International Journal of Mathematical Education in Science and Technology, Vol. 42, No. 6, 15 September 2011, 775–787

Taylor & Francis
Taylor & Francis Group

From recreational mathematics to recreational programming, and back

B.C. Ruiz Jiménez[a]* and M. Ruiz Muñoz[b]

[a]*Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain;* [b]*Departamento de Matemáticas, IES Fuente Lucena, Alhaurín el Grande, Málaga, Spain*

(*Received 20 October 2010*)

Recreational Programming (RecPro) is the discipline that encourages the study of computer programming through ludic problems. Problems that are typically studied within this discipline are similar to those of Recreational Mathematics (RecMat), which sometimes leads to the confusion of these two disciplines. The objective for RecPro is to write programs, while RecMat practitioners can use these programs to state (and prove if possible) conjectures about the solution. This interaction leads to a mathematical quality production. In an educational framework, problems in elemental number theory (those that are formulated with a basic knowledge of arithmetic) are very interesting, leading to the revision of classical unsolved problems. One of these problems is the general form of Zumkeller numbers (those natural numbers as such that their positive divisors can be divided into two disjoint sets with an equal sum). Writing programs by using a programming language that is close to mathematical notation (e.g. Haskell) is the first step to solving the problem, since it is possible to easily write simple and elegant programs so close to the description of the problem that proving their correctness is straightforward.

Keywords: recreational programming; recreational mathematics; functional languages; Haskell

1. Interesting problems to analyse the interaction between RecPro and RecMat

Within the wide collection of problems that share RecPro and RecMat[1] disciplines, we are particularly interested in those that, in addition to motivating the study of these disciplines, lead to a reformulation of unsolved problems. Richard Guy, in his famous *Unsolved Problems in Number Theory* [1] says: 'To pose good unsolved problems is a difficult art. The balance between triviality and hopeless unsolvability is delicate'.

A *smart* programmer may suspect that some *mathematical games* can lead to classic unsolved problems in Number Theory (NT).

Why does NT play a major role? We give two reasons. The first is that it is scientifically sound, and hence success is ensured, for those publications on algorithms or any other branch of science that include a significant amount of

*Corresponding author. Email: blas@lcc.uma.es

ISSN 0020–739X print/ISSN 1464–5211 online

DOI: 10.1080/0020739X.2011.592612
http://www.informaworld.com

work on NT in their references. These include for example [2], in which 40 of its 335 references are on NT, and many more on RecMat! If we dare to conduct a similar analysis on the beautiful texts [3,4] we would be even more surprised.

The second reason is provided by the history of mathematics: every great mathematician has made a significant contribution to NT. Leonard Dickson[2] always said that 'mathematics is the queen of the sciences, and the theory of numbers is the queen of mathematics' [5, p. 333].

Let us analyse the interaction between RecPro and RecMat, which we denote by RecPro $\leftrightarrows$ RecMat. The methods of these two disciplines are, in a sense, complementary; and there is an interaction between them that enriches both. Hence, the symbol $\leftrightarrows$ denotes a bidirectional feedback of results from one discipline to the other. As a subdiscipline of programming, the objective of RecPro is to write good programs, i.e. correct and efficient programs as far as the state of the art allows. The effort in writing good programs could lead to interesting mathematical characterizations of the solutions.

Following guidelines from RecMat, results obtained by writing and running these programs, either (1) lead to conjectures on the solution of the problem, or (2) lead to a direct proof of an essential property of the solution. This is where interaction of *true* mathematics and *experimental mathematics* emerges. Let us recall the famous words of Kolmogorov's disciple, Vladimir I. Arnold (1937–2010) [6]: 'Mathematics is part of physics. Physics is an experimental science, part of natural sciences. Mathematics is the part of physics leading to cheaper experiments.' The interaction between (1) the quality and correctness of functional programs written using modern programming languages, and (2) the theorems of mathematics, provides *cheap experimental research on mathematics*. There are many articles in the literature illustrating this interaction; an excellent one is [7]. For an example, the reader can consult the amazing statement of exercise 1.30, on page 56 of this text [7].

In this article we will see examples of this interaction resulting from the analysis of a curious and original problem:

> The President of the Republic of Zumkia during his long life has gained a *beautiful* collection consisting of a copy of each of the notes in the currency of his country. This collection amounts to a non-negligible total amount of 1,249,920 zumkios. The zumkio, Zumkia official currency, is only available as one zumkio note and as notes of multiples of 3, 5 or 7 zumkios. There exists therefore a note of 15 zumkios, but there is not a 10 zumkios note. How many notes make up the collection? How can he equally divide it between his two children? Do any of the possible distributions have, in addition to the same total value, the same number of notes?

Is this an interesting problem? We will provide an immediate answer by analysing its relation to some *classical* unsolved problems, such as the existence of odd perfect numbers, the cardinality of an even perfect numbers set, or the distribution and density of abundant numbers. Let us start by remembering these concepts and problems.

A number n is *perfect* if it is the sum of its proper factors, ie, its positive divisors, excluding n itself. If we denote with ∂n all factors of n, then $\partial 6 = \{1, 2, 3, 6\}$, and $1 + 2 + 3 = 6$ is perfect. It is also useful to denote with ΣA the sum of the elements from set A. Then n is perfect if and only if $\sigma n = 2n$, where $\sigma n \equiv \Sigma(\partial n)$.

A Zumkeller number is one for which the set of all its factors can be partitioned in two sets with an equal sum.[3] If ζ is the value (in zumkios) of the largest note in the collection of the president of Zumkia, the partition problem will have a solution if ζ is a Zumkeller number.

