ANNALES MATHEMATICAE ET INFORMATICAE borító külső oldala

Contents

Methodological papers

ANNALES MATHEMATICAE ET INFORMATICAE 42. (2013)

ANNALES MATHEMATICAE ET INFORMATICAE

TOMUS 42. (2013)

HUNGARIA, EGER

ANNALES MATHEMATICAE ET INFORMATICAE

International journal for mathematics and computer science

Referred by
Zentralblatt für Mathematik
and
Mathematical Reviews

The journal of the Institute of Mathematics and Informatics of Eszterházy Károly College is open for scientific publications in mathematics and computer science, where the field of number theory, group theory, constructive and computer aided geometry as well as theoretical and practical aspects of programming languages receive particular emphasis. Methodological papers are also welcome. Papers submitted to the journal should be written in English. Only new and unpublished material can be accepted.

Authors are kindly asked to write the final form of their manuscript in LaTeX. If you have any problems or questions, please write an e-mail to the managing editor Miklós Hoffmann: `hofi@ektf.hu`

The volumes are available at `http://ami.ektf.hu`

ANNALES MATHEMATICAE ET INFORMATICAE

VOLUME 42. (2013)

INSTITUTE OF MATHEMATICS AND INFORMATICS
ESZTERHÁZY KÁROLY COLLEGE
HUNGARY, EGER

HU ISSN 1787-5021 (Print)
HU ISSN 1787-6117 (Online)

A kiadásért felelős az
Eszterházy Károly Főiskola rektora
Megjelent az EKF Líceum Kiadó gondozásában
Kiadóvezető: Czeglédi László
Műszaki szerkesztő: Tómács Tibor
Megjelent: 2013. december Példányszám: 30

Készítette az
Eszterházy Károly Főiskola nyomdája
Felelős vezető: Kérészy László

Annales Mathematicae et Informaticae
42 (2013) pp. 3–7
http://ami.ektf.hu

On (log-) convexity of power mean

Antal Bege[a], **József Bukor**, **János T. Tóth**[b*]

[a]Dept. of Mathematics and Informatics, Sapientia Hungarian University of Transylvania,Targu Mures, Romania

[b]Dept. of Mathematics and Informatics, J. Selye University, Komárno, Slovakia
bukorj@selyeuni.sk, tothj@selyeuni.sk

Submitted October 20, 2012 — Accepted March 22, 2013

Abstract

The power mean $M_p(a,b)$ of order p of two positive real values a and b is defined by $M_p(a,b) = ((a^p + b^p)/2)^{1/p}$, for $p \neq 0$ and $M_p(a,b) = \sqrt{ab}$, for $p = 0$. In this short note we prove that the power mean $M_p(a,b)$ is convex in p for $p \leq 0$, log-convex for $p \leq 0$ and log-concave for $p \geq 0$.

Keywords: power mean, logarithmic mean

MSC: 26E60, 26D20

1. Introduction

For $p \in \mathbb{R}$, the power mean $M_p(a,b)$ of order p of two positive real numbers, a and b, is defined by

$$M_p(a,b) = \begin{cases} \left(\frac{a^p+b^p}{2}\right)^{\frac{1}{p}}, & p \neq 0 \\ \sqrt{ab}, & p = 0. \end{cases}$$

Within the past years, the power mean has been the subject of intensive research. Many remarkable inequalities for $M_p(a,b)$ and other types of means can be found in the literature.

It is well known that $M_p(a,b)$ is continuous and strictly increasing with respect to $p \in \mathbb{R}$ for fixed $a, b > 0$ with $a \neq b$.

*The second and the third author was supported by VEGA Grant no. 1/1022/12, Slovak Republic.

Note that $M_p(a,b) = aM_p(1, \frac{b}{a})$. Mildorf [3] studied the function

$$f(p,a) = M_p(1,a) = \left(\frac{1+a^p}{2}\right)^{\frac{1}{p}}$$

and proved that for any given real number $a > 0$ the following assertions hold:

(A) for $p \geq 1$ the function $f(p,a)$ is concave in p,

(B) for $p \leq -1$ the function $f(p,a)$ is convex in p.

The aim of this note is to study the log-convexity of the power mean $M_p(a,b)$ in variable p. As a consequence we get several known inequalities and their generalization.

2. Main results

Theorem 2.1. *Let $f(p,a) = M_p(1,a)$. We have*

(i) *for $p \leq 0$ the function $f(p,a)$ is log-convex in p,*

(ii) *for $p \geq 0$ the function $f(p,a)$ is log-concave in p,*

(iii) *for $p \leq 0$ the function $f(p,a)$ is convex in p.*

Proof. Observe that for any real number t there holds

$$f(pt,a)^t = f(p,a^t). \tag{2.1}$$

Let

$$g(p,a) = \ln f(p,a).$$

Taking the logarithm in (2.1) we have

$$tg(pt,a) = g(p,a^t).$$

Calculating partial derivatives of both sides of the above equation we get

$$t^2 g_1'(pt,a) = g_1'(p,a^t)$$

and

$$t^3 g_{11}''(pt,a) = g_{11}''(p,a^t). \tag{2.2}$$

Specially, taking $p = 1$ in (2.2), we have

$$t^3 g_{11}''(t,a) = g_{11}''(1,a^t). \tag{2.3}$$

Taking into account that the function $f(p,a)$ is increasing and concave in p for $p \geq 1$ (see (A)), the function $g(p,a)$ is also increasing and concave in p for $p \geq 1$. For this reason

$$g_{11}''(1,a^t) \leq 0$$

for an arbitrary $a > 0$ and real t. Let us consider the left hand side of (2.3). We have

$$t^3 g''_{11}(t, a) \leq 0$$

which yields to the facts that the function $g(p, a)$ is concave for $p > 0$, therefore the function $f(p, a)$ is log-concave in this case and the function $g(p, a)$ is convex for $p < 0$. Hence the assertions (i), (ii) follow. Clearly, the assertion (iii) follows immediately from (i). □

The following result is a consequence of the assertion (iii) Theorem 2.1.

Corollary 2.2. *Inequality*

$$\alpha M_p(a, b) + (1 - \alpha)M_q(a, b) \geq M_{\alpha p+(1-\alpha)q}(a, b) \tag{2.4}$$

holds for all $a, b > 0$, $\alpha \in [0, 1]$ *and* $p, q \leq 0$.

Let us denote by $G(a, b) = \sqrt{ab}$ and $H(a, b) = \frac{2ab}{a+b}$ the arithmetic mean and harmonic mean of a and b, respectively. For $\alpha = \frac{2}{3}$, $p = 0$, $q = 1$ in (2.4) we get the inequality

$$\frac{2}{3}G(a, b) + \frac{1}{3}H(a, b) \geq M_{-\frac{1}{3}}(a, b)$$

which was proved in [6].

The next result is a consequence of (ii) in Theorem 2.1.

Corollary 2.3. *For* $\alpha \in [0, 1]$, $p, q \geq 0$ *the inequality*

$$M_p^{\alpha}(a, b)M_q^{(1-\alpha)}(a, b) \leq M_{\alpha p+(1-\alpha)q}(a, b) \tag{2.5}$$

holds for all $a, b > 0$.

Let us denote by $A(a, b) = \frac{a+b}{2}$, $G(a, b) = \sqrt{ab}$,

$$L(a, b) = \begin{cases} \frac{b-a}{\ln b - \ln a}, & a \neq b \\ a, & a = b. \end{cases}$$

the arithmetic mean, geometric mean and logarithmic mean of two positive numbers a and b, respectively. Taking into account the result of Tung-Po Lin [2]

$$L(a, b) \leq M_{\frac{1}{3}}(a, b) \tag{2.6}$$

together with (2.5) we have

$$M_p^{\alpha}(a, b)L^{(1-\alpha)}(a, b) \leq M_{\alpha p+(1-\alpha)\frac{1}{3}}(a, b). \tag{2.7}$$

Specially, for $p = 1$ and $p = 0$ in (2.7) we get the inequalities

$$A^{\alpha}(a, b)L^{(1-\alpha)}(a, b) \leq M_{\frac{1+2\alpha}{3}}(a, b)$$

and

$$G^{\alpha}(a,b)L^{(1-\alpha)}(a,b) \le M_{\frac{1-\alpha}{3}}(a,b)$$

respectively, which results were published in [5].

Denote by

$$I(a,b) = \begin{cases} \frac{1}{e}\left(\frac{a^a}{b^b}\right)^{\frac{1}{a-b}}, & a \neq b \\ a, & a = b. \end{cases}$$

the identric mean of two positive integers. It was proved by Pittenger [4] that

$$M_{\frac{2}{3}}(a,b) \le I(a,b) \le M_{\ln 2}(a,b). \tag{2.8}$$

Using (2.5) together with (2.6) and (2.8) we immeditely have

$$I^{\alpha}(a,b)L^{(1-\alpha)}(a,b) \le M_{\alpha \ln 2 + (1-\alpha)\frac{1}{3}}.$$

Note, in the case of $\alpha = \frac{1}{2}$ our result does not improve the inequality

$$\sqrt{I(a,b)L(a,b)} \le M_{\frac{1}{2}}(a,b)$$

which is due to Alzer [1], but our result is a more general one.

With the help of using Theorem 2.1 more similar inequalities can be proved.

3. Open problems

Finally, we propose the following open problem on the convexity of power mean. The problem is to prove our conjecture, namely

$$\inf_{a,b>0}\{p : M_p(a,b) \text{ is concave for variable } p\} = \frac{\ln 2}{2},$$

$$\sup_{a,b>0}\{p : M_p(a,b) \text{ is convex for variable } p\} = \frac{1}{2}.$$

References

[1] Alzer, H., Ungleichungen für Mittelwerte, *Arch. Math (Basel)*, **47** (5) (1986) 422–426.

[2] Lin, T. P., The power mean and the logarithmic mean, *Amer. Math. Monthly*, **81** (1974) 879–883.

[3] Mildorf, T. J., A sharp bound on the two variable power mean, *Mathematical Reflections*, **2** (2006) 5 pages, Available online at [http://awesomemath.org/wp-content/uploads/reflections/2006_2/2006_2_sharpbound.pdf]

[4] Pittenger, A. O., Inequalities between arithmetic and logarithmic means, *Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat. Fiz.*, **678–715** (1980) 15–18.

[5] Shi, M.-Y., Chu, Y.-M. and Jiang, Y.-P., Optimal inequalities among various means of two arguments, *Abstr. Appl. Anal.*, (2009) Article ID 694394, 10 pages.

[6] Chu, Y.-M., Xia, W.-F.: Two sharp inequalities for power mean, geometric mean, and harmonic mean, *J. Inequal. Appl.* (2009) Article ID 741923, 6 pages.

Annales Mathematicae et Informaticae
42 (2013) pp. 9–21
http://ami.ektf.hu

Cube-and-Conquer approach for SAT solving on grids*

Csaba Biró[a], Gergely Kovásznai[b], Armin Biere[c], Gábor Kusper[a], Gábor Geda[a]

[a]Eszterházy Károly College
birocs,gkusper,gedag@aries.ektf.hu

[b]Vienna University of Technology
kova@forsyte.tuwien.ac.at

[c]Johannes Kepler University
biere@jku.at

Submitted May 10, 2013 — Accepted December 9, 2013

Abstract

Our goal is to develop techniques for using distributed computing resources to efficiently solve instances of the propositional satisfiability problem (SAT). We claim that computational grids provide a distributed computing environment suitable for SAT solving. In this paper we apply the Cube and Conquer approach to SAT solving on grids and present our parallel SAT solver `CCGrid` (Cube and Conquer on Grid) on computational grid infrastructure.

Our solver consists of two major components. The master application runs `march_cc`, which applies a lookahead SAT solver, in order to partition the input SAT instance into work units distributed on the grid. The client application executes an `iLingeling` instance, which is a multi-threaded CDCL SAT solver. We use BOINC middleware, which is part of the SZTAKI Desktop Grid package and supports the Distributed Computing Application Programming Interface (DC-API). Our preliminary results suggest that our approach can gain significant speedup and shows a potential for future investigation and development.

Keywords: grid, SAT, parallel SAT solving, lookahead, march_cc, iLingeling, SZTAKI Desktop Grid, BOINC, DC-API

*Supported by Austro-Hungarian Action Foundation, project ID: 83öu17.

1. Introduction

Propositional satisfiability is the problem of determining, for a formula of the propositional logic, if there is an assignment of truth values to its variables for which that formula evaluates to true. By SAT we mean the problem of propositional satisfiability for formulas in conjunctive normal form (CNF). SAT is one of the most-researched NP-complete problems [8] in several fields of computer science, including theoretical computer science, artificial intelligence, hardware design, and formal verification [5]. Also it should be noted that the hardness of the problem is caused by the possibly increasing number of the variables, since by a fixed set of variables SAT and n-SAT are regular languages and therefore there is a deterministic (theoretical) linear time algorithm to solve them, see, [21, 23].

Modern sequential SAT solvers are based on the Davis-Putnam-Logemann-Loveland (DPLL) [9] algorithm. This algorithm performs Boolean constraint propagation (BCP) and backtrack search, i.e., at each node of the search tree it selects a decision variable, assigns a truth value to it, and steps back when conflict occurs. Conflict-driven clause learning (CDCL) [5, Chpt. 4] is based on the idea that conflicts can be exploited to reduce the search space. If the method finds a conflict, then it analyzes this situation, determines a sufficient condition for this conflict to occur, in form of a learned clause, which is then added to the formula, and thus avoids that the same conflict occurs again. This form of clause learning was first introduced in the SAT solver `GRASP` [19] in 1996. Besides clause learning, lazy data structures are one of the key techniques for the success of CDCL SAT solvers, such as "watched literals" as pioneered in 2001, by the CDCL solver `Chaff` [20, 18] . Another important technique is the use of the VSIDS heuristics and the first-UIP backtracking scheme. In the state-of-the-art CDCL solvers, like `PrecoSAT` and `Lingeling` [3, 4], several other improvements are applied. Besides enhanced preprocessing techniques like e.g. failed literal detection, variable elimination, and blocked clause elimination, clause deletion strategies and restart policies have a great impact to the performance of the CDCL solver.

Lookahead SAT solvers [5, Chpt. 5] combine the DPLL algorithm with lookaheads, which are used in each search node to select a decision variable and at the same time to simplify the formula. One popular way of lookahead measures the effect of assigning a certain variable to a certain truth value: BCP is applied, and then the difference between the original clause set and the reduced clause set is measured (by using heuristics). In general, the variable for which the lookahead on *both* truth values results in a large reduction of the clause set is chosen as the decision variable. The first lookahead SAT solver was `posit` [10] in 1995. It already applied important heuristics for pre-selecting the "important" variables, for selecting a decision variable, and for selecting a truth value for it. The lookahead solvers `satz` [17] and `OKsolver` [16] further optimized and simplified the heuristics, e.g., `satz` does not use heuristics for selecting a truth value (rather prefers *true*), and `OKsolver` does not apply any pre-selection heuristics. Furthermore, `OKsolver` added improvements like local learning and autarky reasoning. In 2002, the solver `march` [13] further improved the data structures and introduced preprocessing techniques. As a variant of `march`, `march_cc` [14] can be considered as a case splitting tool. It produces a set of cubes, where each cube represents a branch cutoff in the DPLL tree constructed by the lookahead solver. It is also worth to mention that `march_cc` outputs learnt clauses as well, which represent refuted branches in the DPLL tree. The resulting set of cubes represents the remaining part of the search tree, which was not refuted by the lookahead solver itself.

There are two types of basic appearance of parallelism in computations, the "and-parallelism" and the "or-parallelism" [22]. The first is used in high performance computing, while the latter is more similar to nondeterministic guesses (data parallel). SAT can (theoretically effectively) be solved by several new computing paradigms using or-parallelism and by using, roughly speaking, exponential number of threads. Since multi-core architectures are common today, the need for parallel SAT solvers using multiple cores has increased considerably.

In essence, there are two approaches to parallel SAT solving [12]. The first group of solvers typically follow a divide-and-conquer approach. They split the search space into several subproblems, sequential DPLL workers solve the subproblems, and then these solutions are combined in order to create a solution to the original problem. This first group uses relatively intensive communication between the nodes. They do for example load balancing, and dynamic sharing of learned clauses.

The second group apply portfolio-based SAT solving. The idea is to run independent sequential SAT solvers with different restart policies, branching heuristics, learning heuristics, etc. `ManySAT` [11] was the first portfolio-based parallel SAT solver. `ManySAT` applies several strategies to the sequential SAT solver `MiniSAT`. `Plingeling` [3, 4] follows a similar approach, and uses the sequential SAT solver `Lingeling`. In most of the state-of-the-art portfolio-based parallel SAT solvers (e.g. `ppfolio`, `pfolioUZK`, `SATzilla`) not only different strategies, but even different sequential solvers compete and, to a limited extent, cooperate on the same formula. In such approaches there is no load balancing and the communication is limited to the sharing of learned clauses.

`GridSAT` [7, 6] was the first complete and parallel SAT solver employing a grid. It belongs to the divide-and-conquer group. It is based on the sequential SAT solver `zChaff`. Besides achieving significant speedup in the case of some (satisfiable and even unsatisfiable) instances, `GridSAT` is able to solve some problems for which sequential `zChaff` exceeds time out. `GridSAT` distributes only the short learned clauses over the nodes, therefore it minimizes the communication overhead. Search space splitting is based on the selection of a so-called pivot variable x on the second decision level, and then creating two subproblems by adding a new decision on x resp. $\neg x$ to the first decision level. If sufficient resources are available, the subproblems can further be partitioned recursively. Each new subproblem is defined by a clause set, including learned clauses, and a decision stack.

[15] proposes a more sophisticated approach, based on using "partition functions", in order to split a problem into a fixed number of subproblems. Two partition functions were compared, a scattering-based and a DPLL-based one with

lookahead. A partition function can be applied even in a recursive way, by repartitioning difficult subproblems (e.g., the ones that exceeds time out). For some of the experiments, an open source grid infrastructure called Nordugrid was used.

`SAT@home` [25] is a large volunteer SAT-solving project on grid, which involves more than 2000 clients. The project is based on the Berkeley Open Infrastructure for Network Computing (BOINC) [1], which is an open source middleware system for volunteer grid computing. On top of BOINC, the project was implemented by using the SZTAKI Desktop Grid [24], which provides the Distributed Computing Application Programming Interface (DC-API), in order to simplify the development, and then also to deploy and distribute applications to multiple grid environments. [25] proposes a rather simple partitioning approach: given a set of n selected variables, called a decomposition, a set of 2^n subproblems is generated. The key issue is how to select a decomposition. One way to solve this issue, is to derive the set of "important" decomposition variables from the original problem formulation, which, however, then is problem-specific, and needs human guidance. For instance, in the context of SAT-based cryptoanalysis of keystream generators, a decomposition set can be obtained from the encoding of the initial state of the linear feedback shift registers [25]. `SAT@home` uses no data exchange among clients.

Our approach, called `CCGrid`, also uses BOINC and the SZTAKI Desktop Grid, as it is detailed in Sect. 3, but is based on the Cube and Conquer approach [14]. For partitioning the input problem, we use `march_cc`. Our approach differs from the previous ones in the fact that it uses a parallel SAT solver, `iLingeling`, for solving the particular subproblems, on each client. In Sect. 4 we present some experiments and preliminary results.

2. Preliminaries

Given a Boolean variable x, there exist two *literals*, the positive literal x and the negative literal $\overline{x}$. A *clause* is a disjunction of literals, a *cube* is a conjunction of literals. Either a clause or a cube can be considered as a finite set of literals.

A *truth assignment* for a (finite) clause set or cube set F is a function ϕ that maps literals in F to $\{0,1\}$, such that if $\phi(x) = v$, then $\phi(\overline{x}) = 1 - v$. A clause resp. cube C is satisfied by ϕ if $\phi(l) = 1$ for some resp. every $l \in C$. A clause set resp. cube set F is satisfied by ϕ if ϕ satisfies C for every resp. some $C \in F$.

For representing the input clause set for a SAT solver, the DIMACS CNF format is commonly used, which references a Boolean variable by its (1-based) index. A negative literal is referenced by the negated reference to its variable. A clause is represented by a sequence of the references to its literals, terminated by a "0". The iCNF format extends the CNF format with a cube set.[1] A cube, called an assumption, is represented by a leading character "a" followed by the references to its literals and a terminating "0".

[1] `http://users.ics.tkk.fi/swiering/icnf/`

3. Architecture

Our application is a variant of the Cube and Conquer approach [14] and consists of two major components: a master application and a client application. The master is responsible for dividing the global input data into smaller chunks and distributing these chunks in the form of work units. Interpreting the output generated by the clients out of the work units and combining them to form a global output is also the job of the master. The architecture is depicted in Fig. 1. Similar to [25], the environment for running our system is the SZTAKI Desktop Grid [24] and BOINC [1], and was implemented by the use of the DC-API.

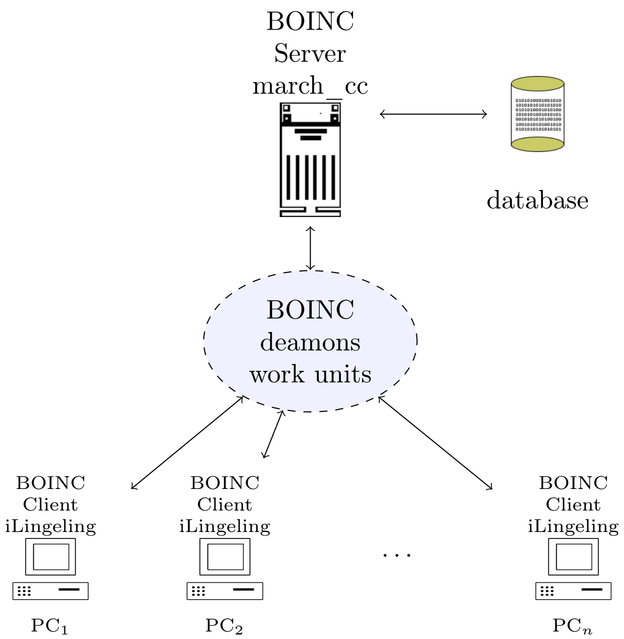

Figure 1: CCGrid architecture

The master

The master executes a partitioning tool called `march_cc` [14], which is based on the lookahead SAT solver `march`. Given a CNF file, `march_cc` primarily tries to refute the input clause set. If this does not succeed, `march_cc` outputs a set of assumptions (cubes) that describe the cutoff branches in the DPLL tree. These assumptions cover all subproblems of the input clause set that have not been refuted during the partitioning procedure. Given these assumptions, the master application creates work units, each of which consists of the input CNF file and a slice of the assumption set. As it can be seen in Fig. 2, if one of the clients reports one of the work units to be satisfiable, then the master outputs the satisfying model and destroys all the running work units. If every clients report unsatisfiability, then the master outputs unsatisfiability.

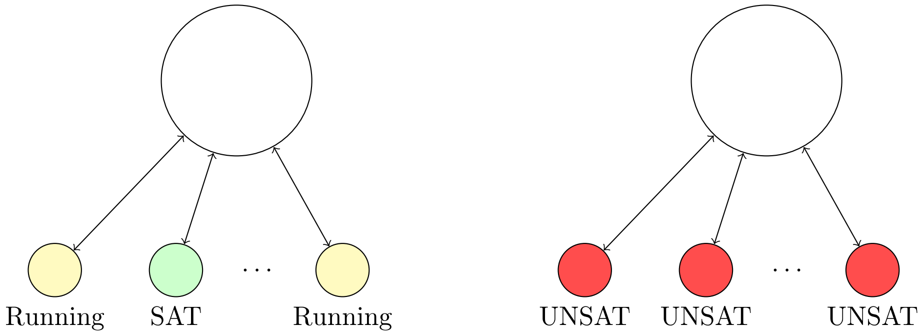

Figure 2: (a) If the problem is SAT, it is enough to find a SAT derived instance. (b) If the problem is UNSAT, one must show all derived instances UNSAT.

The pseudocode below shows how the master application works. It is divided into three procedures; the MAIN procedure is shown in Algorithm 1. It shares two constants with the other procedures: (i) *maxAsmCount* defines the maximum number of assumptions per work unit; (ii) *rfsInterval* gives a refresh interval at which DC-API events are processed. The master application uses several global variables; all of them are self-explanatory. In loop 6-9, work units are created, by calling the procedure CREATEWORKUNIT. Loop 10-13 then processes DC-API events generated by those work units that have finished solving their subproblems. Processing DC-API events is done by calling a callback function which has been previously set to PROCESSWORKUNITRESULT in line 3. The loop stops if either one of the work units returns a SAT result or all the work units completed.

CREATEWORKUNIT, shown in Algorithm 2, creates and submits a work unit to the grid. First, the CNF file is added to the new work unit. Then, in the loop, at most *maxAsmCount* assumptions from *asmFile* are copied into the new file *asmChunkFile*. Note that *asmFile* is global, it has been opened by the MAIN procedure (Algorithm 1, line 4), and therefore its current file position is held. Finally, *asmChunkFile* is added to the work unit, which is then submitted to the grid.

As already mentioned, PROCESSWORKUNITRESULT, shown in Algorithm 3, works as a callback function for DC-API events. It processes the result returned by a work unit.

Algorithm 1 Master: main procedure

```
Require: global constants maxAsmCount, rfsInterval
Require: global variables cnfFile, asmFile, wuCount, res, resFile
procedure MAIN
    initialize DC-API master
    set PROCESSWORKUNITRESULT as result callback
    open asmFile
    wuCount ← 0
    while not EOF(asmFile) do
        CREATEWORKUNIT
        wuCount ← wuCount + 1
    end while
    while res ≠ SAT and wuCount > 0 do
        wait rfsInterval
        process DC-API events
    end while
    if res ≠ SAT then
        res ← UNSAT
        cancel all work units
    end if
end procedure
```

Algorithm 2 Master: creating work units

```
procedure CREATEWORKUNIT
    wu ← new work unit
    wu.cnfFile ← cnfFile
    asmChunkFile ← new file
    for i ← 1 to maxAsmCount do
        if EOF(asmFile) then
            break
        end if
        copy next assumption from asmFile to asmChunkFile
        i ← i + 1
    end for
    wu.asmFile ← asmChunkFile
    submit wu to the grid
end procedure
```

Algorithm 3 Master: processing work unit result

```
procedure PROCESSWORKUNITRESULT(wu)
    if wu.res = SAT then
        res ← SAT
        copy wu.resFile to resFile
    end if
    wuCount ← wuCount − 1
end procedure
```

The client

Each client executes the parallel CDCL solver `iLingeling` [14, 4], for a fixed number of threads. Each thread executes a separate `lingeling` instance. `iLingeling` expects as input an iCNF file, including 1 or more assumptions, which is then loaded into a working queue. Each `lingeling` instance reads the input clause set, and then, in each iteration, gets the first assumption from the working queue.

If one of the `lingeling` instances can prove that the clause set is satisfiable under the given assumptions, then `iLingeling` reports that the clause set itself is satisfiable, the satisfying model is returned, and hence the remaining assumptions in the working queue can be ignored. Otherwise, i.e., if a `lingeling` instance reports unsatisfiability, then the assumption is retrieved from the working queue and the same SAT solver instance continues with the solving procedure. If the working queue becomes empty, then `iLingeling` reports that the clause set under the given set of assumptions is unsatisfiable.

Algorithm 4 shows the client's MAIN procedure. It uses one global constant, *thrCount*, which specifies the number of worker threads to use. First, the procedure creates an `iLingeling` instance with *thrCount* worker threads, loads both the CNF and the assumption files, and runs `iLingeling`. In loop 7-12, the results by all the threads are checked: if any of them is SAT then the result for the work unit is SAT; otherwise it is UNSAT (line 14). The result, as well as the satisfying model, is written into a result file by the procedure CREATERESULTFILE, shown in Algorithm 5.

Algorithm 4 Client: main procedure

Require: global constant *thrCount*
1: **procedure** MAIN(*wu*)
2: initialize DC-API client
3: *iLingeling* ← new iLingeling instance using *thrCount* threads
4: load *wu.cnfFile* into *iLingeling*
5: load *wu.asmFile* into *iLingeling*
6: run *iLingeling*
7: **for** $i \leftarrow 1$ **to** *thrCount* **do**
8: **if** *i*th thread's result is SAT **then**
9: *wu.res* ← SAT
10: **break**
11: **end if**
12: **end for**
13: **if** $i >$ *thrCount* **then**
14: *wu.res* ← UNSAT
15: **end if**
16: CREATERESULTFILE(*wu*, *iLingeling*)
17: **end procedure**

Algorithm 5 Client: creating result file

procedure CreateResultFile(*wu*, *iLingeling*)
  resFile ← new file
  write *wu.res* into *resFile*
  if *wu.res* = SAT **then**
    model ← satisfying assignment from *iLingeling*
    write *model* into *wu.resFile*
  end if
  wu.resFile ← *resFile*
end procedure

4. Results and testing environment

Our implementation consists of a quad-core SUN server with 6 GB memory, used as a master, and 20 quad-core PCs with 2 GB memory, used as clients. In our experiments, we used instances from the SAT Challenge 2012, from the Application (SAT + UNSAT) and the Hard Combinatorial (SAT + UNSAT) tracks. Results are presented in Tab. 1 and Tab. 2. The 1st column represents the instance's name. In the 2nd column, **A** resp. **HC** denotes Application resp. Hard Combinatorial problems. The 4th column shows the number of cubes, generated by `march_cc`. The 3rd resp. 5th column shows the runtime of `march_cc` resp. `iLingeling`, being executed on the master. The 6th column contains the sum of the previous two numbers, which represents the overall runtime of the cube-and-conquer approach running on a single (quad-core) machine. The total runtime of `CCGrid` is shown in the 7th column, while the 8th column measures the speedup as the ratio of the runtimes in the 6th and 7th columns.

In our approach, `CCgrid` have been executed without any communication among clients. Even though they do not cooperate and do not exchange learnt clauses, `CCGrid` shows a wide range of speedups. We achieved speedup up to ca. 8.5 on UNSAT instances (*QG-gensys-icl003.sat05-2715.reshuffled-07*) and up to ca. 7 on SAT instances (*sgen1-sat-160-100*).

Since the master has to distribute quite large work units over the network, communication overhead matters in the case of small instances, where communication costs are significant compared to the input size. Therefore, although we used a 1Gbps LAN in our experiments, cube-and-conquer running on a single machine outperformed `CCgrid` on some instances. If we look at the *battleship-16-31-sat* row in Tab. 1, we can see that `march_cc` and `iLingeling` can solve this problem on 1 client a bit faster than `CCGrid` on 20 clients.

In the case of satisfiable instances, we might be lucky, finding a model quickly, or unlucky. If there are many satisfying models, then it is not worth to distribute the problem over many clients. However, if there exist only a few models, then it is a good idea to use many clients, since the more clients we use, the more probable it is for a client to be lucky enough to find one of those few solutions. Unfortunately, we have no information about how many models the instances in Tab. 1 have.

		march_cc	# of cubes	iLingeling	cube-and-conquer	CCGrid	speedup
# of clients		**1**		**1**	**1**	**20**	
vmpc_26	**A**	8.38	296	40.22	48.6	13.64	**3.56**
AProVE09-07	**A**	65.93	4245	19.12	85.05	79.16	**1.07**
clauses-4	**A**	29.68	25	59.01	88.69	81.98	**1.08**
gss-16-s100	**A**	155.28	6292	201.21	356.49	171.71	**2.08**
IBM_FV_ 2004_rule_ batch_22 _SAT_dat. k65	**A**	17.93	361	148.95	166.88	154.02	**1.08**
ezfact64_3. sat05-450. reshuffled-07	**HC**	458.71	469428	63.71	522.42	505.72	**1.03**
sgen1-sat-160-100	**HC**	10.65	210168	419.92	430.57	62.28	**6.91**
em_7_4 _8_exp	**HC**	20.06	19419	170.9	190.96	46.47	**4.11**
battleship-16-31-sat	**HC**	174.89	91757	2.69	177.58	180.89	**0.98**
Hidoku_ enu_6	**HC**	125.02	256225	91.61	216.63	159.31	**1.36**

Table 1: Runtimes and speedup
all instances are SAT

`CCGrid` seems to be much better in distributing satisfiable instances from the **HC** track than the ones from the **A** track, since `march_cc` seems to generate much more cubes for the previous ones.

In the case of unsatisfiable instances, we cannot be lucky to find an early solution since there is no satisfying model. When comparing the speedups in Tab. 1 and Tab. 2, we can see that speedups around 1 are more frequent on satisfiable instances.

This shows that in the case of unsatisfiable instances there is less risk of wasting resources without any speedup.

5. Future work and conclusion

This paper presents a first attempt of applying the Cube and Conquer approach [14] to computational grids. We presented the parallel SAT solver `CCGrid`, which runs on the MTA SZTAKI Grid using BOINC. In this version, the master application applies `march_cc`, using a lookhead solver, to split a SAT instance. The client application uses the parallel SAT solver `iLingeling` to deal with several assumptions. The client creates a separate `iLingeling` instance for each work unit, and destroys it after completing the work unit. For the sake of improving our current results, in future work, we would like to preserve the state of `iLingeling` instances, including learnt clauses.

In our experience, the cube generation phase implemented in `march_cc` makes up a significant part of the runtime. As a consequence, we were mostly able to achieve significant speedup on such instances on which the cube generation phase

		march_cc	# of cubes	iLingeling	cube-and-conquer	CCGrid	speedup
# of clients		**1**	**1**	**1**	**20**		
counting-clqcolor-unsat-set-b-clqcolor-08-06-07.sat05-1257. reshuffled-07	**A**	5.77	112757	12.77	18.54	8.71	**2.13**
gensys-ukn002.sat05-2744. reshuffled-07	**A**	12.70	21408	230.01	242.71	71.55	**3.39**
Q32inK09	**A**	12.15	5279	35.89	48.04	14.38	**3.34**
QG6-dead-dnd002.sat05-2713. reshuffled-07	**A**	2.38	12147	35.79	38.17	4.74	**8.05**
QG-gensys-icl003.sat05-2715. reshuffled-07	**A**	25.43	38466	291.05	316.48	37.24	**8.49**
instance_n6 i7_pp_ci_ce	**A**	103.78	29290	111.72	215.50	117.20	**1.84**
AProVE07-09.cnf	**A**	34.43	86048	4.55	37.98	37.46	**1.01**
battleship-10-10-unsat	**HC**	0.36	2317	15.47	15.83	4.48	**3.53**
rand_net60-40-10.shuffled	**HC**	111.23	130227	13.24	124.47	115.62	**1.08**
smtlib-qfbv-aigs-ext_con_032 008_0256-tseitin	**HC**	67.24	22384	7.29	74.53	71.47	**1.04**

Table 2: Runtimes and speedup
all instances are UNSAT

took a relatively short time. Therefore, our further aim to reduce the time spent on cube generation by parallelizing the look-ahead solver. We plan to adapt `march_cc` to a cluster infrastructure and to investigate the possibility of merging our BOINC-based approach with a cluster-based master application. We expect further improvement by analyzing the generated cubes and then, based on the result of the analysis, partitioning the cube set in a more sophisticated way.

In order to achieve larger speedup on unsatisfiable instances, it might be useful to call `march_cc` not only while partitioning the original problem, but also for repartitioning difficult subproblems, e.g., those on which a client exceeds a certain time limit. Finally, it might be interesting to apply similar techniques not only to clusters resp. grids, but also to cloud computing platforms.

References

[1] D. P. Anderson, BOINC: A System for Public-Resource Computing and Storage. *Proc. of GRID 2004*, pp. 4–10, 2004.

[2] A. Balint, A. Belov, D. Diepold, A. Gerber, M. Järvisalo, C. Sinz (eds), Proceedings of SAT Challenge 2012: Solver and Benchmark Descriptions. *Department of Computer Science Series of Publications B*, vol. B-2012-2, University of Helsinki, 2012.

[3] A. Biere, Lingeling, Plingeling, PicoSAT and PrecoSAT at SAT Race 2010. Technical Report 10/1, FMV Reports Series, JKU, 2010.

[4] A. Biere, Lingeling and Friends Entering the SAT Challenge 2012. *Haifa Verification Conference*, *Department of Computer Science Series of Publications B*, vol. B-2012-2, pp. 33–34, 2012.

