Exponential Multiplication Schemes

Bernd Borchert
Klaus Reinhardt

WSI-2006-10

Universität Tübingen
Wilhelm-Schickard-Institut für Informatik
Arbeitsbereich Theoretische Informatik/Formale Sprachen
Sand 13
D-72076 Tübingen

borchert/reinhard@informatik.uni-tuebingen.de

ISSN 0946-3852

Exponential Multiplication Schemes

Bernd Borchert Klaus Reinhardt

Universität Tübingen, Sand 13, 72076 Tübingen, Germany

{borchert,reinhard}@informatik.uni-tuebingen.de

Abstract

We present an idea to describe a polynomial with 2^n distinct integer zeros by an n-tuple of integers via a scheme of n recurring equations. We call such an n-tuple an *exponential multiplication scheme* of size n. Exponential multiplication schemes of size $1, 2, 3$, and 4 are presented. Under the assumption that fast exponential multiplication scheme generators exist we suggest a fast randomized heuristic for the factorization problem.

1 Exponential Multiplication Schemes

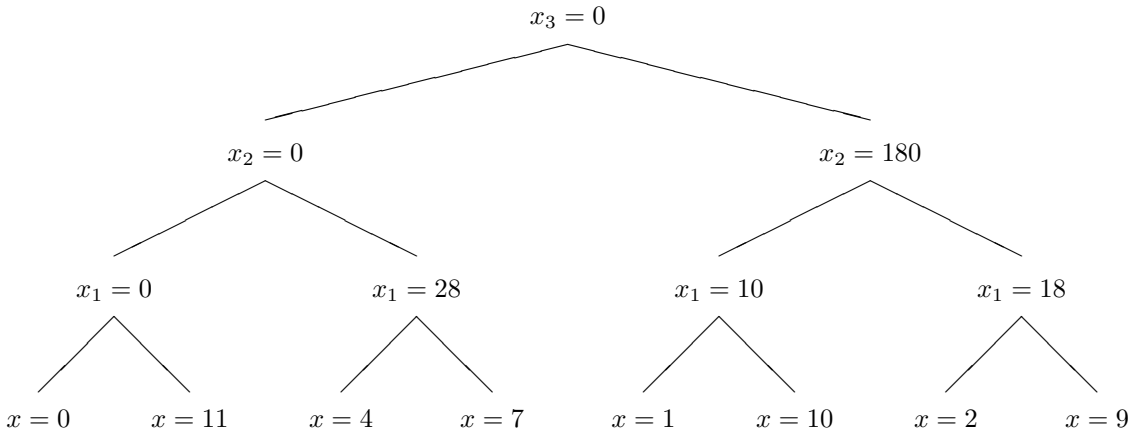
Consider the following value x_3, defined on x via two intermediate values x_1 and x_2:

$$\begin{aligned} x_1 &:= x(x-11) \\ x_2 &:= x_1(x_1-28) \\ x_3 &:= x_2(x_2-180) \end{aligned}$$

The term x_3, seen as an polynomial in x, is of degree 8 and has 8 different integer zeros: 0, 1, 2, 4, 7, 9, 10, 11. In other words:

$$x_3 = x(x-1)(x-2)(x-4)(x-7)(x-9)(x-10)(x-11).$$

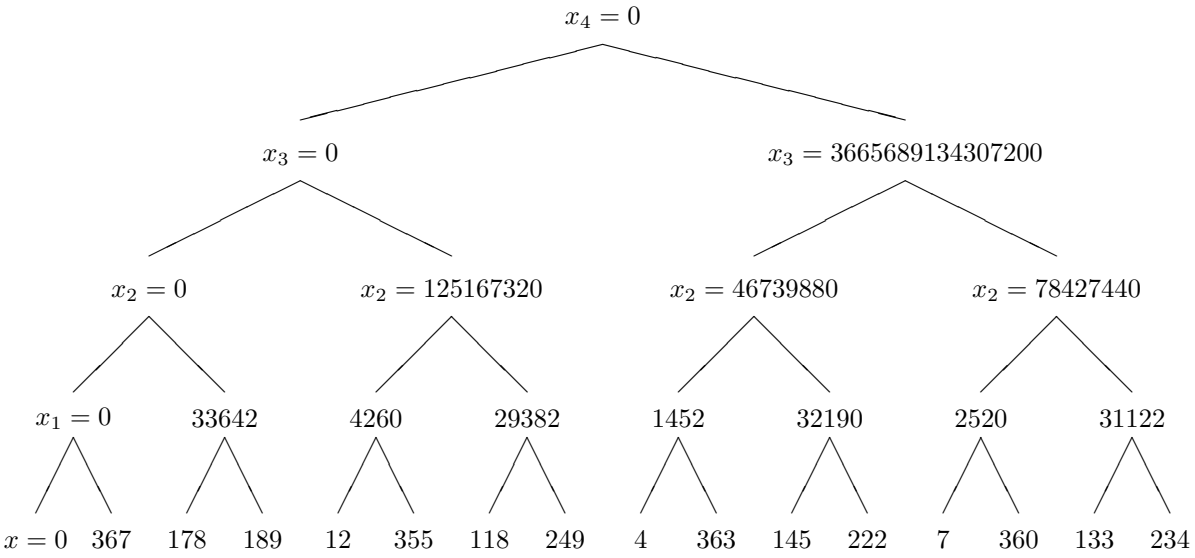
This can be checked by comparing the two expansions of the two sides of the equation. More efficiently, this can be checked the following way: $x_3 = 0$ implies by the last equation either case (A) $x_2 = 0$ or case (B) $x_2 = 180$. The case (A) together with the second equation leads to the case distinction case (A.A) $x_1 = 0$ or case (A.B) $x_1 = 28$. Case (A.A) leads via the first equation to the first two zeros: $x = 0$ and $x = 11$, and case (A.B) leads together with the first equation to the quadratic equation $28 = x(x-11)$ which is solved by $x = 11/2 \pm \sqrt{(11^2 - 4 * 28)/4}$, which gives the next two zeros: $x = 4$ and $x = 7$. Case (B) together with the second equation gives the quadratic equation $180 = x_1(x_1 - 28)$ which has the solutions $x_1 = 10$ (call this case (B.A)) and $x_1 = 18$ (case (B.B)). Together with the first equation case (B.A) gives the quadratic equation $10 = x(x-11)$ solved by $x = 1$ and $x = 10$, and case (B.B) gives the quadratic equation $18 = x(x-11)$ solved by $x = 2$ and $x = 9$. This way, we have found all 8 zeros of x_3. The following picture shows the case distinction tree.



The interesting point about x_3 is that eight integers – namely the zeros of x_3 – are determined by the three integers 11, 28, 180 from the recurring definition above. We will call the triple $(11, 28, 180)$ an *exponential multiplication scheme* of size 3, see the formal definition later. This "compression of information" can be done one step higher, i.e. we will present a exponential multiplication scheme of size 4 as follows. The four constants 367, 33642, 125167320, 3665689134307200 determine the following polynomial x_4:

$$\begin{aligned} x_1 &:= x(x - 367) \\ x_2 &:= x_1(x_1 - 33642) \\ x_3 &:= x_2(x_2 - 125167320) \\ x_4 &:= x_3(x_3 - 3665689134307200) \end{aligned}$$

The polynomial x_4 is of degree 16 and has 16 different integer zeros: 0, 4, 7, 12, 118, 133, 145, 178, 189, 222, 234, 249, 355, 360, 363, 367. These can again be verified by a case distinction process like above. As an example, we follow one path of the case distinction tree: $x_4 = 0$ implies case (A) $x_3 = 0$ or case (B) $x_3 = 3665689134307200$. Case (B) together with the third equation gives the quadratic equation $3665689134307200 = x_2(x_2 - 125167320)$. It has the solutions $x_2 = 62583660 \pm \sqrt{(125167320^2 - 4 * 3665689134307200)/4}$ which are $x_2 = 46739880$ (case (B.A)) and $x_2 = 78427440$ (case (B.B)). Case (B.A) together with the second equation gives the quadratic equation $46739880 = x_1(x_1 - 33642)$. It has the solutions $x_1 = 16821 \pm \sqrt{(33642^2 - 4 * 46739880)/4}$ which are $x_1 = 1452$ (case (B.A.A)) and $x_1 = 32190$ (case (B.A.B)). Case (B.A.B) together with the first equation gives the quadratic equation $32190 = x(x - 367)$. It has the solutions $x = 183.5 \pm \sqrt{(367^2 - 4 * 32190)/4}$ which are $x = 145$ and $x = 222$. These are two of the 16 zeros of x_4, the other 14 are found by continuing the other paths of the case distinction tree, as shown in the following picture.



Call an n-tuple $(c_1, ..., c_n)$ an *exponential multiplication scheme of size* n if the polynomial x_n defined by the recurrence

$$\begin{aligned} x_1 &:= x(x - c_1) \\ x_2 &:= x_1(x_1 - c_2) \\ x_3 &:= x_2(x_2 - c_3) \\ &... \\ x_n &:= x_{n-1}(x_{n-1} - c_n) \end{aligned}$$

has 2^n distinct integer zeros.

We have presented (11,28,180) and (367, 33642, 125167320, 3665689134307200) as examples of exponential multiplication schemes of size 3 and 4, resp. It is easy to verify that the 1-tuple (1) and the 2-tuple (3,2) are exponential multiplication schemes of size 1 and 2, resp. For $n \geq 5$ the authors do not know whether exponential multiplication schemes of size n exist.

