

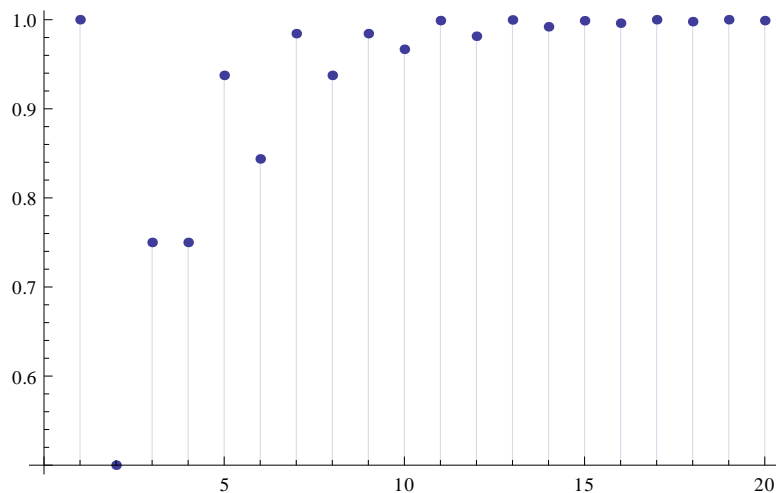
Problem 1: Primitive Words (30 pts.)

Background

Let w be a non-empty word. A word z is the *root* of w if $w \in z^*$ but there is no shorter word with this property. A word is *primitive* if it is its own root. For example, ab is the root of $abababab$. The primitive words over the alphabet $\{a, b\}$ up to length 4 are

- 1 a, b
- 2 ab, ba
- 3 $aab, aba, abb, baa, bab, bba$
- 4 $aaab, aaba, aabb, abaa, abba, abbb, baaa, baab, babb, bbaa, bbab, bbba$

Write $\pi(n)$ for the number of primitive words of length n (over some fixed alphabet). A plot of $\pi(n)/k^n$ for the case $k = 2$ follows.



Task

- A. Show that any two non-empty words u, v that commute (i.e., $uv = vu$) have the same root.
- B. Show that a non-empty word w is primitive if, and only if, $w^2 = xwy$ implies that $x = \varepsilon$ or $y = \varepsilon$.

- C. Count the number $\pi(n)$ of primitive words of length n over a k letter alphabet. What is the asymptotic behavior of π ? Clearly, the picture above suggests an answer.
- D. As part of some algorithm it is necessary to store a few thousand binary words of length 20. Prof. Dr. Wurzelbrunft suggests to speed things up by storing the roots of words, rather than the words directly. What professional advice can you give Wurzelbrunft?

Comment

For part (C) you might want to consider the Möbius function.

Problem 2: Divisibility In Reverse Binary (40 pts.)

Background

We have seen that one can check divisibility by a fixed modulus m on a DFA regardless of the base B . More precisely, we showed how to construct a DFA for words x written in base B notation with the MSD in the first position. Of course, there is no real reason why the MSD should be up front, it might as well be the last digit (reverse binary notation).

Task

1. Construct a DFA that tests divisibility by 3 for numbers written in reverse binary. To do this, find a way to express $\nu(xa)$ in terms of $\nu(x)$ and $|x|$ and choose your state set accordingly.
2. Show how to generalize this construction for arbitrary moduli m and bases B .
3. How does the size of your machine for $m = 5$ and $B = 2$ in reverse binary compare to the machine constructed in class for normal binary notation? What is going on?

Comment

Naturally one would like to know how large the minimal automata are in general, for arbitrary base B and arbitrary modulus m , and for ordinary base B notation as well as reverse base B . Lots of extra credit if you can give a general characterization.

Problem 3: Balance and Majority (30 pts.)**Background**

Write $|x|_0$ for the number of 0's in a binary word x , and likewise $|x|_1$ for the number of 1's. The balance language L_{bal} is the set of all binary words that have the same number of 1's and 0's:

$$L_{bal} = \{ x \in \mathbf{2}^* \mid |x|_1 = |x|_0 \}.$$

The majority language L_{maj} is the set of all binary words that have at least as many 1's as 0's:

$$L_{maj} = \{ x \in \mathbf{2}^* \mid |x|_1 \geq |x|_0 \}.$$

Task

- A. Prove that there is no finite state machine that accepts L_{bal} .
- B. Prove that there is no finite state machine that accepts L_{maj} .

Comment

The essential issue is that in order to recognize these languages a DFA would require some kind of unbounding counting ability – and DFAs can only count in a bounded way. Try to come up with a clean, concise argument. Also note that your argument has to cover all possible finite state machines, you cannot make any assumptions about, say, the size of the machine.