COMBINATORICS ON WORDS

OLGA RADKO MATH CIRCLE
ADVANCED 2
OCTOBER 3, 2021

1. Introduction (the only section where I'm even a little excited about the quarter)

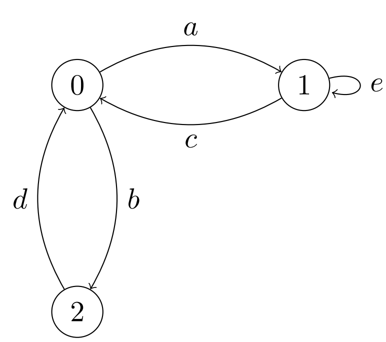
Suppose there is a door that only opens when the correct 2-digit binary (i.e. 0s and 1s) code is entered on a keypad. Suppose further that the door opens as soon as the 2-digit code is entered so, if the code is 00, the door opens after 100 is entered. There are 4 possible codes for this keypad which amounts to 8 binary digits. For instance, we could try 00011011. However, the code 00110 is shorter than 8 digits and guarantees that the door will open.

Problem 1. Now suppose you have to guess a 3-digit binary code on a keypad.

(a) How many different codes are possible?
(b) Try to come up with the shortest binary sequence that is guaranteed to open the door.
(c) (Challenge) Explore the situation for 4-digit codes.

2. Graph Theory Review (do Math Circle instructors know any other areas of math?)

Before we continue, we need to review some vocabulary about graphs. A *directed graph* is a set of vertices and edges (ordered pairs of vertices). We have an example of directed graphs below which we will denote graph A. Graph A has 3 vertices, $\{0, 1, 2\}$, and 5 edges, $\{a, b, c, d, e\}$. Note that a vertex can have an edge to itself.



A *directed path* is a finite sequence of directed edges connecting two vertices. An *Eulerian path* is a traversal of a graph that visits each edge exactly once. An *Eulerian cycle* is an Eulerian path that starts and ends at the same vertex. For example, *aecbd* is an Eulerian cycle of graph A starting at vertex 0.

A directed graph G has an Eulerian cycle if and only if there is a path between any two vertices of G and the number of edges into each vertex equals the number of edges out of each vertex.

Problem 2. Find all Eulerian cycles in graph A starting at the vertex 0.

Given a graph G, we construct the *line graph of* G (denoted $L(G)$) as follows. Every edge in G becomes a vertex in $L(G)$. If the end of edge a is the start of edge b in G, then there is an edge from vertex a to vertex b in $L(G)$. Note that the line graph of a directed graph will be a directed graph.

Problem 3. Draw $L(A)$.

Problem 4. A directed graph G is *connected* if there is a path between any two vertices of G. Show that if G is connected, then $L(G)$ is connected.

3. Words (isn't every section basically just words?)

Suppose we have an *alphabet* A (made up of letters). Today we will be mainly concerned with binary (letters 0, 1) or ternary (letters 0, 1, 2) alphabets. A *word* w (in the alphabet A), is simply a sequence of letters. For example, 001, 101, and 111 are all binary words, and 102 is a ternary word. We will allow a word of length 0, denoted by ε. A word v is a *subword* of a word w if v is contained in w. For instance, 10 is a subword of 101 but 11 is not. If u and v are two words, then we use the natural notation uv to denote the word u followed by the word v. For example, if $u = 01$ and $v = 10$, then $uv = 0110$.

Problem 5. List the subwords of 001.

We let $p_w(n)$ denote the number of distinct subwords of w with length n. If $w = 001$, then

$$\begin{aligned} p_w(0) &= 1, \\ p_w(1) &= 2, \\ p_w(2) &= 2, \\ p_w(3) &= 1 \end{aligned}$$

by Problem 5.

Problem 6. Let $u = 101001$ and $v = 012202$.

(a) Calculate, for $n = 0, 1, \ldots, 6$ the value of $p_u(n)$.
(b) Calculate, for $n = 0, 1, \ldots, 6$ the value of $p_v(n)$.