These numbers generalize perfect numbers. Thus, every perfect number is a Zumkeller one. But the number 12 is not perfect, while $\partial 12 = \{1, 3, 4, 6\} \cup \{2, 12\}$, in such a way that $\Sigma\{1, 3, 4, 6\} = \Sigma\{2, 12\}$, and therefore 12 is a Zumkeller number. There exists many articles on perfect numbers, their generalizations and variants (multiply perfect, abundant, semi perfect, amicable, practical, ...) [1,8]. It is therefore very difficult to provide new ideas, original results, unsolved problems, and it is difficult to know whether an idea, even an unusual one, is really original. Bhaskara and Peng [9] present relations for some of these categories of numbers and summarise the best-known properties of Zumkeller numbers.

Let therefore $\mathcal{P}$ be the set of perfect numbers and let us denote by $\mathcal{Z}$ the set of Zumkeller numbers. We already know that $\mathcal{P} \subset \mathcal{Z}$, and we also know that this is a proper inclusion ($12 \in \mathcal{Z}$, $12 \notin \mathcal{P}$). The first Zumkeller numbers are

6, 12, 20, 24, 28, 30, 40, 42, 48, 54, 56, 60, 66, 70, 78, 80, 84, 88, 90, ...

At present, it is unknown whether the set $\mathcal{P}$ of perfect numbers is finite; on the other hand, it is easy to show that $\mathcal{Z}$ is infinite. For this purpose, let us take any number from $\mathcal{Z}$, for example 12, and a partition of its factors with an equal sum: $\Sigma\{1, 3, 4, 6\} = \Sigma\{2, 12\}$. If we add to each partition each number multiplied by p we get $\Sigma\{1, 3, 4, 6, p, 3p, 4p, 6p\} = \Sigma\{2, 12, 2p, 12p\}$. If p is a prime number > 3, the partition above includes all of the factors for $12p$, and hence $12p \in \mathcal{Z}$. It is enough to apply an infinite primes set to deduce that $\mathcal{Z}$ is also infinite. As we will see, it is reasonable to conjecture that the density of $\mathcal{Z}$ is $\simeq 0.229$.

Let us go back to the distribution problem. Each note of m zumkios can be expressed as $3^i 5^j 7^k$; we should therefore find the largest note in the collection ζ as such, that the sum of its divisors is 1, 249, 920: $\sigma(\zeta) = 1{,}249{,}920$. Taking this information into account, we should check whether it is possible to get two partitions with the same sum for these notes.

According to the Fundamental Theorem of Arithmetic [10, p. 3], all natural numbers allow a unique prime factorization $n = p_1^{k_1} \cdots p_j^{k_j} (k_i > 0,\ p_i$ primes, $p_1 < p_2 < \ldots)$. Once this decomposition is known, it is possible to obtain the sum of the factors of n as a product of sums of a geometric progressions like

$$\sigma(p_1^{k_1} \cdots p_j^{k_j}) = (1 + \cdots + p_1^{k_1}) \cdots (1 + \cdots + p_j^{k_j}) \tag{1}$$

since each factor of n appears exactly once in the expansion of the right-hand side of (1).

Let us now calculate i, j, k so that $\sigma(3^i 5^j 7^k) = 1{,}249{,}920 = 2^7 3^2 5^1 7^1 31^1$. By using (1) and some simple arguments $(7 \nmid 7^{k+1} - 1, \ldots)$ we conclude that $\zeta = 3^3 5^5 7$. Furthermore, the number of its factors is $(3+1)(5+1)(1+1) = 48$, which is the number of notes in the collection. The number of possible divisions is $2^{48}/2 = 140$, 737, 488, 355, 328, and the number of divisions with the same number of notes will be $\binom{48}{24}/2 = 16, 123, 801, 841, 550$. In short, addressing this problem by studying all possible partitions is not feasible.

2. Proving inclusion in $\mathcal{Z}$ and one meditation by Gauss

In many cases, we can directly check $n \in \mathcal{Z}$ (without calculating different partitions of its factors) by simply inspecting the pattern for the prime factorization of n; we will look at some of these cases in Section 5.1. Unfortunately, in most cases it is necessary to inspect partitions. We present in Section 6 an algorithm that turns out to be extremely fast if the factorization of n is known. For the description and implementation of the algorithm we have chosen the programming language Haskell [11,12], a modern functional language widely used both academically and professionally.

Therefore, the problem of checking $n \in \mathcal{Z}$ is virtually solved from a computational perspective if we are able to calculate the factorization in a reasonable time frame, and this is the problem! Let us recall the *famous meditation* that Carl Gauss wrote in 1801 in his *Disquisitiones Arithmetic*æ [13, Article 329, p. 396]; see also [3, p. 398] and [14, p. 301]:

> 'The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic [...] Nevertheless we must confess that all methods that have been proposed thus far are either restricted to very special cases or are so laborious and prolix that even for numbers that do not exceed the limits of tables constructed by estimable men [...] Further, the dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated.'

The commonly used primality and factorization tests are based on developments by Lucas and Lehmer [3, pp. 391–398]. Other efficient algorithms to study primality and for factorizing numbers are ρ family of algorithms by John Pollard and the AKS family (Agrawal, Kayal, and Saxena; [14,15]) that are polynomial time bounded, although due to the factor of proportionality, there is no evidence that they are practical from a computational point of view: the above reflection of Gauss remains valid.

3. Analysis of Zumkeller numbers' pattern and distribution

In this section we will study the necessary properties for inclusion in $\mathcal{Z}$, that while very restrictive, turn out to be insufficient.

