Internship proposal: Algebra and Logic in Automata Theory

Nathanaël Fijalkow Filip Mazowiecki

January 20, 2018

Encadrants: Nathanaël Fijalkow (CNRS) et Filip Mazowiecki

Localisation: LaBRI (Bordeaux)

Sujet du stage: Ce stage propose d'étudier les aspects algébriques et logiques des automates mesurant des propriétés quantitatives.

Thèmes: Théorie des automates, Logique, Algèbre

The proposal starts with a riddle that is in the spirit of our research.

A riddle

Let d be a positive natural number. A linear recurrence sequence (a_n) of order d is given by an equation of the form

$$a_n = c_1 a_{n-1} + \cdots + c_d a_{n-d},$$

for $c_i \in \mathbb{Z}$; and by fixing the values of the first d elements $a_0, \ldots, a_{d-1}$. The most popular example is the Fibonacci sequence F_n of order 2 given by

$$F_n = F_{n-1} + F_{n-2}$$

and by fixing $F_0 = F_1 = 1$. The order d is the depth of recursion we require to compute a new element of the sequence. It is possible to restrict the sequences to order 1 at the cost of adding new sequences obtaining a system of linear recurrence sequences. For example F_n could be given by

$$\begin{cases} F_n = F_{n-1} + G_{n-1} \\ G_n = F_{n-1} \end{cases}$$

and fixing $F_0 = 1$ and $G_0 = 0$. Essentially G_n is a buffer to remember F_{n-2}. This way any linear recurrence sequence can be turned to system of linear recurrence sequences of order 1.

Can we do the opposite? For example the following system defines the sequence $a_n = n^2$

$$\begin{cases} a_n = a_{n-1} + 2b_{n-1} + 1 \\ b_n = b_{n-1} + 1 \end{cases}$$

fixing $a_0 = b_0 = 0$. Can we define it with only one linear recurrence system (possibly of depth larger than 1)? Can we do it for any system of linear recurrence sequences? Hint: linear algebra.

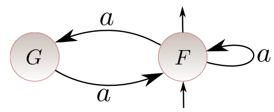

Figure 1: An example of a weighted automaton.

Weighted automata

Weighted automata capture linear recurrence sequences. We can think of a classical automaton as inducing a function $A^* \to \{0, 1\}$, where 1 means that the word is accepted. A weighted automaton generalises this by inducing a function $A^* \to \mathbb{N}$. For example, for the automaton in Figure 1 the alphabet is only the letter a and we associate to the word a^n its number of accepting runs. This automaton computes the Fibonacci sequence.

The goal of the internship is to study the algorithmic properties of weighted automata, for instance we want to determine whether two automata compute the same functions, or whether the function it computes is bounded, and so on. Technically, this involves linear algebra, logic, and combinatorics.