Combined greedy algorithm on construction binary de bruijn sequence

Musthofa1), Agus Maman Abadi2), Karyati3)

Department of Mathematics Education, Universitas Negeri Yogyakarta, Indonesia

1)[musthofa@uny.ac.id](mailto:musthofa@uny.ac.id), 2)[agusmaman@uny.ac.id](mailto:agusmaman@uny.ac.id), 3)[karyati@uny.ac.id](mailto:karyati@uny.ac.id)

**Abstract**. In this paper we construct one type of de bruijn sequence. First, we modify the prefer-one and prefer-zero algorithm to build one cycle. The other cycles are constructed by modified pure circulating register. Finally, we use cycle joining method to get the full length of the sequence.

1. Introduction

De bruijn sequence, especially binary de bruijn sequence is a topic that has many applications in modern technology. A binary de bruijn sequence of order *n* is a binary sequence with property that every *n*-tuple appears exactly once in every one period of the sequence[1]. For any positive integer *n*, there are different sequences [2]. For example, if *n* = 2, then there is exactly one binary de bruijn sequence of order two that is 0011. If *n* = 3, then there are two different sequences, 00010111 and 00011101.

There are some construction methods for this sequence. Golomb [3] proposed Feedback shift register ( FSR) to solve this problem. This method is based on polynomial over finite field. If the characteristic function FSR is linear then it is called linear feedback shift register (LFSR), otherwise is called nonlinear (NFSR). Furthermore, NFSR is more interesting in cryptography point of view. Alhakim [4] proposed the greedy algorithm called prefer opposite to construct de bruijn sequence. Recently, swada, *et*. *al* [5] proposed an algorithm to construct de bruijn efficiently, however not all pattern can be constructed.

Although there are many efforts in constructing de bruijn sequence especially nonlinear types, every method has limitation. For example, the computational aspect and the property of the sequence that should be obtained. Cycle joining method (CJM) is one of the famous algorithms to construct the sequence from many small cycles.

In the process of joining disjoint cycle using CJM, it need the location of conjugate pair or companion pair. Recently, Chang, Ezerman, Ling and Wang [6] have success to find all conjugate pairs in any two cycles from LFSR. Another approach to construct de bruijn sequence is using cross join method. By this method, starting from any de bruijn sequence, theoretically we can generate all pattern of the sequence.

Helleseth and Klove [7] have proved that the number of cross join pairs of any *m*-sequence is (2n-1-1) (2n-1 -2)/6. It has been proved by Mykkeltveit and Szmidt [8] that any two different pattern of the same order de bruijn sequence can be reached it other by implementing several cross join. However, determining the cross join pairs is another problem. Furthermore, coppersmith [9] give a formula to count the pattern of de bruijn sequence resulting from cross join operation starting from LFSR. From this, we consider that constructing NFSR is still interesting in many aspects.

The main focus in the construction methods are based on two criteria, the efficiency of the algorithm and the number of different constructed sequences. In this paper we try to construct different pattern of de bruijn sequence using modified existing algorithm.

1. Binary De Bruijn Sequence

A binary de bruijn sequence of order *n* is a circular string of length 2*n* which every substring of length *n* occurs exactly one. The string consist of 0 and 1. For convenient, we give some simple examples here. First, **01** is the only binary de bruijn sequence of order oneand **0011** is the only binary de bruijn sequence of order two, so there are no other sequences. The substrings of de bruijn sequences of order two are **00**, **01**, **11**, **10**. Furthermore, there are exactly two different binary sequences for order three, that are **00010111** and **00011101** whichevery substring of length 3, *i.e* string from the {000,001, 011, 110, 101, 010, 101,100} appears exactly one**.**

An interesting problem in this topic is how to construct the sequence efficiently. Various approach has been proposed by researchers. However, every algorithms has own advantages and drawbacks. Therefore research on this area is still interesting. Below, we give some methods in constructing de bruijn sequence.

## Prefer One Algorithm

This method has been discussed in [10]. Start with n-zeros, then append 1 as the new bit as long as the new state is not appeared before, otherwise append 0. Here, we list the result of this algorithm for *n* = 2 to *n* = 6.

**Table 1**. Binary De Bruijn Sequence with prefer one algorithm

|  |  |
| --- | --- |
| **N** | **Sequence** |
| 1 | 01 |
| 2 | 0011 |
| 3 | 00011101 |
| 4 | 0000111101100101 |
| 5 | 00000111110111001101011000101001 |
| 6 | 1111110111100111010111000110110100110010110000101010001001000000 |

## Prefer- zero algorithm

This algorithm is similar with the prefer one. Wang [11], summarized this method as follows. Start with string 1n-1 then greedily append a 0 so that the substrings of length n in the resulting sequence are distinct, otherwise append a 1.

The prefer zero algorithm always stop when the length of the sequence reaches 2n + n – 1. However, it is not de bruijn sequence. After removing the initial 1n-1 string we will get de bruijn sequence. The table below presents the result of de bruijn sequence from prefer zero algorithm.

**Tabel 2**. De bruijn sequence result from prefer zero algorithm

|  |  |
| --- | --- |
| **N** | **Sequence** |
| 1 | 01 |
| 2 | 0011 |
| 3 | 00010111 |
| 4 | 0000100110101111 |
| 5 | 00000100011001010011101011011111 |
| 6 | 0000001000011000101000111001001011001101001111010101110110111111 |

## Cycle Joining Method

Suppose S = (s1,s2, s3, …, sn) is substring of de bruijn sequence. The conjugate of S is = s2,s3,….,sn) and the companion of S is = (s1,s2, s3, …, ). Let S belong to cycle C1 and belong to C2. Interchanging the predecessors of S and will join C1 and C2 into one cycle. Moreover, interchanging the successors of S and is also combining C1 and C2 into one cycle.