If $n \in \mathcal{Z}$, then there exists a partition $D \uplus D' = \partial n$ for the set of factors ∂n so that $\Sigma D = \Sigma D'$; hence $2\,(\Sigma D) = \sigma n$ and the sum of factors of n is even. On the other hand, as n is either in D or in D', then $\Sigma D \geq n$, and hence $\sigma n \geq 2n$. In short, if $n \in \mathcal{Z}$, then

$$2 \mid \sigma_n \wedge \sigma_n \geq 2n. \tag{2}$$

These conditions can immediately be checked from (1) and from the prime factors decomposition. Specifically $2 \mid \sigma n$: if the sum of divisors σn is even, some of the sums of progressions of equation (1) must be even. But $1 + \cdots + 2^k$ is always odd, hence some of the remaining sums must be even. Let $s = 1 + \cdots + p^k$ be one of them, being p an odd prime number. As any sum in s is odd, the number of terms must be even, and hence k must be odd. So, neither $2^3 3^2 5^2$, nor $2^i (2k+1)^{2j}$ are Zumkeller numbers.

Let us recall that n is a deficient number if $\sigma n < 2n$, and it is abundant if $\sigma n > 2n$; many results exist with these number properties and their distribution in the literature [1,16,17]. Specifically, no Zumkeller number is deficient.

Of the numbers that meet requirement (2), which are Zumkeller numbers? By using efficient programs to be described in Section 6, we can obtain Table 1. The last column in this table shows the percentage of numbers in $[1 \ldots N]$ that satisfy (2) and are also Zumkeller numbers. For instance, from the first 20 million natural numbers,

Table 1. Frequencies for natural numbers $n \leq N$ satisfying $2 \mid \sigma n \wedge \sigma n \geq 2n$.

N	Ver. $2 \mid \sigma n$	Ver. $\sigma n \geq 2n$	Ver. $2 \mid \sigma n \wedge \sigma n \geq 2n$	Total $\mathcal{Z}$	%
1000	947	249	229	224	97.8
10,000	9830	2492	2420	2294	94.8
100,000	99,461	24,799	24,570	23,051	93.8
1,000,000	998,293	247,549	246,816	229,026	92.8
10,000,000	9,994,602	2,476,741	2,474,422	2,287,889	92.5
20,000,000	19,992,366	4,953,988	4,950,715	4,577,210	92.5

approximately 24.7% (exactly 4,950,715 of them) satisfy (2), and 92.5% of them are also Zumkeller numbers. Thus only a small percentage in those satisfying (2) are not Zumkeller numbers, and from a computational point of view, restriction (2) is essential.

From this table, we also conclude that the condition $2 \mid \sigma n$ is not restrictive (it is verified by 99.6% from the first 20 million natural numbers), and conversely, the condition $\sigma n \geq 2n$ filters almost a fourth of these numbers. Wall [1, p. 46] shows that the density of abundant numbers ($\sigma n > 2n$) is between 0.24750 and 0.24893. The Hungarian mathematician Pál Erdös conjectured that this density is irrational. We conjecture that the density of $\mathcal{Z}$ is approximately 0.229.

It is useful to analyse the condition $\sigma n \geq 2n$ by using the abundance function proposed by Sylvester [18], $hn = (\sigma n)/n$; we will therefore analyse $hn \geq 2$ in the manner of Carmichael, in his study [19,20] on multiplying perfect numbers ($\sigma n = kn$, k natural), and Dickson's [16] study of abundant numbers with a limited number of primes in its representation. From Equation (1) it is easy to conclude that $h(p^k) < \frac{p}{p-1}$, and also that $h(p^k)$ increases with k, but that it decreases with p. So, $h(p^k) < 2$ and no power of a prime number is abundant, nor Zumkeller, nor perfect. Additionally, we find $h(qm) \geq h(q)h(m)$ holds, and hence

- for $k, i \geq 1, h(2^k 3^i) \geq h(2)h(3) = \frac{3}{2} \cdot \frac{4}{3} = 2.$
- for $k \geq 2, i \geq 1, h(2^k 5^i) \geq \frac{7}{4} \cdot \frac{6}{5} > 2,$

So the set of even abundant numbers is vast. For any odd number with two prime factors we have $h(p_1^{k_1} p_2^{k_2}) \leq h(3^{k_1} 5^{k_2}) < \frac{3}{2} \cdot \frac{5}{4} < 2$, where any abundant odd number has at least three different prime numbers: ie, $\omega(n) \geq 3$. In a similar way, we can conclude that if n is odd and abundant and $\omega(n) \leq 6$, then $3 \mid n$. We also obtain a pattern for odd abundant numbers with $\omega(n) \leq 4$:

$3^{k_1} 5^{k_2} p_3^{k_3}$, where $p_3 \in \{7, 11, 13\}$, or $3^{k_1} p_2^{k_2} p_3^{k_3} p_4^{k_4}$, with $p_2 \in \{5, 7\}$.

Most of these numbers are Zumkeller numbers; those, for instance, whose sum of divisors is odd are not excluded, but these are very rare.

4. Unitary combinations of divisors and Zumkeller generators'

The number 12 is a Zumkeller one as factors of 12 can be broken down into two equal sums: $1+3+4+6 = 2+12$, and therefore $1-2+6-12+3+4 = 0$. That means it is possible to obtain the number 0 by combining the factors of 12 with coefficients ± 1. We will call these *unitary combinations*. So a number is a Zumkeller iff 0 can be represented as a unitary combination of its factors. This simple property leads to a very fast analysis. Let us show that the power of a prime p^k is not a Zumkeller number; if it were, some unitary combination of the powers $1, p^2, \dots, p^k$ would be zero; ie it would be possible to choose a sign from $\pm$ in such a way that has $1 \pm p \pm p^2 \pm \cdots \pm p^k = 0$; but this is not possible as the number 1 is not a unitary combination of multiples of p.