Problem 7. Suppose we have an alphabet A of size k. Prove the following facts (where w is any word with letters from A):

(a) $p_w(n) \leq k^n$
(b) $p_w(n) \geq p_w(n-1) - 1$
(c) $p_w(n) \leq k \cdot p_w(n-1)$

Problem 8. The *Champernowne word* of order m, denoted by c_m, is obtained by writing successively the binary representations of the natural numbers $0, 1, \ldots, 2^m - 1$. For example,

$$c_0 = 0, c_1 = 01, c_2 = 011011, c_3 = 011011100101110111.$$

(a) Let $w = c_3$. Compute $p_w(0)$, $p_w(1)$, $p_w(2)$, and $p_w(3)$.
(b) Let $w = c_m$. Prove that $p_w(n) = 2^n$ for $n < m$. Prove also that $p_w(m) < 2^m$.

Problem 9. The *Fibonacci word* of order m (denoted by f_m) is defined recursively as follows

$$f_0 = 0, f_1 = 1, f_n = f_{n-1} f_{n-2}.$$

(a) Write out f_3, f_4, and f_5.
(b) Let $w = f_5$. Calculate $p_w(n)$ for $n = 0, 1, 2, 3, 4, 5$.
(c) (Challenge) Prove by induction that the length of f_n is the $(n+2)$-th Fibonacci number. Recall that the the Fibonacci sequence is recursively defined as $a_0 = 0$, $a_1 = 1$, and $a_{n+1} = a_n + a_{n-1}$.

4. de Bruijn Words (ten bucks if you can pronounce this properly)

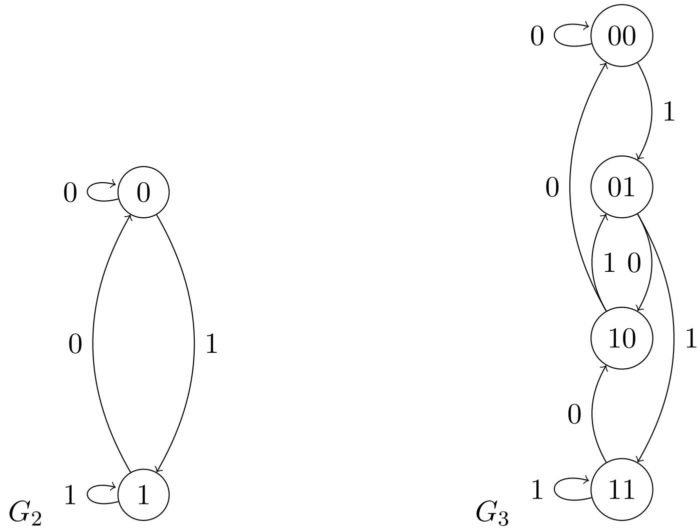
Recall Problem 1 where we tried to find a binary code that contains all length n binary sequences as subwords. The Champernowne word of order m is a solution to this problem for codes of length $m - 1$ by Problem 8(b). It is natural to want to come up with an optimal (i.e. shortest) such code. We will call an optimal word containing all length n binary codes as subwords a *de Bruijn word* of order n. Restated, w is a de Bruijn word if $p_w(n) = 2^n$ and there is no shorter word v with $p_v(n) = 2^n$.

Problem 10. Let L_n be the length of a de Bruijn word of order n. Prove the following bounds.

(a) $L_n \leq n \cdot 2^n$
(b) $L_n \geq 2^n + n - 1$

We now want to show that $L_n = 2^n + n - 1$. Suppose we have a word w. Call u the *prefix* of w if u is all of w except for the last letter. Similarly, v is the *suffix* of w if v is all of w except for the first letter. For example, 0101 has prefix 010 and suffix 101. Further, 1 has prefix and suffix equal to ε.

A *de Bruijn graph* of order n, denoted G_n is constructed as follows. The vertices of the graph are all binary words of length $n-1$. There is an edge from vertex u to vertex v if the suffix of u is equal to the prefix of v. We label this edge by the last letter of v. The order 2 and order 3 de Bruijn graphs are below.



Problem 11. (a) Draw G_4.