[5] A. Biere, M. Heule, H. van Maaren, T. Walsh, Handbook of Satisfiability. IOS Press, Amsterdam, 2009.

[6] W. Chrabakh, R. Wolski, GridSAT: A Chaff-based Distributed SAT Solver for the Grid. *Proc. of SC'03*, pp. 37–49, 2003.

[7] W. Chrabakh, R. Wolski, GrADSAT: A Parallel SAT Solver for the Grid. Technical report, UCSB Computer Science, 2003.

[8] S. A. Cook, The Complexity of Theorem-Proving Procedures. Proc. of STOC'71, pp. 151–158, 1971.

[9] M. Davis, G. Logemann, D. Loveland, A Machine Program for Theorem Proving. *Commun. ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[10] J.W. Freeman, Improvements to Propositional Satisfiability Search Algorithms. Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania, May, 1995.

[11] Y. Hamadi S. Jabbour, L. Sais, ManySAT: a Parallel SAT Solver. *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 6, pp. 245–262, 2009.

[12] Y. Hamadi, S. Jabbour, C. Piette, L. Saïs, Deterministic Parallel DPLL. *JSAT*, vol. 7. no. 4, pp. 127–132, 2011.

[13] M. Heule, March: Towards a Look-ahead SAT Solver for General Purposes. Master thesis, TU Delft, The Netherlands, 2004.

[14] M. Heule, O. Kullmann, S. Wieringa, A. Biere, Cube and Conquer: Guiding CDCL SAT Solvers by Lookaheads. *Proc. of Haifa Verification Conference*, *Lecture Notes in Computer Science*, vol. 7261, pp. 50–65, 2011.

[15] A. E. J. Hyvärinen, T. Junttila, I. Niemelä, Partitioning SAT instances for distributed solving. *Proc. of LPAR'10*, pp. 372–386, 2010.

[16] O. Kullmann, Investigating the Behaviour of a SAT Solver on Random Formulas. Technical Report CSR 23-2002, Swansea University, Computer Science Report Series, October, 2002.

[17] C.M. Li and Anbulagan, Look-Ahead versus Look-Back for Satisfiability Problems. *Lecture Notes in Computer Science*, vol. 1330, pp. 342–356, 1997.

[18] Y. S. Mahajan, Z. Fu, S. Malik, Zchaff2004: An Efficient SAT Solver. *Lecture Notes in Computer Science: Theory and Applications of Satisfiability Testing*, vol. 3542, pp. 360–375, 2005.

[19] J. P. Marques-Silva, K. A. Sakallah, GRASP: A New Search Algorithm for Satisfiability. *Proc. of ICCAD'96*, pp. 220–227, 1996.

[20] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, S. Malik, Chaff: Engineering an Efficient SAT Solver. *Proc. of DAC'01*, pp. 530–535, 2001.

[21] B. Nagy, The Languages of SAT and n-SAT over Finitely Many Variables are Regular. *Bulletin of the EATCS*, no. 82, pp. 286–297, 2004.

[22] B. Nagy, On the Notion of Parallelism in Artificial and Computational Intelligence. *Proc. of HUCI 2006*, pp. 533–541, 2006.

[23] B. Nagy, On Efficient Algorithms for SAT. *Proc. of CMC 2012*, Lecture Notes in Computer Science, vol. 7762, pp. 295–310, 2013.

[24] P. Kacsuk, J. Kovács, Z. Farkas, A. C. Marosi, G. Gombás, Z. Balaton, SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System. *Journal of Grid Computing*, vol. 7, no. 4, pp. 439–461, 2009.

[25] M. Posypkin, A. Semenov, O. Zaikin, Using BOINC Desktop Grid to Solve Large Scale SAT Problems. *Computer Science*, vol. 13, no. 1, pp. 25–34, 2012.

Annales Mathematicae et Informaticae
42 (2013) pp. 23–28
http://ami.ektf.hu

On normals of manifolds in multidimensional projective space*

Marina Grebenyuk[a], Josef Mikeš[b]

[a]National Aviation University, Kiev, Ukraine
ahha@i.com.ua

[b]Palacky University Olomouc, Czech Republic
josef.mikes@upol.cz

Submitted December 4, 2012 — Accepted January 31, 2013

Abstract

In the paper the regular hyper-zones in the multi-dimensional non-Euclidean space are discussed. The determined bijection between the normals of the first and second kind for the hyper-zone makes it possible to construct the bundle of normals of second-kind for the hyper-zone with assistance of certain bundle of normals of first-kind and vice versa. And hence the bundle of the normals of second-kind is constructed in the third-order differential neighbourhood of the forming element for hyper-zone. Research of hyper-zones and zones in multi-dimensional spaces takes up an important place in intensively developing geometry of manifolds in view of its applications to mechanics, theoretical physics, calculus of variations, methods of optimization.

Keywords: non-Euclidean space, regular hyper-zone, bundle of normals, bijection

MSC: 53B05

1. Introduction

In this article we analyze the theory of regular hyper-zone in the extended non-Euclidean space. We derive differential equations that define the hyper-zone SH_r

*Supported by grant P201/11/0356 of The Czech Science Foundation.

with regards to a self-polar normalised frame of space ${}^{\lambda}S_n$. The tensors which determine the equipping planes in the third-order neighborhood of the hyper-zone are introduced. The bundles of the normals of the first and second kind are constructed by an inner invariant method in the third-order differential neighbourhood of the forming element for hyper-zone. The bijection between the normals of the first and second kind for the hyper-zone SH_r is determined.

The concept of zone was introduced by W. Blaschke [1]. V. Wagner [7] was the first who proposed to consider the surface equipped with the field of tangent hyper-planes in the n-dimensional centro-affine space.

We apply the group-theoretical method for research in differential geometry developed by professor G.F. Laptev [4]. At present the method of Laptev remains the most efficient way of research for manifolds, immersed in generalized spaces. We use results obtained in the article [3].

For the past years the methods of generalizations of Theory of regular and singular hyper-zones (zones) with assistance of the Theory of distributions in multidimensional affine, projective spaces and in spaces with projective connections were studied by A.V. Stolyarov, Y.I. Popov and M.M. Pohila. In this article we analyze the theory of regular hyper-zone in the extended non-Euclidean space. We derive differential equations that define the hyper-zone SH_r with regards to a self-polar normalised frame of space ${}^{\lambda}S_n$. The tensors which determine the equipping planes in the third-order neighborhood of the hyper-zone are introduced. The bundles of the normals of the first and second kind are constructed by an inner invariant method in the third-order differential neighbourhood of the forming element for hyper-zone. The bijection between the normals of the first and second kind for the hyper-zone SH_r is determined.

Before M. Grebenyuk and J. Mikeš in the article [2] discussed the theory of the linear distribution in affine space. The bundles of the projective normals of the first kind for the equipping distributions are constructed by an inner invariant method in second and third differential neighbourhoods of the forming element. In the article we apply the group-theoretical method for research in differential geometry developed by G.F. Laptev [4]. At present the method of Laptev remains the most efficient way of research for manifolds, immersed in generalized spaces. We use results obtained in the article [3].

2. Definition of the hyper-zone in the extended non-Euclidean space

Let a non-degenerated hyper-quadric be given in a projective n-dimensional space P_n as

$$q'_{IJ}x^I x^J = 0, \quad q'_{IJ} = q'_{JI}, \quad \det\|q'_{IJ}\| \neq 0, \qquad I, J = 0, 1, \ldots, n,$$

where the smallest number of the coefficients of the same sign is equal to λ. Thus, it is possible to determine a subgroup of collineations for space P_n, which are preserving this hyper-quadric and, hence, it is possible to introduce a projective metrics.

Let us name the obtained in this way metric space with this fundamental group as the extended non-Euclidian space ${}^{\lambda}S_n$ with index λ [5], and the corresponding hyper-quadric as the absolute of the space ${}^{\lambda}S_n$.

Let us consider a plane element (A,τ) in the space ${}^{\lambda}S_n$ which is composed of a point A and a hyper-plane τ, where point A belongs to plane τ.

Definition 2.1. Suppose that the point A defines an r-dimensional surface V_r and the hyper-plane $\tau(A)$ is tangent to the surface V_r in the corresponding points $A \in V_r$. Then the r-parametric manifold of the plane elements (A,τ) is called *r-parametric hyper-zone* $SH_r \subset {}^{\lambda}S_n$. The surface V_r is called the base surface and the hyper-planes $\tau(A)$ are called the principal tangent hyper-planes to the hyper-zone SH_r.

Definition 2.2. The characteristic plane $X_{n-r-1}(A)$ for the tangent hyper-plane $\tau = \tau(u^1,\dots,u^r)$ is called the *characteristic plane* for the hyper-zones SH_r at the point $A(u^1,\dots,u^r)$.

Definition 2.3. The hyper-zone SH_r is called *regular* if the characteristic plane $X_{n-r-1}(A)$ and the tangent plane $T_r(A)$ for directing surface V_r for hyper-zone SH_r at each point $A \in V_r$ have no common straight lines.

The regular hyper-zone SH_r in a self-polar normalized basis $\{A_0, A_1, \dots, A_n\}$ in the space ${}^{\lambda}S_n$ is defined as follows:

$$\omega_o^n = 0, \quad \omega_o^\alpha = 0, \quad \omega_\alpha^n = 0, \quad \omega_n^o = 0, \quad \omega_n^\alpha = 0, \quad \omega_\alpha^o = 0,$$

$$\omega_i^n = a_{ij}\omega^j, \quad \omega_\alpha^i = b_{\alpha j}^i\omega^j, \quad \omega_i^\alpha = b_{ij}^\alpha\omega^j, \quad \omega_i^o = -\varepsilon_{oi}\omega^i, \quad \omega_n^i = \varepsilon_{in}a_{ij}\omega^j,$$

$$\nabla a_{ij} = -a_{ij}\omega_n^n - a_{ijk}\omega^k, \quad \nabla b_{\alpha j}^i = b_{\alpha jk}^i\omega^k, \quad \nabla b_{ij}^\alpha = b_{ijk}^\alpha\omega^k,$$

where

$$b_{\alpha j}^i a_{i\ell} = b_{\alpha\ell}^i a_{ij}, \quad b^\alpha ik = -\varepsilon_{\alpha i} b_a^{ij} a_{jk}, \quad b_{\alpha k}^i = b_\alpha^{ij} a_{jk},$$

and functions $b_{\alpha jk}^i$ are symmetric according to indices j and k.

Systems of objects

$$\Gamma_2 = \{a_{ij}, b_{\alpha j}^i\}, \quad \Gamma_3 = \{\Gamma_2, a_{ijk}, b_{\alpha jk}^i\}$$

make up fundamental objects of second and third orders respectively for hyper-zone $SH_r \subset {}^{\lambda}S_n$.

3. Canonical bundle of projective normals for the hyper-zone

With the help of the components of fundamental geometric object of the third order for hyper-zone $SH_r \subset {}^{\lambda}S_n$ let us construct the quantities

$$d_i = \frac{1}{r+2}\, a_{ijk}a^{jk}, \quad \nabla_\delta d_i = 0,$$

$$d^i = \frac{1}{r+2}\, a^{ijk}a_{jk}, \quad \nabla_\delta d^i = d^i\pi_n^n.$$

The tensors d_i and d^i define dual equipping planes in the third-order neighborhood of the hyper-zone SH_r

$$E_{r-1} \equiv [M_i] = [A_i + d_i A_o], \quad E_{n-r} \equiv [\sigma^i] = [\tau^i + d^i \tau^n].$$

Using the Darboux tensor

$$\mathcal{L}_{ijk} = a_{ijk} - a_{(ij}d_{k)},$$

one builds the symmetric tensor

$$L_{ij} = a^{k\ell} a^{mp} \mathcal{L}_{ikm} \mathcal{L}_{j\ell p}, \qquad \nabla_\delta L_{ij} = 0,$$

which is non-degenerate in general case.

Let us consider a field of straight lines associated with the hyper-zone SH_r

$$h(A_o) = [A_o, P], \qquad P = A_n + x^i A_i + x^\alpha A_\alpha,$$

where each line passes through the respective point A of the directing surface V_r and do not belong to the tangent hyper-plane $\tau(A_o)$.

Let us require that straight line $h = [A_o, P]$ is an invariant line, i.e. $\delta h = \theta h$. The last condition is equivalent to the differential equations:

$$\nabla_\delta \chi^\alpha = \chi^\alpha \pi_n^n \qquad \text{and} \qquad \nabla_\delta \chi^i = \chi^i \pi_n^n.$$

First equations are realized on the condition that $x^\alpha = B^\alpha$, and second equations have two solutions:

$$x^i = -d^i, \quad x^i = B^i.$$

Hence, the system of the differential equations has a general solution of the following form:

$$x^i = -d^i + \sigma(B^i + d^i),$$

where σ is the absolute invariant.

Thus, we obtain the bundle of straight lines, which is associated with the hyper-zone SH_r by inner invariant method:

$$h(\sigma) = [A_o, P(\sigma)] = [A_o, A_n + \{(\sigma - 1)d^i + \sigma B^i\} A_i + B^\alpha A_\alpha],$$

where σ is the absolute invariant.

The constructed projective invariant bundle of straight lines makes it possible to construct the invariant bundle of first-kind normals E_{n-r}, which is associated by the inner method with the hyper-zone SH_r in the differential neighborhood of the third order of its generatrix element.

Consequently, it is possible to represent each invariant first kind normal $E_{n-r}(A_o)$ as the $(n-r)$-plane that encloses the invariant straight line $h(A_o)$ and the characteristic $X_{n-r-1}(A_o)$ for hyper-zone SH_r [6].

$$E_{n-r}(\sigma) \overset{def}{=} [X_{n-r-1}(A_o);\ A_n + \{(\sigma - 1)d^i + \sigma B^i\}\ A_i + B^\alpha A_\alpha,$$

where σ is the absolute invariant.

4. Bijection between first- and second-kind normals of the hyper-zone SH_r

Let us introduce the correspondence between the normals of the first- and second-kind for the hyper-zone SH_r. For that, let us construct a tensor:

$$P_i = -a_{ij}\nu^j + d_i, \quad \nabla_\delta P_j = 0 \tag{4.1}$$

where ν^j is the tensor satisfying the condition $\nabla_\delta \nu^j = \nu^j \pi_n^n$.

The tensor P_i defines the normal of second-kind for hyper-zone SH_r that is determined by the points

$$M_i = A_i + \chi_i A_o, \quad \nabla_\delta \chi_i = 0.$$

Further, the tensor ν^j can be represented using the components of the tensor P_i as follows

$$\nu^j = -P_i a^{ij} + d^j.$$

Therefore, the bijection between the normals of the first- and second-kind for the hyper-zone SH_r is obtained using the relations (4.1). The constructed bijection makes it possible to determine the bundle of second-kind normals, using the bundle of first-kind normals and vice versa. Therefore, we got constructed the bundle of second-kind normals, which is associated by the inner method with the hyper-zone SH_r in the differential neighborhood of the third order of its generating element. So true the following theorem.

Theorem 4.1. *Tensor P_i defines the bijection between the normals of the first- and second-kind for the hyper-zone SH_r.*

Finally, we get the theorem.

Theorem 4.2. *Tensor $\nu^j = -P_i a^{ij} + d^i$ defines the bundle of second- kind normals, which is associated by inner method with the hyper-zone SH_r in the differential neighborhood of the third order of its generating element.*

References

[1] Blaschke, W., *Differential geometry*, I. OSTI, M., L., (1935).

[2] Grebenyuk, M., Mikeš, J., Equipping Distributions for Linear Distribution, *Acta Palacky Univ. Olomuc., Mathematica*, 46 (2007) 35–42.

[3] Kosarenko, M. F., Fields of the geometrical objects for regular hyper-zone $SH_r \subset {}^\lambda S_n$, (Russian), Differential geometry of manifolds of figures. *Interuniversity subject collection of scientific works. Kaliningrad Univ.*, 13 (1982) 38–44.

[4] Laptev, G. F., Differential geometry of embedded manifolds: group-theoretical method for research in differential geometry, *Proc. of the Moscow Math. Soc.*, 2 (1953) 275–382.

[5] Rozenfeld, B. A., Non-Euclidean geometries, *Gostechizdat, Moscow*, (1955).

[6] Stoljarov, A. V., Differential geometry of zones, *Itogi Nauki Tekh., Ser. Probl. Geom.*, 10 (1978) 25–54.

[7] Wagner, V., Field theory for local hyper-zones, *Works of seminar on vector and tensor analysis*, 8 (1950) 197-272.

Annales Mathematicae et Informaticae
42 (2013) pp. 29–44
http://ami.ektf.hu

Complexity metric based source code transformation of Erlang programs*

Roland Király

Eszterházy Károly Collage
Institute of Mathematics and Informatics
kiraly.roland@aries.ektf.hu

Submitted December 4, 2013 — Accepted December 11, 2013

Abstract

In this paper we are going to present how to use an analyzer, which is a part of the *RefactorErl* [10, 12, 13], that reveals inadequate programming style or overcomplicated erlang [14, 15] program constructs during the whole lifecycle of the code using complexity measures describing the program. The algorithm [13], which we present here is also based upon the analysis of the semantic graph built from the source code, but at this stage we can define default complexity measures, and these defaults are compared to the actual measured values of the code, and so the differences can be indicated. On the other hand we show the algorithm measuring code complexity in Erlang programs, that provides automatic code transformations based on these measures. We created a script language that can calculate the structural complexity of Erlang source codes, and based on the resulting outcome providing the descriptions of transformational steps. With the help of this language we can describe automatic code transformations based on code complexity measurements. We define the syntax [11] of the language that can describe those series of steps in these automatic code refactoring that are complexity measurement [7, 9] based, and present the principle of operation of the analyzer and run-time providing algorithm. Besides the introduction of the syntax and use cases, We present the results we can achieve using this language.

Keywords: software metrics, complexity, source code, refactorerl

---

*This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/2-11-1-2012-0001 'National Excellence Program'.

1. Introduction

Functional programming languages, thus Erlang as well, contain several special program constructs, that are unheard of in the realm of object-oriented and imperative languages.

The special syntactic elements make functional languages different, these attributes contribute to those being interesting or extraordinary, but also due to these, some of the known complexity measures are not, or only through modifications usable to measure code.

This does not mean that complexity measures are not developed to these languages, but very few of the existing ones are generic enough to be used with any functional language [3, 4, 8] language-independently, therefore with Erlang as well, because most of these only work well with one specific language, thus have low efficiency with Erlang codes.

For all of this I needed to define the measures of complexity that can be utilized with this paradigm, and create new ones as necessary.

There are tools for measuring software complexity, like Eclipse [6], or the software created by Simon, Steinbrückner and Lewerentz, that implements several complexity measures that help the users in measurement.

The aim of the *Crocodile* [5] project is to create a program that helps to efficiently analyze source code, therefore it can be used quite well to makes measurements after code transformations. *Tidier* [17, 18] is an automatic source code analyzer, and transformer tool, that is capable of automatically correcting source code, eliminating the syntactic errors static analysis can find, but neither software/method uses complexity measures for source code analysis and transformation.

This environment raised the demand for a complex and versatile tool, that is capable of measuring the complexity of Erlang codes, and based on these measurements localize as well as automatically or semi-automatically correct unmanageably complex parts.

We have developed a tool *RefactorErl* [12, 10, 13] which helps to performing refactoring steps. In the new version of the tool we implemented the algorithm and the transformation script language, which enables to write automatic metric based source code transformations.

Problem 1.1 (Automated program transformations)**.** *In this article we examined the feasibility of automated transformation of (functional) Erlang [14, 15] programs' source codes based on some software complexity measures, and if it is possible to develop transformation scheme to improve code quality based on the results of these measurements.*

In order to address the problem we have created an algorithm that can measure the structural complexity of Erlang programs, and can provide automatic code transformations based on the results, we have also defined a script language that offers the description of the transformation steps for the conversion of different program designs.

In our opinion the analysis of complexity measures on the syntax tree created from the source code and the graph including semantic information built from this [12, 16] allows automatic improvement of the quality of the source code.

To confirm this statement, we attempt to make a script that improves a known *McCabe* complexity measure, namely the cyclomatic number (defined in Chapter 2.), in the language described in the first section of Chapter 4., and run this on known software components integrated in Erlang distributions.

We chose *McCabe*'s cyclomatic number for testing, because this measure is well enough known to provide sufficient information on the complexity of the program's source code not only to programmers that are familiar with Erlang or other functional languages.

With the examination of the results of measurements performed in order to validate our hypothesis, and with the analysis of the impact of the transformations we addressed the following questions:

- The modules' cyclomatic number is characterized by the sum of the cyclomatic numbers of the functions. This model cannot take into account the function's call graph, which distorts the resulting value. Is it worthwhile to examine this attribute during the measurements, and to add it to the result?

- Also in relation with the modules, the question arises as to which module is more complex: one that contains ten functions, all of whose *McCabe* value is 1, or one that has a function bearing a *McCabe* value of 10?

- The cyclomatic number for each function is at least one, because it contains a minimum of one path. Then if we extract the more deeply embedded selection terms from within the function, in a way that we create a new function from the selected expression (see Chapter 2.) the cyclomatic number that characterizes the module increases unreasonably (each new function increases it by one). Therefore, each new transformation step is increasingly distorting the results. The question in this case is that this increase should or should not be removed from the end result?

- Taking all these into account, what is the relationship between the cyclomatic number of the entire module, and the sum of the cyclomatic numbers of the functions measured individually?

- How can we best improve the cyclomatic number of Erlang programs, also what modifications should be carried out to improve the lexical structure, the programming style, of the program?

- If a function contains more consecutive selections and another one embeds these into each other, should the cyclomatic numbers of the two functions be regarded as equivalent?

2. Used complexity metrics

In this chapter, for the sake of clarity, we define the complexity metrics that are used during the application of the scripts that manage the transformations. Out of the applicable metrics of the analytical algorithm we have made, in the present writing we only use the *McCabe* cyclomatic number, the *case* statements' maximum embeddedness metric, and the measuring of the number of functions, therefore we only describe these in detail.

The *McCabe* McCabe complexity measure is equivalent to the number of basic routes defined in the control graph [1] constructed by *Thomas J. McCabe*, namely how many types of outputs can a function have not counting the number of the traversal paths of the additionally included functions.

The *McCabe* cyclomatic number was originally developed for the measurement of subprograms in procedural languages. This metric is also suitable for the measurement of functions implemented within modules in functional languages, such as Erlang [14]. *Thomas J. McCabe* defines the cyclomatic number of programs as follows:

Definition 2.1 (McCabe cyclomatic number)**.** The $G = (v, e)$ control flow graph's $V(G)$ cyclomatic number is $V(G) = e - v + 2p$, where p represents the number of the graph's components, which corresponds to the number of linearly connected loops that are located in the strongly connected graph [9].

The *McCabe* number to measure the functions of Erlang programs can be specified as follows:

Definition 2.2 (McCabe in Erlang)**.** Let f_i be the branches (overload versions) of the $fc(f_i)$, function, let $if_{cl}(f_i)$, and $case_{cl}(f_i)$ denote the branches of the if's, and *case*'s within the branches. Then the result of the $McCabe$ cyclomatic number measured for functions is $MCB(f_i) = |fc(f_i)| + |case_{cl}(f_i)| + |if_{cl}(f_i)|$.

The measure can be applied to a group of functions:

$$MCB(f_1, ..., f_k) = \sum_{j=1}^{k} MCB(f_j).$$

The results measured on the module's functions $m_i \in M$ are equal to the sum of the results measured on all the function from within the module:

$$MCB(m_i) = MCB(F(m_i))$$

The next measure of complexity we use measures the maximum embedding of the case statements within the functions.

$$MCB(M) = \sum_{m \in M} MCB(F(m))$$

The next measure of complexity we use measures the maximum embedding of the *case* statements within the functions.

c_0:
case e of
p_1 [when g_1] $\rightarrow e^1_1, \ldots, e^1_{l_1}$;
$\vdots$
p_n [when g_n] $\rightarrow e^n_1, \ldots, e^n_{l_n}$
end

denotes an Erlang *case* case statement, where e, ands $e_i \in E$ are Erlang expressions, $p \in P$ are patterns, $g_i \in G$ are guards in the branches. The e^i_j expressions in branches of the *case* statements may contain nested control structures, including further *case* expressions.

Definition 2.3 (Max depth of cases)**.** In order to measure the embeddedness $T(f_i)$ be the set of all *case* expression located in the f_i function. Let $t(c_1, c_2)$ denote any branch of the *case* expression c_1 that contains *case* expression c_2 and $\nexists c_3$ *case* expression that $t(c_1, c_3) \wedge t(c_3, c_2)$. Let $t_s(c, c_x)$ denote the case that *case* expression c contains in one of its branches, at some depth *case* expression c_x that is $\exists\ c_1, ..., c_n$ *case* expressions so that

$$t(c, c_1), t(c_1, c_2), ..., t(c_{n-1}, c_n), t(c_n, c_x).$$

The $|t_s(c, c_x)|$'s embeddedness depth in this case is $n + 1$. Let $T_0(f_i)$ be the set of those *case* expressions which are not contained in any of the $T(f_i)$ set's *case* expressions (top-level *case* statement). Then the

$$MDC(f_i) = max\{|t_s(c, c_x)| \ \ |c \in T_0(f_i), c_x \in T(f_i)\}.$$

After defining the embeddedness depth let us inspect the third metric we have applied, which measures the number of functions in the modules. This measure is particularly relevant in the characterization of functional programs, since these contain a large number of function-constructions, so in addition to the number of rows, by using this metric we can infer the size of the modules. The general definition of the functions in Erlang can be described by the following formula:

f_0:
$\mathsf{f}^{\mathsf{c}}_{\mathsf{1}}(p_1)$ when $g_1 \ \rightarrow$
$e^1_1, \ldots, e^1_{l_1}$;
$\vdots$
$\mathsf{f}^{\mathsf{c}}_{\mathsf{n}}(p_n)$ when $g_n \ \rightarrow$
$e^n_1, \ldots, e^n_{l_n}$.

where f^c_i is the i^{th} function's branch, $e_i \in E$ are Erlang expressions $g_i \in G$ are guards belonging to the branches, and $p_i \in P$ are patterns that form the function's formal parameter list.

Definition 2.4 (Number of functions)**.** The result of the measurement for all the modules in the semantic graph [12, 16] used to store the source code is $NOF(M) = |F(M)|$, where $F(M)$ denotes all functions in all modules.

In addition to the metrics presented here the analytical algorithms is capable of assessing several other complexity measures, and can apply these measurement results in the construction of various transformations.

3. Transformations used to improve code quality

This chapter describes the operation of the transformation steps from the scripts used to improve the quality of the source code. The scripts automatically transform the program constructions located in the source code, based on the complexity measures presented in Chapter 2.

To improve the *McCabe* cyclomatic number and the programming style we apply the extraction of deeply embedded *case* statements, and in some cases, where, as the effect of the transformations, the number of functions becomes too high, the transformation steps carrying out the movement of functions.

3.1. Conversion of a *case* expression into a function

This transformation step converts the *case* statement designated for extraction into a function, then places a call to the new function in place of the original expression in the way that the bound variables in the expression are converted into parameters (see: Figure 1.).

case statement		function statement
e_0: case e of p_1 when $g_1 \to e^1_1, \dots, e^1_{l_1}$; $\vdots$ p_n when $g_n \to e^n_1, \dots, e^n_{l_n}$ end	$\to$	e_0: m : f(e) or f(e) where the definition of the function is: f(p_1) when $g_1 \to$ $e^1_1, \dots, e^1_{l_1}$; $\vdots$ f(p_n) when $g_n \to$ $e^n_1, \dots, e^n_{l_n}$.

Figure 1: The extraction of a case expression

The transformation in terms of impact affects the complexity of the transformed function, and its modules. The number of functions, of rows, and of characters may increase in the module, but along with this decreases in the function. The transformation is local to the module.

It has a beneficial effect on the rates of embeddedness. As long as when applying, the extractions are kept at bay by limiting the number of functions, good results can be achieved regarding the *McCabe* number and the rate of embeddedness.

3.2. Movement of functions between modules

The moving of functions transformation transfers the selected functions to another module. Of course, these transformation steps (in compliance with pre- and well-defined rules) perform the necessary compensatory measures such as ensuring availability of related records, and macros, and managing or replacing the calls from the function. The transformation is complex, so the structural complexity levels are also markedly changed.

The original code

```
-module(movefun).
-export([f/1, g/1]).

 f(X)->
   X + 1.

 g(X) ->
   X -1.
```

→

The result code

```
-module(movefun).
-export([f/1]).

 f(X)->
   X + 1.

----movefun_new.erl-----

-module(movefun_new).
-export([g/1]).

 g(X)->
   X - 1.
```

Figure 2: Move function to another module

It has an impact on the participating function, but only in the event if it calls other functions or it contains qualified function calls. It affects the function's module, the functions and modules that are linked via function calls, and of course the module that is designated as the intended destination of the move. There is a change in the number of measured values of the relationships between modules, and the inbound and outbound function calls.

4. Transformation scripts

In this chapter, we present a language [19, 20, 21] suitable for automatic program transformations that we have developed and implemented to create scripts aimed to improve the code complexity measurements.

We present the syntax of the language, and also present the operation of the algorithm that was prepared to run it. We show the ways in which the quality of the program code can be improved based on the complexity metrics.

Based on measurements, and taking into account the predefined conditions the transformation language is suited to automatically convert source code stored in the semantic graph [12, 16] constructed from the program code and afterwards restore the code from the modified graph. By using the optimizer scripts and taking into account the changes in the complexity measures, the quality of the source code can be automatically transformed.

$$
\begin{aligned}
\textbf{Query} &\rightarrow \textbf{MetricQuery} \mid \textbf{OptQuery} \\
\textbf{OptQuery} &\rightarrow \textbf{Opti Where Limit} \\
\textbf{Opti} &\rightarrow \textbf{optimize Transformation} \\
\textbf{Transformation} &\rightarrow \mathit{TransformationName} \\
&\mid \mathit{TransformationName}\ \textbf{Params} \\
\textbf{Params} &\rightarrow \textbf{(}\mathit{Attr}\textbf{,}\ \mathit{ValueList}\textbf{)} \\
\textbf{Where} &\rightarrow \textbf{where Cond} \\
\textbf{Cond} &\rightarrow \mathit{Metric}\ \mathit{Rel}\ \mathit{CondValue} \\
&\mid \textbf{Cond}\ \mathit{LogCon}\ \textbf{Cond} \\
\textbf{Limit} &\rightarrow \textbf{limit}\ \mathit{Int}
\end{aligned}
$$

Figure 3: Language of the transformation scripts

In the specification of the syntax *Transformation* denotes a transformation (e.g. `move_fun`), *Rel* stands for a relation or other operator (e.g. $<, <=, >=, >, \mathit{like}$), *LogCon* denotes a logical operator (e.g. "and", "or"), the *CondValue* can be an integer or a designated lexical item (for example, using a modules name with *like*). *Int* in the *limit* section a can be substituted with a positive integer. (The non-terminal elements are with capital initials whereas the keywords of the language begin with small letters.)

In the *optimize* section the applied conversion's transformation steps and its parameters can be specified. The complex condition that can be defined after the *where* keyword, is the one that initiates the measurements, and controls the execution of transformations, namely under what conditions a given transformation step should be re-execute, and also which nodes of the semantic graph should be transformed.

Therefore the "basic condition" that can be specified, which is a logical expression, must contain at least one partial expression, which includes a measurable level of complexity on the modules or functions designated for transformation, an arithmetic operator, as well as a constant value.

The selected elements, which are the subject of the transformation, are not directly defined software constructions or expressions, but program slices that are selected automatically based on the given terms. With this method, the designation of the program parts that need to be transformed is transferred from the lexical level to the level of semantic analysis [11].

During the execution of the script written in the transformation language the analyzer searches for the program parts that fit the conditions, then performs the transformation given in the *optimize* section after which it measures the values of the complexity metrics specified in the criteria for all semantic graph node. The nodes that do not need to be included in further transformations based on the operator, and the constant, are drop out from the scope of the script. If there are no nodes on which the transformation must be re-executed, the script stops running.

Using the transformations we do not always reach the set objective, that is, by executing the script over and over again it always finds graph nodes awaiting another transformation (sometimes the script itself creates these with the application of the transformations).

Under these conditions, there may be cases when the execution does not terminate. To avoid this problem, the maximum number of executable iterations can be defined with the constant given after the *limit* keyword. Therefore if the transformation step does not produce the desired results, the constant of *limit* will definitely stop the execution after the given number of steps.

5. Measurement results

The measured software is the *Dialyzer*, that is part of the Erlang language; it is complex enough to produce results for each of the analyzed measures.

Overall, it consists 19 modules, and the modules contain 1023 functions in total. The number of function's branches is 1453. The most functions within a module is 163, and the highest number of function's branches in one module is 238.

The sum of the measured cyclomatic numbers on the modules was 3024, and with the same measurement the highest value for an individual module was 704, which is an outstanding result. (The source code will not be shared, since it is included in the Erlang distributions, and thus freely available.

The results apply to the release available at time of writing of the article). These figures make the software suitable to test the transformation algorithm on it. In the first experiment, we measured the number of functions, from a module and the *McCabe* number, and then we took the ratio of the two values:

$$\frac{mcCabe(src)}{number\_of\_fun(src)},$$

where $mcCabe(src)$ is *McCabe*'s cyclomatic number measured in the source code, *src* the measured source code, while *nof* is the number of all the functions in the module.

```
                     Mc Cabe number

optimize
   extract_function (exprtype, case_expr)
where
   f_mcCabe > 6
    and
   f_max_depth_of_structures > 2
    and
   f_max_depth_of_cases > 1
limit
   7;
```

Figure 4: The code quality improving script

The result is: $x_1 = \frac{704}{165} = 4.26666666667$. This value was taken as the base and we tried to improve it with the help of the script; that is we tried to improve the module's cyclomatic number in some way.

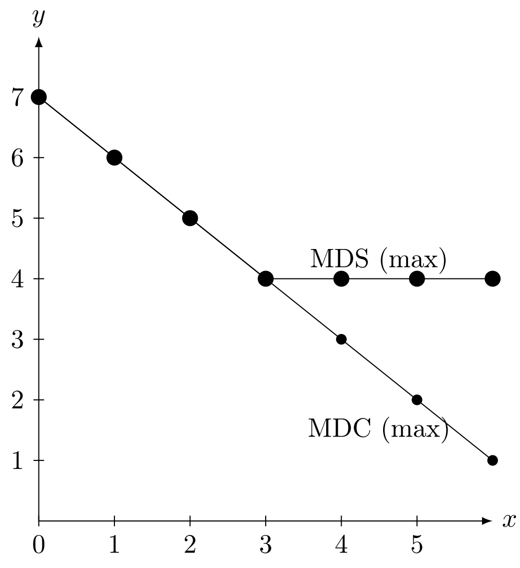

Figure 5: The maximum embeddedness of structures (MDS) and of *case* statements (MDC) (y-axis) during the transformation steps (x-axis)

We divided the cyclomatic number by the number of functions during the test, because the distorting effect that developed due to the increase in the number of functions had to be eliminated.

After running the script on the source code, whose exact task is to extract *case* expressions nested deeper than a given depth and insert in their place a call to functions generated by it, the following results were obtained:

$$x_2 = \frac{mcCabe(src')}{number\_of\_fun(src')} = \frac{794}{255} = 3,1137254901960785$$

$$\frac{x_2}{x_1} = 0.8473122889758293 = 84\% = 16\% \uparrow^{(limit=1)}$$

$$\frac{x_2}{x_1} = 0.729779411764706 = 72\% = 28\% \uparrow^{(limit=2)}$$

To obtain better results, we measured the maximum embeddedness levels of the *case* expressions in the module ($max\_depth\_of\_cases$). The measurement result indicated seven levels, that is the value that we should specify in the script's *limit* section, as this instructs the script to perform the extraction at least seven times.

$$x_3 = \frac{mcCabe(src'')}{number\_of\_fun(src'')} = \frac{868}{329} = 2.6382978723404$$

$$\frac{x_3}{x_1} = 0.6183510638297872 = 61\% = 39\% \uparrow^{(limit=7)}$$

By examining the new results we can draw some important conclusions. The first of which is that the measured values of the modules' cyclomatic number have increased because of the new functions.