Let us now consider the unitary combinations for $p^k m$, with p prime $\nmid m$. In this case, every divisor of $p^k m$ is a product of a divisor of m by a power of p, so that the set $\partial(p^k m)$ can be decomposed into disjoint sets:

$$\partial(p^k m) = D \uplus pD \uplus p^2 D \uplus \cdots \uplus p^k D, \quad \text{where } D = \partial m, \tag{3}$$

where $x \equiv \{x \mid a \in A\}$ and $\uplus$ is the disjoint union. It can be seen that using (3), it is easy to obtain the set of unitary combinations for the divisors of $p^k m$ by properly mixing unitary combinations for divisors of m.

If A is a set of integers, let us denote by $\mathcal{C}A$ the set of its unitary combinations. For disjoint sets, $\mathcal{C}(A \uplus B) = \mathcal{C}A \pm \mathcal{C}B$ holds, where

$$X \pm Y = \{x + y,\ x - y,\ -x + y,\ -x - y \mid x \in X, y \in Y\}.$$

The operator $\pm$ is commutative and associative; additionally, $\mathcal{C}(pA) = p\mathcal{C}A$ holds. By applying these properties to (3) we can finally conclude that

$$\mathcal{C}(p^k m) = M \pm pM \pm p^2 M \pm \cdots \pm p^k M, \quad \text{where } M = \mathcal{C}m, \tag{4}$$

where $\mathcal{C}n$ is a simplification for $\mathcal{C}(\partial n)$.

Let us now show how to apply (4) to *generate* new Zumkeller numbers from known ones. If $0 \in \mathcal{C}m$, by applying (4) we immediately obtain that $0 \in \mathcal{C}(p^k m)$. In short,

$$\text{If } m \in \mathcal{Z}, \text{ with } \gcd(p, m) = 1, \quad \text{then } p^k m \in Z. \tag{5}$$

On the other hand, if in Equation (4) we take $k + s(k+1)$ instead of k, we obtain $\mathcal{C}(p^{k+s(k+1)}m) = \mathcal{C}(p^k m) \pm \cdots \pm p^{s(k+1)}\mathcal{C}(p^k m)$. And, by applying the above reasoning, we arrive at

$$\text{If } p^k m \in \mathcal{Z} \quad \text{with } (p, m) = 1, \quad \text{then } p^{k+s(k+1)} m \in \mathcal{Z}. \tag{6}$$

Number $p^{k+s(k+1)}m$ is called the translated one from $p^k m$ through p; [9] derives properties (5) and (6) with an original method, while we derive it immediately from (4).

Equations (5) and (6) are the only generic method known to us to generate new Zumkeller numbers; we call those that cannot be obtained by this method generators.[4] The first generators are:

6, 12, 20, 28, 40, 48, 56, 70, 80, 88, 90, 104, 112, 126, 176, 192, 198, 208, ...

We can write a simple Haskell program to obtain the sequence of generators of $\mathcal{Z}$ through a method similar to the *sieve of Eratosthenes*: candidates are taken initially as the most approximate ordered sequence of numbers; for example, the one which satisfies $2 \mid \sigma n \wedge \sigma n \geq 2n$. The first Zumkeller number is taken from this sequence, i.e. 6; all its translations and the multiples $6m$ with $\gcd(m, 6) = 1$ are crossed out, i.e. numbers 24, 30, 42, ... The first uncrossed Zumkeller number is the next generator. This one is number 12. Now we also eliminate translations for 12 and the multiples $12m$ with $\gcd(m, 12) = 1$. This process is iterated until the desired sequence is obtained. This process can be written in Haskell in an elemental way thanks to lazy evaluation that simplifies the design and allows us to write the function which computes the infinite list of Zumkeller generators. In just a few minutes, we get 7438 generators for $\mathcal{Z}$ under 1,000,000! From this program, we get the following:

Conjecture 4.1: *The number of Zumkeller generators less than K is about* $0.5 \cdot K^{0.69}$.

4.1. *Iterated quotients*

Equation (4) provides too many combinations and the deleting of some of them simplifies proving $0 \in \mathcal{C}(p^k m)$, that is equivalent to $p^k m \in \mathcal{Z}$. Let us show that we can suppress combinations in the first sum in (4) that are not multiples of p: If $0 \in M \pm p(M \pm pM)$, 0 is represented by summing up three combinations of $M(\equiv \mathcal{C}m)$, $0 = c + pc' + p^2 c''$, and necessarily c must be a multiple of p, hence, the initial test is equivalent to: $0 \in p^{-1}M \pm (M \pm pM)$, being $p^{-1}\,A$ the set of quotients of division by p for the elements in A that are multiples of p; for example: $3^{-1}\{2, 3, -6\} = \{1, -2\}$. Given that the operator $\pm$ is associative we can rewrite our test moving the parentheses to the left: $0 \in (p^{-1}M \pm M) \pm pM$. The same reasoning leads to $0 \in p^{-1}R_1 \pm M$, with $R_1 = p^{-1}M \pm M$. We can also define another group of simplified combinations: $R_2 = p^{-1}R_1 \pm M$, and $0 \in \mathcal{C}(p^3 m)$ is equivalent to $0 \in p^{-1}R_2 \pm M$. In this way, it is easy to prove the following equivalence:

$$0 \in M \pm \cdots \pm p^k M \quad \text{iff } 0 \in R_k \tag{7}$$

where

$$R_0 = M, \quad \text{and for } k > 0,\ R_k = p^{-1}R_{k-1} \pm M. \tag{8}$$

In short, the effort to simplify the computational aspects of the test $0 \in M \pm \cdots \pm p^k M$ yields to another mathematical characterization of the elements of $\mathcal{Z}$, i.e. $p^k m \in \mathcal{Z}$ iff $0 \in R_k$.

