Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

On the extension of the lexicographically greatest de Bruijn sequence to larger alphabets

Tesis de Licenciatura en Ciencias de la Computación

Gabriel Eric Thibeault Lüthy

LU 114/13

Email: gabriel.eric.thibeault@gmail.com

Director: Verónica Becher

Buenos Aires, July 29, 2019

Sobre la extensión de la secuencia de Bruijn lexicográficamente máxima a alfabetos más grandes

Resumen. Una secuencia de Bruijn de orden n en k símbolos es una secuencia en la que cada palabra de longitud n ocurre exactamente una vez. Se sabe que para cada secuencia circular de Bruijn v de orden n en k símbolos hay otra secuencia circular de Bruijn w de orden n pero en $k + 1$ símbolos tal que v es una subsecuencia de w. En esta tesis nos dedicamos a la secuencia de Bruijn lexicográficamente máxima, a la que llamamos dual-Ford. El nombre se debe a que la secuencia de Bruijn lexicográficamente mínima es conocida como la secuencia de Ford. Demostramos que la secuencia dual-Ford de un orden dado y un alfabeto dado es un sufijo de la secuencia dual-Ford del mismo orden en un alfabeto con un símbolo más. Dado que hay un algoritmo óptimo en tiempo (lineal relativo al tamaño de la salida) para generar las secuencias Ford y las duales-Ford, el resultado que presentamos determina un algoritmo óptimo en tiempo para generar la extension de una en la otra.

Palabras claves: secuencias de Bruijn, secuencia Ford, ciclos Eulerianos.

On the extension of the lexicographically greatest de Bruijn sequence to larger alphabets

Abtract. It is known that for each circular sequence of Bruijn v of order n in k symbols there is another circular sequence of Bruijn w of order n but in $k+1$ symbols such that v is a subsequence of w. In this thesis we focus on the lexicographically greatest de Bruijn, which we call dual-Ford. The name follows because the Bruijn lexicographically least de Bruijn sequence is known as the Ford sequence. In this work we show that the dual-Ford sequence of a given order and on a given alphabet is a suffix of the dual-Ford sequence of the same order on an alphabet with an additional symbol. Since there is a time-optimal (linear relative to the size of the output) algorithm that generates the Ford and the dual-Ford sequences, our result yields a time-optimal algorithm to generate the extension of one to the other.

Keywords: de Bruijn sequences, Ford sequences, Eulerian cycles.

CONTENTS

1. THE PROBLEM

The original intent of this thesis was to provide an efficient algorithm to extend a de Bruijn sequence of order n on an alphabet of size k to one of the same order on an alphabet of size $k+1$. In the process of attaining this goal, we found a curious result for the specific case of the lexicographically greatest de Bruijn sequence, the $\overline{\text{Ford}}$ sequence, which is dual to the lexicographically least de Bruijn sequence, the well-known Ford sequence. We found that $\overline{\text{Ford}}_{k,n}$ is a suffix of the $\overline{\text{Ford}}_{k+1,n}$. This thesis consists of the statement and proof of this theorem, along with a corollary employing this result to provide an algorithm whose time complexity is asymptotically optimal that given $\overline{\text{Ford}}_{k,n}$ produces $\overline{\text{Ford}}_{k+1,n}$.

Throughout this thesis we'll extend an alphabet by adding a lexicographically greatest symbol; adding a lexicographically least symbol instead yields the dual result for the Ford sequence.

2. PRIMARY DEFINITIONS

2.1 Generalities on words

An *alphabet* is a finite set and its elements are called *symbols.* For example, the binary alphabet $\{0, 1\}$. A *word* on an alphabet Σ is a sequence of symbols belonging to Σ. For example 0110 is a word on the alphabet $\Sigma = \{0, 1\}$. The length of a word w is denoted with $|w|$. We write w_i to indicate the symbol at position i of w. For an alphabet Σ and a positive interger n, Σ^n is the set of words on Σ of length n. For example, for $\Sigma = \{0, 1\}$, $\Sigma^2 = \{00, 01, 10, 11\}$. Without loss of generality it shall be assumed that alphabets of size k consist of the symbols $\{0, \dots,\ k - 1\}$.

A word v is said to be a subsequence of another word w if there is a sequence of increasing positions in w, $i_1, i_2, \dots i_n$ such that v is equal to $w_{i_1} w_{i_2} \dots w_{i_n}$ A *substring* v of a given string w is a subsequence that is contiguous in the original word w. For example, 11 is a subsequence and 10 is a substring of 1010.