(b) Show that G_n has 2^{n-1} vertices and 2^n edges, each vertex has 2 edges going in and going out, and that there are the same number of edges labeled 0 as there are edges labeled 1.

We can now construct a de Bruijn word w of order n as follows.

(1) Construct G_n.
(2) Find an Eulerian cycle of G_n.
(3) The word w is the concatenation of the Eulerian cycle's starting vertex with the Eulerian cycle.

Recall that a graph has an Eulerian cycle if and only if there is a path between any two vertices of G and the number of edges into each vertex equals the number of edges out of each vertex.

Problem 12. (a) Find the de Bruijn words of orders 2, 3, and 4.

(b) Argue that we actually are constructing de Bruijn words. That is, if we use G_n to construct a word w using the method above, the length of w is $2^n + n - 1$ and $p_w(n) = 2^n$.

Problem 13. Recall the definition of the line graph $L(G)$ from Graph Theory Review. In G_n, let x be the first letter of vertex v and y be the last letter of vertex w. Relabel the edge from $v = xu$ to $w = uy$ as xuy. We will refer to this as the length n labeling of G_n.

(a) Construct $L(G_2)$.
(b) Construct $L(G_3)$.

What do you notice?

5. Sturmian Words (oh shoot this is the section I've been waiting for)

De Bruijn words are the shortest possible words containing all subwords of a certain length. One might want to know if a similar construction exists where, instead of having all possible subwords, the word has some fixed number of distinct subwords. This can be a difficult problem in general, but such a construction exists for words that have $m+1$ distinct length m subwords for each $m \leq n$.

Call a word w a *Sturmian word* of order n if $p_w(m) = m + 1$ for all $m \leq n$. Furthermore, it is called a *minimal Sturmian word* of order n if it is a Sturmian word and there is no shorter Sturmian word.

Problem 14. Show that the length of a Sturmian word of order n is at least $2n$.

Start with G_3. Remove 4 edges and call the remaining graph G_3'. If possible, find an Eulerian path of G_3'. We will study the concatenation of the starting vertex of the Eulerian path with the Eulerian path.

Problem 15. Show that each of the following situations is possible.

(1) The graph G_3' has no Eulerian path.
(2) The graph G_3' has an Eulerian path, leading to a word w with $p_w(3) = 4$ but $p_w(2) = 4$.
(3) The graph G_3' has an Eulerian path, leading to a word w that is a minimal Sturmian word of order 3.

Start with G_2 where the edge labels are length 2. Remove one of the edges to get G_2'. Now compute $L(G_2')$. If $L(G_2')$ has 4 edges, set $G_3' = L(G_2')$. If not, remove one edge from $L(G_2')$, ensuring that it still has an Eulerian path, and call the graph G_3'. Label an edge from vertex u to vertex v in G_3' as the last letter of v. Let w be the concatenation of the starting vertex of the Eulerian path with the Eulerian path.

Problem 16. Attempt the above construction a few times. Is w a minimal Sturmian word?

We will now construct a minimal Sturmian word of order $n \geq 3$ recursively.

(1) Start with G_2 with length 2 edge labels and create G_2' by removing an edge from G_2.
(2) Compute $L(G_2')$. Remove an edge from $L(G_2')$ if necessary while still ensuring we have an Eulerian path. Denote this graph with 4 edges G_3'.
(3) Apply Step 2 to G_{n-1}' with length $n-1$ edge labels to produce G_n', a graph with $n+1$ edges and an Eulerian path. Label an edge from vertex u to vertex v of G_n' as the last letter in v.
(4) Construct a minimal Sturmian word w of order n by concatenating the starting vertex of the Eulerian path of G_n' with the Eulerian path of G_n'.

Problem 17. (a) Use the method above to construct a minimal Sturmian word of order 4.
(b) Use the method above to construct a minimal Sturmian word of order 5.
(c) Argue that we actually are constructing minimal Sturmian words. That is, if we use the above method to construct w, the length of w is $2n$ and $p_w(m) = m + 1$ for all $m \leq n$.

As extra content, you can think about words w for which $p_w(m) = 2m$ for all $m \leq n$. Or you can try to extend the ideas in this worksheet to a ternary alphabet.