5. Examples of RecPro $\leftrightarrows$ RecMat interaction

We devote each part of this section to the study of a group of examples. First, we will introduce a notation to describe unitary combinations. For example, $\mathcal{C}3 = \{-3-1, -3+1, -1+3, 1+3\} = \{-4, -2, 2, 4\}$. As we can see, a unitary set of combinations is always symmetric, i.e. $x \in \mathcal{C}n \Leftrightarrow -x \in \mathcal{C}n$. But a symmetric set can be represented by its non-negative elements via a compact notation: $\langle 0, 1, 2, 4 \rangle \doteq \{-4, -2, -1, 0, 1, 2, 4\}$.

5.1. *Study of $3^i 5^j 7^k$ numbers*

Let us apply the iterated quotients method to the first odd candidate numbers $3^i 5^j 7^k$. Let us start with $3^i 5\ 7$. Taking $p = 3$, $M = \mathcal{C}(5\ 7)$, applying (7) and (8), *and helped by an easy computer program* we find that

$\mathcal{C}(5\ 7) = \langle 22, 24, 32, 34, 36, 38, 46, 48 \rangle$,
$R_1 = \langle 6, \ldots, 64 \rangle \doteq \langle 6, 8, 10, \ldots, 62, 64 \rangle$,
$R_2 = \langle 2, \ldots, 68 \rangle$,
$R_3 = \langle 0, \ldots, 70 \rangle = R_4 = R_5 = \cdots$

and finally $3^i5\ 7 \in \mathcal{Z} \Leftrightarrow i \geq 3$; among these, those corresponding to $i = 3, 4, 5, 6$ are generators. Let us recall that $R_0 \subset R_1$, and by (8), $R_1 \subset R_2 \subset \ldots$. But we need to find the smaller index i such that $0 \in R_i$. This is an easy example of RecPro $\leftrightarrows$ RecMat interaction: Firstly, we have written a correct program to check $0 \in R_i$; the execution of this program for $i = 0, 1, 2$, together with the equivalence $3^i5\ 7 \in \mathcal{Z} \Leftrightarrow 0 \in R_i$ allows us to prove that $3^i5\ 7 \notin \mathcal{Z}$. Next, the same program computationally proves that $R_0 \subset R_1$, in addition to $0 \in R_3$; and therefore we conclude $3^i5\ 7 \in \mathcal{Z} \Leftrightarrow i \geq 3$. We would like to stress the mutual feedback of this methodology: the help provided by a *correct* program is essential in the proof. The same method applied to $M = \mathcal{C}(5^2 7^1)$ leads to

$$R_1 = \langle 20, \ldots, 330 \rangle \subset R_2 = \langle 0, \ldots, 358 \rangle \subset \cdots$$

and hence $3^i 5^2 7^1 \in \mathcal{Z} \Leftrightarrow i \geq 2$.

This suggests a very interesting method with theoretical/practical implications: localizing the smaller i_0 such that $0 \in R_{i_0} \subset R_{i_0+1}$ leads to a proof for $p^i\, m \in Z$, $\forall i \geq i_0$. Similarly, localizing the smaller i_0 such that $0 \notin R_{i_0} \supseteq R_{i_0+1}$, would lead to $p^i m \notin Z$, $\forall i \geq i_0$.

5.2. *Solution to president of Zumkia distribution problem*

Let us see a second example of the RecPro $\leftrightarrows$ RecMat interaction. The biggest note in the collection of the president of Zumkia is $\zeta = 3^3 5^5 7$, a translation of $3^3 5^2 7$, which we showed in the previous section, is a Zumkeller number. Thus ζ is also a Zumkeller number, and the distribution problem has a solution. Is there a distribution with the same number of notes? Since the original $\zeta = 3^3 5^5 7$ has too many divisors, even an efficient Haskell program for finding such a partition is very slow. But for number $3^3 5^2 7$, its execution proves that there is one partition with the same sum and number of notes:

$$\Sigma\{ 1, 3, 5, 7, 9, 15, 25, 27, 35, 45, 63, 3^3 5^2 7\} = \Sigma\{21, 75, 105, 135, 175, 189, 225, 315, 525, 675, 945, 1575\}$$

Hence, we just have to add the products for 5^3, to distribute all the factors of ζ (notes in the collection) in two groups with the same sum and cardinality:

$$\Sigma\{ 1, 3, 5, 7, 9, 15, 25, 27, 35, 45, 63, 3^3 5^2\ 7, 125, 375, 625, 875, 1125, 1875, 3125, 3375, 4375, 5625, 7875, 590625 = \Sigma\{ 21, 75, 105, 135, 175, 189, 225, 315, 525, 675, 945, 1575, 2625, 9375, 13125, 16875, 21875, 23625, 28125, 39375, 65625, 84375, 118125, 196875\}$$

5.3. *A generalization for a result by Euclides/Euler*

It is an elementary exercise to verify that the number $e = 2^k\ (2^{k+1} - 1)$ is perfect if $2^{k+1} - 1$ is prime. Indeed, by Equation (1), the sum of factors of e is $(1 + 2^{k+1} - 1)(1 + \cdots + 2^k) = 2^{k+1}(2^{k+1} - 1) = 2e$, and hence e is perfect. This surprising result was already described by Euclides in Proposition 36 on his IX Book of Elements [21, p. 67], and many centuries later, Euler proved that perfect even numbers are of the form $2^k(2^{k+1} - 1)$, where $2^{k+1} - 1$ a prime number.[5] These special prime numbers are called Mersenne primes [7]. Are there infinitely many Mersenne primes? This is another notorious unsolved problem.