Given two words u, v, respectively on the two alphabets Σ and Δ, the *concatenation* of u and v, noted uv, is a word on $\Sigma \cup \Delta$ consisting of u followed by v. Thus, 0123 is the concatenation of 01 and 23. Given a word or a symbol (seen as a word of length one) a and some length n, a^n is the result of concatenating a with itself n many times. For instance, $(011)^2$ is 011011.

Necklaces are the equivalence classes of words under rotation. A necklace may be represented by any of its members. The notation

$$[0110]$$

denotes the equivalence class containing 0110, 1100, 1001 and 0011.

2.2 Graphs

The following is standard material in graph theory, which can be read from any of these monographs [10, 8, 9]. A *Hamiltonian* cycle of a graph is a cycle in which each vertex of the graph occurs exactly once. An *Eulerian* cycle is a cycle in which each edge of the graph occurs exactly once. A graph that admits an Eulerian cycle is called *Eulerian.* An undirected graph is *connected* if there is a path between every pair of vertices. A directed graph is *strongly connected* if there is a directed path between every pair of vertices. A directed graph is *regular* if each vertex has the same number of incoming and outgoing edges as all other vertices. Given a directed graph G, its *line graph* is a directed graph whose vertices are the edges of G and whose edges correspond to the directed paths of length two of G.

2.3 de Bruijn necklaces and de Bruijn graphs

The de Bruijn necklaces receive their name due to the work of Nicolaas Gover de Bruijn [2]. However they have been discovered by several authors, possibly the first was Camille Flye Sainte-Marie [11]. See [3] for a fine presentation and history.

A necklace is *de Bruijn* of order n on an alphabet Σ of size k if every word on Σ of length n appears exactly once as a substring (possibly wrapping around). One such necklace of order 2 for alphabet size 3 is

$$[021122010]$$

since it contains every word of length 2 over the alphabet $\{0, 1, 2\}$ exactly once.

The *de Bruijn graph* of order n for an alphabet size k, noted $G_{k,n}$, is a directed, labeled graph such that the vertices correspond to the words of length n on Σ, with an edge from u to v with label $b \in \Sigma$ if $u = a_1, a_2, \ldots a_n$ and $v = a_2, \ldots, a_n, b$. More formally,

$G_{k,n} = (V, E)$ for all $V = \Sigma^n$ and

$E = \{(u, v) \in V^2 | \ \exists\ a, b \in \Sigma, w \in \Sigma^{n-1}/u = aw \wedge v = wb\}$

The line graph of the de Bruijn graph of order n on an alphabet Σ of size k is the de Bruijn graph of order $n + 1$ on Σ.

The edges of $G_{k,n}$ can be labeled with sequences of length $n + 1$, such that the edge (w, v) is labeled with $w_1 v = w v_n$. Then, each possible sequence of length $n+1$ in k symbols appears in exactly one edge of $G_{k,n}$. Moreover, the line graph of $G_{k,n}$ is exactly $G_{k,n+1}$. The label of a path $v_1, \ldots, v_t$ in $G_{k,n}$ is the sequence that contains as substrings exactly the sequences $v_1, \ldots, v_t$, in that order. Taking a path of length t in $G_{k,n}$ and consider the set of $t-1$ traversed edges, it is easy to see that they form a path that has the same label in $G_{k,n+1}$. In the same way, the labels of a cycle yield a necklace.

The label of a Hamiltonian cycle in $G_{k,n}$ is a de Bruijn necklace of order n on an alphabet of size k, and the label of an Eulerian cycle in $G_{k,n}$ is a de Bruijn necklace of order $n+1$ on an alphabet of size k, as an Eulerian cycle in $G_{k,n}$ is a Hamiltonian cycle in $G_{k,n+1}$.

By choosing a vertex as the *starting point* of the cycle, a specific sequence of the necklace is obtained. Given $k, n \in \mathbb{N}$, note that the n-length path leading to the starting vertex is visited at the end of the cycle. Since the concatenation of the labels of this path correspond to the label of the vertex, the label of the starting vertex is a suffix of the resulting sequence. For example, for where $k = 2,\ n = 3$, picking 000 as the starting point of the de Bruijn necklace

$$[11100010]$$

the resulting sequence is

$$10111000.$$

The number of de Bruijn necklaces of order n in an alphabet of size k is

$$\frac{k!^{k^{n-1}}}{k^n}$$

Thus, among these many necklaces there is a lexicographically least and a lexicographically greatest.