Comparing the number of functions, and the cyclomatic number before and after the transformation, it is clear that $mcCabe(src) = mcCbae(src') - (nof(src) - nof(src'))$, so with the extractions the cyclomatic number of modules does not change, in the case the degree of embeddedness and the number of functions are not included in the calculated value.

This is so because the "decisions" from the expressions of the functions remain in the module, that is, whether or not a decision inside a function is extracted to a new function it does not disappear from the module (hence earlier we have divided the value by the number of functions).

In addition to the measured values of the modules we have to consider the cyclomatic number of each function in the module measured individually, as well as the maximum and the minimum from these values. If the changes of these results are compared with the values before and after the transformation, only then can we get a clear picture of the impact of the transformation. (Otherwise the average values of the module cannot be called accurate and the number of functions in the original module can greatly influence the results, as each new function adds at least one to this value...)

Analyzing the performed measurements we can see that the sum of cyclomatic numbers measured before the transformations is 704, and 794 after, if it is not divided by the number of functions; also prior to the transformation, the number of functions is 165, and 225 thereafter. Since $794 - 704 = 255 - 165$, it is clear that the newly created functions bring the increase in value.

In light of these we can make the suggestion that when measuring *McCabe*)'s cyclomatic number the values measured in the module should not be taken into account, but rather the highest reading obtained from the module's functions should be compared before the execution of the transformation, and thereafter.

$$max(mcCabe_f(src)) > max(mcCabe_f(src'))$$

We should consider the extent of the nestedness of different control structures, and so we should calculate according to the following formula, also we need to develop the appropriate transformation scripts based on this. Calculation of the result for the initial source code is as follows:

$$mcCabe(src) + sum(max\_dept\_of\_struct(src)) - z$$

$$+number\_of\_exceptions(src)$$

From the maximum of the embeddedness value the number of those functions where the degree of embeddedness is one (or the value we optimized the script to) can be subtracted as they also distort the value (in the formula this value is denoted by z). The $+(number\_of\_exceptions)$ section, which accounts for choices brought in by the exception handlers is optional, but if we use it for the initial condition, we cannot omit it from the calculation of the post-transformation state. (We would have even more accurate results if we would also included the branches of the exception handlers, that is the possible outcomes of exceptions, in the results. At this point, we have introduced a new measurement, but only in order to achieve better results. This metric returns the number of exception handlers located in programs in the module and function type nodes. To implement this measurement the function realising the measuring of the function expressions was converted so that it does not only return the number of expressions (fun_expr), but also the number of exception handlers(try_expr).

In the Erlang language, the exception handling *try* block can contain branches based on pattern matching that are customary for *case* control structures, also in a catch block the program's control can branch in multiple directions. So the solution does not find the decisions in the exception handlers, but rather it only returns the number of exception handlers, therefore it is not entirely accurate, but it is still convenient.) For the transformed text the result can be calculated with the following formula:

$$mcCabe(src') + sum(max\_dept\_of\_struct(src')) - z$$

$$+number\_of\_exceptions(src')$$

Thereafter from the measured maximum value of the functions and from the values calculated with the formula it can be decided with a high degree of certainty whether the result is better or not than the initial value. The calculation method already takes into account the depth of embeddedness, by increasing the cyclomatic number of a given function or module with each level.

Unfortunately this method together with the additional elements is still not perfect, because with regard to the measured values of the module it does not take into account the relationships between functions, and the resulting decision-making situations, which can be mapped to the call graph, but it is definitely better than the previous ones.

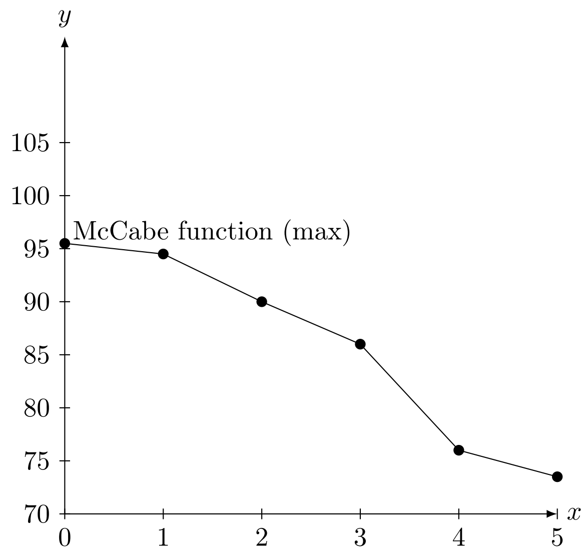

Figure 6: The maximum *McCabe* number of functions (y-axis) after each transformation step (x-axis)

In order to obtain more representative results than the previous one we had to analyze the complete source code of the Dialyzer software, with the source code scanning algorithm and then we transformed it. To perform the sequence of steps the previously used script was applied however, we took into account the proposed changes, so the embeddedness is added to the result, and the minimums and the maximums measured for the functions are also examined when the conclusions are deducted. The measured maximum value of the cyclomatic number of the functions before the transformation $max(mcCabe_f(src)) = 96$, and after restructuring $max(mcCabe_f(src')) = 73$, that is $max(mcCabe_f(src)) > max(mcCabe_f(src'))$.

The results show an improvement, but the script performs the extraction on all the function of each module that has an embeddedness greater than one. This embeddedness depth is not necessarily bad. As far as possible extractions should only be applied to areas where this is absolutely necessary, that is, in those modules in which the measured maximum cyclomatic numbers of the functions is high.

6. Conclusion

We introduced the language we have developed and the operation of the analysis algorithm. The language enables us to write automated program transformation scripts based on the measuring of complexity rates. In Chapter 2 we presented those structural complexity measures that were used to measure the complexity of Erlang source codes.

In Chapters 3 and 4 we examined the possibility of implementing an automated program for transformations based on the measurement and analysis of the complexity levels.

We defined the syntax of the language suitable for describing and executing the sequence of automated transformation steps based on software complexity measurements and described the operating principle of the analyzing and execution conducting algorithm that was constructed for the language.

In Chapter 5 using example programs, and their execution results we demonstrated the operability of automatic code quality improvement.

Beside the syntax and the descriptions of use cases we showed what results can be achieved by using a simple script made up of only a couple of lines.

In summary, the analyzing and the optimizing algorithm based on complexity measurements, which can be used to automatically or semi-automatically improve the source code of software written in Erlang language as well as previously published programs that are awaiting conversion, operated properly during the transformation of large-scale software.

The sequences of transformational steps improved the complexity rates which were designated for optimization. During the transformation the meaning of the source code did not change, and the program worked as expected following the re-translation.

In the following by using the results presented here we would like to test the parser and the transformational language constructed for it, on working client-server based software and programs from the industrial environment, for analyzing and also improving the quality of the source code. In addition we attempt to prove that following the execution of the transformation script, the modified source code's meaning conservation properties and correctness by using mathematical methods.

References

[1] McCabe T. J. A Complexity Measure, *IEE Trans. Software Engineering, SE-2(4), pp.308–320 (1976)*

[2] Frank Simon, Frank Steinbrückner, Claus Lewerentz *Metrics based refactoring* IEEE Computer Society Press 2001 30-38,

[3] Klaas van den Berg.: Software Measurement and Functional Programming, *PhD Thesis University of Twente (1995)*

[4] Ryder, C. Software Measurement for Functional Programming, *PhD thesis, Computing Lab, University of Kent, Canterbury, UK 2004)*

[5] Claus Lewerentz, Frank Simon *A Product Metrics Tool Integrated into a Software Development Environment* Object-Oriented Technology (ECOOP'98 Workshop Reader), LNCS 1543 Springer-Verlag 256–257

[6] Eclipse Foundation
http://www.eclipse.org/

[7] Zoltán Porkoláb, Ádám Sipos, Norbert Pataki, Structural Complexity Metrics on SDL Programs. *Computer Science, CSCS 2006, Volume of extended abstracts, (2006)*

[8] Ryder, C., Thompson, S. *Software Metrics: Measuring Haskell*, In Marko van Eekelen and Kevin Hammond, editors, Trends in Functional Programming (September 2005)

[9] Zoltán Porkoláb Programok Strukturális Bonyolultsági Méröszámai. *PhD thesis Dr Töke Pál, ELTE Hungary, (2002)*

[10] Zoltán Horváth, Zoltán Csörnyei, Roland Király, Róbert Kitlei, Tamás Kozsik, László Lövei, Tamás Nagy, Melinda Tóth, and Anikó Víg.: Use cases *for refactoring in Erlang, To appear in Lecture Notes in Computer Science, (2008)*

[11] Csörnyei Zoltán *Fordítóprogramok* Typotex Kiadó, Budapest, 2006. 3

[12] R. Kitlei, L. Lövei, M Tóth, Z. Horváth, T. Kozsik, T. Kozsik, R. Király, I. Bozó, Cs. Hoch, D. Horpácsi.: Automated Syntax Manipulation in RefactorErl. *14th International Erlang/OTP User Conference. Stockholm, (2008)*

[13] Lövei, L., Hoch, C., Köllö, H., Nagy, T., Nagyné-Víg, A., Kitlei, R., and Király, R.: Refactoring Module Structure *In 7th ACM SIGPLAN Erlang Workshop, (2008)*

[14] Lövei, L., Horváth, Z., Kozsik, T., Király, R., Víg, A., and Nagy, T.: Refactoring in Erlang, a Dynamic Functional Language *In Proceedings of the 1st Workshop on Refactoring Tools, pages 45–46, Berlin, Germany, extended abstract, poster (2007)*

[15] Erlang - Dynamic Functional Language
http://www.erlang.org

[16] T. Kozsik, Z. Horváth, L. Lövei, T. Nagy, Z. Csörnyei, A. Víg, R. Király, M. Tóth, R. Kitlei.. Refactoring Erlang programs. *CEFP'07, Kolozsvár (2007)*

[17] Thanassis Avgerinos, Konstantinos F. Sagonas *Cleaning up Erlang code is a dirty job but somebody's gotta do it.* Erlang Workshop 2009: 1–10

[18] Konstantinos F. Sagonas, Thanassis Avgerinos *Automatic refactoring of Erlang programs.* PPDP '09 Proceedings of the 11th ACM SIGPLAN conference on Principles and practice of declarative programming 2009: 13–24

[19] Király, R., Kitlei R.: *Complexity measurments for functional code* 8th Joint Conference on Mathematics and Computer Science (MaCS 2010) refereed, and the proceedings will have ISBN classification July 14–17, 2010

[20] Király, R., Kitlei R.: *Implementing structural complexity metrics in Erlang.* '10 ICAI 2010 – 8th International Conference on Applied Informatics to be held in Eger, Hungary January 27-30, 2010

[21] Roland Király, Róbert Kitlei: *Metrics based optimization of functional source code* a research paper in Annales Mathematicae et Informaticae 38 (2011) Pages: 59–74

Annales Mathematicae et Informaticae
42 (2013) pp. 45–55
http://ami.ektf.hu

Performance evaluation of wireless networks speed depending on the encryption

Tamás Krausz, János Sztrik

Faculty of Informatics, University of Debrecen
krausz.tamas@inf.unideb.hu, sztrik.janos@inf.unideb.hu

Submitted September 2, 2013 — Accepted November 28, 2013

Abstract

We can use a variety of encryption standards to encrypt data traffic to ensure the safety of wireless networks. The question is to what extent the security of the network affects network performance. For answering this question, experiments were performed without data encryption, and the use of various encryption standards. IEEE 802.11g and 802.11n wireless networking standards were used in the experiment. The answer of the question is that encryption should be used because it does not cause significantly slower speed.

Keywords: Wireless networking, security, encryption, WEP, WPA/TKIP, WPA2/AES

MSC: 68M20, 68M12, 94A60

1. Introduction

Wireless networks are increasingly exposed to the risk of unauthorized access. The reason for this is that the information runs instead of cable into the air. So it is enough to be in radio signal propagation range, and eavesdropping is easy (password and file contents can be stolen). You can use other internet subscriptions, and perform various illegal activities.

Avoiding illegal access to our network, we can encrypt the data flow. We can read about various wireless security tools in books [6, 7]. Wireless network security was examined in [2, 3, 5]. Paper [1] discovers the effects of the IEEE802.11i security specification on the performance of wireless networks. In [4], the throughput performance of IPv4 and IPv6 using UDP for wireless LAN networks with 802.11n and with and without security for two client-server networks were compared.

The question arises as to the security of wireless networks influences the speed of data transfer, that is, the network performance. To answer this question, experiments were performed without data encryption, and the use of various encryption standards.

At first, a wireless router was connected directly (USB 2.0) to hard disk and the file transfer speeds between client and disk were measured, than the file transfer speeds between two wireless clients were tested using a modern wireless router for home use.

The number of clients was increased for further examination of the network performance. In experiments, the number and type of clients were changing and the ftp service speed was measured in conjunction with encryption.

The following encryption standards were used in the experiments:

WEP (Wired Equivalent Privacy) is a security algorithm for IEEE 802.11 wireless networks. Obsolete, it is not safe in today's circumstances. Each 802.11 packet is encrypted separately with an RC4 cipher stream generated by a 64-bit RC4 key.

WPA/TKIP (Wi-Fi Protected Access, Wi-Fi Protected Access), which is similar to the WEP uses RC4 coder 128-bit key and 48-bit initialization vector, but this has been introduced in accessing the TKIP (Temporal Key Integrity Protocol, temporary secure key protocol), which continuously rotates keys used in the link.

WPA2/AES (Advanced Encryption Standard) uses a new coder instead of the old RC4.

2. The effect of encryption for the wireless network speed

During the experiments ca. 50 MB (50 298 448 bytes) transfer file was used.

2.1. First experiment

Copy to laptop from hard drive and back.

The laptop was placed close to the router, a SATA hard disk was connected to the router with USB port. We set up the router smb share. The wireless settings 2.4 GHz band and b / g / n mixed mode were used.

laptop 1: dell studio 1557 (Dell 1520 wireless N card, Core i720Qm, 8GB RAM, windows7 x64 operating system

router: TP-LINK WR2543ND wireless router (Atheros AR7242@400MHz CPU 64MB RAM)

The following speeds were measured:

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
copy to laptop (sec)	17,16	16,94	16,88	16,81	17,03	16,96
copy back to USB hdd (sec)	29,05	29,12	28,97	29,67	29,93	29,35
copy to laptop (MB/sec)	2,93114	2,96921	2,97977	2,99217	2,95352	2,96501
copy back to USB hdd (MB/sec)	1,73144	1,72728	1,73623	1,69526	1,68054	1,71386

Table 1: Without encryption

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
copy to laptop (sec)	27,33	25,94	26,18	25,77	26,84	26,412
copy back to USB hdd (sec)	38,06	38,74	38,11	37,92	38,55	38,276
copy to laptop (MB/sec)	1,84041	1,93903	1,92125	1,95182	1,87401	1,90438
copy back to USB hdd (MB/sec)	1,32156	1,29836	1,31982	1,32644	1,30476	1,3141

Table 2: WEP 64 bit encryption (no n)

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
copy to laptop (sec)	29,49	28,81	28,11	29,22	28,79	28,884
copy back to USB hdd (sec)	39,67	38,49	39,12	39,08	39,53	39,178
copy to laptop (MB/sec)	1,70561	1,74587	1,78934	1,72137	1,74708	1,74139
copy back to USB hdd (MB/sec)	1,26792	1,30679	1,28575	1,28706	1,27241	1,28384

Table 3: WPA/TKIP (no n)

	1. m.	2. m.	3. m.	4. m.	5. m.	average
copy to laptop (sec)	19,29	18,31	18,95	19,75	18,54	18,968
copy back to USB hdd (sec)	32,13	31,94	32,75	32,76	32,48	32,412
copy to laptop (MB/sec)	2,60749	2,74705	2,65427	2,54676	2,71297	2,65175
copy back to USB hdd (MB/sec)	1,56547	1,57478	1,53583	1,53536	1,5486	1,55185

Table 4: WPA2/AES

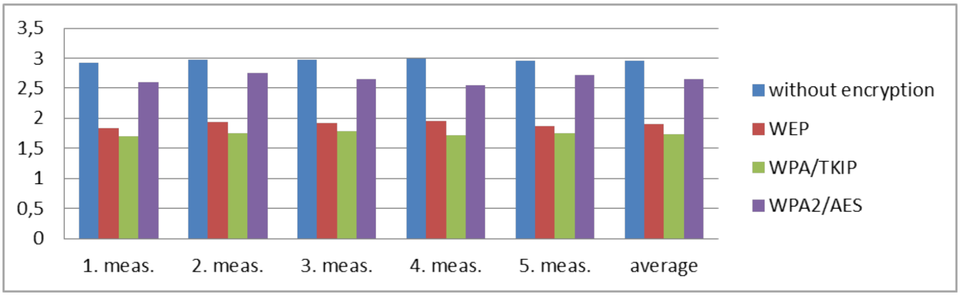

Figure 1: Copy to laptop (MB/sec)

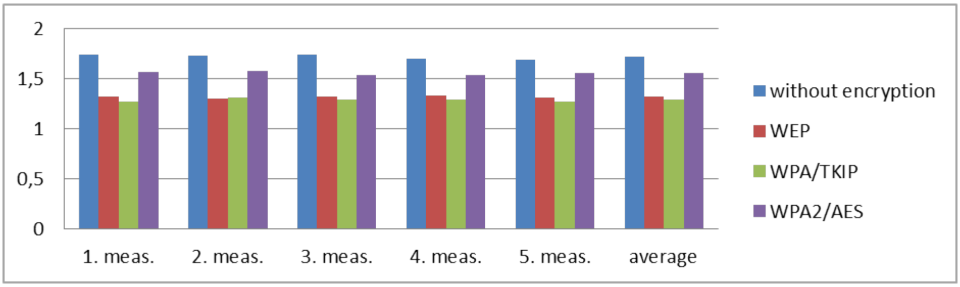

Figure 2: Copy back to USB (MB/sec)

3. Second experiment

In the second experiment, we copied the file between the two laptops using the TP-LINK WR2543ND wireless router.

laptop 1: dell studio 1557 (Dell 1520 wireless N card, Core i720Qm, 8GB RAM, window7 x64 operating system

laptop 2: fujisu amilo Pa1538 (TP-Link TL-W722N usb wireless card, AMD turion xl-50 processor 4GB RAM, windows 7 x64 operating system)

router: TP-link wr2543ND wireless router (Atheros AR7242@400MHz CPU 64MB RAM)

The following speeds were measured:

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
from laptop1 to laptop2 (sec)	13,87	14,05	14,69	14,13	14,54	14,256
from laptop2 to laptop1 (sec)	17,61	17,92	16,99	17,51	17,44	17,494
from laptop1 to laptop2 (MB/sec)	3,62642	3,57996	3,42399	3,55969	3,45932	3,52823
from laptop2 to laptop1 (MB/sec)	2,85624	2,80683	2,96047	2,87256	2,88409	2,87518

Table 5: Without encryption

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
from laptop1 to laptop2 (sec)	41,69	39,98	40,22	40,89	40,92	40,74
from laptop2 to laptop1 (sec)	39,5	39,88	40,13	39,64	40,02	39,834
from laptop1 to laptop2 (MB/sec)	1,20649	1,25809	1,25058	1,23009	1,22919	1,23462
from laptop2 to laptop1 (MB/sec)	1,27338	1,26124	1,25339	1,26888	1,25683	1,2627

Table 6: WEP

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
from laptop1 to laptop2 (sec)	46,07	45,16	45,54	45,93	46,12	45,764
from laptop2 to laptop1 (sec)	45,03	44,59	45,15	45,37	45,42	45,112
from laptop1 to laptop2 (MB/sec)	1,09178	1,11378	1,10449	1,09511	1,0906	1,09908
from laptop2 to laptop1 (MB/sec)	1,117	1,12802	1,11403	1,10863	1,10741	1,11497

Table 7: WPA/TKIP

	1. meas.	2. meas.	3. meas.	4. meas.	5. meas.	average
from laptop1 to laptop2 (sec)	15,87	16,17	16,43	16,01	16,23	16,142
from laptop2 to laptop1 (sec)	19,89	20,32	20,51	20,88	19,97	20,314
from laptop1 to laptop2 (MB/sec)	3,1694	3,1106	3,06138	3,14169	3,0991	3,116
from laptop2 to laptop1 (MB/sec)	2,52883	2,47532	2,45239	2,40893	2,5187	2,47605

Table 8: WPA2/AES

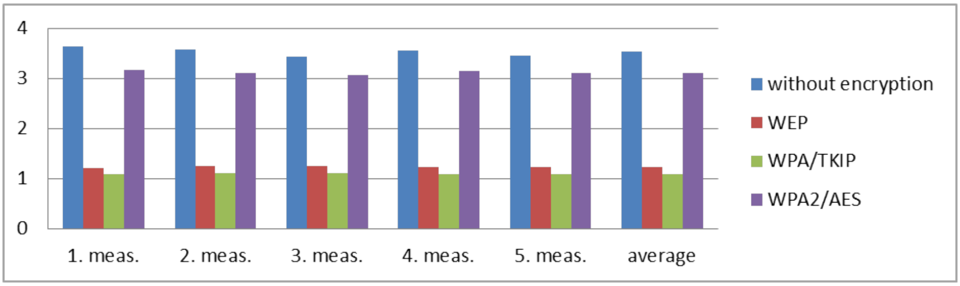

Figure 3: Copy from laptop1 to laptop2 (MB/sec)

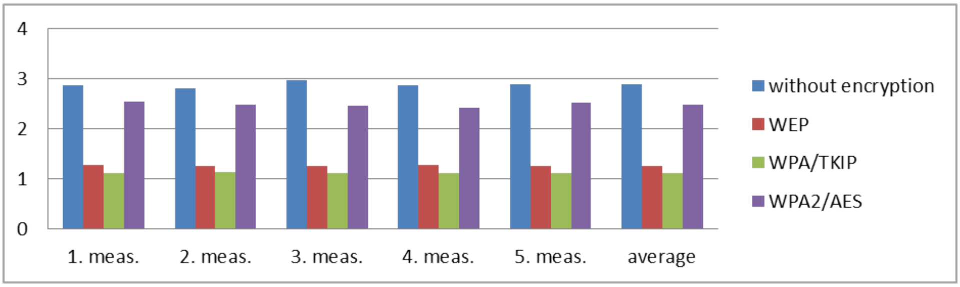

Figure 4: Copy from laptop2 to laptop1 (MB/sec)

3.1. Conclusions

Based on the first measurement, WPA2/AES causes slowdown of 10-30 percent, depending on the direction of the copy.

The 802.11n does not allow WEP and WPA/TKIP encryption, so the router will switch back to 802.11 g mode, so despite the weaker encryption much slower speeds are obtained. The WEP is no longer secure only marginally faster than the WPA/TKIP encryption.

On modern devices, WPA2/AES encryption should be used because it does not cause too significantly slower speed when transferring files.

In the second experiment, WPA2/AES encryption with the 802.11n causes 10-15 percent slowdown of copying in both directions. WPA/TKIP is 12-13 percent slower than WEP because the stronger encryption makes more load on the network card and the router.

4. FTP speed change depending on the number of clients and encryption

In these experiments, increasing the number of clients, we have examined the data traffic rate in the context of encryption. We have used TP-LINK WR2543nd router built-in FTP server to which USB 2.0 hard drive was connected. The transfer file was approximately 100 MB (100 769 606 bytes). The wireless router setting was 2.4 GHz band and b / g / n mixed mode.

During the measurements, the following devices were used:

laptop 1: Lenovo R500 (Atheros AR5006X wireless a/b/g card, Core2 Dou P8400 CPU, 4GB RAM Windows7 x64 operating system)

laptop 2: Dell studio 1557 (Dell 1520 wireless N card, Core i720Qm 8GB RAM, Windows7 x64 operating system)

desktop: Pentium dual core E6500 (TL-WN721N 150 MB usb wireless card, 4GB RAM, window8 x64 operating system)

router: TP-Link WR2543ND wireless router (Atheros AR7242@400MHz CPU 64MB RAM)

The following speeds were measured:

4.1. Download

	Lenovo	Dell	deskt	Dell + desktop			all three			
				Dell	deskt	avg	Lenovo	Dell	deskt	avg
transmission time (sec)	55	27	24	40	39	39,5	73	66	65	68,0
transmission rate (KB/sec)	1832	3732	4199	2519	2584	2551	1380	1527	1550	1482

Table 9: WPA2/AES encryption

	Lenovo	Dell	deskt	Dell + desktop			all three			
				Dell	deskt	avg	Lenovo	Dell	deskt	avg
transmission time (sec)	45	23	22	38	38	38,0	69	53	54	58,7
transmission rate(KB/sec)	2239	4381	4580	2652	2652	2652	1460	1901	1866	1718

Table 10: Download without encryptions

	Lenovo	Dell	deskt	Dell + desktop			all three			
				Dell	deskt	avg	Lenovo	Dell	deskt	avg
transmission time (sec)	48	63	57	80	79	79,5	119	120	119	119,3
transmission rate(KB/sec)	2099	1600	1768	1260	1276	1268	847	840	847	844

Table 11: WEP 64 bit download

4.2. Upload

	Lenovo	Dell	deskt	Dell + desktop			all three			
				Dell	deskt	avg	Lenovo	Dell	deskt	avg
transmission time (sec)	86	68	96	115	116	115,5	178	177	178	177,7
transmission rate(KB/sec)	1172	1482	1050	876	869	872	566	569	566	567

Table 12: WPA2/AES upload

	Lenovo	Dell	deskt	Dell + desktop			all three			
				Dell	deskt	avg	Lenovo	Dell	deskt	avg
transmission time (sec)	69	65	93	109	115	112,0	176	176	175	175,7
transmission rate(KB/sec)	1460	1550	1084	924	876	900	573	573	576	574

Table 13: No encryption upload

	Lenovo	Dell	deskt	Dell + desktop			all three			
				Dell	deskt	avg	Lenovo	Dell	deskt	avg
transmission time (sec)	73	78	57	125	112	118,5	184	184	180	182,7
transmission rate(KB/sec)	1380	1292	1768	806	900	850	548	548	560	552

Table 14: WEP 64 bit upload

4.3. Download speed rates

download speed	Lenovo	Dell	desktop	Dell + desktop average	all three average
WPA2/AES	1832	3732	4199	2551	1482
no encryption	2239	4381	4580	2652	1718
WEP 64 bit	2099	1600	1768	1268	844

Table 15: Download speed rates

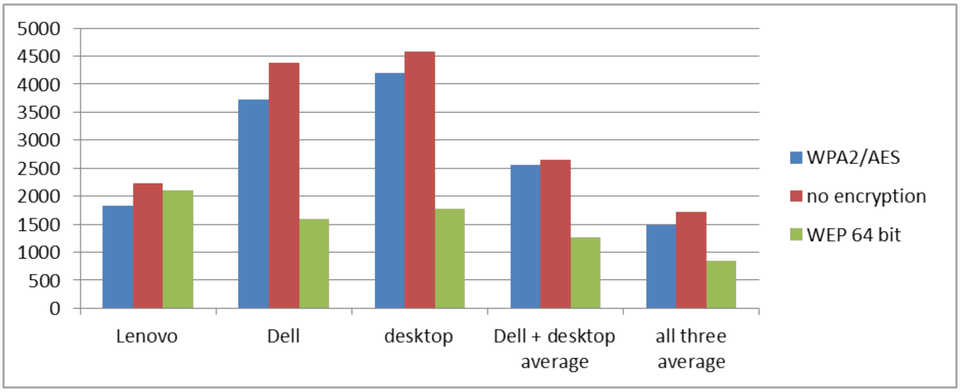

Figure 5: Download transfer speeds

Compared to the unencrypted case, the download speeds are slowed somewhat by increasing the number of clients at WPA2/AES case. The rate reduction of computers with 802.11n card is bigger if the computers are used alone compared to the case when we use them together. The three computers one-time download speed loss is similar to that of the single download.

4.4. Upload speed rates

In case of 802.11n there is no significant difference among the speed of type of encryption, because the upload speed is slow. Lenovo uses 802.11g speed in the upload. WPA2/AES is 24 percent slower than unencrypted. When all three computers upload simultaneously the speed was slow and therefore it did not significantly slow down.

upload speeds	Lenovo	Dell	desktop	Dell + desktop average	all three average
WPA2/AES upload	1172	1482	1050	872	567
no encryption upload	1460	1550	1084	900	574
WEP 64 bit upload	1380	1292	1072	850	552

Table 16: Upload speed rates

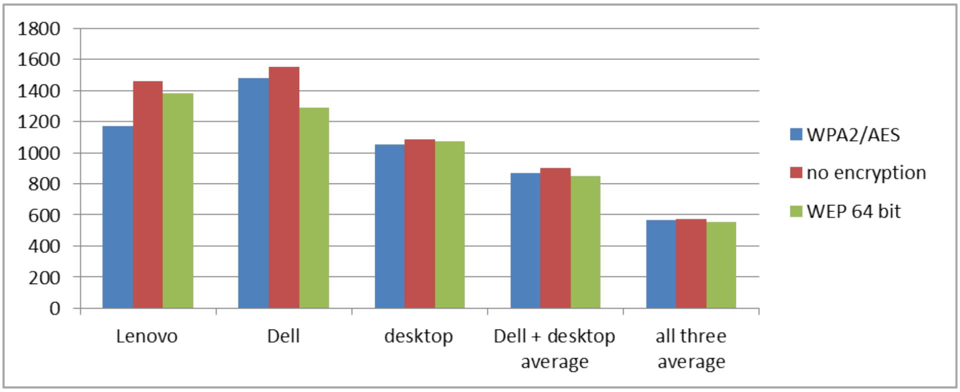

Figure 6: Upload transfer speeds

4.5. Conclusions

The WEP security is poor and 802.11n switches back to 802.11g, and therefore the speed is significantly reduced. The only exception from this is Lenovo, which originally used the 802.11g standard.

Using the FTP service when security matters, WPA2/AES encryption should always be used. If speed is more important than safety (such as anonymous FTP service), you can disable the encryption and speed of 10-20 per cent gain can be obtained.

5. Summary

We got similar result to paper [1] using more modern hardware and operating system with 802.11n wireless standard. The encryption and decryption takes time so that is the main cause of slowing down the traffic. (The packet size does not change significantly.)

In wireless networks where devices on the network are compatible and security matters, WPA2/AES encryption should always be used. The weaker encryptions switch back the more modern devices, on the older devices do not give a significantly better rate, but their security is worse. If speed is more important than safety (e.g., media playback with wireless), with disabling the encryption 10-30 percent speed gain can be obtained.

After these results we can raise the question what is more responsible for slowing down the transmission speed, either the encryption or the full bandwidth of the device.

Acknowledgment. Tamás Krausz was supported by the TÁMOP 4.2.2. C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

The work of János Sztrik was realized in the frames of TÁMOP 4.2.4. A/2-11-1-2012-0001 National Excellence Program – Elaborating and operating an inland student and researcher personal support system. The project was subsidized by the European Union and co-financed by the European Social Fund.

References

[1] Gin, R. Hunt, Performance Analysis of Evolving wireless IEEE 802.11 Security Architectures, The International Conference on Mobile Technology, Applications and Systems, ACM, Ilan, Taiwan, 2008.

[2] P. Georgopoulos, B. McCarthy, C. Edwards, *Providing Secure and Accountable Privacy to Roaming* 802.11 Mobile, ACM, MPM'12, Bern, 2012.

[3] Bruce Potter, Wireless Hotspots: Petri Dish of Wireless Security, *Communication of the ACM*, 2006, vol. 49, no.6, pp. 51–56.

[4] Samad S. Kolahi, Zhang Qu, Burjiz K. Soorty, and N. Chand, *The Performance of IPv4 and IPv6 using UDP on IEEE 802.11n WLANs with WPA2 Security*, ACM, 2009.

[5] T. Chenoweth, R. Minch, S. Tabor, Wireless Insecurity: Examining User Behavior on Public Networks, *Communication of the ACM*, 2010, vol. 53, pp. 134–138.

[6] Vivek Ramachandran, *BackTrack 5 Wireless Penetration Testing*, PACKT Publishing, 2011.

[7] Willie Pritchett, David De Smet, *BackTrack 5 Cookbooks, Networking & Telephony, Open Source*, PACKT Publishing, 2012.

Annales Mathematicae et Informaticae
42 *(2013) pp. 57–64*
http://ami.ektf.hu

On pseudoconformal models of fibrations determined by the algebra of antiquaternions and projectivization of them*

Irina Kuzmina[a], Josef Mikeš[b]

[a]Kazan Federal University, Russian Federation
iranina@mail.ru

[b]Palacky University Olomouc, Czech Republic
josef.mikes@upol.cz

Submitted December 4, 2012 — Accepted January 31, 2013

Abstract

In the present article we study the principal bundles determined by the algebra of antiquaternions in the projective model. The projectivizations of the pseudoconformal models of fibrations determined by the subalgebra of complex numbers is considered as example.

Keywords: pseudoconformal models, conformal mapping, pseudo-euclidean space, metric form

MSC: 53B20; 53B30; 53C21

1. Introduction

A. P. Norden developed the theory of normalization which appeared useful in applications to conformal, non-Euclidean and linear geometry [4]. By means of the normalization theory, A. P. Shirokov [8] succeeded to construct conformal models of non-Euclidean spaces. We show here basic steps of this construction.

*Supported by grant P201/11/0356 of The Czech Science Foundation

Let a real non-degenerate hyperquadric Q be given in the projective space $\mathbb{P}^{n+1}$. Let us choose a projective frame $(E_0, \dots, E_{n+1})$ such that E_{n+1} is the pole of the hyperplane $y^{n+1} = 0$, and the straight line E_nE_{n+1} intersects the hyperquadric Q in two real points N and N', and the points $E_0, \dots, E_{n-1}$ belong to the polar of the straight line E_nE_{n+1}.

Then the analytic expression of the hyperquadric Q reads

$$\mathbf{y}^2 = a_{pq}y^py^q + (y^n)^2 - (y^{n+1})^2 = 0, \tag{1.1}$$

where $p, q = 0, \dots, n-1$. The hyperquadric (1.1) intersects the hyperplane $y^{n+1} = 0$ in a hypersphere $\tilde{Q}$

$$a_{pq}y^py^q + (y^n)^2 = 0,$$

which can be either real or imaginary.

Let us construct the stereographic projection with the pole $N(0 : \dots : 0 : 1 : 1)$ of the hyperplane $\mathbb{P}^n : y^{n+1} = 0$ into the hyperquadric Q. If $U(y^0 : \dots : y^n : 0) \in \mathbb{P}^n$ take the straight line

$$\lambda U + \mu N = (\lambda y^0 : \dots : \lambda y^{n-1} : \lambda y^n + \mu : \mu);$$

coordinates of its intersection point with Q satisfy

$$\lambda^2 a_{pq}y^py^q + (\lambda y^n + \mu)^2 - \mu^2 = 0, \quad \lambda \neq 0.$$

Setting $k = \frac{\mu}{\lambda}$ we can write the previous equation as

$$a_{pq}y^py^q + (y^n)^2 + 2ky^n = 0.$$

If $y^n \neq 0$, i. e. the point $U \notin \mathbb{P}^{n-1}$, then

$$k = -\frac{a_{pq}y^py^q + (y^n)^2}{2y^n}.$$

Hence the intersection point of the straight line UN with the hyperquadric Q is uniquely determined. In the hyperplane $y^{n+1} = 0$, consider the $(n-1)$–plane $\mathbb{P}^{n-1}$: $y^n = 0$ as an ideal hyperplane; we obtain the structure of affine space $\mathbb{A}^n$ on the rest. In $\mathbb{A}^n$, we can introduce Cartesian coordinates $u^i = y^i/y^n$. Moreover, in $\mathbb{A}^n$ there exists the structure of Euclidean space $\mathbb{E}^n$ with the metric form

$$ds_0^2 = \pm a_{pq}du^pdu^q. \tag{1.2}$$

The point $U(u^0 : u^1 : \dots : u^{n-1} : 1 : 0)$ is mapped into the point

$$X_1(2u^0 : \dots : 2u^{n-1} : 1 - a_{pq}u^pu^q : -1 - a_{pq}u^pu^q).$$

Let us normalize the hyperquadric (1.1) self-polar, taking the lines of the sheaf of lines with a fixed center $Z = E_{n+1}$ as normals of the first-order, and their polar

$(n-1)$–planes belonging to the hyperplane $y^{n+1}=0$ as second-order normals. The straight line $E_{n+1}X_1$ intersects the hyperplane $y^{n+1}=0$ in the point

$$X(2u^0:\ldots:2u^{n-1}:1-a_{pq}u^pu^q:0).$$

Note that the polar of the point X related to the hyperquadric (1.1) intersects the hyperplane $y^{n+1}=0$ exactly in the $(n-1)$–dimensional second-order normal which corresponds to the first-order normal X_1E_{n+1}. Hence in the hyperplane $y^{n+1}=0$, a point X in general position is in correspondence with an $(n-1)$–plane, and the hyperplane $y^{n+1}=0$ appears to be a polary normalized projective space $\mathbb{P}^n$ with the same geometry as the quadric itself.