Not every number of the form $2^{k+1} - 1$ is prime: $2^{12} - 1 = 4095$. It is well known that if $a^k - 1$ is prime, then $a = 2$ and k is also prime [10, Th. 18, p. 15].

The existence of odd perfect numbers is currently unknown, and the following sentence by Guy [1, p. 44] still holds: 'The existence or otherwise of odd perfect numbers is one of the more notorious unsolved problems of number theory'. We must contend ourselves with knowing an amazing amount of requirements that should be met by any odd perfect number [1,22].

By using the method of iterated quotients, we will prove a variation of Euclides/Euler result for Zumkeller numbers:

Theorem 5.1: *For any odd prime p,*

$$2^k p^j \text{ is a Zumkeller number iff } j \text{ is odd } \wedge p \leq 2^{k+1} - 1.$$

Proof: We will prove this theorem applying the interaction between RecPro $\leftrightarrows$ RecMat.

Step A: Let $M = \mathcal{C}(2^k)$; we will use the equivalence $2^k p \in \mathcal{Z}$ iff $0 \in p^{-1}M \pm M$, which was suggested by a computational analysis of the efficiency of the test $0 \in p^{-1}M \pm M$.
Step B: By observing unitary combinations $\mathcal{C}(2)$, $\mathcal{C}(2^2), \ldots$ generated by the same Haskell program used in sections above, we can conjecture that these combinations are constituted by odd consecutive natural numbers, i.e. $\mathcal{C}(2^k) = \langle 1, \ldots, 2^{k+1} - 1 \rangle$.
Step C: Once this conjecture has been stated, by applying Equation (4), it is easy to prove it by induction on natural k.
Step D: Now it is easy to carry out a simple analysis using elemental properties of the set $p^{-1}M \pm M$. We consider now two cases: (*a*) If $p \leq 2^{k+1} - 1$ then a number in the sequence of odd numbers $1, \ldots, 2^{k+1} - 1$ provides the unity when it is divided by p. That is, $1 \in p^{-1}M$; but because $1 \in M$, we conclude that $0 \in p^{-1}M \pm M$; hence, $2^k p \in \mathcal{Z}$, as well as its translations $2^k p^j$ (with j is odd), and this proves the sufficient condition; (*b*) If $p > 2^{k+1} - 1$, $p > \sigma(2^k)$, and then $p^{-1}M$ is empty and hence $p^{-1}M \pm M$ is also empty; so $R_j = \varnothing$, $\forall j \geq 1$, and hence $2^k p^j \notin \mathcal{Z}$, $\forall j \geq 1$. □

The last part of the proof for Theorem 5.1 *suggests* a criterion to discard inclusion in set $\mathcal{Z}$:

Theorem 5.2: *Let us consider a prime number $p \nmid m$, with $p > \sigma m$, then*

$$m \notin \mathcal{Z} \Leftrightarrow \forall j : j \geq 1 : p^j m \notin \mathcal{Z}.$$

Proof: Let $M = \mathcal{C}m$. Because $p > \sigma m$, the greater element in M is less than p, and hence

$$p^{-1}M = \begin{cases} \varnothing, & \text{if } 0 \notin M, \\ \{0\}, & \text{if } 0 \in M. \end{cases}$$

If $m \notin \mathcal{Z}$ then $0 \notin M$, hence $p^{-1}M = \varnothing$, and so $R_j =$, $\forall j \geq 1$, and $p^j m \notin \mathcal{Z}$. □

Let us show some example applications for this theorem: (*i*) $\sigma 2 = 3$, so that, $2p^j \notin \mathcal{Z}$, $\forall p$ prime > 3; (*ii*) $\sigma(2^1 5^1) = 18$, hence $2^1 5^1 p^k \notin \mathcal{Z}$, $\forall k$, $\forall p$ prime > 18.

Theorem 5.2 can be used to speed up the computation for the sequence of numbers in $\mathcal{Z}$ by using sieve based methods because if $m \notin \mathcal{Z}$, we can eliminate multiples $p^i m$, for every prime $p > \sigma m$.

6. An *almost* greedy test for the study of Zumkeller numbers

Simple algorithms for locating Zumkeller numbers are those which scrutinize all possible partitions of divisors. If we are not very careful in the design, usually the

```
type Factors = [Integer]
type Sum = Integer

canBePartitioned :: Factors → Sum → Sum → Bool
canBePartitioned []          _ _ = False
canBePartitioned (d : ds) s s'
 | d == s || d == s'             = True
 | otherwise                     = d ≤ s' && canBePartitioned ds s (s' − d) ||
                                   d ≤ s && canBePartitioned ds (s − d) s'

isZumkeller :: Integer → Bool
isZumkeller n = even sigma && sigma ≥ 2 ∗ n &&
                canBePartitioned ods ms ms
  where fs    = factorize n
        sigma = sumOfDivisors fs
        ods   = oddDivisors fs
        ms    = sigma ‘quot‘ 2
```

Figure 1. An *almost* greedy test.

result is an inefficient algorithm, that turns out to be useless even for small numbers. Algorithms based on iterated quotients or the sieve one described above are better alternatives. These algorithms are also inefficient for $n > 100{,}000$. Is there an efficient algorithm for at least a large amount of numbers? We study in this section a curious algorithm that checks in an *almost* direct way whether a number is a Zumkeller one. We will use the programming language Haskell.