Given a directed graph $G = (V, E)$, an *in-arborescence* rooted at $v \in V$ is a subgraph of G such that there exists exactly one path from every vertex in V to v. Hence, an *in-arborescence* rooted at $v \in V$ it is a directed spanning tree in which every vertex points to the root v.

2.4 The *BEST* Theorem

In graph theory, the *BEST* theorem gives a formula for the number of Eulerian cycles in directed graphs. The name is an acronym of the names of people who discovered it: de Bruijn, van Aardenne-Ehrenfest, Smith and Tutte, see [10, 9]. Though the theorem concerns only the amount of Eulerian cycles in directed graphs, its proof demonstrates an even stronger result which shall be presently stated as a corollary.

Corollary (BEST Theorem)**.** Let $G = (V, E)$ be a directed, Eulerian graph. Let $e_1 = (s, \tilde{s})$ be an arbitrary starting edge, to be called the *special edge*. There exists a bijection between in-arborescences T converging to s and sets $\mathscr{W}(T)$ of Eulerian cycles starting with the edge e_1. Each set $\mathscr{W}(T)$ is of size

$$d := \prod_{v \in V} (d^-(v) - 1)!,$$

where $d^-(v)$ is the out-degree of v.

The mapping from an Eulerian cycle to an in-arborescence is as follows. Given an Eulerian cycle W, order its edges starting with the special edge e_1. For each vertex $v \neq s$, let $e(v)$ be the last edge leaving v (the one with the greatest index in the ordering). Then the subgraph

$$T(W) := \big(V, \{e(v) \mid v \neq s\}\big)$$

is W's corresponding in-arborescence.

The mapping from an in-arborescence to a set of Eulerian cycle is as follows. Let T be an in-arborescence converging to s. For every vertex $u \in V$, label the $d^-(u)$ edges leaving u,

1. Among all edges (s, v) leaving s, the special edge $e_1 = (s, \tilde{s})$ is labelled with the number 1.

2. For all v diferent from s, the tree-edge $(v, v') \in E(G)$ is labelled with the greatest number (which is $d^-(v)$).

Otherwise, the numbering is arbitrary. Construct a cycle W by starting in $(s, \tilde{s})$ and, at every vertex, taking the edge with the smallest number that has not yet been taken. Each different ordering (following the two conditions) results in a different cycle, and there are

$$d := \prod_{v \in V} (d^-(v) - 1)!$$

cycles for each in-arborescence.

2.5 The Ford sequence and its dual $\overline{\mathrm{Ford}}$

Given two words u, v; u is *lexicographically less* than v if and only if u is a strict prefix of v or, for the first index i for which u and v differ, $u_i < v_i$.

Given $k, n \in \mathbb{N}$, the lexicographically least de Bruijn sequence of order n on an alphabet of size k is called the $\mathrm{Ford}_{k,n}$, while the lexicographically greatest is called the $\overline{\mathrm{Ford}}_{k,n}$. In Figure 2.5 some examples are given.

There is a greedy algorithm, optimal in time computational complexity, to produce the Ford and $\overline{\mathrm{Ford}}$ sequences. Given $k, n \in \mathbb{N}$, the Eulerian cycle in $G_{k,n}$ obtained by starting at the lexicographically least vertex and always taking the available edge with the greatest lexicographic label yields the $\overline{\mathrm{Ford}}_{k,n+1}$ sequence. Starting at the lexicographically greatest

$$
\begin{aligned}
\mathrm{Ford}_{2,2} &= 0011\\
\overline{\mathbf{Ford}}_{2,2} &= 1100
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Ford}_{2,3} &= 00010111\\
\overline{\mathbf{Ford}}_{2,3} &= 11101000
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Ford}_{3,2} &= 001021122\\
\overline{\mathbf{Ford}}_{3,2} &= 221201100
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Ford}_{3,3} &= 000100201101202102211121222\\
\overline{\mathbf{Ford}}_{3,3} &= 222122021121020120011101000
\end{aligned}
$$

Fig. 2.1: Ford and $\overline{\mathrm{Ford}}$ sequences for $k = 2, 3$ and $n = 2, 3$.