Let us define a second-order normal by basic points $Y_i=\partial_iX-l_iX$. We find the scalar product $(X,X)=(1+a_{pq}u^pu^q)^2$. The points X and Y_i are polar conjugate, i. e. the scalar product $(X,Y_i)=0$. From these conditions we calculate the normalizator l_i:

$$l_i=\frac{2a_{is}u^s}{1+a_{pq}u^pu^q}.$$

The decompositions

$$\partial_jY_i=l_jY_i+\Gamma^s_{ij}Y_s+p_{ij}X$$

determine components of the projective-Euclidean connection Γ^k_{ij} and the tensor p_{ij} [4]. Then the differential equations of the normalized space $\mathbb{P}^n: y^{n+1}=0$ read

$$\partial_iX=Y_i+l_iX,\qquad \nabla_jY_i=l_jY_i+p_{ij}X. \tag{1.3}$$

Covariant differentiation of the equation $(X,Y_i)=0$ gives

$$(\partial_jX,Y_i)+(X,\nabla_jY_i)=0.$$

By (1.3) we get

$$\begin{aligned}(X,\nabla_jY_i)&=-(\partial_jX,Y_i)=-(Y_j,Y_i)-l_j(X,Y_i)\\&=-(\partial_iX-l_iX,\partial_jX-l_jX)=-(\partial_iX,\partial_jX)-l_il_j(X,X).\end{aligned}$$

Therefore

$$p_{ij}=\frac{(X,\nabla_jY_i)}{(X,X)}=-\frac{(\partial_iX,\partial_jX)}{(X,X)}+l_il_j=-\frac{4a_{ij}}{(1+a_{pq}u^pu^q)^2}. \tag{1.4}$$

Hence considering in $\mathbb{A}^n$ the structure of the Euclidean space $\mathbb{E}^n$ with the Cartesian coordinates u^i we obtain a conformal model of a polar normalized projective space $\mathbb{P}^n$, i.e. a non-Euclidean space with the metric tensor

$$ds^2=g_{ij}du^idu^j=\frac{\pm a_{ij}du^idu^j}{(1+a_{pq}u^pu^q)^2}. \tag{1.5}$$

As we can see from (1.2) and (1.5), the obtained non-Euclidean space is conformally equivalent to the Euclidean space.

Quadrics of a special type in the projective spaces have been also studied in [1, 2].

2. On pseudoconformal models of fibrations determined by the algebra of antiquaternions and projectivization of them

Consider the associative unital 4-dimensional algebra $\mathbb{A}$ of antiquaternions [5, 6] with the basis $1, f, e, i$ and the multiplication table

	1	f	e	i
1	1	f	e	i
f	f	1	i	e
e	e	$-i$	1	$-f$
i	i	$-e$	f	-1

As well known, any antiquaternion can be uniquely expressed as $\mathbf{x} = x^0 + x^1 f + x^2 e + x^3 i$, conjugation is given by $\mathbf{x} \mapsto \bar{\mathbf{x}} = x^0 - x^1 f - x^2 e - x^3 i$, $\overline{\mathbf{xy}} = \bar{\mathbf{y}}\bar{\mathbf{x}}$ holds, the number $\mathbf{x}\bar{\mathbf{x}} = (x^0)^2 - (x^1)^2 - (x^2)^2 + (x^3)^2$ is real, and $\mathbf{x} \mapsto |\mathbf{x}| = \sqrt{\mathbf{x}\bar{\mathbf{x}}}$ defines a norm corresponding to the scalar product $\mathbf{xy} = \frac{1}{2}(\mathbf{x}\bar{\mathbf{y}} + \mathbf{y}\bar{\mathbf{x}})$ that turns $\mathbb{A}$ into the four-dimensional Pseudoeuclidean space $\mathbb{E}^4_2$. $|1| = |i| = 1$, $|e| = |f| = i$. For any $\mathbf{x}$ with $|\mathbf{x}| \neq 0$ there exists the inverse element $\mathbf{x}^{-1} = \frac{\bar{\mathbf{x}}}{|\mathbf{x}|^2}$. The set of all invertible elements from $\mathbb{A}$

$$\tilde{\mathbb{A}} = \{\mathbf{x} \mid |\mathbf{x}|^2 \neq 0\}$$

is a Lie group [7].

The group of antiquaternions of the unit norm $\mathbf{x}\bar{\mathbf{x}} = 1$ can be interpreted as the unit sphere $S^3_2(1)$

$$(x^0)^2 - (x^1)^2 - (x^2)^2 + (x^3)^2 = 1 \tag{2.1}$$

in the Pseudoeuclidean space $\mathbb{E}^4_2$.

We extend $\mathbb{E}^4_2$ into $\mathbb{P}^4$; taking

$$x^0 = \frac{y^0}{y^4},\ x^1 = \frac{y^1}{y^4},\ x^2 = \frac{y^2}{y^4},\ x^3 = \frac{y^3}{y^4},$$

we introduce homogeneous coordinates $(y^0 : y^1 : y^2 : y^3 : y^4)$. The quadric $S^3_2(1)$ has coordinate expression

$$\mathbf{y}^2 = (y^0)^2 - (y^1)^2 - (y^2)^2 + (y^3)^2 - (y^4)^2 = 0. \tag{2.2}$$

The quadric (2.2) intersects the hyperplane $y^0 = 0$ in the sphere S^2_1

$$(y^1)^2 + (y^2)^2 - (y^3)^2 + (y^4)^2 = 0.$$

The point E_0 of the projective frame $(E_0, \dots, E_4)$ is the pole of the hyperplane $y^0 = 0$, the straight line E_0E_4 intersects the quadric in two real points $N(1 : 0 : 0 :$

$0 : 1)$ and $N'(-1 : 0 : 0 : 0 : 1)$, and the points E_1, E_2, E_3 belong to the polar $\mathbb{P}^2$ of the straight line E_0E_4.

The tangent plane at the point N has the equation $y^0 - y^4 = 0$ and intersects the sphere in the real cone $-(y^1)^2 - (y^2)^2 + (y^3)^2 = 0$. Also it intersects $\mathbb{P}^3$ in the 2-plane $\mathbb{P}^2$: $y^0 = 0$, $y^4 = 0$. Hence in the hyperplane $\mathbb{P}^3$ there is a structure of affine space $\mathbb{A}^3$ for which $\mathbb{P}^2$ is the improper plane. Consequently, under the assumption $y^4 \neq 0$ we can introduce Cartesian coordinates

$$u^i = \frac{y^i}{y^4}, \ i = 1, 2, 3.$$

The sphere S_1^2 determines in $\mathbb{A}^3$ the structure of Pseudoeuclidean space $\mathbb{E}_1^3$ with the metric form

$$ds_0^2 = -(du^1)^2 - (du^2)^2 + (du^3)^2. \tag{2.3}$$

Consider the stereographic projection of the hyperplane $y^0 = 0$ from the pole $N(1 : 0 : 0 : 0 : 1)$ onto the quadric (2.2). The point $U(0 : u^1 : u^2 : u^3 : 1)$ is mapped into the point

$$X_1(-1 + r^2 : 2u^1 : 2u^2 : 2u^3 : 1 + r^2),$$

where $r^2 = -(u^1)^2 - (u^2)^2 + (u^3)^2$ is the square of distance of the point U from the origin of the Pseudoeuclidean metric of the space $\mathbb{E}_1^3$.

Let us normalize the quadric (2.2) self-polar, taking as the first-order normals straight lines passing through E_0, and as second-order normals their polar two-planes belonging to the hyperplane $y^0 = 0$. The straight line E_0X_1 intersects the hyperplane $y^0 = 0$ in the point

$$X(0 : 2u^1 : 2u^2 : 2u^3 : 1 + r^2).$$

The polar of the point X related to the quadric (2.2) intersects the hyperplane $y^0 = 0$ in the normal of the second order. Hence in the hyperplane $y^0 = 0$, a point X in general position corresponds to a two-plane, and the hyperplane $y^0 = 0$ is the normalized projective space $\mathbb{P}^3$.

Let us define the second-order normal by basic points $Y_i = \partial_i X - l_i X$. Points X and Y_i are polar conjugate, i. e. $(X, Y_i) = 0$. From this condition and since $(X, X) = -(r^2 - 1)^2$ we find coordinates of the normalizer:

$$l_1 = -\frac{2u^1}{r^2 - 1}, \quad l_2 = -\frac{2u^2}{r^2 - 1}, \quad l_3 = \frac{2u^3}{r^2 - 1}.$$

Then by (1.4) we obtain finally

$$p_{11} = p_{22} = -\frac{4}{(r^2 - 1)^2}, \quad p_{33} = \frac{4}{(r^2 - 1)^2}.$$

Now introducing in $\mathbb{A}^3$ the structure of Pseudoeuclidean space $\mathbb{E}_1^3$ with u^i as Cartesian coordinates we find the pseudoconformal model of the sphere $S_2^3(1)$ with the metric form

$$ds^2 = g_{ij}du^i du^j = \frac{-(du^1)^2 - (du^2)^2 + (du^3)^2}{(r^2 - 1)^2}. \tag{2.4}$$

The corresponding Riemannian (Levi-Civita) connection of this Pseudoriemannian metric form appears. Non-vanishing components (Christoffel symbols) of it are

$$\Gamma^1_{11} = -\Gamma^1_{22} = \Gamma^1_{33} = \Gamma^2_{21} = \Gamma^3_{13} = \frac{2u^1}{r^2-1},$$
$$\Gamma^2_{22} = \Gamma^1_{12} = -\Gamma^2_{11} = \Gamma^2_{33} = \Gamma^3_{32} = \frac{2u^2}{r^2-1},$$
$$-\Gamma^3_{33} = -\Gamma^3_{22} = -\Gamma^1_{13} = -\Gamma^3_{11} = -\Gamma^2_{23} = \frac{2u^3}{r^2-1}.$$

The connection is of constant curvature $K = -1$.

As an example, we obtain equations of fibres in model of the fibration defined by the subalgebra of complex numbers.

3. Example

Let us write an antiquaternion in the form

$$\mathbf{x} = x^0 + x^3 i + f(x^1 + x^2 i) = z_1 + f z_2, \quad z_1, z_2 \in \mathbb{R}(i),$$

where $\mathbb{R}(i)$ is a 2–dimensional subalgebra of complex numbers with basis $\{1, i\}$. The set of its invertible elements

$$\tilde{\mathbb{R}}(i) = \{\lambda = a + bi \mid \lambda \neq 0\}, \quad a,\, b \in \mathbb{R}$$

turns out to be a Lie subgroup of the group $\tilde{\mathbb{A}}$, a 2–plane with exception of one point.

The canonical projection $\pi : \tilde{\mathbb{A}} \to \tilde{\mathbb{A}}/\tilde{\mathbb{R}}(i)$ takes the form

$$\pi(\mathbf{x}) = (\bar{z}_1 : z_2).$$

The factorspace $\tilde{\mathbb{A}}/\tilde{\mathbb{R}}(i)$ is a subset M of a complex projective line $P(i)$ and

$$M = \{[z_1 : z_2] \in P(i) \mid z_1\bar{z}_1 - z_2\bar{z}_2 \neq 0\}.$$

It is covered by two charts

$$U_1 = \{[z_1 : z_2] \mid z_2 \neq 0\} \quad \text{with the coordinate} \quad z = \frac{\bar{z}_1}{z_2}\,,$$

where $|z|^2 \neq 1$, since $z_1\bar{z}_1 - z_2\bar{z}_2 \neq 0$;

$$U_2 = \{[z_1 : z_2] \mid z_1 \neq 0\} \quad \text{with the coordinate} \quad \tilde{z} = \frac{z_2}{\bar{z}_1}\,,$$

where $|\tilde{z}|^2 \neq 1$ by the same reason.

Let the point $z = u + iv \in M \subset P(i)$ is in U_1. Then the coordinate expression of the projection π in real coordinates is

$$\pi(z_1, z_2) = z = \left(\frac{x^0x^1 - x^2x^3}{(x^1)^2 + (x^2)^2}, \frac{-(x^0x^2 + x^1x^3)}{(x^1)^2 + (x^2)^2} \right). \tag{3.1}$$

Then $z = \frac{\bar{z}_1}{z_2}$, where in homogeneous coordinates

$$z_1 = \frac{y^0 + y^3 i}{y^4}, \quad z_2 = \frac{y^1 + y^2 i}{y^4}.$$

The projection $\pi(\mathbf{y}) = z$ can be written as

$$\pi(\mathbf{y}) = \left(\frac{y^0y^1 - y^2y^3}{(y^1)^2 + (y^2)^2}, \frac{-(y^0y^2 + y^1y^3)}{(y^1)^2 + (y^2)^2} \right),$$

which is equivalent to (3.1), and 2–planes $L_2 : \bar{z}_1 - zz_2 = 0$ are given by a system of equations

$$\begin{cases} y^0 - uy^1 + vy^2 = 0, \\ y^3 + vy^1 + uy^2 = 0. \end{cases} \tag{3.2}$$

These 2–planes are the fibres of this fibration. By intersection with the sphere (2.2), we obtain a 2–parameter family of second order curves

$$\begin{cases} (y^0)^2 - (y^1)^2 - (y^2)^2 + (y^3)^2 - (y^4)^2 = 0, \\ y^0 - uy^1 + vy^2 = 0, \\ y^3 + vy^1 + uy^2 = 0, \end{cases}$$

which define the fibration. Excluding y^0 we find the projection of the family of fibres into the space $\mathbb{E}^3_1$. Passing to the Cartesian coordinates we obtain

$$\begin{cases} -(x^1)^2 - (x^2)^2 + (x^3)^2 + (ux^1 - vx^2)^2 = 1, \\ x^3 + vx^1 + ux^2 = 0. \end{cases} \tag{3.3}$$

There is a correspondence of these equations with the equations (21) ([3, p. 89]). If $\mathbf{y}$ is a point on the quadric distinct from N (i.e. $y^0 - y^4 \neq 0$ holds), the corresponding point ξ in $\mathbb{E}^3$: $y^0 = 0$ is uniquely determined by the homogeneous coordinates $(0 : y^1 : y^2 : y^3 : y^4 - y^0)$, that is

$$\xi(0 : \frac{y^1}{y^4 - y^0} : \frac{y^2}{y^4 - y^0} : \frac{y^3}{y^4 - y^0} : 1),$$

and in the space $\mathbb{A}^3$: $y^4 \neq 0$ the point ξ has the Cartesian coordinates

$$u^1 = \frac{x^1}{1 - x^0}, \quad u^2 = \frac{x^2}{1 - x^0}, \quad u^3 = \frac{x^3}{1 - x^0}.$$

The inverse mapping is characterized by the formulas

$$x^0 = \frac{\xi^2 - 1}{\xi^2 + 1},\ x^1 = \frac{2u^1}{\xi^2 + 1},\ x^2 = \frac{2u^2}{\xi^2 + 1},\ x^3 = \frac{2u^3}{\xi^2 + 1},$$

$$\xi^2 = -(u^1)^2 - (u^2)^2 + (u^3)^2,\ \xi^2 + 1 \neq 0,$$

similar to the formulas (18) (cf. [3, p. 88]). Hence the coordinates of the points $\mathbf{y}$ and ξ are related by the conformal mapping. Substituting these expressions into (3.3) we obtain the equations of the family of fibres in the form

$$\begin{cases} (u^1)^2 + (u^2)^2 - (u^3)^2 + 2(uu^1 - vu^2)^2 + 1 = 0, \\ vu^1 + uu^2 + u^3 = 0. \end{cases} \tag{3.4}$$

These equations coincide with the system (21) (cf. [3, p. 89]). So, we have the following result.

Theorem 3.1. *In the projective model the equations of fibres of the fibration defined by the subalgebra of complex numbers are* (3.4)*.*

References

[1] Jukl, M., On homologies of Klingenberg projective spaces over special commutative local rings, *Publ. Math. Univ. Debrec.*, 55 (1999) 113–121.

[2] Jukl, M., Snášel, V., Projective equivalence of quadrics in Klingenberg projective spaces over a special local ring, *Int. Electr. J. Geom.*, 2 (2009) 34–38.

[3] Kuzmina, I. A., Shapukov, B. N., Conformal and elliptic models of the Hopf fibration, *Tr. Geom. Sem. Kazan. Univ.*, 24 (2003) 81–98.

[4] Norden, A. P., Spaces of Affine Connection, *Nauka, Moscow*, (1976).

[5] Rozenfeld, B. A., Higher-dimensional Spaces, *Nauka, Moscow*, (1966).

[6] Rozenfeld, B. A., Geometry of Lie Groups, *Kluwer, Dordrecht–Boston–London*, (1997).

[7] Shapukov, B. N., Connections on a differential fibred bundle, *Tr. Geom. Sem. Kazan. Univ.*, 12 (1980) 97–109.

[8] Shirokov, A. P., Non-Euclidean Spaces, *Kazan University, Kazan*, (1997).

Annales Mathematicae et Informaticae
42 (2013) pp. 65–70
http://ami.ektf.hu

Method for computing angle constrained isoptic curves for surfaces*

Ferenc Nagy, Roland Kunkli

Faculty of Informatics, University of Debrecen, Hungary
nferenc13@gmail.com
kunkli.roland@inf.unideb.hu

Submitted October 10, 2013 — Accepted December 13, 2013

Abstract

In computer graphics and geometric modeling for good quality displaying of an object it is required that the object must fit on the screen. It often happens, for example when we are using modeling software, that the object we would like to rotate around an axis, or edit from another point of view, is partly out of the screen, and thus some parts are not visible. In two dimensions the isoptic curve of a curve is constructed by involving lines with a given angle intersect each other at a certain point of the isoptic curve. In three dimensions the points of an isoptic curve may be the admissible positions of the camera. So from these points we can watch the object with respect to the given viewing angle. The purpose of this paper is to find a general method and computational algorithm that helps to locate the closest possible position of the camera, which positions form a closed curve around the surface. The developed algorithm produces this curve for a special case.

Keywords: isoptic curve, Bézier surface

MSC: 65D17, 68U07

1. Introduction

Let us overview briefly the planar case of the isoptic problem.

*This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4.A/2-11-1-2012-0001 'National Excellence Program'.

Definition 1.1. For a given curve C, consider the locus of points from where the tangents to C meet at a fixed given angle.

The isoptic of quadratic curves can be determined from the definition by calculation based on elementary geometric results [5] and several further results are also known, mainly about classical curves [6, 7]. For freeform curves, the intersections of the appropriate tangents of the given curve determine the points of the isoptic curve. In this case we need to derivate the given curve to find the involving lines. With this algorithm we can find the set of points which form the isoptic curve of convex curves (see Figure 1).

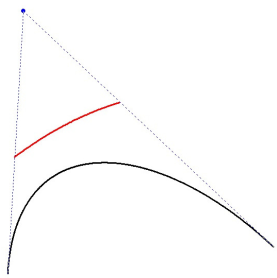

Figure 1: The isoptic curve with 90° (red one) of a Bézier curve

In case of Bézier curves there is a calculation for obtaining an exact formula of the isoptic curve based on [1]. This method defines the isoptic as the envelope of envelopes of families of isoptic circles over the chords of the Bézier curve. Although it cannot always be resolved exactly, the numerical algorithms can provide sufficient results as well.

In this paper we provide a way of generalization of isoptics in 3D and compute a curve around a convex surface from which the surface can be seen under a given angle.

2. Isoptic of surfaces

The first difficulty is that the generalization of the two-dimensional definition is not straightforward and not unique in the three dimensional space. It is not evident how to define the angle of view in 3D. One possibility is looking for points, from where we can draw tangent lines to the surface, which intersect each other at the given angle, but this computation is not uniquely determined. So we are looking for points in a special case for Bézier surfaces.

Figure 2 shows the special circumstances. At first, we suppose that the surface is convex and it is entirely above (but as close as possible to) the base plane which is given by the equation $Y = 0$ in the coordinate system. Moreover, we suppose that the origin of the coordinate system is in the orthogonal projection of the surface onto the base plane. We will search for points of this plane around the surface, from which the viewing angle, that is the angle of the plane $Y = 0$ and the deepest tangent line to the surface from this point is a predefined angle. We will call these points isoptic points of the surface. To locate an isoptic point we use two vectors: **m** and **v**. The vector **m** is parallel to the X-axis; the vector **v** is rotated around the X-axis. This angle of the rotation needs to be selected as follows. If we look from one isoptic point to the origin, the surface should fit on the viewing screen on the top. For all points it is true that vector **v** and one of the normal vectors of the surface are orthogonal. It is also true for the vector **m**. There are two ways to do the scan: by rotating the surface and by rotating vectors **m** and **v** around the Y-axis. With this condition, we can find several points on the plane $Y = 0$, and based on these points we can produce a curve around the surface (see Figure 3) interpolating these points by a closed B-spline curve [2]. In the next subsections details of the two computational approach are provided.

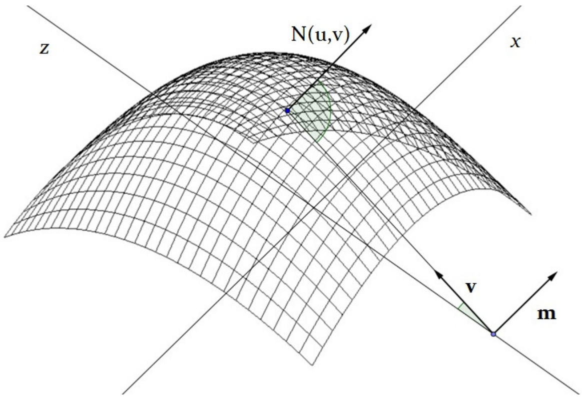

Figure 2: The special case for finding isoptic points of Bézier surface

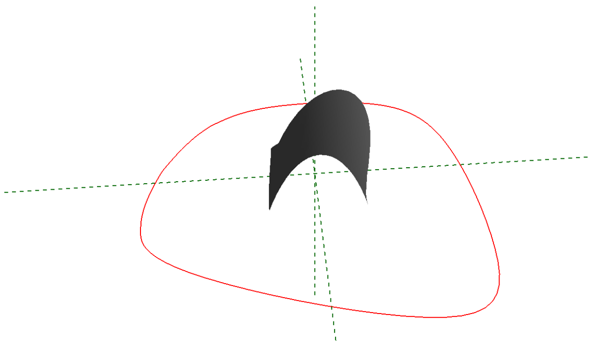

Figure 3: Result of the algorithm

2.1. Rotating the control points of the surface

In this algorithm we have to rotate the control points of the Bézier surface, which is given by the following equation:

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} B_i^n(u)B_j^n(v)P_{i,j}.$$

We need to compute the normal vector by the partial derivatives of the surface, with respect to the parameters u and v: $U(u,v) = \frac{\partial S(u,v)}{\partial u}$ and $V(u,v) = \frac{\partial S(u,v)}{\partial v}$. The normal vector will be the cross product of the two partial derivatives: $N(u,v) = U(u,v){\times}V(u,v)$.

We can write the equation of a plane that touches the Bézier surface in the point from where the normal vector perpendicular to vectors $\mathbf{m}$ and $\mathbf{v}$:

$$N_x(u,v)(X - X_0) + N_z(u,v)(Z - Z_0) = N_y(u,v)Y_0,$$

where X_0, Y_0, Z_0 are the coordinates of the points of the Bézier surface. The isoptic point is lying on the line in which this plane and the base plane given by the equation $Y = 0$ intersect each other. By solving this equation with $X = 0$, we obtain the distance from the origin. This will be the Z coordinate of the isooptic point:

$$Z = \frac{N_y(u,v)Y_0 - N_x(u,v)X_0}{N_z(u,v)} + Z_0.$$

The X and Y coordinates of this point are equal to zero. We need to rotate this point with the same angle as we used to rotate the surface. While we rotate the control points around the Y-axis the following conditions need to be satisfied: $\langle N(u,v), \mathbf{m}\rangle = 0$ and $\langle N(u,v), \mathbf{v}\rangle = 0$, that means the following:

$$N_x(u,v) = 0$$
$$\sin(\alpha){\cdot}N_y(u,v) = \cos(\alpha){\cdot}N_z(u,v).$$

If we can solve this equation for u and v we obtain an exact formula to compute the normal vector and from this we can calculate exactly the position of each isoptic point. Unfortunately the solution cannot be given in closed form in most of the cases, but numerical methods work sufficiently.

2.2. Rotating vectors m and v

The other way to find the isoptic points is to rotate the vector m and v around the Y-axis. The coordinates of the vectors will be the following:

$$\mathbf{m} = (\cos(\beta), 0, -\sin(\beta)) \quad \text{and} \quad \mathbf{v} = (-\sin(\beta){\cdot}\cos(\alpha), \sin(\alpha), -\cos(\beta){\cdot}\cos(\alpha)),$$

where β is the angle of rotation around the Y-axis, in the interval $[0, 2\pi]$. The condition of the search is that $\langle N(u,v), \mathbf{m}\rangle = 0$ and $\langle N(u,v), \mathbf{v}\rangle = 0$ have to be fulfilled. From these we obtain the following:

$$N_x(u,v) = \tan\beta{\cdot}N_Z(u,v)$$

$$N_y(u,v) = \cot\alpha \cdot \frac{1}{\cos\beta} \cdot N_z(u,v).$$

We also tried to resolve this equation for u and v, but it cannot be obtained in closed form as well. But there is no need to rotate the isoptic points so we can compute exactly the X and Y cordinates of these points.

Let e be the intersection line of the plane that touches the surface and the base plane given by the equation $Y = 0$, and let f be the line which has one point in the origin and its normal vector m. The isoptic point is the intersection of these lines:

$$X = \frac{\sin(\beta) \cdot c}{\cos(\beta) \cdot N_z(u,v) - \sin(\beta) \cdot N_x(u,v)}$$

$$Z = \frac{\cos(\beta) \cdot c}{\cos(\beta) \cdot N_z(u,v) - \sin(\beta) \cdot N_x(u,v)},$$

where $c = N_x(u,v)X_0 + N_y(u,v)Y_0 + N_z(u,v)Z_0$. Since $\langle N(u,v), \mathbf{m}\rangle = 0$, we can compute the coordinates by the following equations:

$$X = \frac{cN_x(u,v)}{\langle N_x(u,v), N_x(u,v)\rangle + \langle N_z(u,v), N_z(u,v)\rangle}$$

$$Z = \frac{cN_z(u,v)}{\langle N_x(u,v), N_x(u,v)\rangle + \langle N_z(u,v), N_z(u,v)\rangle}.$$

2.3. Calculation by linear combination

There is another way to compute the isoptic points. To find these points we still need to use the vectors $\mathbf{m}$ and $\mathbf{v}$. If the normal vector of the surface is the cross product of the vector $U(u,v)$ and $V(u,v)$, and $N(u,v)$ is perpendicular to the vector $\mathbf{m}$ and $\mathbf{v}$, then $\mathbf{v}$, $\mathbf{m}$, $U(u,v)$ and $\mathbf{v}$, $\mathbf{m}$, $V(u,v)$ are linearly dependent. Thus $U(u,v)$ and $V(u,v)$ can be obtained by linear combination of vectors $\mathbf{m}$ and $\mathbf{v}$. This means the following:

$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & \sin(\alpha) & -\cos(\alpha) \\ U_x(u,v) & U_y(u,v) & U_z(u,v) \end{vmatrix} = 0,$$

$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & \sin(\alpha) & -\cos(\alpha) \\ V_x(u,v) & V_y(u,v) & V_z(u,v) \end{vmatrix} = 0.$$

From this we obtain the following equations:

$$\sin(\alpha) \cdot U_z(u,v) + \cos(\alpha) \cdot U_y(u,v) = 0$$
$$\sin(\alpha) \cdot V_z(u,v) + \cos(\alpha) \cdot V_y(u,v) = 0.$$

By solving this system we acquire a complex fifth order equation for u and v, but it is not possible approximate the roots even by computer algebra systems.

3. Conclusions

The possibilities of 3 dimensional generalization of isoptic curves are considered. We provided a special scene where isoptic points of a surface can be computed, although in some cases only by numerical methods. This method can be applied in circumstances when a convex surface is above a given plane and the method provides a curve in this plane from the point of which one can see the surface under a given angle. Further investigations in terms of computational efficiency and generalization can be subject of future work.

References

[1] Kunkli, R., Papp, I., Hoffmann, M.: Isoptics of Bézier curves, *Computer Aided Geometric Design*, 30 (2013), 78–84.

[2] Les A. Piegl, Wayne Tiller: The Nurbs book 2nd Edition, 2012.

[3] Qingxiang Fu: The intersection of a bicubic Bézier patch and a plane, *Computer Aided Geometric Design*, 7 (1990), 475–488.

[4] Hoffmann, M., Juhász, I.: On interpolation by spline curves with shape parameters, *Lecture Notes in Computer Science*, 4975 (2008), 205–215.

[5] Loria, G.: Spezielle algebraische und transzendente ebene Kurven: Theorie und Geschichte, B.G. Teubner, Leipzig-Berlin, 1911.

[6] Cieślak, W., Miernowski A. and Mozgawa M.: Isoptics of a closed strictly convex curve, *Lecture Notes in Mathematics*, 1481 (1991), 28–35.

[7] Odehnal, B.: Equioptic curves of conic sections, *J. Geom. Graphics* 14 (2010), 29–43.

Annales Mathematicae et Informaticae
42 (2013) pp. 71–82
http://ami.ektf.hu

Note on formal contexts of generalized one-sided concept lattices*

Jana Pócsová

Technical University of Košice, BERG Faculty
Institute of Control and Informatization of Production Processes
Košice, Slovakia
jana.pocsova@tuke.sk

Submitted June 6, 2013 — Accepted August 21, 2013

Abstract

Generalized one-sided concept lattices represent one of the conceptual data mining methods, suitable for an analysis of object-attribute models with the different types of attributes. It allows to create FCA-based output in form of concept lattice with the same interpretation of concept hierarchy as in the case of classical FCA. The main aim of this paper is to investigate relationship between formal contexts and generalized one-sided concept lattices. We show that each one uniquely determines the other one and we also derive the number of generalized one-sided concept lattices defined within the given framework of formal context. The order structure of all mappings involved in some Galois connections between a power set and a direct product of complete lattices is also dealt with.

Keywords: Galois connection, generalized one-sided concept lattice, formal context.

MSC: 06A15

1. Introduction

Handling uncertainty, imprecise data or incomplete information has become an important research topic in the recent years. One of the frequent solutions, how to

*This work was supported by the Slovak Research and Development Agency under contract APVV–0482–11 and by the Slovak VEGA Grants 1/0729/12 and 1/0497/11.

deal with "imperfect" information, usually leads to the development of the fuzzified versions of several well-known standard structures or approaches. In this paper, we focus on the area of the formal concept analysis, specifically, on the approach known as generalized one-sided concept lattices.

Formal Concept Analysis (FCA [9]) represents a method of data analysis for identifying conceptual structures among data sets. As an efficient tool, Formal Concept Analysis has been successfully applied to domains such as decision systems, information retrieval, data mining and knowledge discovery. Classical FCA is suitable for crisp case, where object-attribute model is based on binary relation (object has/has-not the attribute). In practice there are natural examples of object-attribute models for which relationship between objects and attributes are represented by many-valued (fuzzy) relations. Therefore, several attempts to fuzzify FCA have been proposed. As an example we mention work of Bělohlávek [2, 3, 4] or other approaches [12, 14, 15]. One-sided concept lattices play a special role in fuzzy FCA, where usually objects are considered as crisp subsets and attributes obtain fuzzy values. In this case the interpretation of object clusters is straightforward as in classical FCA, instead of fuzzy approaches with fuzzy subsets of objects, where interpretability often becomes problematic.

Recently, there was a generalization of all known one-sided approaches [1, 10, 11], so called generalized one-sided concept lattices, cf. [7, 8]. This approach is, in contrary with the previous one-sided approaches, convenient for the analysis of object-attribute models with different truth value structures. From this point of view it is applicable to a wide spectrum of real object-attribute models where methods of classical FCA are appropriate, cf. [5, 6, 16, 17]. In this note we deal with theoretical question, whether correspondence between formal contexts, which represent object-attribute models, and concept lattices on the other side is one-to-one or equivalently injective.

In order to make this paper as self-contained as possible, in the next section we give a brief overview of the notions like formal context, Galois connections, complete lattices, direct product, etc. We also describe the basic definitions and the results concerning generalized one-sided concept lattices.

Our main results are in Section 3. Firstly we prove that the correspondence between formal context and generalized one-sided concept lattices is injective, i.e., that each generalized one-sided concept lattice also uniquely determines formal context. Based on this result, we deduce the formula expressing number of generalized one-sided concept lattices defined within the fixed framework of a given formal context. Further, we are studying the order structure of mappings involved in some Galois connections between a power set and a direct product of complete lattices. In particular, we show that the lattice of all such mappings and the lattice of all incidence relations are isomorphic.

2. Formal contexts and generalized one-sided concept lattices

In this section we examine the notion of the object-attribute model and its mathematical counterpart formal context. Further, based on the notion of formal context we define generalized one-sided concept lattices as fuzzy generalization of classical concept lattices.

Firstly, we briefly describe the object-attribute models. Generally, by object we understand any item that can be individually selected and manipulated, e.g., person, car, document, etc. In general, an attribute is a property or characteristic of given object, e.g., height of a person, colour of a car or frequency of occurrence of a given word in some document. We will consider that each particular attribute under consideration has defined its range of possible values. Hence, if we measure the height in cm, then any person has assigned the height as integer value from interval $[0, 280]$. Similarly, color of a car can be from some given set of prescribed colors $\{\mathsf{red}, \mathsf{blue}, \mathsf{white}, \dots\}$ and frequency of occurrence of some word w can be given as the ratio $\frac{N_w}{N_{\text{all}}}$ from the interval $[0, 1]$ of rationals. In this case N_w denotes the number of the occurrences of the word w and N_{all} denotes the number of all words in the considered document.

In our understanding object-attribute model consists of the set of objects, set of the attributes with prescribed ranges and values which characterizes objects by the given attributes, e.g., John is tall 183 cm.

In order to apply methods of FCA, we will need one restriction on the ranges of all attributes belonging to object-attribute models. This restriction is given by the usage of fuzzy logic in the theory of fuzzy concept lattices. The main idea of fuzzifications of classical FCA is the usage of graded truth. In classical logic, each proposition is either true or false, hence classical logic is bivalent. In fuzzy logic, to each proposition there is assigned a truth degree from some scale L of truth degrees. The structure L of the truth degrees is partially ordered and contains the smallest and the greatest element. If to the propositions ϕ and ψ are assigned truth degrees $\parallel \phi \parallel = a$ and $\parallel \psi \parallel = b$, then $a \leq b$ means that ϕ is considered less true than ψ. In the object-attribute models typical propositions are of the form "object has attribute in degree a".

In the theory of fuzzy concept lattices it is always assumed that the structure L of the truth degrees assigned to each attribute forms complete lattice.

Now we recall some basic facts concerning partially ordered sets and lattices. By the partially ordered set $(P, \leq)$ we understand non-empty set $P \neq \emptyset$ together with binary relation $\leq$ satisfying:

i) $x \leq x$ for all $x \in P$, i.e., the relation $\leq$ is reflexive,

ii) $x \leq y$ and $y \leq x$ then $x = y$, i.e., antisymmetry of $\leq$,

iii) $x \leq y$ and $y \leq z$ then $x \leq z$, i.e., transitivity of the relation $\leq$.

