Using Automata and a Decision Procedure to Prove Results in Pattern Matching

Jeffrey Shallit
School of Computer Science, University of Waterloo, Canada

Abstract

The first-order theory of automatic sequences with addition is decidable, and this means that one can often prove combinatorial properties of these sequences "automatically", using the free software Walnut written by Hamoon Mousavi. In this talk I will explain how this is done, using as an example the measure of minimize size string attractor, introduced by Kempa and Prezza in 2018.

Using the logic-based approach, we can also prove more general properties of string attractors for automatic sequences. This is joint work with Luke Schaeffer.

2012 ACM Subject Classification Mathematics of computing → Combinatorics on words; Theory of computation → Regular languages; Theory of computation → Logic and verification

Keywords and phrases finite automata, decision procedure, automatic sequence, Thue-Morse sequence, Fibonacci word, string attractor

Digital Object Identifier 10.4230/LIPIcs.CPM.2022.2

Category Invited Talk


Funding Research supported by NSERC 2018-04118.

1 Introduction

Many famous sequences, such as the Thue-Morse sequence $t = 01101001 \cdots$ and the Fibonacci infinite word $f = 01001010 \cdots$ appear as fundamental examples in combinatorial pattern matching.

As just a few examples, I point to [5, 1, 12], where the Thue-Morse sequence makes an appearance, and [13], where the Fibonacci infinite word is studied.

A fundamental result, essentially due to Büchi [4] and Bruyère et al. [3], tells us that the first-order theory of such sequences, with addition, is decidable, and there is a relatively simple decision procedure based on automata. This decision procedure has been implemented in free software called Walnut, originally created by Hamoon Mousavi [11]. Therefore, in many cases, we can prove properties of such sequences of interest to the CPM community “automatically”, merely by stating the desired property in first-order logic, and invoking Walnut.

Recently there has been interest in a certain measure of repetitivity, based on string attractors, originally introduced by Kempa and Prezza [6], and studied further in [9, 7, 8, 10, 2]. A string attractor of a finite word $w = w[0..n-1]$ is a subset $S \subseteq \{0, 1, \ldots, n-1\}$ such that every nonempty factor $f$ of $w$ has an occurrence that touches at least one of the indices of $S$. For example, $\{2, 3, 4\}$ is a string attractor of minimum size for the French word entente.

In this talk I will introduce Walnut, and explain how to obtain results on string attractors using it and the theory behind it. This is joint work with Luke Schaeffer [14].
2:2 Using Automata and a Decision Procedure to Prove Results in Pattern Matching

2 Results

As an example of the kind of thing we can prove with Walnut, here is one theorem:

► Theorem 1. Let \( a_n \) denote the size of the smallest string attractor for the length-\( n \) prefix of the Thue-Morse word \( t \). Then

\[
a_n = \begin{cases} 
1, & \text{if } n = 1; \\
2, & \text{if } 2 \leq n \leq 6; \\
3, & \text{if } 7 \leq n \leq 14 \text{ or } 17 \leq n \leq 24; \\
4, & \text{if } n = 15, 16 \text{ or } n \geq 25.
\end{cases}
\]

More generally, we can prove

► Theorem 2. Let \( w \) be a \( k \)-automatic sequence. Either

- every factor \( w[i..i+\ell-1] \) has a string attractor of constant size, and there exists a finite automaton outputting the minimum size given \( i \) and \( \ell \), or
- for all \( n \geq 1 \), the minimum size string attractor for the length-\( n \) prefix \( w[0..n-1] \) grows as \( \Theta(\log n) \),

and we can decide which is the case for \( w \).

For more about Walnut and its applications in combinatorics on words, see my forthcoming book [15].

---

References