Algorithm P(**in:** $k \in \mathbb{N}$, $n \in \mathbb{N}$)
out: $\overline{\mathrm{Ford}}_{k,n+1}$, the lexicographically greatest de Bruijn sequence of order $n+1$ in k symbols

```
    G_{k,n} ← the de Bruijn graph of order k, n with labeled edges
    result ← emptyWord
    // Start with the lexicographically least vertex in G_{k,n}
    currentVertex ← 0^n
    while currentVertex has edges not yet taken in result do
        nextEdge ← the greatest edge not yet taken in result
        append nextEdge's label to result
        currentVertex ← nextEdge's destination vertex
    return result
```

vertex and always taking the available edge with the least lexicographic label yields the $\mathrm{Ford}_{k,n+1}$ sequence. The algorithm **P** presented above produces the $\overline{\mathrm{Ford}}_{k,n+1}$ sequence.

A beautiful result characterizes the Ford sequence of order n on an alphabet Σ as the concatenation of the Lyndon words (in lexicographic order) on Σ whose lengths divide n. A Lyndon word is a non-empty string that is strictly smaller in lexicographic order than all of its rotations. For example, for alphabet $\Sigma = \{0, 1\}$ the concatenation of the Lyndon words whose length divides four is

$$0\ 0001\ 0011\ 01\ 0111\ 1$$

This construction, together with the efficient generation of Lyndon words, provides an efficient method for constructing a particular de Bruijn sequence [7, 5].

3. ON THE EXTENSION OF THE LEXICOGRAPHICALLY GREATEST DE BRUIJN SEQUENCE TO LARGER ALPHABETS

Theorem 1. *The* $\overline{\text{Ford}}$ *sequence of any given order in an alphabet of a given size is a suffix of the* $\overline{\text{Ford}}$ *sequence of the same orden in an alphabet with one more symbol.*

Corollary. *There is a time-optimal algorithm such that for any choice of positive integers* k *and* n *it extends the sequence* $\overline{\text{Ford}}_{k,n}$ *to the sequence* $\overline{\text{Ford}}_{k+1,n}$.

3.1 Proof of Theorem 1

The $\overline{\text{Ford}}$ sequence of order $n+1$ for alphabet size k, $\overline{\text{Ford}}_{k,n+1}$, can be obtained by the greedy algorithm $\mathbf{P}$ presented in the previous section, which starts at the lexicographically least vertex in $G_{k,n}$, 0^n, and always takes the edge with the greatest available label. We first argue that this procedure computes a de Bruijn sequence. Then it is immediate to see that that the resulting sequence is the lexicographically greatest de Bruijn sequence.

An in-arborescence, henceforth termed A, pointed at the root 0^n can be constructed as follows. Select, for every vertex, the last edge taken by $\mathbf{P}$ (namely, the lexicographically least, 0). Therefore the selected edges are of the form

$$au \to u0$$

for $a \in \Sigma, u \in \Sigma^{n-1}$.

Thus, there are $k^n - 1$ selected edges because, except for 0^n, for each of the k^n vertices a single edge is selected. Also note that there is a path from every vertex $a_1 a_2 ... a_n$ (with $a_i \in \Sigma$) to the root, namely the edges

$$a_1 a_2 \dots a_n \to a_2 \dots a_n 0$$

$$a_2 \dots a_n 0 \to a_3 \dots a_n 00$$

and so on.

From these two observations follows that the construction is indeed an in-arborescence, as depicted in Figure 3.1

The special edge first leaving 0^n is chosen to have the lexicographically greatest label, namely

$$0^n \to 0^{n-1}(k-1)$$

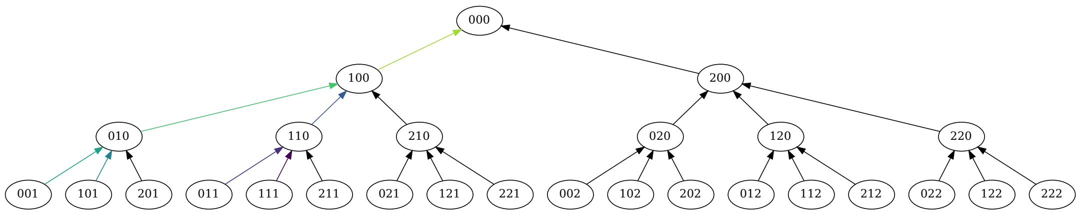

Fig. 3.1: In-arborescence A for the graph $G_{3,3}$.

and this is exactly the first edge selected by $\mathbf{P}$.