Let $(P, \leq)$ be a partially ordered set and $H \subseteq P$ be an arbitrary subset. An element $a \in P$ is said to be the *least upper bound* or *supremum* of H, if a is the upper bound of the subset H ($h \leq a$ for all $h \in H$) and a is the least of all elements majorizing H ($a \leq x$ for any upper bound x of H). We shall write $a = \sup H$ or $a = \bigvee H$. The concepts of the *greatest lower bound* or *infimum* is similarly defined and it will be denoted by $\inf H$ or $\bigwedge H$.

A partially ordered set $(L, \leq)$ is a *lattice* if $\sup\{a, b\} = a \vee b$ and $\inf\{a, b\} = a \wedge b$ exist for all $a, b \in L$. A lattice L is called *complete* if $\bigvee H$ and $\bigwedge H$ exist for any subset $H \subseteq L$. Obviously, each finite lattice is complete. Note that any complete lattice contains the greatest element $1_L = \sup L = \inf \emptyset$ and the smallest element $0_L = \inf L = \sup \emptyset$. In what follows we will denote the class of all complete lattices by CL.

Now we are able to define formal context which represents mathematical formalization of the notion object-attribute model.

Definition 2.1. A 4-tuple $\big(B, A, \mathcal{L}, R\big)$ is said to be a generalized one-sided formal context if the following conditions are fulfilled:

a) B is a non-empty set of objects and A is a non-empty set of attributes.

b) $\mathcal{L}\colon A \to \mathsf{CL}$,

c) $R\colon B \times A \to \bigcup_{a \in A} \mathcal{L}(a)$ is a mapping satisfying $R(b, a) \in \mathcal{L}(a)$ for all $b \in B$ and $a \in A$.

Second condition says that $\mathcal{L}$ is a mapping from the set of attributes to the class of all complete lattices. Hence, for any attribute a, $\mathcal{L}(a)$ denotes the complete lattice, which represents structure of truth values for attribute a, i.e., $\mathcal{L}(a)$ denotes the range of attribute a. As it is explicitly given, we require that all ranges form complete lattices. The symbol R denotes so-called (generalized) incidence relation, i.e., $R(b, a)$ represents a degree from the structure $\mathcal{L}(a)$ in which the element $b \in B$ has the given attribute a.

As an example of simple formal context, consider four-element set of objects $B = \{a, b, c, d\}$ and eight-element set of attributes $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$. We will assume that the attributes in our model are binary or real, i.e., ranges of these attributes are represented either two-element chain $\mathbf{2} = \{0, 1\}$ with $0 < 1$ or real unit interval $[0, 1]$. Particularly we have $\mathcal{L}(a_1) = \mathcal{L}(a_3) = \mathcal{L}(a_5) = \mathcal{L}(a_6) = \mathbf{2}$ and $\mathcal{L}(a_2) = \mathcal{L}(a_4) = \mathcal{L}(a_7) = \mathcal{L}(a_8) = [0, 1]$. The generalized incidence relation R of each formal context is usually described as data table. In this case the value $R(b, a)$ can be found on the intersection of b-th row and a-th column of the table. The incidence relation of our example is depicted in Table 1.

Further we define generalized one-sided concept lattices derived from given generalized one-sided formal context. Since the theory of concept lattices is based on the notion of Galois connections, we recall this notion at first, cf. [13] or [9].

Definition 2.2. Let $(P, \leq)$ and $(Q, \leq)$ be partially ordered sets and let

$$\varphi\colon P \to Q \quad \text{and} \quad \psi\colon Q \to P$$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a	0	0.2	1	0.3	1	0	0.1	0.5
b	1	0.6	0	0.6	0	1	0.5	0.3
c	1	1.0	0	0.7	0	0	0.5	0.0
d	0	0.2	0	0.3	1	0	0.1	0.5

Table 1: Data table of object-attribute model

be maps between these ordered sets. Such a pair (φ, ψ) of mappings is called a Galois connection between the ordered sets if:

(a) $p_1 \leq p_2$ implies $\varphi(p_1) \geq \varphi(p_2)$,

(b) $q_1 \leq q_2$ implies $\psi(q_1) \geq \psi(q_2)$,

(c) $p \leq \psi(\varphi(p))$ and $q \leq \varphi(\psi(q))$.

Let us remark that the conditions (a), (b) and (c) are equivalent to the following one:

$$p \leq \psi(q) \quad \text{iff} \quad \varphi(p) \geq q. \tag{2.1}$$

These two maps are also called dually adjoint to each other. An important property of Galois connections is captured in the following expressions (see [9] for the proof).

$$\varphi = \varphi \circ \psi \circ \varphi \qquad \text{and} \qquad \psi = \psi \circ \varphi \circ \psi \tag{2.2}$$

Moreover the dual adjoint is determined uniquely, i.e., if (φ_1, ψ) forms Galois connection as well as (φ_2, ψ) then $\varphi_1 = \varphi_2$. The same is true if (φ, ψ_1) and (φ, ψ_2) form Galois connections, then $\psi_1 = \psi_2$.

Now we describe the partially ordered sets, where we define appropriate Galois connection. On the side of objects, we will consider the set $\mathbf{P}(B)$ as a domain of one part of Galois connection. Let us note that $\mathbf{P}(B)$ denotes the power set of all subsets of the set B partially ordered by the set theoretical inclusion. It is well known fact that $\mathbf{P}(B)$ forms complete lattice. In this case, clusters of objects are represented by classical subsets, hence this is the reason for the name "one-sided concept lattices".

If L_i for $i \in I$ is a family of lattices the *direct product* $\prod_{i \in I} L_i$ is defined as the set of all functions

$$f : I \to \bigcup_{i \in I} L_i \tag{2.3}$$

such that $f(i) \in L_i$ for all $i \in I$ with the "componentwise" order, i.e, $f \leq g$ if $f(i) \leq g(i)$ for all $i \in I$. If $L_i = L$ for all $i \in I$ we get a direct power L^I. In this case the direct power L^I represents the structure of L-fuzzy sets, hence direct product of lattices can be seen as a generalization of the notion of L-fuzzy sets. The direct product of lattices forms complete lattice if and only if all members of the family are complete lattices. The straightforward computations show that the lattice operations in the direct product $\prod_{i\in I} L_i$ of complete lattices are calculated componentwise, i.e., for any subset $\{f_j : j \in J\} \subseteq \prod_{i\in I} L_i$ we obtain

$$\Big(\bigvee_{j\in J} f_j\Big)(i) = \bigvee_{j\in J} f_j(i) \quad \text{and} \quad \Big(\bigwedge_{j\in J} f_j\Big)(i) = \bigwedge_{j\in J} f_j(i), \tag{2.4}$$

where these equalities hold for each index $i \in I$.

Generalized one-sided concept lattices were designed to handle with different types of attributes, hence the appropriate domain for second part of Galois connection consists of direct product of attribute lattices $\prod_{a\in A} \mathcal{L}(a)$.

Definition 2.3. Let $\big(B, A, \mathcal{L}, R\big)$ be a generalized one-sided formal context. We define a pair of mappings $\uparrow\colon \mathbf{P}(B) \to \prod_{a\in A} \mathcal{L}(a)$ and $\downarrow\colon \prod_{a\in A} \mathcal{L}(a) \to \mathbf{P}(B)$ as follows:

$$\uparrow\!\big(X\big)(a) = \bigwedge_{b\in X} R(b,a), \text{ for all } X \subseteq B, \tag{2.5}$$

$$\downarrow\!(g) = \{b \in B : \forall a \in A,\ g(a) \le R(b,a)\}, \text{ for all } g \in \prod_{a\in A} \mathcal{L}(a). \tag{2.6}$$

The main result concerning such defined pair of mappings is stated in the following proposition.

Proposition 2.4. *The pair* $(\uparrow,\downarrow)$ *forms a Galois connection between* $\mathbf{P}(B)$ *and* $\prod_{a\in A} \mathcal{L}(a)$.

Proof. We prove that $\uparrow(X) \ge g$ if and only if $X \subseteq \downarrow(g)$ for all $X \subseteq B$ and all $g \in \prod_{a\in A} \mathcal{L}(a)$.

Since $\uparrow\!\big(X\big) \ge g$ if and only if $\uparrow\!\big(X\big)(a) \ge g(a)$ for all $a \in A$, according to the Definition (2.5) of the map $\uparrow$ and expression (2.4) we obtain

$$\forall a \in A, \uparrow\!\big(X\big)(a) = \bigwedge_{b\in X} R(b,a) \ge g(a) \quad \text{iff} \quad \forall a \in A, \forall b \in X,\ R(b,a) \ge g(a).$$

Due to the definition (2.6) of the map $\downarrow$, this is equivalent to

$$X \subseteq \{b \in B : \forall a \in A,\ g(a) \le R(b,a)\} = \downarrow\!(g). \qquad \square$$

The result of this proposition allows to define generalized one-sided concept lattices. Let $\big(B, A, \mathcal{L}, R\big)$ be a generalized one-sided formal context. Denote by $\mathcal{C}\big(B, A, \mathcal{L}, R\big)$ the set of all pairs (X, g), $X \subseteq B$, $g \in \prod_{a\in A} \mathcal{L}(a)$ which form fixed points of the Galois connection $(\uparrow,\downarrow)$, i.e., satisfying

$$\uparrow\!\big(X\big) = g \quad \text{and} \quad \downarrow\!(g) = X.$$

In this case the ordered pair (X, g) is said to be a concept, the set X is usually referred as extent and g as intent of the concept (X, g).

Further we define partial order on the set $\mathcal{C}\big(B, A, \mathcal{L}, R\big)$ as follows:

$$(X_1, g_1) \le (X_2, g_2) \quad \text{iff} \quad X_1 \subseteq X_2 \quad \text{iff} \quad g_1 \ge g_2. \tag{2.7}$$

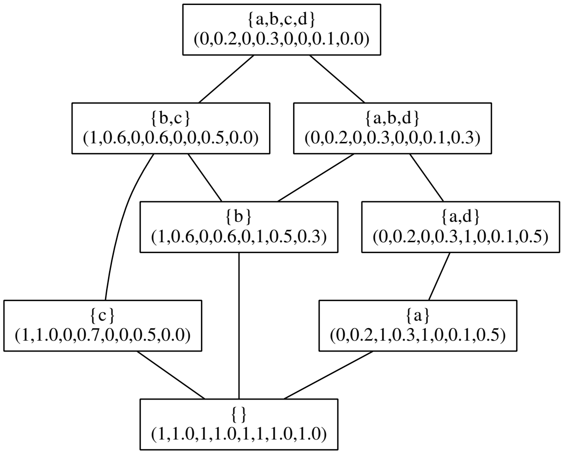

Figure 1: Generalized one-sided concept lattice

Proposition 2.5. *The set* $\mathcal{C}\big(B, A, \mathcal{L}, R\big)$ *with the partial order defined by (2.7) forms a complete lattice, where*

$$\bigwedge_{i\in I}(X_i, g_i) = \Big(\bigcap_{i\in I} X_i, \uparrow\downarrow\big(\bigvee_{i\in I} g_i\big)\Big) \quad \textit{and} \quad \bigvee_{i\in I}(X_i, g_i) = \Big(\downarrow\uparrow\big(\bigcup_{i\in I} X_i\big), \bigwedge_{i\in I} g_i\Big)$$

for each family $(X_i, g_i)_{i\in I}$ *of elements from* $\mathcal{C}\big(B, A, \mathcal{L}, R\big)$.

Proof of this proposition is based on the fact that any Galois connection between complete lattices induces dually isomorphic closure systems (see [13]). Consequently, this dual isomorphism maps infima on the one side onto suprema in a closure system on the other side and vice versa.

Remark that the algorithm for generation of generalized one-sided concept lattices can be found in [7] or [8].

The Hasse diagram of the generalized one-sided concept lattice determined by Table 1 is shown on Figure 1. Let us remark that we denote the elements of direct product as ordered tuples, as it is common in lattice theory.

3. On relationship between incidence relations and generalized one-sided concept lattices

In this section we present our results concerning incidence relations and corresponding one-sided concept lattices. We also describe the order structure of the set of all mappings involving in some Galois connection between power set and the direct product of complete lattices. Firstly, we show that the correspondence

$$\text{generalized one-sided context} \;\mapsto\; \text{generalized one-sided concept lattice}$$

is injective or equivalently one-to-one. We already know how to define generalized one-sided lattice from given formal context. However, there is an interesting theoretical question, whether different formal contexts yield different one-sided concept lattices. The positive answer means that not only formal context fully characterizes generalized one-sided context, but the converse is also true, i.e., given generalized one-sided concept lattice fully determines formal context. Hence, generalized one-sided concept lattice contains all information about object-attribute model.

We recall the definition of injective mapping. A mapping $f : A \to B$ is said to be injective (one-to-one) if

$$x \neq y \quad \text{implies} \quad f(x) \neq f(y)$$

Evidently, this condition is equivalent to the condition $f(x) = f(y)$ implies $x = y$.

In what follows, we will consider that the set of objects B is fixed, as well as the set of all attributes A (together with truth value structures $\mathcal{L}(a)$). Consider that we have two generalized one-sided formal contexts $(B, A, \mathcal{L}, R_1)$ and $(B, A, \mathcal{L}, R_2)$. The corresponding concept lattices are denoted by $C_1 = \mathcal{C}\big(B, A, \mathcal{L}, R_1\big)$ and $C_2 = \mathcal{C}\big(B, A, \mathcal{L}, R_2\big)$.

Theorem 3.1. *The correspondence* $(B, A, \mathcal{L}, R) \mapsto \mathcal{C}\big(B, A, \mathcal{L}, R\big)$*, which assign to each generalized one-sided formal context the corresponding generalized one-sided concept lattice is injective.*

Proof. We prove this theorem in two steps. Firstly we show that the correspondence $(B, A, \mathcal{L}, R) \mapsto (\uparrow, \downarrow)$, which maps formal context onto the Galois connection given by (2.5) and (2.6) respectively, is injective. Next we show that the correspondence $(\uparrow, \downarrow) \mapsto \mathcal{C}\big(B, A, \mathcal{L}, R\big)$, which maps Galois connection to the concept lattice is injective too. Since the composition of two injective mappings is injective, this will satisfy to prove our result.

Suppose that incidence R_1 and R_2 differ, i.e., there exist $b \in B$, $a \in A$ such that $R_1(b, a) \neq R_2(b, a)$. Note, that we will recognize the corresponding Galois connection by subscript. According to the definition (2.5) of mapping $\uparrow$ we obtain:

$$\uparrow_1(\{b\}) = \bigwedge_{b' \in \{b\}} R_1(b', a) = R_1(b, a) \neq R_2(b, a) = \bigwedge_{b' \in \{b\}} R_2(b', a) = \uparrow_2(\{b\}).$$

This equation shows that we have found one-element subset $\{b\}$ with $\uparrow_1(\{b\}) \neq \uparrow_2(\{b\})$ and consequently $(\uparrow_1, \downarrow_1) \neq (\uparrow_2, \downarrow_2)$. Hence, the first correspondence between formal contexts and Galois connections is injective.

Further, assume that $C_1 = C_2$, i.e., that the generalized one-sided concept lattices equal. This means that the sets of fixed points coincide, i.e., for all $X \subseteq B$ and $g \in \prod_{a \in A} \mathcal{L}(a)$ it holds

$$\uparrow_1(X) = g \text{ and } \downarrow_1(g) = X \quad \text{iff} \quad \uparrow_2(X) = g \text{ and } \downarrow_2(g) = X. \tag{3.1}$$

Let $X \subseteq B$ be an arbitrary subset. From the property (2.2) of Galois connections we have $\uparrow_1(X) = \uparrow_1(\downarrow_1(\uparrow_1(X)))$, thus ordered pair $(\downarrow_1(\uparrow_1(X)), \uparrow_1(X))$ forms a fixed point of Galois connection $(\uparrow_1, \downarrow_1)$. Then, due to condition (3.1) we obtain that $\downarrow_2(\uparrow_1(X)) = \downarrow_1(\uparrow_1(X))$. Consequently, we have $X \subseteq \downarrow_1(\uparrow_1(X)) = \downarrow_2(\uparrow_1(X))$ which yields the first half of the condition (c) of the Definition 2.2.

Similarly, using (2.2) we obtain for each element $g \in \prod_{a\in A} \mathcal{L}(A)$ the pair $(\downarrow_2(g), \uparrow_2(\downarrow_2(g))$ forms fixed point of $(\uparrow_2, \downarrow_2)$. Again, due to condition (3.1) we obtain $\uparrow_2(\downarrow_2(g)) = \uparrow_1(\downarrow_2(g))$, which yields $g \leq \uparrow_2(\downarrow_2(g)) = \uparrow_1(\downarrow_2(g))$. Since the mappings $\uparrow_1$ and $\downarrow_2$ are order reversing, we have proved that the pair $(\uparrow_1, \downarrow_2)$ forms Galois connection. Now using the fact that dual adjoint is unique, we obtain $\uparrow_1 = \uparrow_2$ and $\downarrow_1 = \downarrow_2$, which completes the proof. □

It was proved in [7] that for any Galois connection (Φ, Ψ) between $\mathbf{P}(B)$ and $\prod_{a\in A} \mathcal{L}(a)$ there exists a generalized formal context $(B, A, \mathcal{L}, R)$ that $\uparrow = \Phi$ and $\downarrow = \Psi$. Hence the correspondence between formal contexts and generalized one-sided concept lattices is surjective, too. Since we have shown that it is injective, in fact this correspondence is bijective. Using this fact we can prove the following theorem about number of all concept lattices.

Theorem 3.2. *Let $B \neq \emptyset$ be set of objects, $A = \{a_1, a_2, \ldots, a_m\}$ be set of attributes. Denote by $n = |B|$ number of objects and for all $i = 1, \ldots, m$ denote by $n_i = |\mathcal{L}(a_i)|$ the cardinality of the complete lattice $\mathcal{L}(a_i)$. Then there is $(\prod_{i=1}^{m} n_i)^n$ generalized one-sided concept lattices.*

Proof. There is a bijection between set of all generalized incidence relations and one-sided concept lattices, thus it is sufficient to count all generalized incidence relations. For each object b and each attribute a the value $R(b, a)$ can obtain $n_i = |\mathcal{L}(a_i)|$ values. Since we have n objects, there is ${n_i}^n$ possibilities for columns in data table (which represents incidence relation). Together we have

$$ {n_1}^n \cdot {n_2}^n \cdot \ldots \cdot {n_m}^n \cdot \ldots = (\prod_{i=1}^{m} n_i)^n $$

possibilities to define incidence relation. □

This result generalizes the similar assertion for classical concept lattices. Suppose there is given a formal context (B, A, I). If we have n objects and m attributes, then there is $2^{n\cdot m}$ concept lattices. Any classical concept lattice can be characterized as generalized one-sided concept lattice by setting $\mathcal{L}(a) = \mathbf{2}$ ($\mathbf{2} = \{0, 1\}$ denotes two-element chain) and $R(b, a) = 1$ if and only if $(b, a) \in I$ (see [14] for details). Hence applying the result of Theorem 3.2 we obtain $\prod_{i=1}^{m} 2^n = (2^n)^m = 2^{m\cdot n}$.

Similarly, if one will consider $\mathcal{L}(a_i) = L$ for all $i = 1, \ldots, m$, than generalized one-sided concept lattices, represent one-sided concept lattices. Hence, applying Theorem 3.2 we obtain that there is $\prod_{i=1}^{m} |L|^n = |L|^{m\cdot n}$ different one-sided concept lattices.

Next we show that formal contexts also characterize order properties of the Galois connections between power sets and complete lattices. Firstly we prove the following lemma, concerning the closure property of Galois connections. Let L and

M be complete lattices. Denote by $\mathsf{Gal}(L, M)$ the set of all $\varphi\colon L \to M$ such that there exists $\psi\colon M \to L$ dually adjoint to φ.

Lemma 3.3. *Let L, M be complete lattices. The set $\mathsf{Gal}(L, M)$ forms a closure system in complete lattice M^L.*

Proof. We show that the set $\mathsf{Gal}(L, M)$ is closed under arbitrary infima. Let $\{\varphi_i : i \in I\} \subseteq \mathsf{Gal}(L, M)$ be an arbitrary system. Denote by $\varphi = \bigwedge_{i\in I} \varphi_i$. In this case $\varphi(x) = \bigwedge_{i\in I} \varphi(x)$ for all $x \in L$. In order to prove that $\varphi \in \mathsf{Gal}(L, M)$ we show that there is a dual adjoint $\psi\colon M \to L$. Define $\psi = \bigwedge_{i\in I} \psi_i$ where ψ_i is dually adjoint to φ_i for all $i \in I$.

Let $x_1, x_2 \in L$ be elements such that $x_1 \leq x_2$. Since $\varphi_i(x_1) \geq \varphi_i(x_2)$ for all $i \in I$, we obtain

$$\varphi(x_1) = \bigwedge_{i\in I} \varphi_i(x_1) \geq \bigwedge_{i\in I} \varphi_i(x_2) = \varphi(x_2).$$

Similarly, for all $y_1, y_2 \in M$ condition $y_1 \leq y_2$ implies $\psi(y_2) \geq \psi(y_1)$.

Finally, we show that $x \leq \psi(\varphi(x))$ for all $x \in L$. Let $j \in I$ be an arbitrary index. Then for all $x \in L$ we have

$$x \leq \psi_j\bigl(\varphi_j(x)\bigr) \leq \psi_j\bigl(\bigwedge_{i\in I} \varphi(x)\bigr),$$

since ψ_j is order reversing and $\varphi_j(x) \geq \bigwedge_{i\in I} \varphi_i(x)$. This yields

$$x \leq \bigwedge_{j\in I} \psi_j\bigl(\bigwedge_{i\in I} \varphi_i(x)\bigr) = \bigwedge_{j\in I} \psi_j\bigl(\varphi(x)\bigr) = \psi\bigl(\varphi(x)\bigr).$$

In similar way, one can prove $y \leq \varphi(\psi(y))$ for all $y \in M$. □

Since $\mathsf{Gal}(L, M)$ forms a closure system in complete lattice M^L, it forms complete lattice too. In this case meets in $\mathsf{Gal}(L, M)$ coincide with the meets in M^L, but this is not valid for joins in general. In particular, if $(\varphi_i : i \in I\} \subseteq \mathsf{Gal}(L, M)$ then

$$\sup\{\varphi_i : i \in I\} = \bigwedge\{\varphi \in \mathsf{Gal}(L, M) : \varphi \geq \bigvee_{i\in I} \varphi_i\}$$

where the symbols $\bigwedge$ and $\bigvee$ denote operations of meet and join in M^L.

Let us note that $\mathsf{Gal}(L, M)$ and $\mathsf{Gal}(M, L)$ forms isomorphic posets. This follows from the fact that the correspondence $\varphi \mapsto \psi$ where ψ denotes the dual adjoint of φ is bijective. Moreover it is order preserving in both directions. Suppose $\varphi_1(x) \leq \varphi_2(x)$ for all $x \in L$. Let $y \in M$ be an arbitrary element. Then $y \leq \varphi_1(\psi_1(y)) \leq \varphi_2(\psi_1(y))$ and according to the condition (2.1) it follows $\psi_1(y) \leq \psi_2(y)$. The opposite implication can be proved analogously, hence $\varphi_1 \leq \varphi_2$ if and only if $\psi_1 \leq \psi_2$.

Further assume that $B, A \neq \emptyset$ and $\mathcal{L}\colon A \to \mathsf{CL}$ are fixed. In order to describe the structure of the lattice $\mathsf{Gal}(\mathbf{P}(B), \prod_{a\in A} \mathcal{L}(a))$ we denote by $\mathsf{R}(B, A, \mathcal{L})$ the set of all relations R such that $(B, A, \mathcal{L}, R)$ forms generalized one-sided formal context. Obviously the set $\mathsf{R}(B, A, \mathcal{L})$ forms complete lattice. In this case, if $\{R_i : i \in I\}$ is a system of relations, then relation R where $R(b,a) = \bigwedge_{i\in I} R_i(b,a)$ ($R(b,a) = \bigvee_{i\in I} R_i(b,a)$) corresponds to the infimum (supremum).

Theorem 3.4. *The lattice* $\mathsf{Gal}(\mathbf{P}(B), \prod_{a\in A} \mathcal{L}(a))$ *is isomorphic to the lattice of all incidence relations* $\mathsf{R}(B, A, \mathcal{L})$*.*

Proof. Define $F\colon \mathsf{R}(B, A, \mathcal{L}) \to \mathsf{Gal}(\mathbf{P}(B), \prod_{a\in A} \mathcal{L}(a))$ for all $R \in \mathsf{R}(B, A, \mathcal{L})$ by $F(R) = \uparrow_R$, where $\uparrow_R$ is defined by (2.5). As we already know, the mapping F is bijective. We show, that it also preserves the lattice operations, i.e., $F\big(R_1 \wedge R_2\big) = F(R_1) \wedge F(R_2)$ and $F(R_1 \vee R_2) = \sup\{F(R_1), F(R_2)\}$.

Let $X \subseteq B$ be any subset and $a \in A$ be an arbitrary element. Then we obtain

$$\uparrow_{R_1 \wedge R_2}(X)(a) = \bigwedge_{b\in X} \Big(R_1(b,a) \wedge R_2(b,a)\Big) =$$

$$= \bigwedge_{b\in X} R_1(b,a) \ \wedge \ \bigwedge_{b\in X} R_2(b,a) = \uparrow_{R_1}(X)(a) \ \wedge \ \uparrow_{R_2}(X)(a).$$

Hence the mapping F preserves meets.

In order to prove that F preserves joins, we use the fact that the mapping F is surjective, i.e., for any Galois connection (φ, ψ) between $\mathbf{P}(B)$ and $\prod_{a\in A} \mathcal{L}(a)$ there is some relation R with $\varphi = \uparrow_R$ and $\psi = \downarrow_R$.

Let $\varphi \in \mathsf{Gal}(\mathbf{P}(B), \prod_{a\in A} \mathcal{L}(a))$ be a mapping satisfying $\varphi \geq \uparrow_{R_1}, \uparrow_{R_2}$. Then $\varphi = \uparrow_R$ for some $R \in \mathsf{R}(B, A, \mathcal{L})$ and for all $b \in B$ and $a \in A$ we obtain

$$\varphi(\{b\})(a) = \uparrow_R(\{b\})(a) = \bigwedge_{b'\in\{b\}} R(b,a) = R(b,a).$$

Since $\varphi(\{b\}) \geq \uparrow_{R_1}(\{b\}), \uparrow_{R_2}(\{b\})$ for all $b \in B$ we have $R(b,a) \geq R_1(b,a) \vee R_2(b,a)$ for all $b \in B$ and $a \in A$. This yields

$$\varphi(X)(a) = \uparrow_R(X)(a) = \bigwedge_{b\in X} R(b,a) \geq \bigwedge_{b\in X} \Big(R_1(b,a) \vee R_2(b,a)\Big) = \uparrow_{R_1 \vee R_2}(X)(a)$$

for all $X \subseteq B$ and for all $a \in A$. Obviously $\uparrow_{R_1 \vee R_2}$ is the upper bound of $\uparrow_{R_1}$ and $\uparrow_{R_2}$ and we have shown that it is in fact the least upper bound of $\uparrow_{R_1}$ and $\uparrow_{R_2}$. Hence in the lattice $\mathsf{Gal}(\mathbf{P}(B), \prod_{a\in A} \mathcal{L}(a))$ the assertion $F(R_1 \vee R_2) = \sup\{F(R_1), F(R_2)\}$ is valid. □

References

[1] Ben Yahia, S; Jaoua, A., Discovering knowledge from fuzzy concept lattice. *Data Mining and Computational Intelligence*, Physica-Verlag, Heidelberg, Germany (2001), 167–190.

[2] Bělohlávek, R., Lattices generated by binary fuzzy relations., *Tatra Mt. Math. Publ.* 16, (1999), 11–19.

[3] Bělohlávek, R., Lattices of Fixed Points of Fuzzy Galois Connections, *Math. Log. Quart.* 47(1), (2001), 111–116.

[4] Bělohlávek, R.; Vychodil, V., What is a fuzzy concept lattice?, *Proceedings of the CLA 2005*, Olomouc (2005), 34–45.

[5] Butka, P., Use of FCA in the ontology extraction step for the improvement of the semantic information retrieval. *Proceedings of the SOFSEM 2006 : Theory and Practice of Computer Science*, Prague (2006), 74–82.

[6] Butka, P.; Sarnovsky, M.; Bednar, P., One approach to combination of FCA-based local conceptual models for text analysis Grid-based approach. *Proceedings of the 6-th international conference SAMI 2008*, Budapest (2008), 131–135.

[7] Butka, P; Pócs, J., Generalization of one-sided concept lattices. *Computing and Informatics*, 32 (2013), 355–370.

[8] Butka, P; Pócs, J.; Pócsová J., Use of Concept Lattices for Data Tables with Different Types of Attributes. *Journal of Information and Organizational Sciences*, 36(1) (2012), 1–12.

[9] Ganter, B.; Wille, R., *Formal concept analysis. Mathematical foundations.* Springer, Berlin, (1999).

[10] Jaoua, A.; Elloumi, S., Galois connection, formal concepts and Galois lattice in real relations: application in a real classifier. *The Journal of Systems and Software*, Vol. 60 (2002), 149–163.

[11] Krajči, S., Cluster based efficient generation of fuzzy concepts. *Neural Network World*, 13(5), (2003), 521–530.

[12] Krajči, S., A generalized concept lattice. *Logic Journal of the IGPL*, 13(5) (2005), 543–550.

[13] Ore, O., Galois Connexions. *Transactions of the American Mathematical Society*, 55(3) (1944), 493–513.

[14] Pócs, J., Note on generating fuzzy concept lattices via Galois connections. *Information Sciences*, 185(1) (2012), 128–136.

[15] Pócs, J., On possible generalization of fuzzy concept lattices using dually isomorphic retracts. *Information Sciences* 210 (2012), 89–98.

[16] Sarnovský, M.; Butka, P.; Paralič, J., Grid-based support for different text mining tasks *Acta Polytechnica Hungarica.*, 6(4) (2009), 5–27.

[17] Sarnovský, M.; Butka, P., Cloud computing as a platform for distributed data analysis *Proceedings of the 7th Workshop on Intelligent and Knowledge Oriented Technologies*, Smolenice (2012), 177–180.

Annales Mathematicae et Informaticae
42 (2013) pp. 83–92
http://ami.ektf.hu

Bi-periodic incomplete Fibonacci sequences

José L. Ramírez*

Instituto de Matemáticas y sus Aplicaciones
Universidad Sergio Arboleda, Bogotá, Colombia
josel.ramirez@ima.usegioarboleda.edu.co

Submitted October 07, 2013 — Accepted December 06, 2013

Abstract

In this paper, we define the bi-periodic incomplete Fibonacci sequences, we study some recurrence relations linked to them, some properties of these numbers and their generating functions. In the case $a = k = b$, we obtain the incomplete k-Fibonacci numbers. If $a = 1 = b$, we have the incomplete Fibonacci numbers.

Keywords: bi-periodic incomplete Fibonacci sequence, bi-periodic Fibonacci sequence, generating function

MSC: 11B39, 11B83, 05A15

1. Introduction

Fibonacci numbers and their generalizations have many interesting properties and applications to almost every field of science and art [10]. The Fibonacci numbers F_n are defined by the recurrence relation

$$F_0 = 0, \quad F_1 = 1, \quad F_{n+1} = F_n + F_{n-1}, \; n \geqslant 1.$$

There exist a lot of properties about Fibonacci numbers. In particular, there is a beautiful combinatorial identity

$$F_n = \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-i-1}{i} \tag{1.1}$$

*The author was partially supported by Universidad Sergio Arboleda.

for Fibonacci numbers [10].

In analogy with (1.1), Filipponi [6] introduced the incomplete Fibonacci numbers $F_n(s)$ and the incomplete Lucas numbers $L_n(s)$. They are defined by

$$F_n(s) = \sum_{j=0}^{s} \binom{n-1-j}{j} \quad \left(n = 1, 2, 3, \ldots; 0 \le s \le \left\lfloor \frac{n-1}{2} \right\rfloor\right),$$

and

$$L_n(s) = \sum_{j=0}^{s} \frac{n}{n-j}\binom{n-j}{j} \quad \left(n = 1, 2, 3, \ldots; 0 \le s \le \left\lfloor \frac{n}{2} \right\rfloor\right).$$

Further in [11], generating functions of the incomplete Fibonacci and Lucas numbers are determined. In [2] Djordević gave the incomplete generalized Fibonacci and Lucas numbers. In [3] Djordević and Srivastava defined incomplete generalized Jacobsthal and Jacobsthal-Lucas numbers. In [15] the authors define the incomplete Fibonacci and Lucas p-numbers. Also the authors define the incomplete bivariate Fibonacci and Lucas p-polynomials in [16]. In [13] we introduce the incomplete k-Fibonacci and k-Lucas numbers and in [12] we study incomplete $h(x)-$Fibonacci and $h(x)-$Lucas polynomials.

On the other hand, many kinds of generalizations of Fibonacci numbers have been presented in the literature. In particular, a generalization is the bi-periodic Fibonacci sequence [4]. For any two nonzero real numbers a and b, the bi-periodic Fibonacci sequence, say $\{q_n\}_{n=0}^{\infty}$, is determined by:

$$q_0 = 0, \quad q_1 = 1, \quad q_n = \begin{cases} aq_{n-1} + q_{n-2}, & \text{if } n \equiv 0 \pmod 2; \\ bq_{n-1} + q_{n-2}, & \text{if } n \equiv 1 \pmod 2; \end{cases} \quad n \geqslant 2. \tag{1.2}$$

These numbers have been studied in several papers; see [1, 4, 5, 8, 9, 17]. In [17], the explicit formula to bi-periodic Fibonacci numbers is

$$q_n = a^{\xi(n-1)} \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-i-1}{i} (ab)^{\lfloor \frac{n-1}{2} \rfloor - i}, \tag{1.3}$$

where $\xi(n) = n - 2\lfloor \frac{n}{2} \rfloor$, i.e., $\xi(n) = 0$ when n is even and $\xi(n) = 1$ when n is odd. From equation (1.3) we introduce the bi-periodic incomplete Fibonacci numbers and we obtain new recurrent relations, new identities and generating functions.