Let us consider function *canBePartitioned* in Figure 1; *canBePartitioned ds s s′* returns *True* if the decreasing list of natural numbers *ds* can be partitioned into two lists whose element sums are s and s'. We will thus assume that $s + s'$ coincides with the sum of values in *ds* and that $s, s' \geq 0$.

This program is extremely fast if factorization for the number at hand is already known. The reason is that our program is ‘almost’ greedy. To illustrate this, let us consider the following analysis. Let us first analyse correction for the function *canBePartitioned*. This function successively distributes divisors in a decreasing list among both partitions, for which only the sum of remaining factors is known. Let us observe the functioning of || operator in the last equation of *canBePartitioned*: search will be done in a prioritized way, placing the greatest divisor d, as yet unassigned in the partition corresponding to the second sum. If the factor cannot be assigned to this partition, either because $d > s'$, or because the first election fails, then d is tried, to be assigned to the first partition through test $d \leq s$ *&& canBePartitioned ds* $(s - d)\ s'$. This proves correctness. Let us now shown that the second election is very infrequent. For this purpose, we define a new function *almostCanBePartitioned* whose last equation is

```
almostCanBePartitioned (d : ds) s s'
 | d == s || d == s'   = True
 | d ≤ s'              = almostCanBePartitioned ds s (s' − d)
 | d ≤ s               = almostCanBePartitioned ds (s − d) s'
almostIsZumkeller n    = even sigma && sigma ≥ 2 ∗ n && almostCanBePartitioned ods ms ms
where fs = ... -- see Figure 1
```

Table 2. Execution times (in s) and *failures* for function *almostIsZumkeller*.

N	*isZumkeller*	*almostIsZumkeller*	Number of failures	Total in $\mathcal{Z}$	Failures %
10,000	0.06	0.06	10	2294	0.45
100,000	1.11	1.11	224	23,051	0.97
1,000,000	17.02	16.40	4764	229,026	2.08
10,000,000	359.05	291.03	81,914	2,287,889	3.58
20,000,000	8339.50	746.03	185,256	4,577,210	4.05

This new algorithm is a greedy one and the number of steps for computing *almostCanBePartitioned ds s s′* is proportional to the number of divisors. In addition, if *almostIsZumkeller n* returns *True*, *n* will be a Zumkeller number and this function performs the same computation as the original *isZumkeller* function. Hence, for these cases the algorithm is very efficient. The bad news is that *almostIsZumkeller n* can return *False* for some Zumkeller numbers, and hence, they are not detected by this function. We will call these numbers *failures*. This is not such bad news because these failures are very rare. To analyse them, we write a simple test:

```
Main> [ n | n ∈ [1 .. 10000], isZumkeller n ≠ almostIsZumkeller n]
[1190, 1430, 1575, 2090, 3410, 4408, 4510, 5775, 8228, 9765]
```

So, function *almostIsZumkeller* only misses 10 Zumkeller numbers from the first 10,000 natural numbers: this is less than 0.5% from the total amount of Zumkeller numbers in that interval. For wider intervals, the proportion of failures increases, but not too quickly. Hence, in the last column of Table 2, we show the percentage of Zumkeller numbers not detected by function *almostIsZumkeller* from the total amount of numbers in $\mathcal{Z}$ less than N. We estimate that for small values for $K(\leq 12)$, the proportion of failures less than 10^k is about $0{,}00003125 \cdot 2^K$. Also in Table 2, the second and third columns show time (in s) to compute Zumkeller numbers less than N, for different values for N, using two functions above and 'The Glasgow Haskell Compiler' ghc-6.10.1 [11] on a Sony VAIO laptop equipped with Intel U1400 @ 1.20Ghz CPU and 1Gb of RAM (i.e. a very modest computer). In less than 2 min, we obtain a sorted sequence with the first million Zumkeller numbers!

Small changes in the function *canBePartitioned* can be introduced to return built partitions, or even to only localize partitions with the same cardinality.

7. Conclusions

We have studied a difficult problem from Elementary Theory of Numbers and from a computational point of view: the general pattern and distribution of Zumkeller numbers. We have characterized the computational limits that can be found, some conjectures on their distribution and essential properties of set $\mathcal{Z}$; this has been possible after finding a *semi-greedy* algorithm to determine the test $n \in \mathcal{Z}$.

The way that our method, based on iterated quotients, can be used for studying the structure of $\mathcal{Z}$ and to derive effective sieve based programs must be emphasized.

Nowadays, there is no question that the use of programming languages close to mathematical notation, such as Haskell, are of great help in mathematical education at any level: these languages allow us to motivate and introduce a great deal of mathematical concepts and attitudes. Both the writing of simple programs, close to the statement of the problem, and the execution of these programs, resulting in a fruitful interaction between mathematical and computational thinking. The effort put into learning a programming language such as Haskell, pays off when we assess its value as a tool for mathematical education.

In this article we have shown that by using only rudimentary methods based on elementary arithmetic, and the research resulting from RecPro $\leftrightarrows$ RecMat interaction we achieve a *mathematical productivity* of excellent quality. In other words, elementary experimental mathematics leads to high quality mathematical productivity. Therefore, we think that ideas presented in this article may be useful to develop typical features from elementary number theory and functional programming, both in secondary and university education.