Let C1 be a cycle 001= 001 → 010 → 100 → 001 and C2 = 011 = 011 → 110 → 101 → 011. We can join these two cycles since 010 and 011 is companion pair. After interchange the predecessor, we get 110 → 101 → 010 → 100 → 001 → 011→ 110 = 110100. On the other hand, 010 and 110 is conjugate pair, therefore interchange the successor will also join the cycles. Thus the result is 001→ 010 → 101 → 011 → 110 → 100 → 001 = 001011.

This is the basic concept of the cycle joining method proposed by Golomb [3]. However, in real implementation of this method, the position of companion pairs or conjugate pairs shared by the cycles need to be found.

1. Pure Circulating Register

Consider the cyclic sequence (10010) → (00101) → (01010) → (10100) → (01001) → (10010). It can be derived from the rule Si+1 = ( si+1,si+2,  …, si+n-1, si). It has been discussed in [3],[12], pure cycling register or pure circulating register is a type of feedback shift register with special feedback function f(s0, s1, …, sn-1) = s0.

By this method, de bruijn cycle can be obtain by combining the all cycles from pure circulating register. For example, we want to construct de bruijn sequence of order 4. There are five cycles: 0000, 0001, 0101, 0011, 0111, 1111 that can be seen in the figure below.



**Figure 1**. pure circulating register of n= 3

Finally, by Cycle joining method, we will get de bruijn sequence as presented in the figure below.



**Figure 2**. CJM of pure cycling register

1. Result and Discussion

We construct one cycle using prefer one and prefer zero algorithm as follows. Suppose we want to construct binary d bruijn sequence of order *n*.

1. Apply the prefer one algorithm until we get .
2. The next state is
3. Apply prefer one algorithm until we get
4. Apply prefer zero algorithm until we get

**Example 1.** Suppose *n*= 5.

1. Applying the prefer one algorithm, we get the sequence: 00000, 00001, 00011, 00111, 01111, 11111.
2. The next state is 11110.
3. Applying the prefer one algorithm, we get the sequence: 11101, 11011, 10111
4. Applying the prefer zero algorithm, we get the sequence: 01110, 11100, 11000, 10000

Now, we get one cycle : 00000, 00001, 00011, 00111, 01111, 11111, 11110, 11101, 11011, 10111, 01110, 11100, 11000, 10000.

From the above result, it can be seen that the type of state appeared on the cycle are:

1. and
2. The pure circulating register of

Furthermore, the other cycle can be done by applying modified pure circulating register as follows.

1. If the state not appear in the 4 type above, then do pure circulating register.
2. If we do pure circulating register and one or more the state appear in the 4 types above, then take the companion state for the state.

**Example 2.** Take n = 5.

By implementing the above rule , we get all other cycles as follows:



**Figure 3**. Modified Pure Cycling Register

Finally, applying the cycle joining method, we get the full cycle (de bruijn sequence).

**Example 3**. Below we give an example of the process of CJM of above cycle.



Figure 4. CJM for Modified Method

1. Conclusion

In this paper, we success to construct binary de bruijn sequence by applying combined greedy algorithm, especially prefer one and prefer zero. The process also involved modified pure circulating register and cycle joining method. Although we do not analyze the efficient aspect of this method, we hope this result useful in term of developing research in this subject.

References

1. Z. Chang, M.F Ezerman, S. Ling, and H. Wang. Construction of de Bruijn sequences from product of two irreducible polynomials. *Cryptography and Communications*, pages 1–25, 2017.
2. N. G. de Bruijn. A combinatorial problem. *Proc. Kon. Ned. Akad. Wetensh.*, vol. 49, no. 7, pp. 758–764, Jun. 1946N. G. de Bruijn. A combinatorial problem. *Proc. Kon. Ned. Akad. Wetensh.*, vol. 49, no. 7, pp. 758–764, Jun. 1946.
3. S. W. Golomb, Shift Register Sequences, San Fransisco, CA : Holden- Day Inc, 1967.
4. A.Alhakim, A simple combinatorial algorithm for de Bruijn sequences, Amer. Math. Monthly 117 (8) (2010) 728–732.
5. J. Sawada, A. Williams, and D. Wong. A surprisingly simple de Bruijn sequence construction. Discrete Math., 339:127–131, 2016.
6. Z. Chang, M. F. Ezerman, S. Ling, H. Wang, On binary de Bruijn sequences from LFSRs with arbitrary characteristic polynomials, Designs, Codes and Cryptography 87 (5) (2019) 1137–1160.
7. T. Helleseth, T. Kløve. The number of cross-join pairs in maximum length linear sequences.

IEEE Trans. on Inform. Theory, 31(1991), pp. 1731-1733.

1. J. Mykkeltveit and J. Szmidt.On cross joining de Bruijn sequences, Contemporary Mathematics,vol. 632, pp. 333-344 (2015).
2. D. Coppersmith, R. C. Rhoades, J. M. Vanderkam. Counting de Bruijn sequences as perturbations of linear recursions *arXiv e-prints*, May 2017.
3. H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms. SIAM review,vol. 24, no. 2, pp. 195–221, 1982.
4. X. Wang, D. Wong, W. G. Zhang, A simple greedy de Bruijn sequence construction, presented at Sequences and Their Applications (SETA), Hong Kong, Oct. 1–6 (2018).
5. J. Mykkeltveit, Generating and counting the double adjacencies in a pure cycling shift register, IEEE Trans.Comput., C-24 (1975), pp. 299-304.