As stated in the corollary of the BEST theorem, the in-arborescence A can be mapped to a set of Eulerian cycles on $G_{k,n}$. By fixing the order of the edges that are not in the arborescence to be the descending lexicographic one (and thus the one followed by algorithm $\mathbf{P}$), the in-arborescence A can be mapped to a single Eulerian cycle. Since at every point in the construction the edge taken is the greatest available, this Eulerian cycle is then the lexicographically greatest, namely the one given by the $\overline{\mathrm{Ford}}_{k,n+1}$ sequence.

Now we show that the the sequence $\overline{\mathrm{Ford}}_{k-1,n+1}$ is a suffix of $\overline{\mathrm{Ford}}_{k,n+1}$. Let C be the cycle in $G_{k,n}$ given by $\overline{\mathrm{Ford}}_{k-1,n+1}$ and let $E(C)$ be the set of edges of C. Consider now the subgraph of $G_{k,n}$

$$R := \Big(V(G_{k,n}), E(G_{k,n}) \setminus E(C)\Big)$$

which is the graph with the vertices of $G_{k,n}$ and the edges $G_{k,n}$ that are not in the cycle given by $\overline{\mathrm{Ford}}_{k-1,n+1}$, as depicted in Figure 3.1. Every vertex v in C has a single incoming and outgoing edge in R, respectively the edges

$$\begin{aligned} (k-1)v_1 \dots v_{n-1} &\rightarrow v_1 \dots v_n \\ v_1 \dots v_n &\rightarrow v_2 \dots v_n(k-1) \end{aligned}$$

Note that the remaining outgoing edge is the lexicographically greatest amongst all edges leaving v in $G_{k,n}$.

Now an algorithm is given to produce $\overline{\mathrm{Ford}}_{k,n+1}$. Consider the algorithm $\mathbf{P}$ applied to the graph R. This shall henceforth be called the algorithm $\mathbf{P}_R$. We define an in-arborescence named A_R, with the same root 0^n as in A.

Some of the edges in $\overline{\mathrm{Ford}}_{k,n+1}$ are also in $\overline{\mathrm{Ford}}_{k-1,n+1}$, namely those that correspond to edges between vertices having no symbol $k-1$. For every such vertex v, the branch in

A with v as root will be taken and attached in A_R to the erstwhile leaf $v_2 \dots v_n(k-1)$. This means that the new edge in A_R for the vertex is

$$v \to v_2 \dots v_n(k-1)$$

The arborescence invariant is not breached through this procedure since these erstwhile leaves end with $k-1$, therefore the last edge in their path to the root in the arborescence is

$$(k-1)0^{n-1} \to 0^n$$

and so they will not belong to the same branch as any vertex in $\overline{\text{Ford}}_{k-1,n}$.