2. Bi-Periodic Incomplete Fibonacci Sequence

Definition 2.1. For $n \geqslant 1$, the bi-periodic incomplete Fibonacci numbers are defined as

$$q_n(l) = a^{\xi(n-1)} \sum_{i=0}^{l} \binom{n-i-1}{i} (ab)^{\lfloor \frac{n-1}{2} \rfloor - i}, \quad 0 \le l \le \left\lfloor \frac{n-1}{2} \right\rfloor. \tag{2.1}$$

For $a = b$, $q_n(l) = F^l_{k,n}$, we get incomplete k-Fibonacci numbers [13]. If $a = b = 1$, we obtained incomplete Fibonacci numbers [6]. In Table 1, some values of bi-periodic incomplete k-Fibonacci numbers are provided, with $a = 3$ and $b = 2$.

n/l	0	1	2	3	4	5	6
1	1						
2	3						
3	6	7					
4	18	24					
5	36	54	55				
6	108	180	189				
7	216	396	432	433			
8	648	1296	1476	1488			
9	1296	2808	3348	3408	3409		
10	3888	9072	11340	11700	11715		
11	7776	19440	25488	26748	26838	26839	
12	23328	62208	85536	91584	92214	92232	
13	46656	132192	190512	208656	211176	211302	211303
14	139968	419904	633744	711504	725112	726120	726141
15	279936	886464	1399680	1613520	1658880	1663416	1663584
16	839808	2799360	4618944	5474304	5688144	5715360	5716872

Table 1: Numbers $q_n(l)$, for $1 \leqslant n \leqslant 16$, and $a = 3, b = 2$

Some special cases of (2.1) are

$$q_n(0) = a^{\xi(n-1)}(ab)^{\lfloor \frac{n-1}{2} \rfloor}; \ (n \geq 1) \tag{2.2}$$

$$q_n(1) = a^{\xi(n-1)}(ab)^{\lfloor \frac{n-1}{2} \rfloor} + a^{\xi(n-1)}(n-2)(ab)^{\lfloor \frac{n-1}{2} \rfloor - 1}; \ (n \geq 3) \tag{2.3}$$

$$q_n\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right) = q_n; \ (n \geq 1) \tag{2.4}$$

$$q_n\left(\left\lfloor \frac{n-3}{2} \right\rfloor\right) = \begin{cases} q_n - \frac{na}{2}, & \text{if } n \equiv 0 \pmod 2; \\ q_n - 1, & \text{if } n \equiv 1 \pmod 2; \end{cases} \quad n \geq 3. \tag{2.5}$$

2.1. Some recurrence properties of the numbers $q_n(l)$

Proposition 2.2. *The non-linear recurrence relation of the bi-periodic incomplete Fibonacci numbers $q_n(l)$ is*

$$q_{n+2}(l+1) = \begin{cases} aq_{n+1}(l+1) + q_n(l), & \text{if } n \equiv 0 \pmod 2; \\ aq_{n+1}(l+1) + q_n(l), & \text{if } n \equiv 1 \pmod 2; \end{cases} \quad 0 \leq l \leq \frac{n-2}{2}. \tag{2.6}$$

The relation (2.6) *can be transformed into the non-homogeneous recurrence relation*

$$q_{n+2}(l) = \begin{cases} aq_{n+1}(l) + q_n(l) - a\binom{n-l-1}{l}(ab)^{\lfloor \frac{n-1}{2} \rfloor - l}, & \text{if } n \equiv 0 \pmod 2; \\ bq_{n+1}(l) + q_n(l) - \binom{n-l-1}{l}(ab)^{\lfloor \frac{n-1}{2} \rfloor - l}, & \text{if } n \equiv 1 \pmod 2. \end{cases} \tag{2.7}$$

Proof. If n is even, then $\lfloor \frac{n}{2} \rfloor = \lfloor \frac{n-1}{2} \rfloor + 1$. Use the Definition 2.1 to rewrite the right-hand side of (2.6) as

$$\begin{aligned}
&a\left(a^{\xi(n)}\sum_{i=0}^{l+1}\binom{n-i}{i}(ab)^{\lfloor\frac{n}{2}\rfloor-i}\right)+a^{\xi(n-1)}\sum_{i=0}^{l}\binom{n-i-1}{i}(ab)^{\lfloor\frac{n-1}{2}\rfloor-i}\\
&=a^{\xi(n+1)}\sum_{i=0}^{l+1}\binom{n-i}{i}(ab)^{\lfloor\frac{n}{2}\rfloor-i}+a^{\xi(n+1)}\sum_{i=1}^{l+1}\binom{n-i}{i-1}(ab)^{\lfloor\frac{n-1}{2}\rfloor-(i-1)}\\
&=a^{\xi(n+1)}\left(\sum_{i=0}^{l+1}\left[\binom{n-i}{i}+\binom{n-i}{i-1}\right](ab)^{\lfloor\frac{n}{2}\rfloor-i}\right)-a^{\xi(n+1)}\binom{n}{-1}(ab)^{\lfloor\frac{n+1}{2}\rfloor}\\
&=a^{\xi(n+1)}\sum_{i=0}^{l+1}\binom{n-i+1}{i}(ab)^{\lfloor\frac{n-1}{2}\rfloor-i}-0\\
&=q_{n+2}(l+1).
\end{aligned}$$

If n is odd, the proof is analogous. On the other hand, equation (2.7) is clear from (2.6). In fact, if n is even

$$\begin{aligned}
q_{n+2}(l)&=aq_{n+1}(l)+q_n(l-1)=aq_{n+1}(l)+q_n(l)+(q_n(l-1)-q_n(l))\\
&=aq_{n+1}(l)+q_n(l)-a\binom{n-l-1}{l}(ab)^{\lfloor\frac{n-1}{2}\rfloor-l}.
\end{aligned}$$

If n is odd, the proof is analogous. □

Proposition 2.3. *One has*

$$\sum_{i=0}^{s}\binom{s}{i}q_{n+i}(l+i)a^{\lfloor\frac{i+\xi(n+1)}{2}\rfloor}b^{\lfloor\frac{i+\xi(n)}{2}\rfloor}=q_{n+2s}(l+s),\quad 0\le l\le\frac{n-s-1}{2}. \tag{2.8}$$

Proof. (By induction on s.) The sum (2.8) clearly holds for $s=0$ and $s=1$ (see (2.6)). Now suppose that the result is true for all $j<s+1$, we prove it for $s+1$. If n is even, then

$$\begin{aligned}
&\sum_{i=0}^{s+1}\binom{s+1}{i}q_{n+i}(l+i)a^{\lfloor\frac{i+1}{2}\rfloor}b^{\lfloor\frac{i}{2}\rfloor}\\
&=\sum_{i=0}^{s+1}\left[\binom{s}{i}+\binom{s}{i-1}\right]q_{n+i}(l+i)a^{\lfloor\frac{i+1}{2}\rfloor}b^{\lfloor\frac{i}{2}\rfloor}\\
&=\sum_{i=0}^{s+1}\binom{s}{i}q_{n+i}(l+i)a^{\lfloor\frac{i+1}{2}\rfloor}b^{\lfloor\frac{i}{2}\rfloor}+\sum_{i=0}^{s+1}\binom{s}{i-1}q_{n+i}(l+i)a^{\lfloor\frac{i+1}{2}\rfloor}b^{\lfloor\frac{i}{2}\rfloor}\\
&=q_{n+2s}(l+s)+\binom{s}{s+1}q_{n+s+1}(l+s+1)a^{\lfloor\frac{s+2}{2}\rfloor}b^{\lfloor\frac{s+1}{2}\rfloor}
\end{aligned}$$

$$+\sum_{i=-1}^{s}\binom{s}{i}q_{n+i+1}(l+i+1)a^{\lfloor\frac{i+2}{2}\rfloor}b^{\lfloor\frac{i+1}{2}\rfloor}$$
$$=q_{n+2s}(l+s)+0+a\sum_{i=0}^{s}\binom{s}{i}q_{n+i+1}(l+i+1)a^{\lfloor\frac{i}{2}\rfloor}b^{\lfloor\frac{i+1}{2}\rfloor}+\binom{s}{-1}q_n(l)a^{\lfloor\frac{1}{2}\rfloor}b^0$$
$$=q_{n+2s}(l+s)+a\sum_{i=0}^{s}\binom{s}{i}q_{n+i+1}(l+i+1)a^{\lfloor\frac{i}{2}\rfloor}b^{\lfloor\frac{i+1}{2}\rfloor}+0$$
$$=q_{n+2s}(l+s)+aq_{n+2s+1}(l+s+1)$$
$$=q_{n+2s+2}(l+s+1).$$

If n is odd, the proof is analogous. □

Proposition 2.4. *For $n\geq 2l+2$,*

$$\sum_{i=0}^{s-1}a^{\lfloor\frac{s-\xi(n+1)}{2}\rfloor-\lfloor\frac{i+\xi(n)}{2}\rfloor}b^{\lfloor\frac{s-\xi(n)}{2}\rfloor-\lfloor\frac{i+\xi(n+1)}{2}\rfloor}q_{n+i}(l)$$
$$=q_{n+s+1}(l+1)-a^{\lfloor\frac{s+\xi(n+1)}{2}\rfloor}b^{\lfloor\frac{s+\xi(n)}{2}\rfloor}q_{n+1}(l+1). \quad (2.9)$$

Proof. (By induction on s.) Sum (2.9) clearly holds for $s=1$ (see (2.6)). Now suppose that the result is true for all $i<s$. We prove it for s. If n is even, then

$$\sum_{i=0}^{s}a^{\lfloor\frac{s}{2}\rfloor-\lfloor\frac{i}{2}\rfloor}b^{\lfloor\frac{s+1}{2}\rfloor-\lfloor\frac{i+1}{2}\rfloor}q_{n+i}(l)$$
$$=\sum_{i=0}^{s-1}a^{\lfloor\frac{s}{2}\rfloor-\lfloor\frac{i}{2}\rfloor}b^{\lfloor\frac{s+1}{2}\rfloor-\lfloor\frac{i+1}{2}\rfloor}q_{n+i}(l)+q_{n+s}(l)$$
$$=a^{\xi(s+1)}b^{\xi(s)}\sum_{i=0}^{s-1}a^{\lfloor\frac{s-1}{2}\rfloor-\lfloor\frac{i}{2}\rfloor}b^{\lfloor\frac{s}{2}\rfloor-\lfloor\frac{i+1}{2}\rfloor}q_{n+i}(l)+q_{n+s}(l)$$
$$=a^{\xi(s+1)}b^{\xi(s)}\Big(q_{n+s+1}(l+1)-a^{\lfloor\frac{s+1}{2}\rfloor}b^{\lfloor\frac{s}{2}\rfloor}q_{n+1}(l+1)\Big)+q_{n+s}(l)$$
$$=\Big(a^{\xi(s+1)}b^{\xi(s)}q_{n+s+1}(l+1)+q_{n+s}(l)\Big)-a^{\xi(s+1)+\lfloor\frac{s+1}{2}\rfloor}b^{\xi(s)+\lfloor\frac{s}{2}\rfloor}q_{n+1}(l+1)$$
$$=\Big(a^{\xi(s+1)}b^{\xi(s)}q_{n+s+1}(l+1)+q_{n+s}(l)\Big)-a^{\lfloor\frac{s+2}{2}\rfloor}b^{\lfloor\frac{s+1}{2}\rfloor}q_{n+1}(l+1)$$
$$=q_{n+s+2}(l+1)-a^{\lfloor\frac{s+2}{2}\rfloor}b^{\lfloor\frac{s+1}{2}\rfloor}q_{n+1}(l+1).$$

If n is odd, the proof is analogous. □

Following proposition shows the sum of the nth row of the array in Table 1.

Proposition 2.5. *One has*

$$\sum_{l=0}^{\lfloor\frac{n-1}{2}\rfloor}q_n(l)=(l+1)q_n(l)-a^{\xi(n-1)}\sum_{i=0}^{\lfloor\frac{n-1}{2}\rfloor}i\binom{n-i-1}{i}(ab)^{\lfloor\frac{n-1}{2}\rfloor-i}. \quad (2.10)$$

Proof. Let $h = \lfloor \frac{n-1}{2} \rfloor$, then

$$
\begin{aligned}
\sum_{l=0}^{h} q_n(l) &= q_n(0) + q_n(1) + \cdots + q_n(h) \\
&= a^{\xi(n-1)} \binom{n-1-0}{0} (ab)^h \\
&+ a^{\xi(n-1)} \left[\binom{n-1-0}{0} (ab)^h + \binom{n-1-1}{1} (ab)^{h-1} \right] + \cdots \\
&+ a^{\xi(n-1)} \left[\binom{n-1-0}{0} (ab)^h + \cdots + \binom{n-1-h}{h} (ab)^{h-h} \right] \\
&= a^{\xi(n-1)} \left[(h+1) \binom{n-1-0}{0} (ab)^h + h \binom{n-1-1}{1} (ab)^{h-1} + \right. \\
&\left. \cdots + \binom{n-1-h}{h} (ab)^{h-h} \right] \\
&= a^{\xi(n-1)} \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} (h+1-i) \binom{n-1-i}{i} (ab)^{h-i} \\
&= a^{\xi(n-1)} (h+1) \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-1-i}{i} (ab)^{h-i} \\
&- a^{\xi(n-1)} \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} i \binom{n-1-i}{i} (ab)^{h-i} \\
&= (h+1)\, q_n(l) - a^{\xi(n-1)} \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} i \binom{n-1-i}{i} (ab)^{h-i}. \qquad \square
\end{aligned}
$$

3. Generating function of the bi-periodic incomplete Fibonacci numbers

In this section, we give the generating functions of bi-periodic incomplete Fibonacci numbers.

Lemma 3.1. *Let $\{s_n\}_{n=0}^{\infty}$ be a complex sequence satisfying the following non-homogeneous and non-linear recurrence relation:*

$$
s_n = \begin{cases} as_{n-1} + s_{n-2} + ar_n, & \text{if } n \equiv 1 \pmod 2; \\ bs_{n-1} + s_{n-2} + s_{n-1}, & \text{if } n \equiv 0 \pmod 2; \end{cases} \quad (n > 1), \tag{3.1}
$$

where a and b are complex numbers and $\{r_n\}_{n=0}^{\infty}$ is a given complex sequence. Then *the generating function $U(t)$ of the sequence $\{s_n\}_{n=0}^{\infty}$ is*

$$U(t) = \frac{aG(t) + s_0 - r_0 + (s_1 - as_0 - ar_1)t + (b-a)tf(t) + (1-a)R(t)}{1 - at - t^2}, \quad (3.2)$$

where $G(t)$ denotes the generating function of $\{r_n\}_{n=0}^{\infty}$, $f(t)$ denotes the generating function of $\{s_{2n+1}\}_{n=0}^{\infty}$ and $R(t)$ denotes the generating function of $\{r_{2n}\}_{n=0}^{\infty}$. Moreover,

$$f(t) = \frac{atR(t) + a(1-t^2)R'(t) + (s_1 - a(r_1 + r_0))t + (a(s_0 + r_1) - s_1)t^3}{1 - (ab+2)t^2 + t^4}, \quad (3.3)$$

where $R'(t)$ denotes the generating function of $\{r_{2n-1}\}_{n=1}^{\infty}$.

Proof. We begin with the formal power series representation of the generating function for $\{s_n\}_{n=0}^{\infty}$ and $\{r_n\}_{n=0}^{\infty}$,

$$U(t) = s_0 + s_1 t + s_2 t^2 + \cdots + s_k t^k + \cdots ,$$
$$G(t) = r_0 + r_1 t + r_2 t^2 + \cdots + r_k t^k + \cdots .$$

Note that,

$$atU(t) = as_0 t + as_1 t^2 + as_2 t^3 + \cdots + as_k t^{k+1} + \cdots ,$$
$$t^2U(t) = s_0 t^2 + s_1 t^3 + s_2 t^4 + \cdots + s_k t^{k+1} + \cdots ,$$

and,

$$aG(t) = ar_0 + ar_1 t + ar_2 t^2 + \cdots + ar_k t^k + \cdots .$$

Since $s_{2k+1} = as_{2k} + s_{2k-1} + ar_{2k+1}$, we get

$$(1 - at - t^2)U(t) - aG(t)$$
$$= (s_0 - ar_0) + (s_1 - a(s_0 + r_1))t + \sum_{m=1}^{\infty} (s_{2m} - as_{2m-1} - s_{2m-2} - ar_{2m})\, t^{2m}.$$

Since $s_{2k} = bs_{2k-1} + s_{2k-2} + r_{2k}$, we get

$$(1 - at - t^2)U(t) - aG(t)$$
$$= (s_0 - ar_0) + (s_1 - a(s_0 + r_1))t + \sum_{m=1}^{\infty} ((b-a)s_{2m-1} + (1-a)r_{2m})\, t^{2m}$$
$$= (s_0 - ar_0) + (s_1 - a(s_0 + r_1))t + (b-a)t \sum_{m=1}^{\infty} s_{2m-1}t^{2m-1} + (1-a) \sum_{m=1}^{\infty} r_{2m}t^{2m}$$
$$= (s_0 - ar_0) + (s_1 - a(s_0 + r_1))t + (b-a)tf(t) + (1-a)R(t) - (1-a)r_0$$
$$= (s_0 - r_0) + (s_1 - a(s_0 + r_1))t + (b-a)tf(t) + (1-a)R(t).$$

Then equation (3.2) is clear.

On the other hand,

$$\begin{aligned} s_{2m-1} &= as_{2m-2} + s_{2m-3} + ar_{2m-1} \\ &= a(bs_{2m-3} + s_{2m-4} + r_{2m-2}) + s_{2m-3} + ar_{2m-1} \\ &= (ab+1)s_{2m-3} + as_{2m-4} + a(r_{2m-2} + r_{2m-1}) \\ &= (ab+1)s_{2m-3} + s_{2m-3} - s_{2m-5} - ar_{2m-3} + a(r_{2m-2} + r_{2m-1}) \\ &= (ab+2)s_{2m-3} - s_{2m-5} + a(-r_{2m-3} + r_{2m-2} + r_{2m-1}). \end{aligned}$$

Then

$$\begin{aligned} &(1-(ab+2)t^2+t^4)f(t) - atR(t) + a(t^2-1)R'(t) \\ &= (s_1 - a(r_0+r_1))t + (s_3 - (ab+2)s_1 - ar_2 + a(r_1 - r_3))t^3 \\ &+ \sum_{m=3}^{\infty}(s_{2m-1} - (ab+2)s_{2m-3} + s_{2m-5} - ar_{2m-2} \\ &+ a(r_{2m-3} - r_{2m-1}))t^{2m-1} \\ &= (s_1 - a(r_0+r_1))t + (s_3 - (ab+2)s_1 - ar_2 + a(r_1 - r_3))t^3 \\ &= (s_1 - a(r_0+r_1))t + (a(s_0+r_1) - s_1)t^3. \end{aligned}$$

Therefore equation (3.3) is obtained. □

Theorem 3.2. *The generating function of the bi-periodic incomplete Fibonacci numbers $q_n(l)$ is given by*

$$Q_l(t) = \sum_{i=0}^{\infty} q_i(l)t^i \tag{3.4}$$

$$= \frac{aG(t) + q_{2l+1} + (q_{2l+2} - aq_{2l+1})t + (b-a)tf(t) + (1-a)R(t)}{1 - at - t^2}, \tag{3.5}$$

where

$$G(t) = -\frac{1}{2}\left(\frac{t^2}{(1-(ab)^{1/2}t)^{l+1}}(1+(ab)^{-1/2}) + \frac{t^2}{(1+(ab)^{1/2}t)^{l+1}}(1-(ab)^{-1/2})\right), \tag{3.6}$$

$$f(t) = \frac{q_{2l+2}t + (aq_{2l+1} - q_{2l+2})t^3 + atR(t) + a(1-t^2)R'(t)}{1-(ab+2)t^2+t^4} \tag{3.7}$$

and

$$R(t) = -\frac{1}{2}\left(\frac{t^2}{(1-(ab)^{1/2}t)^{l+1}} + \frac{t^2}{(1+(ab)^{1/2}t)^{l+1}}\right), \tag{3.8}$$

$$R'(t) = -\frac{1}{2(ab)^{1/2}}\left(\frac{t^2}{(1-(ab)^{1/2}t)^{l+1}} - \frac{t^2}{(1+(ab)^{1/2}t)^{l+1}}\right). \tag{3.9}$$

Proof. Let l be a fixed positive integer. From (2.1) and (2.7), $q_n(l) = 0$ for $0 \leq n < 2l+1$, $q_{2l+1}(l) = q_{2l+1}$, and $q_{2l+2}(l) = q_{2l+2}$, and

$$q_n(l) = \begin{cases} aq_{n-1}(l) + q_{n-2}(l) - a\binom{n-l-3}{l}(ab)^{\lfloor \frac{n-3}{2} \rfloor - l}, & \text{if } n \equiv 0 \pmod 2; \\ bq_{n-1}(l) + q_{n-2}(l) - \binom{n-l-3}{l}(ab)^{\lfloor \frac{n-3}{2} \rfloor - l}, & \text{if } n \equiv 1 \pmod 2. \end{cases} \tag{3.10}$$

Now let

$$s_0 = q_{2l+1}(l) = q_{2l+1}, \quad s_1 = q_{2l+2}(l) = q_{2l+1}, \text{ and}$$

$$s_n = q_{n+2l+1}(l).$$

Also let

$$r_0 = r_1 = 0 \text{ and } r_n = \binom{n+l-2}{n-2}(ab)^{\lfloor \frac{n}{2} \rfloor - 1}.$$

The generating function of the sequence $\{-r_n\}$ is

$$G(t) = -\frac{1}{2}\left(\frac{t^2}{(1-(ab)^{1/2}t)^{l+1}}(1+(ab)^{-1/2}) + \frac{t^2}{(1+(ab)^{1/2}t)^{l+1}}(1-(ab)^{-1/2})\right)$$

See [14, p. 355] and bisection generating functions [7]. Thus, from Lemma 3.1, we get the generating function $Q_l(t)$ of sequence $\{q_n(l)\}_{n=0}^{\infty}$. □

4. Conclusion

In this paper, we introduce the notion of bi-periodic incomplete Fibonacci numbers, and we obtain new identities. An open question is to evaluate the right sum in Proposition 2.5. On the other hand, in [9], authors introduced the bi-periodic Lucas numbers. They are defined by the recurrence relation

$$p_0 = 2, \quad p_1 = 1, \quad p_n = \begin{cases} ap_{n-1} + p_{n-2}, & \text{if } n \equiv 0 \pmod 2; \\ bp_{n-1} + p_{n-2}, & \text{if } n \equiv 1 \pmod 2; \end{cases} \quad n \geqslant 2. \tag{4.1}$$

It would be interesting to study the bi-periodic incomplete Lucas numbers and research their properties.

Acknowledgements. The author thanks the anonymous referee for his careful reading of the manuscript and his fruitful comments and suggestions.

References

[1] Alp, M., Irmak, N., Szalay, L., Two-Periodic ternary recurrences and their Binet-formula, *Acta Math. Univ. Comenianae* 2(2012), 227–232.

[2] Djordjević, G.B., Generating functions of the incomplete generalized Fibonacci and generalized Lucas numbers, *Fibonacci Quart.* 42(2)(2004), 106–113.

[3] Djordjević, G.B., Srivastava, H.M., Incomplete generalized Jacobsthal and Jacobsthal-Lucas numbers, *Math. Comput. Modelling* 42(9-10)(2005).

[4] Edson, M., Yayenie, O., A new generalization of Fibonacci sequences and extended Binet's Formula, *Integers*, 9(A48)(2009), 639–654.

[5] Edson, M., Lewis, S., Yayenie, O., The k-periodic Fibonacci sequence and extended Binet's formula, *Integers*, 11(A32)(2011), 739–751.

[6] Filipponi, P., Incomplete Fibonacci and Lucas numbers, *Rend. Circ. Mat. Palermo* 45(2)(1996), 37–56.

[7] Hoggatt,V., Anaya, J., A primer for the Fibonacci number Part XI, In: *A primer for the Fibonacci numbers*, Ed. Bicknell, M., Hoggatt, V., The Fibonacci Association, (1973), 159–165.

[8] Irmak, N., Szalay, L., On k-periodic binary recurrences, *Ann. Math. Inform.* 40(2012), 25–35.

[9] Irmak, N., Alp, M., Some identities for generalized Fibonacci and Lucas sequences, *Hacet. J. Math. Stat.*, 42(4)(2013), 331–338.

[10] Koshy, T., *Fibonacci and Lucas Numbers with Applications*, A Wiley-Interscience Publication, 2001.

[11] Pintér, Á., Srivastava, H.M., Generating functions of the incomplete Fibonacci and Lucas numbers, *Rend. Circ. Mat. Palermo* 48(2)(1999), 591–596.

[12] Ramírez, J., Incomplete Generalized Fibonacci and Lucas Polynomials, arXiv:1308.4192, (2013).

[13] Ramírez, J., Incomplete k-Fibonacci and k-Lucas Numbers, *Chinese Journal of Mathematics*, (2013).

[14] Srivastava, H.M., Manocha, H.L., *A Treatise on Generating Functions*, Halsted Pres (Ellis Horwood Limited, Chichester), John Wiley and Sons, New York, 1984.

[15] Tasci, D., Cetin Firengiz, M., Incomplete Fibonacci and Lucas p-numbers, *Math. Comput. Modelling* 52(9-10)(2010), 1763–1770.

[16] Tasci, D., Cetin Firengiz, M., Tuglu, N., Incomplete Bivariate Fibonacci and Lucas p-polynomials, *Discrete Dynam. Nat. Soc.* vol 2012, (2012).

[17] Yayenie, O., A note on generalized Fibonacci sequence, *Applied. Math. Comp.* 217(12)(2011), 5603–5611.

Annales Mathematicae et Informaticae
42 (2013) pp. 93–101
http://ami.ektf.hu

On the zeros of some polynomials with combinatorial coefficients

Mark Shattuck

Department of Mathematics, University of Tennessee, Knoxville
shattuck@math.utk.edu

Submitted March 1, 2013 — Accepted May 23, 2013

Abstract

We consider two general classes of second-order linear recurrent sequences and the polynomials whose coefficients belong to a sequence in either of these classes. We show for each such sequence $\{a_i\}_{i\geq 0}$ that the polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$ always has the smallest possible number of real zeros, that is, none when the degree is even and one when the degree is odd. Among the sequences then for which this is true are the Motzkin, Riordan, Schröder, and Delannoy numbers.

Keywords: zeros of polynomials, Motzkin number, Schröder number

MSC: 11C08, 13B25

1. Introduction

Garth, Mills, and Mitchell [3] considered the Fibonacci coefficient polynomial $p_n(x) = F_1x^n + F_2x^{n-1} + \cdots + F_nx + F_{n+1}$ and showed that it has no real zeros if n is even and exactly one real zero if n is odd. Later, Mátyás [5, 6] extended this result to polynomials whose coefficients are given by more general second order recurrences (having constant coefficients), and Mátyás and Szalay [7] showed the same holds true for the Tribonacci coefficient polynomials $q_n(x) = T_2x^n + T_3x^{n-1} + \cdots + T_{n+1}x + T_{n+2}$. The latter result has been extended to k-Fibonacci polynomials by Mansour and Shattuck [4].

In the apparent absence of a general criterion for determining when a polynomial having real coefficients has the smallest possible number of real zeros, one might wonder as to what other sequences a_i for which this result holds true for the polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$. Here, we consider this question for sequences belonging to two general classes and show that it holds in all cases. Among the sequences belonging to these classes are the Motzkin [9], Riordan [1], Schröder [2], and Delannoy [10].

We note that the sequences under consideration in the current paper are all given by second order linear recurrences, but with variable instead of constant coefficients. Thus, instead of multiplying $f(x)$ by a characteristic polynomial to obtain another polynomial whose coefficients are mostly zero (as was done in [3] and in subsequent papers in the case when $a_i = F_i$ for all i), we first apply a different linear operator to f, namely, one that is of a first-order differential nature. This yields a differential equation for f which can then be used to express it in an integral form that we find more convenient.

Recall that the Motzkin numbers m_n and the Riordan numbers r_n are given by

$$(n+2)m_n = (2n+1)m_{n-1} + 3(n-1)m_{n-2}, \qquad n \geq 2,$$

with $m_0 = m_1 = 1$, and by

$$(n+1)r_n = (n-1)(2r_{n-1} + 3r_{n-2}), \qquad n \geq 2,$$

with $r_0 = 1$ and $r_1 = 0$. See entries A001006 and A005043 in OEIS [8].

Recall that the (little) Schröder numbers s_n and the (central) Delannoy numbers d_n are given by

$$(n+1)s_n = 3(2n-1)s_{n-1} - (n-2)s_{n-2}, \qquad n \geq 2,$$

with $s_0 = s_1 = 1$, and by

$$nd_n = 3(2n-1)d_{n-1} - (n-1)d_{n-2}, \qquad n \geq 2,$$

with $d_0 = 1$ and $d_1 = 3$. See entries A001003 and A001850 in [8].

We will prove the following result in the next two sections.

Theorem 1.1. *If a_i denotes any one of the sequences m_i, r_i, s_i, or d_i, then the polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$, $n \geq 2$, has no real zeros if n is even and one real zero if n is odd.*

The first two parts of Theorem 1.1 are shown in the next section as special cases of a more general result, while the last two parts are shown in a comparable manner in the third section.

2. Motzkin family polynomials

Let u_n, $n \geq 0$, denote the sequence defined by the recurrence

$$(n+a)u_n = (2(n-1)+b)u_{n-1} + 3(n-1)u_{n-2}, \qquad n \geq 2, \tag{2.1}$$

with the initial values $u_0 = 1$ and $u_1 = c$, where a, b, and c are constants. Note that u_n reduces to the Motzkin sequence when $a = 2$, $b = 3$, $c = 1$ and to the Riordan sequence when $a = 1$, $b = c = 0$. Let

$$f_n(x) = \sum_{i=0}^{n} u_i x^i, \qquad n \geq 0.$$

We will need the following integral representation of $f_n(x)$.

Lemma 2.1. *If $-1 < x < 0$, then*

$$f_n(x) = j(x)^{-1} \left(\int_{x_o}^{x} \frac{j(t)h_n(t)}{t(1+t)(1-3t)} dt + j(x_o) f_n(x_o) \right), \tag{2.2}$$

where $-1 < x_o < 0$ is any fixed number,

$$j(x) = |x|^a (1+x)^{\frac{3-a-b}{4}} (1-3x)^{\frac{1-3a+b}{4}},$$

and

$$h_n(x) = a + ((1+a)c - b)x - (n+a+1)u_{n+1}x^{n+1} - (3n+3)u_n x^{n+2}.$$

Proof. Let $f = f_n(x)$. By the recurrence (2.1), we have

$$\begin{aligned}
&xf' + af - 2x^2 f' - bxf - 3x^3 f' - 3x^2 f \\
&\quad = \sum_{i=1}^{n} i u_i x^i + a \sum_{i=0}^{n} u_i x^i - 2 \sum_{i=2}^{n+1} (i-1) u_{i-1} x^i - b \sum_{i=1}^{n+1} u_{i-1} x^i \\
&\quad\quad - 3 \sum_{i=3}^{n+2} (i-2) u_{i-2} x^i - 3 \sum_{i=2}^{n+2} u_{i-2} x^i \\
&\quad = a + ((1+a)c - b)x - ((2n+b)u_n + 3nu_{n-1})x^{n+1} - (3n+3)u_n x^{n+2} \\
&\quad\quad + \sum_{i=2}^{n} \left[(i+a)u_i - (2(i-1)+b)u_{i-1} - 3(i-1)u_{i-2} \right] x^i \\
&\quad = a + ((1+a)c - b)x - (n+1+a)u_{n+1}x^{n+1} - (3n+3)u_n x^{n+2},
\end{aligned}$$

where the prime denotes differentiation. The final equality may be rewritten in the form

$$f_n'(x) + \frac{a - bx - 3x^2}{x(1+x)(1-3x)} f_n(x) = \frac{h_n(x)}{x(1+x)(1-3x)}, \tag{2.3}$$

where $h_n(x)$ is as given. Note that, by partial fractions, we have

$$\frac{a - bx - 3x^2}{x(1+x)(1-3x)} = \frac{a}{x} + \frac{3-a-b}{4(1+x)} - \frac{3-9a+3b}{4(1-3x)},$$

which gives the antiderivative

$$\int \frac{a - bx - 3x^2}{x(1+x)(1-3x)} dx = \log\left(|x|^a |1+x|^{\frac{3-a-b}{4}} |1-3x|^{\frac{1-3a+b}{4}}\right).$$

Formula (2.2) now follows from solving the first order linear differential equation (2.3) on the open interval $(-1, 0)$ by the usual method. □

In the next two lemmas, we assume that the constants a, b, and c used in defining u_n are non-negative real numbers such that $a \leq b + 1$ and $c \leq \frac{b}{a+1}$.

Lemma 2.2. *If $n \geq 2$ is even, then the polynomial $f_n(x) = \sum_{i=0}^{n} u_i x^i$ has no zeros on the interval $(-\infty, -1]$.*

Proof. First note that $f_2(-1) > 0$ if and only if $a + 5 > (a - b)c$. The latter inequality clearly holds if $a - b \leq 0$. It also holds if $a - b > 0$, since in this case we have

$$(a-b)c \leq \frac{(a-b)b}{a+1} \leq \frac{a^2}{4(a+1)} < a + 5.$$

Let $k_n(x) = f_n(-x)$. Observe that

$$k_2(x) = 1 - u_1 x + u_2 x^2 > 0, \qquad x \geq 1,$$

since $k_2(1) > 0$ and

$$k_2'(x) = 2u_2 x - u_1 \geq 2u_2 - u_1 = \frac{(2b - a + 2)c + 6}{a + 2} \geq \frac{(b+1)c + 6}{a + 2} > 0.$$

Using recurrence (2.1) and the assumption $a - b \leq 1$, one can show by induction that $u_n > u_{n-1}$ if $n \geq 4$. If $x \geq 1$, then

$$\begin{aligned} k_n(x) &= (1 - u_1 x + u_2 x^2) + \sum_{i=2}^{\frac{n}{2}} (u_{2i} x^{2i} - u_{2i-1} x^{2i-1}) \\ &\geq (1 - u_1 x + u_2 x^2) + \sum_{i=2}^{\frac{n}{2}} (u_{2i} - u_{2i-1}) x^{2i-1} > 0, \end{aligned}$$

being the sum of positive terms, whence $f_n(x) > 0$ for $x \leq -1$. □

Lemma 2.3. *If $n \geq 3$ is odd and $f_n(-1) \geq 0$, then $f_n(x)$ has exactly one negative zero.*

Proof. Let $k_n(x) = f_n(-x)$, $x > 0$. Note that as in the previous proof, we have $u_n \geq u_{n-1}$ if $n \geq 3$, with equality possible only when $n = 3$. Therefore, if $0 < x < 1$, we have

$$u_{2i+1} x^{2i+1} - u_{2i} x^{2i} = u_{2i+1} x^{2i}(x - 1) + x^{2i}(u_{2i+1} - u_{2i}) < u_{2i+1} - u_{2i}, \qquad i \geq 1,$$

which implies for $n \geq 3$ odd that

$$k_n(x) = (u_0 - u_1x + u_2x^2 - u_3x^3) + \sum_{i=2}^{\frac{n-1}{2}} (u_{2i}x^{2i} - u_{2i+1}x^{2i+1})$$

$$> (u_0 - u_1 + u_2 - u_3) + \sum_{i=2}^{\frac{n-1}{2}} (u_{2i} - u_{2i+1}) = k_n(1) \geq 0.$$

Thus, $k_n(x) > 0$ if $0 < x < 1$.

If $x \geq 1$, then

$$k'_n(x) = -u_1 + \sum_{i=1}^{\frac{n-1}{2}} (2iu_{2i}x^{2i-1} - (2i+1)u_{2i+1}x^{2i}) < 0,$$

since $2iu_{2i} < (2i+1)u_{2i+1}$ for $i \geq 1$. Then $k_n(x)$ has one zero for $x \geq 1$ since $k_n(1) \geq 0$ and $k'_n(x) < 0$, which implies $f_n(x)$ has one negative real zero. □

We now prove the main result of this section.

Theorem 2.4. *Suppose a, b, and c are non-negative real numbers such that $a \leq b+1$ and $c \leq \frac{b}{a+1}$. If $n \geq 2$, then the polynomial $f_n(x) = \sum_{i=0}^{n} u_i x^i$ has no real zeros if n is even and one real zero if n is odd.*

Proof. Clearly, $f_n(x)$ has no positive zeros since it has non-negative coefficients. First suppose n is even. By Lemma 2.2, we may restrict our attention to the case $-1 < x < 0$. By Lemma 2.1, we have

$$f_n(x) = j(x)^{-1}\alpha_n(x), \qquad -1 < x < 0, \tag{2.4}$$

where

$$\alpha_n(x) = \int_{x_o}^{x} \frac{j(t)h_n(t)}{t(1+t)(1-3t)}dt + j(x_o)f_n(x_o), \qquad -1 < x < 0,$$

$x_o \in (-1, 0)$ is fixed, and $j(x), h_n(x)$ are as above. By (2.4), to complete the proof in the even case, it suffices to show that $\alpha_n(x) > 0$ for $-1 < x < 0$ as $j(x) > 0$ on this interval. Since $\alpha_n(x) = j(x)f_n(x)$, with $f_n(0), f_n(-1) > 0$, we first see that $\alpha_n(x) > 0$ for all x sufficiently close to either -1 or 0.

When n is even, note that the polynomial $h_n(x)$ has one negative zero, by Descartes' rule of signs and the assumption $c \leq \frac{b}{a+1}$. Since $h_n(0) \geq 0$, we must have either (i) $h_n(x) > 0$ if $-1 < x < 0$, or (ii) $h_n(x) > 0$ if $r < x < 0$ and $h_n(x) < 0$ if $-1 < x < r$, for some $r \in (-1, 0)$. Note that

$$\alpha'_n(x) = \frac{j(x)h_n(x)}{x(1+x)(1-3x)}, \qquad -1 < x < 0.$$

If (i) occurs, then $\alpha_n'(x) < 0$, which implies $\alpha_n(x) > 0$ for $-1 < x < 0$, since it is positive for all x sufficiently close to either endpoint of this interval. If (ii) occurs, then $\alpha_n'(x) > 0$ for $-1 < x < r$ and $\alpha_n'(x) < 0$ for $r < x < 0$, which again implies $\alpha_n(x) > 0$ for $-1 < x < 0$, since in this case the minimum value of $\alpha_n(x)$ on the interval is achieved as x approaches one of the endpoints.

