously, and any egg from previous day turns into a baby duck, and any baby duck from previous day turns into an adult duck. So, we put them in a stretchable line... Oh, did I tell you, the ducks don’t die? Yep. They exist forever till the end of time...

**AUTOMATIC SEQUENCES, Their kernels & automata**

Then, we get the sequence of the letters assigned as debbdedebddebdddebde... This is called a **morphic sequence**!

Based on certain rules (MORPHISMS), the automatic sequences are morphic sequences which are accepted by a k-DFAO. Now, whatever is this DFAO, you wonder?

Automaton is a machine that accepts or rejects an automatic sequence. DFAO is an automaton with an output. It looks a bit like this—

The **states** hold input, the **edges** direct based on the next input, and / signifies the output.

This directed graph counts the sum of digits of the binary representation of integers starting with 0 in mod 2, which generates the famous automatic sequence known as the Thue-Morse sequence. This is a 2-DFAO machine.

Now, every automatic sequence has a **kernel** — a set of subsequences, and COBHAM states (with proof of course) that they are finite.

Now, let us call the adult duck by ‘d’, an egg by ‘e’, and the baby duck by ‘b’. We could rewrite the sequence on day 7 as follows. Also, notice that the terms at any position in the line do not change from day 1 to day 7 and theoretically day \( \infty \).