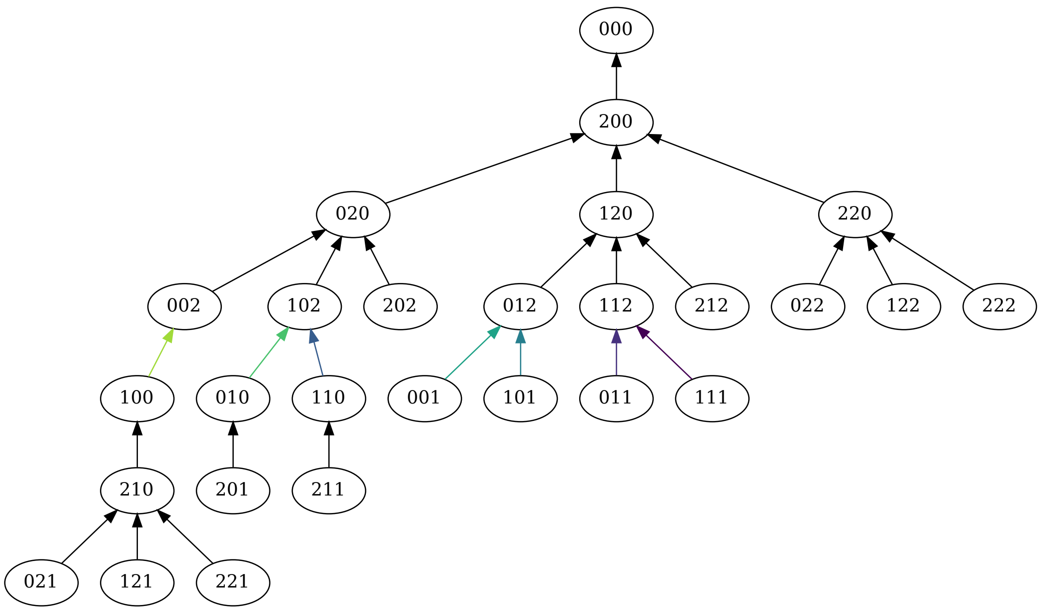

Fig. 3.2: In-arborescence A_R for the graph $G_{3,3}$. The colored edges in Figure 3.1 are replaced here by the edges of the same color.

Note that R is strongly connected since it is regular and its underlying graph is connected [1], as every vertex has a path to the root of A_R. Thus, it is Eulerian.

Since $\mathbf{P}_R$ follows at every step the same choices as $\mathbf{P}$, the only way they may differ before $\mathbf{P}_R$ depletes the edges of R is if it gets stuck doing so, and so $\overline{\text{Ford}}_{k,n+1}$ would be forced to take an edge in $\overline{\text{Ford}}_{z,n+1}$.

This is absurd however because algorithm $\mathbf{P}_R$ follows the in-arborescence A_R and by the corollary of the BEST theorem, this determines an Eulerian cycle in R. Thus it is concluded that $\overline{\mathrm{Ford}}_{k,n+1}$ is comprised of the lexicographically greatest label of all the cycles in R followed by $\overline{\mathrm{Ford}}_{k-1,n+1}$.

3.2 Proof of Corollary

We shall show the algorithm $\mathbf{P}$ to be optimal (linear relative to the size of the output) in time complexity. The de Bruijn graph can be left implicit and computed as needed in constant time. Each run of the loop computes a character of the output sequence; by keeping an array for the k^n vertices in the graph, the greatest edge not yet taken for a given vertex can be known in constant time. Thus, each run of the loop takes constant time, resulting in an optimal time complexity of $O(k^{n+1})$, the size of the output sequence.

By Theorem 1 the $\overline{\mathrm{Ford}}$ sequence of order n on an alphabet of size k is a suffix of the $\overline{\mathrm{Ford}}$ sequence of order n on an alphabet of size $k+1$. Thus, the time-optimal algorithm $\mathbf{P}$ can be used to extend the $\overline{\mathrm{Ford}}_{k,n}$ to $\overline{\mathrm{Ford}}_{k+1,n}$.

An asymptotically equivalent alternative is to employ Algorithm $\mathbf{P}_R$ and then append the $\overline{\mathrm{Ford}}_{k,n}$.

BIBLIOGRAPHY

[1] V. Becher and P.A. Heiber. On extending de Bruijn sequences, *Information Processing Letters* 111: 930–932, 2011.

[2] Nicolaas Gover de Bruijn. A combinatorial problem, *Koninklijke Nederlandse Akademie v. Wetenschappen*, 49 (1946), 758-764. (=Indagationes Math. 8 (1946) 461-467).

[3] Jean Berstel, Dominique Perrin. The origins of combinatorics on words. *European journal of Combinatorics*, 28 (2007), 996-1022.

[4] Harold Fredricksen. A survey of full length nonlinear shift register cycle algorithms. *SIAM Review*, 24:2 (1982), 195-221.

[5] Fredricksen, Harold; Maiorana, James (1978), Necklaces of beads in k colors and k-ary de Bruijn sequences, *Discrete Mathematics*, 23 (3): 207–210,

[6] M. Fleury. Deux problèmes de géométrie de situation, *Journal de mathématiques elémentaires*, (1883), 257-261.

[7] Martin, M. H. (1934), A problem in arrangements, *Bulletin of the American Mathematical Society*, 40 (12): 859–864.

[8] Jonathan Gross, Jay Yellen. *Graph Theory and Its Applications*, Second Edition, CRC Press, 2006.

[9] Frank Harary. *Graph Theory*, Addison-Wesley, 1969.

[10] William T. Tutte. *Graph Theory, Reading*, Addison-Wesley, 1984.

[11] Camille Flye Sainte-Marie, Question 48, L'intermédiaire des mathématiciens, 1 (1894) 107-110.