Now suppose n is odd. We'll show in this case that $f_n(x)$ possesses exactly one negative zero. By Lemma 2.3, we may assume $f_n(-1) < 0$. Note that $f_n(-1) < 0$ implies $f_n(x) < 0$ for all $x \leq -1$ since $f_n'(x) > 0$ if $x \leq -1$ and n is odd. Thus, we may again restrict attention to when $-1 < x < 0$, and we'll show in this case that $\alpha_n(x)$, and thus $f_n(x)$, possesses exactly one zero. Note first that $\alpha_n(x)$ is positive for all x near zero and negative for all x near -1 since $f_n(0) > 0$ and $f_n(-1) < 0$.

We claim that $h_n(x)$ must possess at least one zero on the interval $(-1, 0)$ when $a > 0$. Suppose that this is not the case. By Descartes' rule and the assumption $c \leq \frac{b}{a+1}$, the polynomial $h_n(x)$ when n is odd has either two negative zeros or none at all. Then $h_n(0) \geq 0$ and $\lim_{x \to -\infty} h_n(x) = \infty$ would imply $h_n(x) > 0$ if $-1 < x < 0$ and thus $\alpha_n'(x) < 0$. But this would contradict the fact that $\alpha_n(x)$ is negative for x near -1 and positive for x near 0. Thus, $h_n(x)$ possesses two negative zeros and at least one of these zeros lies in the interval $(-1, 0)$ when $a > 0$. Therefore, we must have either (a) $h_n(x) > 0$ if $r < x < 0$ and $h_n(x) < 0$ if $-1 < x < r$ for some $-1 < r < 0$, or (b) $h_n(x) > 0$ if $-1 < x < r$ or $s < x < 0$, with $h_n(x) < 0$ if $r < x < s$ for some $-1 < r < s < 0$.

If (a) occurs, then $\alpha_n(x)$ initially increases going to the right from $x = -1$ and crosses the x-axis before it decreases in its approach to $x = 0$ from the left. If (b) occurs, then $\alpha_n(x)$ traces out a similar curve in going from $x = -1$ to $x = 0$ except that it initially decreases some from its negative value near $x = -1$ before it starts to increase. In each case, we see that $\alpha_n(x)$, and thus $f_n(x)$, possesses exactly one zero for $-1 < x < 0$ when $a > 0$.

If $a = 0$, then a similar argument applies if $c < b$. If $a = 0$ and $c = b$, then $h_n(x)$ possesses exactly one negative zero, which we will denote by t. Note that $h_n(x) < 0$ if $t < x < 0$ and $h_n(x) > 0$ if $x < t$ since $h_n(0) = 0$ and $\lim_{x \to -\infty} h_n(x) = \infty$. If $t \leq -1$, then $h_n(x) < 0$ on $(-1, 0)$ and thus $\alpha_n(x)$ is increasing on $(-1, 0)$, which implies it has a single zero there. If $-1 < t < 0$, then $\alpha_n(x)$ is decreasing on $(-1, t)$ and increasing on $(t, 0)$, which yields the same conclusion. This completes the odd case and the proof. □

Taking $a = 2, b = 3, c = 1$ and $a = 1, b = c = 0$ in the prior theorem gives the first two parts of Theorem 1.1 above concerning the Motzkin and the Riordan sequences.

3. Schröder family polynomials

Let v_n, $n \geq 0$, denote the sequence defined by the recurrence

$$(n + a)v_n = 3(2n - 1)v_{n-1} - (n - 2 + b)v_{n-2}, \qquad n \geq 2, \qquad (3.1)$$

with the initial values $v_0 = 1$ and $v_1 = c$, where a, b, and c are constants. Note that v_n reduces to the (little) Schröder sequence when $a = 1, b = 0, c = 1$ and to the (central) Delannoy sequence when $a = 0, b = 1, c = 3$. Let

$$g_n(x) = \sum_{i=0}^{n} v_i x^i, \qquad n \geq 0.$$

We will need the following integral representation of $g_n(x)$.

Lemma 3.1. *If $x < 0$, then*

$$g_n(x) = j(x)^{-1} \left(\int_{x_o}^{x} \frac{j(t)h_n(t)}{t(1-6t+t^2)} dt + j(x_o)g_n(x_o) \right), \tag{3.2}$$

where $x_o < 0$ is any fixed number,

$$j(x) = |x|^a (1-6x+x^2)^{\frac{b-a}{2}} \left(\frac{x-3-2\sqrt{2}}{x-3+2\sqrt{2}} \right)^{\frac{3(a+b-1)}{4\sqrt{2}}},$$

and

$$h_n(x) = a + ((1+a)c-3)x - (n+a+1)v_{n+1}x^{n+1} + (n+b)v_n x^{n+2}.$$

Proof. Let $g = g_n(x)$. By (3.1), we have

$$\begin{aligned}
&xg' + ag - 6x^2g' - 3xg + x^3g' + bx^2g \\
&\quad = a + ((1+a)c-3)x - (3(2n+1)v_n - (n-1+b)v_{n-1})x^{n+1} + (n+b)v_n x^{n+2} \\
&\qquad + \sum_{i=2}^{n} \left[(i+a)v_i - 3(2i-1)v_{i-1} + (i-2+b)v_{i-2} \right] x^i \\
&\quad = a + ((1+a)c-3)x - (n+1+a)v_{n+1}x^{n+1} + (n+b)v_n x^{n+2}.
\end{aligned}$$

This may be rewritten in the form

$$g_n'(x) + \frac{a-3x+bx^2}{x(1-6x+x^2)} g_n(x) = \frac{h_n(x)}{x(1-6x+x^2)}, \qquad x < 0, \tag{3.3}$$

where $h_n(x)$ is as given. Note that, by partial fractions, we have

$$\begin{aligned}
\frac{a-3x+bx^2}{x(1-6x+x^2)} &= \frac{a}{x} + \frac{(b-a)(x-3)}{1-6x+x^2} \\
&\quad + \frac{3(a+b-1)}{4\sqrt{2}} \left(\frac{1}{x-3-2\sqrt{2}} - \frac{1}{x-3+2\sqrt{2}} \right),
\end{aligned}$$

which gives the antiderivative

$$\int \frac{a-3x+bx^2}{x(1-6x+x^2)}dx = \log\left(|x|^a|1-6x+x^2|^{\frac{b-a}{2}}\left|\frac{x-3-2\sqrt{2}}{x-3+2\sqrt{2}}\right|^{\frac{3(a+b-1)}{4\sqrt{2}}}\right).$$

Formula (3.2) now follows from solving (3.3) for $x<0$ by the usual method. □

Theorem 3.2. *Suppose a, b, and c satisfy $0 \le a \le 2$, $0 \le b \le 7-a$, and $1 \le c \le \frac{3}{a+1}$. If $n \ge 2$, then the polynomial $g_n(x) = \sum_{i=0}^{n} v_i x^i$ has no real zeros if n is even and one real zero if n is odd.*

Proof. Using (3.1) and the assumptions $c \ge 1$ and $a+b \le 7$, one can show by induction that $v_n \ge v_{n-1}$ for all $n \ge 1$, with equality possible only when $n=1$ or $n=2$. Then $g_n(x)$ clearly has no positive zeros since it has positive coefficients.

Suppose n is even. If $x \le -1$, then

$$g_n(x) = 1 + \sum_{i=1}^{\frac{n}{2}} x^{2i-1}(v_{2i-1}+v_{2i}x) > 0,$$

so we may restrict attention to the case $-1<x<0$. By Lemma 3.1, we have

$$g_n(x) = j(x)^{-1}\beta_n(x), \qquad x<0, \tag{3.4}$$

where

$$\beta_n(x) = \int_{x_o}^{x} \frac{j(t)h_n(t)}{t(1-6t+t^2)}dt + j(x_o)g_n(x_o), \qquad x<0,$$

x_o is a fixed negative number, and $j(x), h_n(x)$ are as stated in this lemma.

Since $j(x)>0$ for $x<0$, to complete the proof in the even case, it suffices to show that $\beta_n(x)>0$ for $x<0$, by (3.4). Since $\beta_n(x) = j(x)g_n(x)$, with $g_n(0), g_n(-1) > 0$, we see that $\beta_n(x)>0$ for $x=-1$ and all x sufficiently close to 0. Next observe that

$$\beta_n'(x) = \frac{j(x)h_n(x)}{x(1-6x+x^2)}, \qquad x<0,$$

and that $h_n(x)>0$ if $x<0$ for n even, by the assumption $c \le \frac{3}{a+1}$. Thus $\beta_n'(x)<0$, which implies $\beta_n(x)>0$ for $-1<x<0$, since it is positive at $x=-1$ and for all x near 0. This completes the even case.

Now suppose n is odd. We'll show in this case that $g_n(x)$ possesses exactly one negative zero. Note first that $g_n(-1)<0$, which implies $g_n(x)<0$ for all $x \le -1$ since $g_n'(x)>0$ if $x \le -1$ and n is odd. To complete the proof, we will show that $\beta_n(x)$, and thus $g_n(x)$, possesses exactly one zero on the interval $(-1,0)$. Observe that $\beta_n(x)$ is positive for all x sufficiently close to zero and that $\beta_n(-1)<0$ since $g_n(0)>0$ and $g_n(-1)<0$.

By Descartes' rule, the polynomial $h_n(x)$ has one negative zero when n is odd and $a > 0$. Reasoning as in the proof of Theorem 2.4 above then shows that this zero must belong to the interval $(-1, 0)$. Since $h_n(0) > 0$, it must be the case that $h_n(x) > 0$ if $r < x < 0$ and $h_n(x) < 0$ if $-1 < x < r$ for some $-1 < r < 0$. Reasoning now as in the proof of Theorem 2.4 shows that $\beta_n(x)$, and thus $g_n(x)$, possesses exactly one zero on the interval $(-1, 0)$ when $a > 0$. A similar argument applies to the case when $a = 0$ and $c < 3$. If $a = 0$ and $c = 3$, then $h_n(x) < 0$ if $x < 0$, which implies $\beta_n(x)$ is increasing on $(-1, 0)$ and thus has one zero there. This completes the odd case and the proof. □

Taking $a = 1, b = 0, c = 1$ and $a = 0, b = 1, c = 3$ in the prior theorem gives the last two parts of Theorem 1.1 above concerning the Schröder and Delannoy sequences.

References

[1] Chen, W.Y.C., Deng, E.Y.P., Yang, L.L.M., Riordan paths and derangements, *Discrete Math.* **308** (2008), 2222–2227.

[2] Foata, D., Zeilberger, D., A classic proof of a recurrence for a very classical sequence, *J. Combin. Theory Ser. A* **80** (1997), 380–384.

[3] Garth, D., Mills, D., Mitchell, P., Polynomials generated by the Fibonacci sequence, *J. Integer Seq.* **10** (2007), Art. 07.6.8.

[4] Mansour, T., Shattuck, M., Polynomials whose coefficients are k-Fibonacci numbers, *Ann. Math. Inform.* **40** (2012), 57–76.

[5] Mátyás, F., On the generalization of the Fibonacci-coefficient polynomials, *Ann. Math. Inform.* **34** (2007), 71–75.

[6] Mátyás, F., Further generalizations of the Fibonacci-coefficient polynomials, *Ann. Math. Inform.* **35** (2008), 123–128.

[7] Mátyás, F., Szalay, L., A note on Tribonacci-coefficient polynomials, *Ann. Math. Inform.* **38** (2011), 95–98.

[8] Sloane, N.J., The On-Line Encyclopedia of Integer Sequences, `http://oeis.org`, 2010.

[9] Sulanke, R.A., Bijective recurrences for Motzkin paths, *Adv. in Appl. Math.* **27** (2001), 627–640.

[10] Sulanke, R.A., Objects counted by the central Delannoy numbers, *J. Integer Seq.* **6** (2003), Art. 03.1.5.

Methodological papers

Annales Mathematicae et Informaticae
42 (2013) pp. 105–117
http://ami.ektf.hu

Averaging sums of powers of integers and Faulhaber polynomials

José Luis Cereceda[a]

[a]Distrito Telefónica, Madrid, Spain
jl.cereceda@movistar.es

Submitted May 13, 2013 — Accepted September 29, 2013

Abstract

As an application of Faulhaber's theorem on sums of powers of integers and the associated Faulhaber polynomials, in this article we provide the solution to the following two questions: (1) when is the average of sums of powers of integers itself a sum of the first n integers raised to a power? and (2), when is the average of sums of powers of integers itself a sum of the first n integers raised to a power, times the sum of the first n squares? In addition to this, we derive a family of recursion formulae for the Bernoulli numbers.

Keywords: sums of powers of integers, Faulhaber polynomials, matrix inversion, Bernoulli numbers

MSC: 11C08, 11B68

1. Introduction

Recently Pfaff [1] investigated the solutions of the equation

$$\frac{\sum_{i=1}^{n} i^a + \sum_{i=1}^{n} i^b}{2} = \left(\sum_{i=1}^{n} i \right)^c, \tag{1.1}$$

for positive integers a, b, and c, and found that the only solution (a, b, c) to (1.1) with $a \neq b$ is $(5, 7, 4)$ (the remaining solutions being the trivial one $(1, 1, 1)$ and the well-known solution $(3, 3, 2)$). Furthermore, Pfaff provided some necessary conditions for

$$\frac{\sum_{i=1}^{n} i^{a_1} + \sum_{i=1}^{n} i^{a_2} + \cdots + \sum_{i=1}^{n} i^{a_{m-1}}}{m-1} = \left(\sum_{i=1}^{n} i \right)^{a_m}, \tag{1.2}$$

to hold. Specifically, by assuming that $a_1 \leq a_2 \leq \cdots \leq a_{m-p-1} < a_{m-p} = \cdots = a_{m-1}$ (with $a_1, a_2, \ldots, a_m$ positive integers), Pfaff showed that any solution to (1.2) must fulfil the condition

$$\frac{p}{m-1} = \frac{a_m}{2^{a_m - 1}}. \tag{1.3}$$

There are infinitely many solutions to (1.3). For example, for $a_m = 8$, the set of solutions to (1.3) is given by $(a_m, p, m-1) = (8, p, 16p)$, with $p \geq 1$. However, as Pfaff himself pointed out [1], it is not known if any given solution to (1.3) also yields a solution to (1.2), so that solving this problem for $m > 3$ will require some other approach. In this article we show that, for any given value of $a_m \geq 1$, there is indeed a unique solution to equation (1.2), on the understanding that the fraction $\frac{p}{m-1}$ is given in its lowest terms. Interestingly, this is done by exploiting the properties of the coefficients of the so-called Faulhaber polynomials [2, 3, 4, 5, 6]. Although there exist more direct ways to arrive at the solution of equation (1.2) (for example, by means of the binomial theorem or by mathematical induction), our pedagogical approach here will serve to introduce the (relatively lesser known) topic of the Faulhaber polynomials to a broad audience.

In addition to the equation (1.2) considered by Pfaff, we also give the solution to the closely related equation

$$\frac{\sum_{i=1}^{n} i^{a_1} + \sum_{i=1}^{n} i^{a_2} + \cdots + \sum_{i=1}^{n} i^{a_{m-1}}}{m-1} = \left(\sum_{i=1}^{n} i^2 \right) \left(\sum_{i=1}^{n} i \right)^{a_m}, \tag{1.4}$$

where now $a_m \geq 0$. Obviously, for $a_m = 0$, we have the trivial solution $a_1 = a_2 = \cdots = a_{m-1} = 2$. In general, it turns out that all the powers $a_1, a_2, \ldots, a_{m-1}$ on the left-hand side of (1.4) must be even integers, whereas those appearing in the left-hand side of (1.2) must be odd integers. This is a straightforward consequence of the following theorem.

2. Faulhaber's theorem on sums of powers of integers

Let us denote by S_r the sum of the first n positive integers each raised to the integer power $r \geq 0$, $S_r = \sum_{i=1}^{n} i^r$. The key ingredient in our discussion is an old result concerning the S_r's which can be traced back to Johann Faulhaber (1580–1635), an early German algebraist who was a close friend of both Johannes Kepler and René Descartes. Faulhaber discovered that, for even powers $r = 2k$ ($k \geq 1$), S_{2k} can be put in the form

$$S_{2k} = S_2 \big[F_0^{(2k)} + F_1^{(2k)} S_1 + F_2^{(2k)} S_1^2 + \cdots + F_{k-1}^{(2k)} S_1^{k-1} \big], \tag{2.1}$$

whereas, for odd powers $r = 2k+1$ $(k \geq 1)$, S_{2k+1} can be expressed as

$$S_{2k+1} = S_1^2\big[F_0^{(2k+1)} + F_1^{(2k+1)}S_1 + F_2^{(2k+1)}S_1^2 + \cdots + F_{k-1}^{(2k+1)}S_1^{k-1}\big], \tag{2.2}$$

where $\{F_j^{(2k)}\}$ and $\{F_j^{(2k+1)}\}$, $j = 0, 1, \ldots, k-1$, are sets of numerical coefficients. Equations (2.1) and (2.2) can be rewritten in compact form as

$$S_{2k} = S_2 F^{(2k)}(S_1), \tag{2.3}$$

$$S_{2k+1} = S_1^2 F^{(2k+1)}(S_1), \tag{2.4}$$

where both $F^{(2k)}(S_1)$ and $F^{(2k+1)}(S_1)$ are polynomials in S_1 of degree $k-1$. Following Edwards [2] we refer to them as Faulhaber polynomials and, by extension, we call $F_j^{(2k)}$ and $F_j^{(2k+1)}$ the Faulhaber coefficients. Next we quote the first instances of S_{2k} and S_{2k+1} in Faulhaber form as

$$\begin{aligned}
S_2 &= S_2,\\
S_3 &= S_1^2,\\
S_4 &= S_2\big[-\tfrac{1}{5} + \tfrac{6}{5}S_1\big],\\
S_5 &= S_1^2\big[-\tfrac{1}{3} + \tfrac{4}{3}S_1\big],\\
S_6 &= S_2\big[\tfrac{1}{7} - \tfrac{6}{7}S_1 + \tfrac{12}{7}S_1^2\big],\\
S_7 &= S_1^2\big[\tfrac{1}{3} - \tfrac{4}{3}S_1 + 2S_1^2\big],\\
S_8 &= S_2\big[-\tfrac{1}{5} + \tfrac{6}{5}S_1 - \tfrac{8}{3}S_1^2 + \tfrac{8}{3}S_1^3\big],\\
S_9 &= S_1^2\big[-\tfrac{3}{5} + \tfrac{12}{5}S_1 - 4S_1^2 + \tfrac{16}{5}S_1^3\big],\\
S_{10} &= S_2\big[\tfrac{5}{11} - \tfrac{30}{11}S_1 + \tfrac{68}{11}S_1^2 - \tfrac{80}{11}S_1^3 + \tfrac{48}{11}S_1^4\big],\\
S_{11} &= S_1^2\big[\tfrac{5}{3} - \tfrac{20}{3}S_1 + \tfrac{34}{3}S_1^2 - \tfrac{32}{3}S_1^3 + \tfrac{16}{3}S_1^4\big].
\end{aligned}$$

Note that, from the expressions for S_5 and S_7, we quickly get that $\frac{S_5+S_7}{2} = S_1^4$.

Let us now write the equations (1.2) and (1.4) using the notation S_r for the sums of powers of integers,

$$\frac{S_{a_1} + S_{a_2} + \cdots + S_{a_{m-1}}}{m-1} = S_1^{a_m}, \tag{2.5}$$

and

$$\frac{S_{a_1} + S_{a_2} + \cdots + S_{a_{m-1}}}{m-1} = S_2 S_1^{a_m}. \tag{2.6}$$

Since $S_1 = n(n+1)/2$ and $S_2 = (2n+1)S_1/3$, from (2.1) and (2.2) we retrieve the well-known result that S_r is a polynomial in n of degree $r+1$. From this result, it in turn follows that the maximum index a_{m-1} on the left-hand side of (2.5) is given by $a_{m-1} = 2a_m - 1$, a condition already established in [1]. In fact, in order for equation (2.5) to hold, it is necessary that all the indices $a_1, a_2, \ldots, a_{m-1}$ appearing in the left-hand side of (2.5) be odd integers. To see this, suppose on the contrary that one of the indices is even, say a_j. Then, from (2.3) and (2.4), the left-hand side of (2.5) can be expressed as follows:

$$L(S_1, S_2) = \frac{S_2 F^{(a_j)}(S_1) + S_1^2 P(S_1)}{m-1},$$

where $F^{(a_j)}(S_1)$ and $P(S_1)$ are polynomials in S_1. On the other hand, for nonnegative integers u and v, it is clear that $S_2 S_1^u \neq S_1^v$ irrespective of the values of u and v, as $S_2 S_1^u$ (S_1^v) is a polynomial in n of odd (even) degree. This means that $S_2 F^{(a_j)}(S_1)$ cannot be reduced to a polynomial in S_1 from which we conclude, in particular, that $L(S_1, S_2) \neq S_1^{a_m}$.

Similarly, using (2.3) and (2.4) it can be seen that, in order for equation (2.6) to hold, all the indices $a_1, a_2, \ldots, a_{m-1}$ in the left-hand side of (2.6) have to be even integers, the maximum index a_{m-1} being given by $a_{m-1} = 2a_m + 2$.

3. Faulhaber's coefficients

The sets of coefficients $\{F_j^{(2k)}\}$ and $\{F_j^{(2k+1)}\}$ satisfy several remarkable properties, a number of which will be described below. As it happens with the binomial coefficients and the Pascal triangle, the properties of the Faulhaber coefficients are better appreciated and explored when they are arranged in a triangular array. In Table 1 we have displayed the set $\{F_j^{(2k+1)}\}$ for $k = 1, 2, \ldots, 10$, while the corresponding coefficients $\{F_j^{(2k)}\}$ (also for $k = 1, 2, \ldots, 10$) are given in Table 2. The numeric arrays in Tables 1 and 2 also can be viewed as lower triangular matrices, with the rows being labelled by k and the columns by j. The following list of properties of the Faulhaber coefficients are readily verified for the coefficients shown in Tables 1 and 2. They are, however, completely general.

1. The Faulhaber coefficients are nonzero rational numbers.

2. The entries in a row have alternating signs, the sign of the leading coefficient (which is situated on the main diagonal) being positive.

3. The sum of the entries in a row is equal to unity, $\sum_{j=0}^{k-1} F_j^{(2k+1)} = \sum_{j=0}^{k-1} F_j^{(2k)} = 1$.

4. The entries on the main diagonal are given by $F_{k-1}^{(2k+1)} = \frac{2^k}{k+1}$ and $F_{k-1}^{(2k)} = \frac{3 \cdot 2^{k-1}}{2k+1}$, and the entries in the $j = 0$ column are $F_0^{(2k+1)} = 2(2k+1)B_{2k}$ and $F_0^{(2k)} = 6B_{2k}$, where B_{2k} denotes the $2k$-th Bernoulli number. Furthermore, the entries in the $j = 1$ column are connected to those in the $j = 0$ column by the simple relations $F_1^{(2k+1)} = -4F_0^{(2k+1)} = -8(2k+1)B_{2k}$ and $F_1^{(2k)} = -6F_0^{(2k)} = -36B_{2k}$.

$k\backslash j$	0	1	2	3	4	5	6	7	8	9
1	1									
2	$-\frac{1}{3}$	$\frac{4}{3}$								
3	$\frac{1}{3}$	$-\frac{4}{3}$	2							
4	$-\frac{3}{5}$	$\frac{12}{5}$	-4	$\frac{16}{5}$						
5	$\frac{5}{3}$	$-\frac{20}{3}$	$\frac{34}{3}$	$-\frac{32}{3}$	$\frac{16}{3}$					
6	$-\frac{691}{105}$	$\frac{2764}{105}$	$-\frac{944}{21}$	$\frac{4592}{105}$	$-\frac{80}{3}$	$\frac{64}{7}$				
7	35	-140	$\frac{718}{3}$	$-\frac{704}{3}$	$\frac{448}{3}$	-64	16			
8	$-\frac{3617}{15}$	$\frac{14468}{15}$	$-\frac{4948}{3}$	$\frac{24304}{15}$	$-\frac{9376}{9}$	$\frac{1408}{3}$	$-\frac{448}{3}$	$\frac{256}{9}$		
9	$\frac{43867}{21}$	$-\frac{175468}{21}$	$\frac{1500334}{105}$	$-\frac{210656}{15}$	$\frac{45264}{5}$	$-\frac{144512}{35}$	$\frac{6944}{5}$	$-\frac{1024}{3}$	$\frac{256}{5}$	
10	$-\frac{1222277}{55}$	$\frac{4889108}{55}$	$-\frac{5016584}{33}$	$\frac{24655472}{165}$	$-\frac{3180688}{33}$	44096	-15040	$\frac{11776}{3}$	-768	$\frac{1024}{11}$

Table 1: The set of coefficients $\{F_j^{(2k+1)}\}$ for $1 \leq k \leq 10$.

5. There exists a relation between $F_j^{(2k+1)}$ and $F_j^{(2k)}$, namely,

$$F_j^{(2k+1)} = \frac{2(2k+1)}{3(j+2)} F_j^{(2k)}, \quad j = 0, 1, \dots, k-1. \tag{3.1}$$

This formula allows us to obtain the k-th row in Table 1 from the k-th row in Table 2, and vice versa.

6. The entries $F_{k-2}^{(2k+1)}, F_{k-3}^{(2k+1)}, \dots, F_0^{(2k+1)}$ within the k-th row in Table 1 can be successively obtained by the rule

$$\sum_{j=0}^{q} 2^j \binom{k+1-j}{2q+1-2j} F_{k-j-1}^{(2k+1)} = 0, \quad 1 \leq q \leq k-1, \tag{3.2}$$

given the initial condition $F_{k-1}^{(2k+1)} = \frac{2^k}{k+1}$. By applying the bijection (3.1) to the coefficients $F_{k-j-1}^{(2k+1)}$, one gets the corresponding rule for the entries in the k-th row in Table 2.

7. For any given $k \geq 3$, and for each $j = 0, 1, \dots, k-3$, we have

$$\sum_{r=1}^{k} \mathrm{odd}(r) \binom{k}{r} F_j^{(2k-r)} = 0, \tag{3.3}$$

where $\mathrm{odd}(r)$ restricts the summation to odd values of r, i.e., $\mathrm{odd}(r) = 1\,(0)$ for odd (even) r. Similarly, by applying the bijection (3.1) to the coefficients $F_j^{(2k-r)}$, it can be seen that, for any given $k \geq 1$ and for each $j = 0, 1, \dots, k-1$,

$$\sum_{r=0}^{k+1} \mathrm{even}(r) \binom{k+2}{r+1} (2k+3-r) F_j^{(2k+2-r)} = 0, \tag{3.4}$$

$k\backslash j$	0	1	2	3	4	5	6	7	8	9
1	1									
2	$-\frac{1}{5}$	$\frac{6}{5}$								
3	$\frac{1}{7}$	$-\frac{6}{7}$	$\frac{12}{7}$							
4	$-\frac{1}{5}$	$\frac{6}{5}$	$-\frac{8}{3}$	$\frac{8}{3}$						
5	$\frac{5}{11}$	$-\frac{30}{11}$	$\frac{68}{11}$	$-\frac{80}{11}$	$\frac{48}{11}$					
6	$-\frac{691}{455}$	$\frac{4146}{455}$	$-\frac{1888}{91}$	$\frac{328}{13}$	$-\frac{240}{13}$	$\frac{96}{13}$				
7	7	-42	$\frac{1436}{15}$	$-\frac{352}{3}$	$\frac{448}{5}$	$-\frac{224}{5}$	$\frac{64}{5}$			
8	$-\frac{3617}{85}$	$\frac{21702}{85}$	$-\frac{9896}{17}$	$\frac{12152}{17}$	$-\frac{9376}{17}$	$\frac{4928}{17}$	$-\frac{1792}{17}$	$\frac{384}{17}$		
9	$\frac{43867}{133}$	$-\frac{263202}{133}$	$\frac{3000668}{665}$	$-\frac{105328}{19}$	$\frac{407376}{665}$	$-\frac{216768}{95}$	$\frac{83328}{95}$	$-\frac{4608}{19}$	$\frac{768}{19}$	
10	$-\frac{174611}{55}$	$\frac{1047666}{55}$	$-\frac{10033168}{231}$	$\frac{12327736}{231}$	$-\frac{454384}{11}$	22048	$-\frac{60160}{7}$	$\frac{17664}{7}$	$-\frac{3840}{7}$	$\frac{512}{7}$

Table 2: The set of coefficients $\{F_j^{(2k)}\}$ for $1 \leq k \leq 10$.

where even$(r) = 1\,(0)$ for even (odd) r picks out the even power terms. Informally, we may call the property embodied in equations (3.3) and (3.4) the *sum-to-zero column property*, as the coefficients $F_j^{(2k-r)}$ $[F_j^{(2k+2-r)}]$ entering the summation in (3.3) [(3.4)] pertain to a given column j. This is to be distinguished from the *sum-to-zero row property* in equation (3.2), where the coefficients $F_{k-j-1}^{(2k+1)}$ belong to a given row k.

8. For completeness, next we write down the explicit formula for $F_j^{(2k+1)}$ which was originally obtained in [7, Section 12]. Adapting the notation in [7] to ours, we have that

$$F_j^{(2k+1)} = (-1)^j \frac{2^{j+2}}{j+2} \sum_{r=0}^{\lfloor j/2 \rfloor} \binom{2j+1-2r}{j+1} \binom{2k+1}{2r+1} B_{2k-2r}, \tag{3.5}$$

for $j = 0, 1, \ldots, k-1$, and where $\lfloor j/2 \rfloor$ denotes the floor function of $j/2$, namely the largest integer not greater than $j/2$. The set of coefficients $\{F_j^{(2k)}\}$ can then be found through relation (3.1). We shall use relation (3.5) in Section 6 to derive a family of recursion formulae for the Bernoulli numbers.

4. Averaging sums of powers of integers

Interestingly enough, the sum-to-zero column property in equations (3.3) and (3.4) provides the solution to the problem of averaging sums of powers of integers in equations (2.5) and (2.6). For the sake of brevity, next we focus on the connection between (3.3) and (2.5). An analogous reasoning can be made to establish the link between (3.4) and (2.6). To grasp the meaning of equation (3.3), consider a concrete example where $k = 7$. Then the column index j takes the values $j = 0, 1, 2, 3, 4$, and (3.3) gives rise to the following five equalities:

$$\begin{aligned}
\tbinom{7}{1}F_0^{(13)} + \tbinom{7}{3}F_0^{(11)} + \tbinom{7}{5}F_0^{(9)} + \tbinom{7}{7}F_0^{(7)} &= 0 \\
\tbinom{7}{1}F_1^{(13)} + \tbinom{7}{3}F_1^{(11)} + \tbinom{7}{5}F_1^{(9)} + \tbinom{7}{7}F_1^{(7)} &= 0 \\
\tbinom{7}{1}F_2^{(13)} + \tbinom{7}{3}F_2^{(11)} + \tbinom{7}{5}F_2^{(9)} + \tbinom{7}{7}F_2^{(7)} &= 0 \\
\tbinom{7}{1}F_3^{(13)} + \tbinom{7}{3}F_3^{(11)} + \tbinom{7}{5}F_3^{(9)} &= 0 \\
\tbinom{7}{1}F_4^{(13)} + \tbinom{7}{3}F_4^{(11)} &= 0.
\end{aligned}$$

For a reason that will become clear in just a moment, we add to this list of equalities a last one to include the value of $\binom{7}{1}F_5^{(13)}$, namely, $\binom{7}{1}F_5^{(13)} = 2^6$. Furthermore, we multiply the first equality by S_1^0, the second equality by S_1^1, the third equality by S_1^2, and so on, that is,

$$\begin{aligned}
\tbinom{7}{1}F_0^{(13)}S_1^0 + \tbinom{7}{3}F_0^{(11)}S_1^0 + \tbinom{7}{5}F_0^{(9)}S_1^0 + \tbinom{7}{7}F_0^{(7)}S_1^0 &= 0 \\
\tbinom{7}{1}F_1^{(13)}S_1^1 + \tbinom{7}{3}F_1^{(11)}S_1^1 + \tbinom{7}{5}F_1^{(9)}S_1^1 + \tbinom{7}{7}F_1^{(7)}S_1^1 &= 0 \\
\tbinom{7}{1}F_2^{(13)}S_1^2 + \tbinom{7}{3}F_2^{(11)}S_1^2 + \tbinom{7}{5}F_2^{(9)}S_1^2 + \tbinom{7}{7}F_2^{(7)}S_1^2 &= 0 \\
\tbinom{7}{1}F_3^{(13)}S_1^3 + \tbinom{7}{3}F_3^{(11)}S_1^3 + \tbinom{7}{5}F_3^{(9)}S_1^3 &= 0 \\
\tbinom{7}{1}F_4^{(13)}S_1^4 + \tbinom{7}{3}F_4^{(11)}S_1^4 &= 0 \\
\tbinom{7}{1}F_5^{(13)}S_1^5 &= 2^6 S_1^5.
\end{aligned}$$

Now we can see that the sum of the entries in the first column is just $\binom{7}{1}$ times the Faulhaber polynomial $F^{(13)}(S_1)$, the sum of the entries in the second column is $\binom{7}{3}$ times $F^{(11)}(S_1)$, the sum of the third column is $\binom{7}{5}$ times $F^{(9)}(S_1)$, and the sum of the fourth column is $\binom{7}{7}$ times $F^{(7)}(S_1)$. Then we have

$$\tbinom{7}{1}F^{(13)}(S_1) + \tbinom{7}{3}F^{(11)}(S_1) + \tbinom{7}{5}F^{(9)}(S_1) + \tbinom{7}{7}F^{(7)}(S_1) = 2^6 S_1^5.$$

Next we multiply both sides of this equation by S_1^2 and divide them by 2^6. Thus, taking into account (2.4), we finally obtain

$$\frac{\binom{7}{7}S_7 + \binom{7}{5}S_9 + \binom{7}{3}S_{11} + \binom{7}{1}S_{13}}{2^6} = S_1^7. \tag{4.1}$$

Since $\binom{7}{1} + \binom{7}{3} + \binom{7}{5} + \binom{7}{7} = 2^6$, the identity (4.1) constitutes the solution to (2.5) for the particular case $a_m = 7$. In this case we have that $\frac{p}{m-1} = \frac{7}{2^6}$, in accordance with condition (1.3).

In general, for an arbitrary exponent $a_m \geq 1$, the solution to equation (2.5) is given by

$$\frac{\sum_{r=1}^{a_m} \mathrm{odd}(r)\binom{a_m}{r} S_{2a_m - r}}{2^{a_m - 1}} = S_1^{a_m}. \tag{4.2}$$

A few comments are in order concerning the solution in (4.2). In the first place, by the constructive procedure we have used to obtain the solution (4.1) for the case $a_m = 7$, it should be clear that the solution (4.2) is unique for each $a_m \geq 1$, the quotient $\frac{p}{m-1}$ characterizing the solution being determined (when expressed in lowest terms) by the relation $\frac{p}{m-1} = \frac{a_m}{2^{a_m-1}}$. Secondly, for odd (even) a_m, the numerator of (4.2) involves $\frac{a_m+1}{2}$ ($\frac{a_m}{2}$) different sums S_j with j being an odd integer ranging in $a_m \leq j \leq 2a_m - 1$ ($a_m + 1 \leq j \leq 2a_m - 1$). Furthermore, the binomial coefficients fulfil the identity $\sum_{r=1}^{a_m} \text{odd}(r)\binom{a_m}{r} = 2^{a_m-1}$, thus ensuring that the overall number of terms appearing in the numerator of (4.2) equals 2^{a_m-1}.

