Mapping the Discrete Logarithm

Daniel R. Cloutier
Cincinnati, OH, USA
Daniel.R.Cloutier@
alumni.rose-hulman.edu

Joshua Holden
Rose-Hulman Institute of Technology
Terre Haute, IN 47803, USA
Joshua.Holden@rose-hulman.edu

1 Graphs Produced by Discrete Exponentiation: A Comparison to Random Graphs

We investigate the functional graph produced by the discrete exponentiation transformation

$$g^a \mod p.$$

This is the inverse of the discrete logarithm, which is used in many cryptographic algorithms. We predict that these graphs behave like random graphs with the same in-degrees and out-degrees for each node.

2 Terminology and Background

A *functional graph* is a directed graph such that each vertex must have exactly one edge directed out from it. An m*-ary functional graph* is a graph where each node has in-degree of exactly zero or m.

There are a number of statistics of interest derived from functional graphs. These include:

- number of connected components
- number of cyclic nodes
- number of terminal nodes
- average cycle length
- maximum cycle length
- average tail length
- maximum tail length

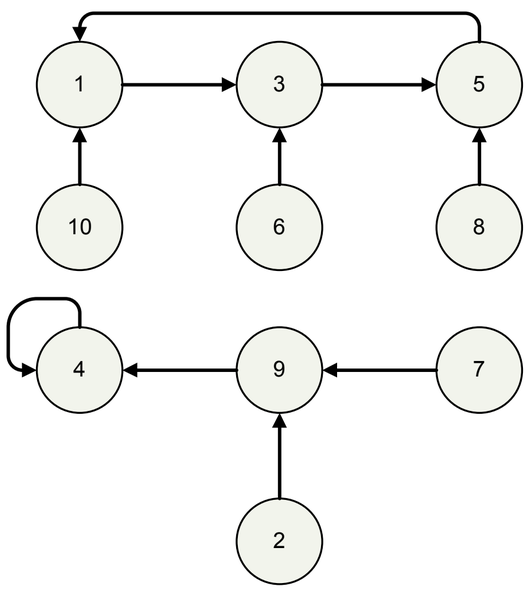

Figure 1: The graph generated using $f(x) = 3^x \mod 11$. This graph has two connected components: one containing a cycle of length three (1,3,5) and the other containing a cycle of length one (fixed point) at 4.

3 Theoretical Predictions for Random Binary Functional Graphs: The Basis for Comparison

The methods in [2] can be extended to develop estimates for these parameters in random binary functional graphs. Define the following:

```
BinFunGraph  = set(Components)
Component    = cycle(Node*BinaryTree)
BinaryTree   = Node + Node*set(BinaryTree, cardinality = 2)
Node         = Atomic Unit
```

Imitating [2, Section 2.1], the generating functions of interest are

$$f(z) = e^{c(z)} \qquad \text{(Binary functional graphs)} \tag{1}$$

$$c(z) = \ln \frac{1}{1 - zb(z)} \qquad \text{(Connected components)} \tag{2}$$

$$b(z) = z + \frac{1}{2} z b^2(z) \qquad \text{(Binary trees)} \tag{3}$$

A bivariate generating function, $\xi(u, z)$, is defined with parameter u marking the elements of interest. The mean value generating function, $\Xi(z)$, is found as

$$\Xi(z) = \left. \frac{\partial \xi}{\partial u} \right|_{u=1}.$$

This yields the following results

$$\Xi_1(z) = \frac{1}{1 - zb(z)} \ln \left(\frac{1}{1 - zb(z)} \right) \qquad \text{(Number of components)} \tag{4}$$

$$\Xi_2(z) = \frac{zb(z)}{(1 - zb(z))^2} \qquad \text{(Number of cyclic nodes)} \tag{5}$$

$$\Xi_3(z) = \frac{z^2}{(1 - 2z^2)^{3/2}}. \qquad \text{(Number of terminal nodes)} \tag{6}$$

We compute an asymptotic form for each of these by performing a singularity analysis[1] as in [2, Section 2]. (See [1] for full proofs.)

Theorem 1. *The expected values of the following parameters in a random binary functional graph of size n, as $n \to \infty$, are asymptotic to*

Number of components	$\frac{\ln(2n) + \gamma}{2}$	(i)
Number of cyclic nodes	$\sqrt{\pi n/2} - 1$	(ii)
Number of terminal nodes	$n/2$	(iii)

In part (i), γ represents the Euler constant which is approximately 0.57721566. Note that the formula in part (iii) can be proved to be exact.

Using similar techniques, we can calculate:

Theorem 2. *The expected values for the following parameters as seen from a random node in a random binary functional graph of size n, as $n \to \infty$, are asymptotic to*

Average cycle length	$\sqrt{\pi n/8}$	(i)
Average tail length	$\sqrt{\pi n/8}$	(ii)

Theorem 3. *The expected values of the largest cycle and the largest tail in a random binary functional graph of size n, as $n \to \infty$, are asymptotic to*

Largest cycle	$\sqrt{\frac{\pi n}{2}} \int_0^\infty \left[1 - \exp\left(- \int_v^\infty e^{-u} \frac{du}{u} \right) \right] dv \approx 0.78248\sqrt{n}$	(i)
Largest tail	$\sqrt{2\pi n} \ln 2 - 3 + 2 \ln 2 \approx 1.73746\sqrt{n} - 1.61371$	(ii)

4 Success in Predicting the Observed Results

We generated experimental data for the parameters described by these theoretical predictions. The generation and analysis of each of the discrete exponentiation graphs was handled by C++ code written by the first author. The computation took approximately one week for each prime. The statistics observed (Table 1) seem to support the claim that binary functional graphs induced by exponentiation modulo a prime behave in the same fashion as random binary functional graphs.

	100043		**100057**		**106261**	
	Observed	**Error**	**Observed**	**Error**	**Observed**	**Error**
Components	6.389	0.047%	6.364	0.437%	6.370	0.810%
Cyclic Nodes	395.303	0.029%	395.858	0.105%	408.433	0.217%
Terminal Nodes	50021	0%	50028	0%	53130	0%
Avg Cycle	198.319	0.056%	197.766	0.230%	202.651	0.795%
Avg Tail	197.961	0.125%	197.550	0.339%	202.422	0.907%
Max Cycle	247.261	0.094%	247.302	0.082%	256.986	0.754%
Max Tail	541.827	1.115%	549.588	1.145%	566.370	1.744%

Table 1: The observed results for the three primes over all binary functional graphs generated and the corresponding percent errors.

References

[1] Daniel R. Cloutier and Joshua Holden. Mapping the discrete logarithm. http://xxx.lanl.gov/abs/math.NT/0605024.

[2] Philippe Flajolet and Andrew M. Odlyzko. Random mapping statistics. In *Advances in cryptology—EUROCRYPT '89 (Houthalen, 1989)*, volume 434 of *Lecture Notes in Comput. Sci.*, pages 329–354. Springer, Berlin, 1990.

[1]The analyses in this paper have been performed using the computer algebra program Maple and the packages created as part of the Algorithms Project at INRIA, Rocquencourt, France. The packages can be found online at http://pauillac.inria.fr/algo/libraries/software.html.