A Binary de Bruijn Sequence Generator from Product of Irreducible Polynomials

Martianus Frederic Ezerman and Adamas Aqsa Fahreza

Abstract

This is a basic software implementation that generate binary de Bruijn sequences from products of irreducible polynomials.

I. Introduction

Technical Reference

This is a basic implementation of the procedures proposed in the preprint

Z. Chang, M. F. Ezerman, S. Ling, and H. Wang, "On binary de Bruijn sequences from LFSRs with arbitrary characteristic polynomials," Available at http://arxiv.org/pdf/1611.10088.

We retain the definitions and notations in the paper and recommend, upon usage, that users cite it and provide a link to the source code.

License and Proper Attribution

II. Ingredients and Procedures

One needs a working `Python 2.7` since `sympy` is required. An auxiliary file `mathhelper.py` defines some functions and classes that support the main file `debruijn.py`. The latter contains routines that implement Algorithms 1 to 4, producing the complete adjacency graph G, as well as the main procedure to generate the sequences detailed in Section 6 of the reference.

Users can choose from several options to output.

1) Switch the verbose mode on `-v`.
2) Generate the sequences randomly, *i.e.*, randomizing the choice of tree and an initial n-string state to use in the generation of the actual de Bruijn sequences `-r`.
3) Halt the run after all basic information regarding the adjacency graph is known or computed. No actual de Bruijn sequence is generated `-n`.
4) Write the output to a file `-o`.
5) Specify the number of sequences to output `-t`. The default is $\min\{1024, \zeta_G\}$.
6) Generate a single de Bruijn sequence chosen using Broder's Algorithm `-R`.

For example, running the command

```
python debruijn.py -v -r -t 100 -o outputfile.txt 11 111 1101
```

turns on the verbose mode and writes 100 (out of 5280 constructible) de Bruijn sequences of order 6 based on some random choice of trees in $\widehat{G}$ and random 6-string initial states to a file named outputfile.txt with $f(x) = (x+1)(x^2+x+1)(x^3+x^2+1)$ as the characteristic polynomial of the LFSR.

The files containing the complete source code can be requested from the authors or downloaded directly from Github at https://github.com/adamasstokhorst/debruijn.

Research Grants TL-9014101684-01 support the authors' work.

M. F. Ezerman and A. A. Fahreza are with the School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371, e-mails: {fredezerman,adamas}@ntu.edu.sg.