Haskell programs used in this work are available from the authors for interested readers.

Acknowledgements

This research was partially supported by the projects ReSCUE TIN2008-05932 and P07-TIC03131.

Notes

1. The term *Recreational Programming* is usually not used – not established! – Although the related term *Computer Recreations* was popularized by Alexander Dewdney in his famous section in the 80's *Scientific American*. This section replaced the no less celebrated one in the same journal that Martin Gardner wrote over 24 years entitled *Mathematical Games*, a similar term to Recreational Mathematics. Interestingly, the tradition of the famous Journal of Recreational Mathematics is to have a high number of papers on Recreational Programming.
2. Leonard Eugene Dickson (1874–1954) was one of the most prolific mathematicians. At the age of 30, he had already written 100 relevant papers. His scientific output consists of 285 scientific publications [5], 18 of which are books, including his monumental *History of the Theory of Numbers* [23,24].
3. These numbers are called Zumkeller numbers after the mathematician Reinhard Zumkeller, who published some results and conjectures on these numbers in 2003.
4. Dickson [16] introduces the concept of primitive abundant number as those abundant numbers the proper divisors of which are deficient. Any multiple of an abundant number is also abundant; in addition [16] as proof, for any K, the set of odd primitives satisfying $\omega(n) \leq K$ is finite. It is not easy to generalize a primitive concept for Zumkeller numbers similar to the one used in [16].
5. The reader can find compiled in [8] up to six different proofs for this result, including the original one by Euler and the most elegant and simple one by Dickson from 1911.

References

[1] R. Guy, *Unsolved Problems in Intuitive Mathematics*, *Unsolved Problems in Number Theory*, 2nd ed., Vol. 1, Springer-Verlag, New York, 1994.
[2] G. Brassard and P. Bratley, *Fundamentals of Algorithmics*, Prentice Hall, New York, 1996.
[3] D. Knuth, *The Art of Computer Programming*, *Seminumerical Algorithms*, 2nd ed., Vol. 2, Addison Wesley, Reading, MA, 1980.
[4] E.R. Berlekamp, J.H. Conway, and R.K. Guy, *Winning Ways for your Mathematical Plays*, 2nd ed., Wellesley, Massachusetts, 2001–2004.

[5] A.A. Albert, *Leonard Eugene Dickson: A biographical Memoir*, in *Biographical Memoir*, National Academy of Sciences, Washington, 1955, pp. 329–345.

[6] M. de León, *Vladímir I. Arnold, el matemático que amaba los problemas*, EL País, viernes 11 de Junio, Madrid, 2010.

[7] R. Crandrall and C. Pomerance, *Prime Numbers: A Computational Perspective*, 2nd ed., Springer, New York, 2005.

[8] J. Voight, *Perfect Numbers: An Elementary Introduction*, Department of Mathematics, University of California, Berkeley, 1998.

[9] K.P.S. Bhaskara and Y. Peng, *On Zumkeller Numbers*, Cornell University Library, arXiv:0912.0052v1 [math.NT], 2009.

[10] G.H. Hardy and E.M. Wright, *Introduction to the Theory of Numbers*, 4th ed., Oxford University Press, London, 1968.

[11] The Haskell Language website: http://www.haskell.org; the Glasgow Haskell Compiler website: http://www.haskell.org/ghc/

[12] B.C. Ruiz, F. Gutiérrez, P. Guerrero, and J. Gallardo, *Razonando con Haskell*, Thomson, 2004. Available at http://www.lcc.uma.es/RazonandoConHaskell

[13] C.F. Gauss, *Disquisitiones Arithmeticæ* (translated by Arthur A. Clarke), Springer-Verlag, New York, 1985.

[14] C. Pomerance, *Primality testing: Variations on a them of Lucas*, Proceedings of the 13th Meeting of the Fibonacci Association, Congres. Numerat. 201 (2010), pp. 301–312.

[15] M. Agrawal, N. Kayal, and N. Saxena, *PRIMES is in P*, Ann. Math. 160 (2004), pp. 781–793.

[16] L.E. Dickson, *Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors*, Am. J. Math. 35 (1913), pp. 413–422.

[17] L.E. Dickson, *Even abundant numbers*, Am. J. Math. 35 (1914), pp. 423–426.

[18] J.J. Sylvester, *Sur l'imposibilité de l'existence d'un nombre parfait impair qui ne contient pas au moins 5 diviseurs premiers distincts*, C.R. Acad. Sci. París 106 (1888), pp. 522–526.

[19] R.D. Carmichael, *A table of multiply perfect numbers*, Bull. Am. Math. Soc. 13 (1907), pp. 383–386.

[20] R.D. Carmichael, *Even multiply perfect numbers of five different prime factors*, Bull. Am. Math. Soc. 15(1) (1908), pp. 7–8.

[21] S. Hawking, *God Created the Integers*, Penguin, Philadelphia, 2005.

[22] J. Voight, *On the nonexistence of odd perfect numbers*, in *MASS Selecta: Teaching and Learning Advanced Undergraduate Mathematics*, S, Katok, A. Sossinsky, and S. Tabachnikov, eds., American Mathematical Society, 2003, pp. 293–300.

[23] L.E. Dickson, *History of the Theory of Numbers*, Vol. I, Carnegie Institute of Washington, 1919.

[24] L.E. Dickson, *History of the Theory of Numbers*, Vol. II, Carnegie Institute of Washington, 1920.