The trees for the examples of exponential multiplication schemes confirm the following observation: An exponential multiplication scheme of size n exists if and only if there exists a complete binary tree of depth n labeled with integers such that all leaf labels are distinct, every label of a non-leaf node is the product of the labels of the two sons, the sums of the two labels of two brothers are the same for every level of the tree, and the leftmost path is labeled with 0's. The n integers for the exponential multiplication scheme can be read as the labels of the right sons of the nodes of the leftmost path.

The exponential multiplication scheme $(11, 28, 180)$ was found "manually" by the authors, the exponential multiplication scheme (367, 33642, 125167320, 3665689134307200) was found with the help of a computer (a Java applet running some hours on a PC).

The exponential multiplication scheme (11,28,180) is the lexicographically smallest one of size 3.

The following exponential multiplication schemes of size 3, in lexicographical order, are (13,42,360), (15,50,504), (16,63,720), (17,72,1260), (18,77,1440), (19,78,1080), (19,88,1260), (21,110,1800), and (22,112,2880), the last being a "multiple" of the first: (22,112,2880) = (2∗11,4∗28,16∗180). Computer simulations suggest that for every $n \geq 22$ there is an exponential multiplication scheme starting with $c_1 = n$. The authors do not know a proof of this. It should be mentioned that the multitude seems on average to grow with n: for example, there are 9 exponential multiplication schemes of size 3 with $c_1 = 47$.

The quadruple (367, 33642, 125167320, 3665689134307200) is the lexicographically smallest exponential multiplication scheme of size 4. The next one is (474, 44933, 500669280, 58651026148915200).

2 Factorization

The factorization problem is the following computational problem:

Input: a composite integer $n \geq 2$ in binary representation.

Output: an integer a with $2 < a < n$ which divides n.

For the prime number problem (a "yes/no" problem) there are fast algorithms known, the most efficient of them are randomized. Although the factorization problem seems to be just a little generalization of the prime number problem, no fast algorithm is known for it, see any book on complexity or cryptography, for example [Pa94] or [Sti95].

Call a program which computes a function $0^n \to (c_1, ..., c_n)$ such that $(c_1, ..., c_n)$ is a exponential multiplication schemes an *exponential multiplication scheme generator*. Note that the binary length of the last component c_n will grow exponentially with n. Therefore, an exponential multiplication scheme generator running in polynomial time does not exist, even if exponential multiplication schemes of size $n \geq 5$ do exist. We will call an exponential multiplication scheme generator *fast* if for every $n \geq 2$ the integers $c_1 \bmod n, ..., c_n \bmod n$ are computable in polynomial time in $|n|$.

Under the assumption that a fast exponential multiplication scheme generator exists we suggest the following heuristic for the factorization problem.

int factor(n);

Input: a binary composite number n of length $|n|$.

Do all following computations modulo n.

Consider a fast exponential multiplication scheme $(c_1, \ldots, c_m)$ of size $m = |n|$.

Randomly choose an integer x between 1 and n.

Compute $x_1 := x(x - c_1)$, $x_2 := x_1(x_1 - c_2)$, $x_3 := x_2(x_2 - c_3)$, ...

until the first i is found such that $a :=\gcd(x_i, n) \neq 1$ (or $i = m$).

(use the well-known Euclid algorithm for gcd (greatest common divisor))

If $a > 1$ output a, otherwise output "don't know" (and retry with another x).

We believe that the probability that the algorithm reaches an $a \neq 1$ converges to 1 with growing n since there are at least $\sqrt{n}$ numbers $< n$ which have a common factor with n. With the intuitive

assumption that the factors produced by the exponential multiplication scheme are randomly distributed modulo n we expect that such a factor hits a number with a common factor with n already for $i = m/2$. The algorithm finds a non-trivial factor of n if not all factors of n are hit in the same step (in this case $a = 0$ and we assume that the probability is independent for the next x).

We call this randomized procedure only a heuristic (and not a randomized algorithm) because it is unverified in two ways: not only that it assumes the existence of a fast exponential multiplication scheme generator but moreover we are unable to prove that in fact enough (=exponentially many) zeros remain when considering the polynomial as a polynomial modulo n.

Note that in the case $a = 0$ we would be able to find a factorization for sure if we could also calculate any single integer in the tree and decend until factors come from different subtrees.

3 Open Question and Acknowledgements

The main open question is if there are exponential multiplication schemes of size $n \geq 5$. And if they exists, can they be computed fast modulo a given number?

A way to find a solution may be to understand the size 3 case better. Is every integer $a \geq 22$ the first component $(a, \ , \ )$ of an exponential multiplication scheme of size 3? Once this is understood one may try to attack the size 4 case and higher size cases.

Thanks to Andreas Krebs for some discussions.

References

[Pa94] Chr. Papadimitriou. Computational Complexity. Wiley, 1994.

[Sti95] D. Stinson. Crytography - Theory and Practice. CRC Press, 1995.