For example, for $a_m = 3$, from (4.2) we get the solution $\frac{S_3+3S_5}{4} = S_1^3$, which was also found in [1]. For $a_m = 4$, noting that $\frac{p}{m-1} = \frac{4}{8} = \frac{1}{2}$, we get the solution $\frac{S_5+S_7}{2} = S_1^4$ which, as we saw, corresponds to the one denoted as $(a, b, c) = (5, 7, 4)$ in [1]. More sophisticated examples are, for instance,

$$\frac{S_9 + 36S_{11} + 126S_{13} + 84S_{15} + 9S_{17}}{256} = S_1^9,$$

and

$$\frac{1}{131072}\big(5S_{21} + 285S_{23} + 3876S_{25} + 19380S_{27} + 41990S_{29} + 41990S_{31} + 19380S_{33} + 3876S_{35} + 285S_{37} + 5S_{39}\big) = S_1^{20}.$$

On the other hand, starting with equation (3.4) and making an analysis similar to that leading to equation (4.2), one can deduce the following general solution to equation (2.6), namely,

$$\frac{\frac{1}{a_m+2}\sum_{r=0}^{a_m+1} \text{even}(r)\binom{a_m+2}{r+1}(2a_m + 3 - r)S_{2a_m+2-r}}{3 \cdot 2^{a_m}} = S_2 S_1^{a_m}. \tag{4.3}$$

Now, for each $a_m \geq 0$, the quotient $\frac{p}{m-1}$ characterizing the solution (4.3) turns out to be $\frac{p}{m-1} = \frac{2a_m+3}{3\cdot 2^{a_m}}$. Further, for odd (even) a_m, the numerator of (4.3) involves $\frac{a_m+3}{2}$ ($\frac{a_m}{2} + 1$) different sums S_j with j being an even integer ranging in $a_m + 1 \leq j \leq 2a_m + 2$ ($a_m + 2 \leq j \leq 2a_m + 2$). Moreover, the following identity holds

$$\sum_{r=0}^{a_m+1} \text{even}(r)\binom{a_m+2}{r+1}(2a_m + 3 - r) = 3 \cdot 2^{a_m}(a_m + 2),$$

and then the overall number of terms in the numerator of (4.3) is $3 \cdot 2^{a_m}$. For example, for $a_m = 17$, from equation (4.3) we find

$$\frac{1}{393216}\big(S_{18} + 189S_{20} + 4692S_{22} + 35700S_{24} + 107406S_{26} + 140998S_{28} + 82212S_{30} + 20196S_{32} + 1785S_{34} + 37S_{36}\big) = S_2 S_1^{17}.$$

Finally we note that, by combining (4.2) and (4.3), we obtain the double identity (with $a_m \geq 1$):

$$(a_m+2)\sum_{r=1}^{a_m}\text{odd}(r)\binom{a_m}{r}S_2S_{2a_m-r}$$
$$=\frac{1}{6}\sum_{r=0}^{a_m+1}\text{even}(r)\binom{a_m+2}{r+1}(2a_m+3-r)S_{2a_m+2-r}$$
$$=2^{a_m-1}(a_m+2)S_2S_1^{a_m}.$$

5. Matrix inversion

It is worth pointing out that, for any given a_m, we can equally obtain $S_1^{a_m}$ ($S_2S_1^{a_m}$) by inverting the corresponding triangular matrix formed by the Faulhaber coefficients in Table 1 (Table 2). This method was originally introduced by Edwards [3] (see also [2]) to obtain the Faulhaber coefficients themselves by inverting a matrix related to Pascal's triangle. As a concrete example illustrating this fact, consider the equation (2.2) written in matrix format up to $k=6$:

$$\begin{pmatrix} S_3\\ S_5\\ S_7\\ S_9\\ S_{11}\\ S_{13}\end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ -\frac{1}{3} & \frac{4}{3} & 0 & 0 & 0 & 0\\ \frac{1}{3} & -\frac{4}{3} & 2 & 0 & 0 & 0\\ -\frac{3}{5} & \frac{12}{5} & -4 & \frac{16}{5} & 0 & 0\\ \frac{5}{3} & -\frac{20}{3} & \frac{34}{3} & -\frac{32}{3} & \frac{16}{3} & 0\\ -\frac{691}{105} & \frac{2764}{105} & -\frac{944}{21} & \frac{4592}{105} & -\frac{80}{3} & \frac{64}{7}\end{pmatrix}\begin{pmatrix} S_1^2\\ S_1^3\\ S_1^4\\ S_1^5\\ S_1^6\\ S_1^7\end{pmatrix}. \tag{5.1}$$

Let us call the square matrix of (5.1) $\mathbf{F}$. Clearly, $\mathbf{F}$ is invertible since all the elements in its main diagonal are nonzero. Then, to evaluate the column vector on the right of (5.1), we pre-multiply by the inverse matrix of $\mathbf{F}$ on both sides of (5.1) to get

$$\begin{pmatrix} S_1^2\\ S_1^3\\ S_1^4\\ S_1^5\\ S_1^6\\ S_1^7\end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ \frac{1}{4} & \frac{3}{4} & 0 & 0 & 0 & 0\\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0\\ 0 & \frac{1}{16} & \frac{5}{8} & \frac{5}{16} & 0 & 0\\ 0 & 0 & \frac{3}{16} & \frac{5}{8} & \frac{3}{16} & 0\\ 0 & 0 & \frac{1}{64} & \frac{21}{64} & \frac{35}{64} & \frac{7}{64}\end{pmatrix}\begin{pmatrix} S_3\\ S_5\\ S_7\\ S_9\\ S_{11}\\ S_{13}\end{pmatrix},$$

from which we obtain the powers $S_1^2, S_1^3, S_1^4, \dots$, expressed in terms of the odd power sums $S_3, S_5, S_7, \dots$. Of course the resulting formula for S_1^7 agrees with that in equation (4.1). Conversely, by inverting the matrix $\mathbf{F^{-1}}$ we get the corresponding Faulhaber coefficients. Note that the elements of $\mathbf{F^{-1}}$ are nonnegative, and that the sum of the elements in each of the rows is equal to one. In fact, the row elements of $\mathbf{F^{-1}}$ are given by the corresponding coefficients $\binom{a_m}{r}/2^{a_m-1}$ appearing in the left-hand side of (4.2).

Similarly, writting the equation (2.1) in matrix format up to $k = 6$, we have

$$\begin{pmatrix} S_2 \\ S_4 \\ S_6 \\ S_8 \\ S_{10} \\ S_{12} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{5} & \frac{6}{5} & 0 & 0 & 0 & 0 \\ \frac{1}{7} & -\frac{6}{7} & \frac{12}{7} & 0 & 0 & 0 \\ -\frac{1}{5} & \frac{6}{5} & -\frac{8}{3} & \frac{8}{3} & 0 & 0 \\ \frac{5}{11} & -\frac{30}{11} & \frac{68}{11} & -\frac{80}{11} & \frac{48}{11} & 0 \\ -\frac{691}{455} & \frac{4146}{455} & -\frac{1888}{91} & \frac{328}{13} & -\frac{240}{13} & \frac{96}{13} \end{pmatrix} \begin{pmatrix} S_2 \\ S_2S_1 \\ S_2S_1^2 \\ S_2S_1^3 \\ S_2S_1^4 \\ S_2S_1^5 \end{pmatrix}. \tag{5.2}$$

Let us call the square matrix of (5.2) $\mathbf{G}$. Then, pre-multiplying both sides of (5.2) by the inverse matrix of $\mathbf{G}$, we obtain

$$\begin{pmatrix} S_2 \\ S_2S_1 \\ S_2S_1^2 \\ S_2S_1^3 \\ S_2S_1^4 \\ S_2S_1^5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{6} & \frac{5}{6} & 0 & 0 & 0 & 0 \\ 0 & \frac{5}{12} & \frac{7}{12} & 0 & 0 & 0 \\ 0 & \frac{1}{24} & \frac{7}{12} & \frac{3}{8} & 0 & 0 \\ 0 & 0 & \frac{7}{48} & \frac{5}{8} & \frac{11}{48} & 0 \\ 0 & 0 & \frac{1}{96} & \frac{9}{32} & \frac{55}{96} & \frac{13}{96} \end{pmatrix} \begin{pmatrix} S_2 \\ S_4 \\ S_6 \\ S_8 \\ S_{10} \\ S_{12} \end{pmatrix},$$

from which we can determine S_2 times the powers $S_1, S_1^2, S_1^3, \dots$, in terms of the even power sums $S_2, S_4, S_6, \dots$. Likewise, we can see that the elements of $\mathbf{G^{-1}}$ are nonnegative and that the sum of the elements in each row is equal to one.

In view of this example, it is clear that the formulae (4.2) and (4.3) can be regarded as a rule for calculating the inverse of the triangular matrices in Tables 1 and 2, respectively. Moreover, the uniqueness of the solutions in (4.2) and (4.3) follows ultimately from the uniqueness of the inverse of such triangular matrices.

6. A family of recursion formulae for the Bernoulli numbers

As a last important remark we note that the sum-to-zero column property allows us to derive a family of recursive relationships for the Bernoulli numbers. Consider initially the equation (3.3) for the column index $j = 0$. So, recalling that $F_0^{(2k+1)} = 2(2k+1)B_{2k}$, we will have (for odd r) that $F_0^{(2k-r)} = 2(2k-r)B_{2k-r-1}$, and then equation (3.3) becomes (for $j = 0$)

$$\sum_{r=1}^{k} \operatorname{odd}(r) \binom{k}{r} (2k-r) B_{2k-r-1} = 0, \tag{6.1}$$

which holds for any given $k \geq 3$. For example, for $k = 13$, from (6.1) we obtain

$$B_{12} + 90B_{14} + 935B_{16} + 2508B_{18} + 2079B_{20} + 506B_{22} + 25B_{24} = 0,$$

and so, knowing $B_{12}, B_{14}, B_{16}, B_{18}, B_{20}$, and B_{22}, we can get B_{24}. On the other hand, from equation (3.5) we obtain

$$F_2^{(2k+1)} = \frac{4}{3}(2k+1)\big[30B_{2k} + k(2k-1)B_{2k-2}\big],$$

from which we in turn deduce that, for odd r,

$$F_2^{(2k-r)} = 40(2k-r)B_{2k-r-1} + 4\binom{2k-r}{3}B_{2k-r-3}.$$

Therefore, recalling (6.1), from equation (3.3) with $j = 2$ we obtain the recurrence relation

$$\sum_{r=1}^{k} \text{odd}(r)\binom{k}{r}\binom{2k-r}{3}B_{2k-r-3} = 0, \tag{6.2}$$

which holds for any given $k \geq 5$. On the other hand, from equation (3.5) we obtain

$$F_4^{(2k+1)} = \frac{16}{45}(2k+1)\big[3780B_{2k} + 210k(2k-1)B_{2k-2} + k(k-1)(2k-1)(2k-3)B_{2k-4}\big],$$

from which we in turn deduce that, for odd r,

$$F_4^{(2k-r)} = 1344(2k-r)B_{2k-r-1} + 224\binom{2k-r}{3}B_{2k-r-3} + \frac{32}{3}\binom{2k-r}{5}B_{2k-r-5}.$$

Thus, taking into account (6.1) and (6.2), we see that, for $j = 4$, equation (3.3) yields the recurrence relation

$$\sum_{r=1}^{k} \text{odd}(r)\binom{k}{r}\binom{2k-r}{5}B_{2k-r-5} = 0, \tag{6.3}$$

which holds for any given $k \geq 7$. The pattern is now clear. Indeed, by assuming that $F_{2s}^{(2k-r)}$ (for odd r) is of the form

$$F_{2s}^{(2k-r)} = \sum_{q=0}^{s} f_q^{(2k-r)}\binom{2k-r}{2q+1}B_{2k-r-2q-1},$$

with the $f_q^{(2k-r)}$'s being nonzero rational coefficients, from equation (3.3) one readily gets the following general recurrence relation for the Bernoulli numbers:

$$\sum_{r=1}^{k} \text{odd}(r)\binom{k}{r}\binom{2k-r}{2s+1}B_{2k-r-2s-1} = 0, \tag{6.4}$$

which holds for any given $k \geq 2s+3$, with $s = 0, 1, 2, \ldots$. Formulae (6.1), (6.2), and (6.3) are particular cases of the recurrence (6.4) for $s = 0, 1$, and 2, respectively.

Similarly, starting from equation (3.4), it can be shown that

$$\sum_{r=0}^{k+1} \text{even}(r) \binom{k+2}{r+1} \binom{2k+3-r}{2s+1} B_{2k+2-r-2s} = 0, \tag{6.5}$$

which holds for any given $k \geq 2s+1$, with $s = 0, 1, 2, \ldots$. It is easy to see that relations (6.4) and (6.5) are equivalent to each other. Moreover, we note that the recurrence (6.4) is essentially equivalent to the one given in [8, Theorem 1.1].

7. Conclusion

In this article we have tackled the problem of averaging sums of powers of integers as considered by Pfaff [1]. For this purpose, we have expressed the S_r's in the Faulhaber form and then we have used certain properties of the coefficients of the Faulhaber polynomials. Indeed, as we have seen, the sum-to-zero column property in equations (3.3) and (3.4) constitutes the skeleton of the solutions displayed in (4.2) and (4.3). It is to be noted, on the other hand, that the formulae (4.2) and (4.3) can be obtained in a more straightforward way by a proper application of the binomial theorem (for a derivation of the counterpart to the formulae (4.2) and (4.3) using this method, see [5, Subsections 3.2 and 3.3]). Furthermore, a demonstration by mathematical induction of the identities in (4.2) and (4.3) (although expressed in a somewhat different manner) already appeared in [9].

We believe, however, that our approach here is worthwhile since it introduces an important topic concerning the sums of powers of integers that may not be widely known, namely, the Faulhaber theorem and the associated Faulhaber polynomials. We invite the interested reader to prove some of the properties listed above, and to pursue the subject further [3, 4, 5, 6, 10, 11, 12].

References

[1] Pfaff, T. J., Averaging sums of powers of integers, *College Math. J.*, 42 (2011) 402–404.

[2] Edwards, A. W. F., Sums of powers of integers: a little of the history, *Math. Gaz.*, 66 (1982) 22–28.

[3] Edwards, A. W. F., A quick route to sums of powers, *Amer. Math. Monthly*, 93 (1986) 451–455.

[4] Knuth, D. E., Johann Faulhaber and sums of powers, *Math. Comp.*, 61 (1993) 277–294.

[5] Kotiah, T. C. T., Sums of powers of integers—A review, *Int. J. Math. Edu. Sci. Technol.*, 24 (1993) 863–874.

[6] Beardon, A. F., Sums of powers of integers, *Amer. Math. Monthly*, 103 (1996) 201–213.

[7] Gessel, I. M., Viennot, X. G., Determinants, paths, and plane partitions. Preprint, 1989. Available at `http://people.brandeis.edu/~gessel/homepage/papers/pp.pdf`.

[8] Zékiri, A., Bencherif, F., A new recursion relationship for Bernoulli numbers, *Annales Mathematicae et Informaticae*, 38 (2011) 123–126.

[9] Piza, P. A., Powers of sums and sums of powers, *Math. Mag.*, 25 (1952) 137–142.

[10] Krishnapriyan, H. K., Eulerian polynomials and Faulhaber's result on sums of powers of integers, *College Math. J.*, 26 (1995) 118–123.

[11] Shirali, S. A., On sums of powers of integers, *Resonance*, 12(7) (2007) 27–43.

[12] Chen, W. Y. C., Fu, A. M., Zhang, I. F., Faulhaber's theorem on power sums, *Discrete Math.*, 309 (2009) 2974–2981.

Annales Mathematicae et Informaticae
42 (2013) pp. 119–127
http://ami.ektf.hu

Linear recurrence relations with the coefficients in progression

Mircea I. Cîrnu

Department of Mathematics, Faculty of Applied Sciences
Polytechnic University of Bucharest, Romania
cirnumircea@yahoo.com

Submitted March 7, 2012 — Accepted January 31, 2013

Abstract

The aim of this paper is to solve the linear recurrence relation

$$x_{n+1} = a_0x_n + a_1x_{n-1} + \cdots + a_{n-1}x_1 + a_nx_0,\ n = 0, 1, 2, \dots,$$

when its constant coefficients are in arithmetic, respective geometric progression. Rather surprising, when the coefficients are in arithmetic progression, the solution is a sequence of certain generalized Fibonacci numbers, but not of usual Fibonacci numbers, while if they are in geometric progression the solution is again a geometric progression, with different ratio. In both cases the solution will be found by generating function method. Alternatively, in the first case it will be obtained by reduction to a generalized Fibonacci equation and in the second case by mathematical induction. Finally, the case is considered when both the coefficients and solutions form geometric progressions with generalized Fibonacci numbers as terms. The paper has a didactical purpose, being intended to familiarize the students with the usual procedures for solving linear recurrence relations. Another algebraic, differential and integral recurrence relations were considered by the author in the papers cited in the references.

Keywords: linear recurrence relations, arithmetic and geometric progressions, generalized Fibonacci numbers.

MSC: 11C08, 11B39.

1. Introduction

In this paper we apply the usual methods for solving linear recurrence relations with constant coefficients of special form - progressions. The method of characteristic equation, of generating function and of mathematical induction are used. The relationship between the considered relations and the generalized Fibonacci numbers is also specified. The numbers considered in this paper are complex. We remember that one calls *generalized Fibonacci numbers* or *Horadam numbers* (see [5, 6, 7]) of orders α and β, the numbers $x_n,\ n = 0, 1, 2, \ldots$, satisfying the *generalized Fibonacci recurrence relation* $x_{n+1} = \alpha x_n + \beta x_{n-1},\ n = 1, 2, \ldots$, with arbitrary initial data x_0 and x_1. If $\alpha = \beta = 1$, hence when the numbers x_n satisfy the *usual Fibonacci recurrence relation* $x_{n+1} = x_n + x_{n-1}$, these numbers are called *Fibonacci type numbers*. Particularly, when the initial data are $x_0 = 0$ and $x_1 = 1$, the *usual Fibonacci numbers* are obtained. When the coefficients of the linear recurrence relation of order n are in arithmetic progression, then its solutions are generalized Fibonacci numbers of certain orders. When the coefficients are in geometric progression, then the solutions are also in such a progression. In the final Section, this last situation is particularly considered when both the coefficients and solutions are generalized Fibonacci numbers. Aspects of the theory of recurrence relations and Fibonacci numbers can be found in the works listed in References.

2. Linear recurrence relations with coefficients in arithmetic progression

Theorem 2.1. *The numbers x_n are solutions of the linear recurrence relation with the coefficients in arithmetic progression*

$$x_{n+1} = ax_n + (a+r)x_{n-1} + \cdots + (a+(n-1)r)x_1 + (a+nr)x_0,\ n = 0, 1, 2, \ldots, \tag{2.1}$$

with initial data x_0, if and only if they are the generalized Fibonacci numbers given by the Binet type formula

$$x_n = \frac{x_0}{\lambda_1 - \lambda_2}\left[(b - a\lambda_2)\lambda_1^{n-1} - (b - a\lambda_1)\lambda_2^{n-1}\right],\ n = 1, 2, \ldots, \tag{2.2}$$

where

$$b = a^2 + a + r,\ \lambda_{1,2} = \frac{a + 2 \pm \sqrt{a^2 + 4r}}{2}. \tag{2.3}$$

Proof. (*By reduction to a generalized Fibonacci recurrence relation*) We suppose that the numbers x_n satisfy the recurrence relation (2.1). Then we have $x_1 = ax_0$, $x_2 = bx_0$ and

$$x_{n+1} - x_n = ax_n + rx_{n-1} + rx_{n-2} + \cdots + rx_1 + rx_0,$$
$$x_n - x_{n-1} = ax_{n-1} + rx_{n-2} + \cdots + rx_1 + rx_0,$$

$$(x_{n+1} - x_n) - (x_n - x_{n-1}) = ax_n + (r - a)x_{n-1}.$$

Denoting $c = a - r$, one obtains the generalized Fibonacci recurrence relation

$$x_{n+1} = (a + 2)x_n - (c + 1)x_{n-1},\ n = 2, 3, \ldots. \tag{2.4}$$

This equation has the solution $x_n = C_1\lambda_1^n + C_1\lambda_2^n$, where $\lambda_{1,2}$ are the roots, given by (2.3), of the characteristic equation $\lambda^2 - (a + 2)\lambda + c + 1 = 0$. The initial conditions $x_1 = C_1\lambda_1 + C_2\lambda_2 = ax_0$ and $x_2 = C_1\lambda_1^2 + C_2\lambda_2^2 = bx_0$ give $C_{1,2} = \pm\dfrac{x_0(b - a\lambda_{2,1})}{\lambda_{1,2}(\lambda_1 - \lambda_2)}$, hence the solutions of the recurrence relation (2.1) are given by the formula (2.2). □

Remark. The linear recurrence relation (2.4) fails for $n = 1$ and therefore its initial conditions are x_1 and x_2 instead of x_0 and x_1.

Proof. (*By generating function method*) Working with formal series, we denote by $X(t) = \sum_{n=0}^{\infty} x_n t^n$, the generating function of the sequence x_n. Then the recurrence relation (2.1) takes the form $\sum_{n=0}^{\infty} x_{n+1}t^{n+1} = \sum_{n=0}^{\infty}\sum_{k=0}^{n}(a+kr)x_{n-k}t^{n+1}$. Using the formula for the product of two power series, one obtains

$$X(t) - x_0 = t\sum_{n=0}^{\infty}(a + nr)t^n \sum_{n=0}^{\infty} x_n t^n = t\sum_{n=0}^{\infty}(a + nr)t^n X(t).$$

Because

$$\sum_{n=0}^{\infty}(a + nr)t^n = a\sum_{n=0}^{\infty} t^n + rt\sum_{n=0}^{\infty}\frac{d}{dt}(t^n) = a\frac{1}{1-t} + rt\frac{d}{dt}\Big(\frac{1}{1-t}\Big) = \frac{a - ct}{(1-t)^2},$$

we have $X(t) - x_0 = \dfrac{t(a - ct)}{(1-t)^2}X(t)$. One obtains

$$X(t) = \frac{x_0(t-1)^2}{(c+1)t^2 - (a+2)t + 1} = \frac{x_0}{c+1} + \frac{x_0((r-c)t + c)}{\sqrt{c+1}(t - t_1)(t - t_2)},$$

where $t_1 = \dfrac{1}{\lambda_1}$ and $t_2 = \dfrac{1}{\lambda_2}$ are the roots of the equation $(c+1)t^2 - (a+2)t + 1 = 0$, the numbers $\lambda_{1,2}$ been given by the relation (2.3). We have

$$\begin{aligned}
X(t) &= \frac{x_0}{c+1} + \frac{x_0}{(c+1)^2(t_1 - t_2)}\left[\frac{(r-c)t_1 + c}{t - t_1} - \frac{(r-c)t_2 + c}{t - t_2}\right] \\
&= \frac{x_0}{\lambda_1\lambda_2} + \frac{x_0}{\lambda_1\lambda_2(\lambda_1 - \lambda_2)}\left[\frac{c\lambda_1 + r - c}{1 - \lambda_1 t} - \frac{c\lambda_2 + r - c}{1 - \lambda_2 t}\right] \\
&= \frac{x_0}{\lambda_1\lambda_2} + \frac{x_0}{\lambda_1\lambda_2(\lambda_1 - \lambda_2)}\left[(c\lambda_1 + r - c)\sum_{n=0}^{\infty}\lambda_1^n t^n - (c\lambda_2 + r - c)\sum_{n=0}^{\infty}\lambda_2^n t^n\right]
\end{aligned}$$

$$= x_0 + \frac{x_0}{\lambda_1\lambda_2(\lambda_1-\lambda_2)}\Big[(c\lambda_1+r-c)\sum_{n=1}^{\infty}\lambda_1^n t^n - (c\lambda_2+r-c)\sum_{n=1}^{\infty}\lambda_2^n t^n\Big].$$

Therefore the coefficients of the generating function $X(t)$ are given by the relation

$$x_n = \frac{x_0(c\lambda_1+r-c)}{\lambda_2(\lambda_1-\lambda_2)}\lambda_1^{n-1} - \frac{x_0(c\lambda_2+r-c)}{\lambda_1(\lambda_1-\lambda_2)}\lambda_2^{n-1},\ n=1,2,\ldots.$$

Taking into account the identities $\dfrac{c\lambda_1+r-c}{\lambda_2} = b-a\lambda_2$ and $\dfrac{c\lambda_2+r-c}{\lambda_1} = b-a\lambda_1$, from the above expression of x_n one obtains formula (2.2). □

Proof. (*Reciprocal*) If the sequence x_n is given by formula (2.2), it satisfies the recurrence relation (2.4). Indeed, using (2.2) and the relation $\lambda_j^2 = (a+2)\lambda_j - (c+1)$, $j=1,2$, which results by the definition of the numbers $\lambda_{1,2}$, one obtain

$$\begin{aligned}
(a+2)x_n - (c+1)x_{n-1} &= (a+2)\frac{x_0}{\lambda_1-\lambda_2}\big[(b-a\lambda_2)\lambda_1^{n-1} - (b-a\lambda_1)\lambda_2^{n-1}\big]-\\
&- (c+1)\frac{x_0}{\lambda_1-\lambda_2}\big[(b-a\lambda_2)\lambda_1^{n-2} - (b-a\lambda_1)\lambda_2^{n-2}\big]\\
&= \frac{x_0}{\lambda_1-\lambda_2}(b-a\lambda_2)\big[(a+2)\lambda_1-(c+1)\big]\lambda_1^{n-2}-\\
&- \frac{x_0}{\lambda_1-\lambda_2}(b-a\lambda_1)\big[(a+2)\lambda_2-(c+1)\big]\lambda_2^{n-2}\\
&= \frac{x_0}{\lambda_1-\lambda_2}\big[(b-a\lambda_2)\lambda_1^{n} - (b-a\lambda_1)\lambda_2^{n}\big] = x_{n+1},\ n=1,2,\ldots.
\end{aligned}$$

Now we prove by induction that the sequence x_n given by (2.2) satisfies the recurrence relation (2.1). We first show that (2.1) is satisfied for $n=0,1,2$. Indeed, from (2.2) it follows

$$\begin{aligned}
x_1 &= \frac{x_0}{\lambda_1-\lambda_2}(b-a\lambda_2-b+a\lambda_1) = ax_0,\\
x_2 &= \frac{x_0}{\lambda_1-\lambda_2}\big[(b-a\lambda_2)\lambda_1 - (b-a\lambda_1)\lambda_2\big] = bx_0\\
&= (a^2+a+r)x_0 = ax_1 + (a+r)x_0,\\
x_3 &= \frac{x_0}{\lambda_1-\lambda_2}\big[(b-a\lambda_2)\lambda_1^2 - (b-a\lambda_1)\lambda_2^2\big]\\
&= \frac{x_0}{\lambda_1-\lambda_2}\big[b(\lambda_1^2-\lambda_2^2) - a\lambda_1\lambda_2(\lambda_1-\lambda_2)\big]\\
&= x_0\big[b(\lambda_1+\lambda_2) - a\lambda_1\lambda_2\big] = x_0\big[b(a+2) - a(c+1)\big]\\
&= abx_0 + x_0\big[2(a^2+a+r) - a(1+a-r)\big]\\
&= ax_2 + x_0(a^2+ar+a+2r) = ax_2 + (a+r)x_1 + (a+2r)x_0.
\end{aligned}$$

For a fixed index $n \geq 2$, we suppose that the formula (2.1) is true when $k \leq n$, hence we have

$$x_{k+1} = \sum_{j=0}^{k}(a+(k-j)r)x_j,\ k \leq n. \tag{2.5}$$

Using (2.4) and (2.5), one obtains

$$\begin{aligned}
x_{n+2} &= (a+2)x_{n+1} - (c+1)x_n \\
&= (a+2)\sum_{k=0}^{n}(a+(n-k)r)x_k - (c+1)\sum_{k=1}^{n}(a+(n-k)r)x_{k-1} \\
&= \sum_{k=2}^{n}(a+(n-k)r)\big[(a+2)x_k - (c+1)x_{k-1}\big] + (a+2)(a+(n-1)r)x_1 + \\
&\quad + (a+2)(a+nr)x_0 - (c+1)(a+(n-1)r)x_0 = \sum_{k=2}^{n}(a+(n-k)r)x_{k+1} + \\
&\quad + a(a+2)(a+(n-1)r)x_0 + a(a+nr)x_0 + \\
&\quad + 2(a+nr)x_0 + (r-a-1)(a+(n-1)r)x_0 \\
&= \sum_{k=2}^{n}(a+(n-k)r)x_{k+1} + (a+(n-1)r)x_2 + (a+nr)x_1 + (a+(n+1)r)x_0 \\
&= \sum_{k=0}^{n+1}(a+(n+1-k)r)x_k,
\end{aligned}$$

hence formula (2.1) is true for the index $n+1$. According to the induction axiom, (2.1) is true for any natural number n. □

Remarks. 1) The sequence of usual Fibonacci numbers can not be solution of the equation (2.1). Indeed, for this would be that $a+2 = -c-1 = r-a-1 = 1$, for the equation (2.4) to reduce to well-known Fibonacci recurrence relation $x_{n+1} = x_n + x_{n-1}$ and to have the initial conditions $x_1 = ax_0 = 1$ and $x_2 = bx_0 = (a^2+a+r)x_0 = 1$. But these conditions are contradictory, leading to the false equality $x_0 = 1 = -1$.
2) An arithmetic progression x_n cannot be solution of the equation (2.1). Indeed, this requires that $x_{n+1} = 2x_n - x_{n-1}$, therefore $a+2 = 2$ and $-c-1 = r-a-1 = -1$, which leads to the trivial case $a = r = x_n = 0$, for $n = 1, 2, \ldots$.

Corollary 2.2. *The linear recurrence relation*

$$x_{n+1} = x_n + 2x_{n-1} + \cdots + nx_1 + (n+1)x_0,\ n = 0, 1, 2, \ldots, \tag{2.6}$$

with the initial data $x_0 = 1$, *has the solution*

$$x_n = \frac{1}{\sqrt{5}}\left[\left(\frac{3+\sqrt{5}}{2}\right)^n - \left(\frac{3-\sqrt{5}}{2}\right)^n\right],\ n = 1, 2, \ldots. \tag{2.7}$$

Proof. For $a = r = x_0 = 1$, from Theorem 2.1 and its proof it results that the recurrence relation (2.6) reduces to the generalized Fibonacci relation

$$x_{n+1} = 3x_n - x_{n-1},\ n = 2, 3, \ldots, \tag{2.8}$$

with the initial data $x_1 = 1$ and $x_2 = 3$, hence it has the solution (2.7). Particularly, both (2.6) and (2.7) give $x_3 = 8$, $x_4 = 21$ and so on. □

Remark. The recurrence relation (2.6) from above corollary was considered as problem 9 (ii) in F. Lazebnik, Combinatorics and Graphs Theory, I, (Math 688). Problems and Solutions, 2006, a work appearing on the Internet at the address `www.math.udel.edu/~lazebnik/papers/688hwsols.pdf`. Unfortunately, in the cited work one obtains the wrong solution

$$x_n = \frac{5-\sqrt{5}}{10}\left(\frac{3+\sqrt{5}}{2}\right)^n + \frac{5+\sqrt{5}}{10}\left(\frac{3-\sqrt{5}}{2}\right)^n,\ n = 0, 1, 2, \ldots,$$

with particular solutions $x_0 = x_1 = 1$, $x_2 = 2$, $x_3 = 5$ and so on, the last two being false. The explanation of this mistake is that the recurrence relation (2.8) was wrongly considered for $n = 1, 2, \ldots,$, with the initial data $x_0 = x_1 = 1$, leading to the wrong solution mentioned above. Indeed, for $n = 1$, the obtined recurrence relation $x_2 = 3x_1 - x_0$ is false. This mistake shows the importance of the correct initialization of the recurrence relations.

3. Linear recurrence relations with coefficients in geometric progression

Theorem 3.1. *The numbers x_n are solutions of the linear recurrence relation with constant coefficients in geometric progression*

$$x_{n+1} = ax_n + aqx_{n-1} + \cdots + aq^{n-1}x_1 + aq^n x_0,\ n = 0, 1, 2, \ldots, \tag{3.1}$$

with initial data x_0, if and only if they form the geometric progression given by the formula

$$x_n = ax_0(a+q)^{n-1},\ n = 1, 2, \ldots \tag{3.2}$$

Proof. (*By induction*). From (3.1) we obtain $x_1 = ax_0$ and $x_2 = ax_0(a+q)$. For a fixed natural number n we suppose formula (3.2) true for every $k \le n$. Therefore we have $x_k = ax_0(a+q)^{k-1}$, for $k \le n$. Then, from the recurrence relation (3.1) one obtains

$$\begin{aligned} x_{n+1} &= a^2x_0(a+q)^{n-1} + a^2x_0q(a+q)^{n-2} + \cdots + \\ &\quad + a^2x_0q^{n-2}(a+q) + a^2x_0q^{n-1} + ax_0q^n \\ &= a^2x_0(a+q)\big[(a+q)^{n-2} + q(a+q)^{n-3} + \cdots + q^{n-3}(a+q) + q^{n-2}\big] + \\ &\quad + ax_0q^{n-1}(a+q) \\ &= a^2x_0(a+q)\frac{(a+q)^{n-1} - q^{n-1}}{a} + ax_0q^{n-1}(a+q) = ax_0(a+q)^n, \end{aligned}$$

hence the formula (3.2) is true for $n+1$. According to the induction axiom it results that formula (3.2) is true for every natural number n. □

Proof. (*By generating function method*) Denoting $X(t) = \sum_{n=0}^{\infty} x_n t^n$, from the recurrence relation (3.1) one obtains $\sum_{n=0}^{\infty} x_{n+1} t^{n+1} = at \sum_{n=0}^{\infty} \sum_{k=0}^{n} q^k x_{n-k} t^n$. Using the formula for the product of two power series, one obtains

$$X(t) - x_0 = at \sum_{n=0}^{\infty} q^n t^n \sum_{n=0}^{\infty} x_n t^n = \frac{at}{1-qt} X(t).$$

Therefore

$$X(t) = x_0 \frac{qt-1}{(a+q)t-1} = \frac{x_0 q}{a+q} + \frac{a x_0}{(a+q)(1-(a+q)t)}$$
$$= \frac{x_0 q}{a+q} + \frac{a x_0}{a+q} \sum_{n=0}^{\infty} (a+q)^n t^n = x_0 + a x_0 \sum_{n=1}^{\infty} (a+q)^{n-1} t^n,$$

from which it results the formula (3.2). □

Proof. (*Reciprocal*) If x_n is given by the formula (3.2), then we have

$$a \sum_{k=0}^{n} q^{n-k} x_k = a^2 x_0 \sum_{k=1}^{n} q^{n-k} (a+q)^{k-1} + a x_0 q^n$$
$$= a^2 x_0 q^{n-1} \sum_{k=1}^{n} \Big(\frac{a+q}{q}\Big)^{k-1} + a x_0 q^n$$
$$= a^2 x_0 q^{n-1} \frac{\Big(\frac{a+q}{q}\Big)^n - 1}{\frac{a+q}{q} - 1} + a x_0 q^n = a x_0 (a+q)^n = x_{n+1},\ n = 1, 2, \ldots,$$

hence the sequence x_n satisfies the recurrence equation (3.1). □

4. Linear recurrence relations having as coefficients generalized Fibonacci numbers in geometric progression

Lemma 4.1. *The terms $a_n = aq^n$, $n = 0, 1, 2, \ldots,$ of a geometric progression are generalized Fibonacci numbers of orders α and β if and only if the progression ratio is given by the formula*

$$q = \frac{\alpha \pm \sqrt{\alpha^2 + 4\beta}}{2}. \tag{4.1}$$

Proof. If the terms a_n of the geometric progression are generalized Fibonacci numbers of orders α and β, then $a_{n+1} = \alpha a_n + \beta a_{n-1}$, relation which becomes $aq^{n+1} = \alpha a q^n + \beta a q^{n-1}$. One obtains the quadratic equation $q^2 - \alpha q - \beta = 0$, with the roots given by formula (4.1). Reciprocally, if the number q is given by formula (4.1), it satisfies the above quadratic equation. Multiplying this equation by aq^{n-1}, one obtains the relation $a_{n+1} = \alpha a_n + \beta a_{n-1}$, hence a_n are generalized Fibonacci numbers of orders α and β. □

Example. If $\alpha = 2i$, with $i = \sqrt{-1}$ and $\beta = 1$, then (4.1) gives $q = i$, therefore the terms of the geometric progression $a_n = ai^n$ are generalized Fibonacci numbers of orders $2i$ and 1. Indeed, we have $2ia_n + a_{n-1} = 2ai^{n+1} + ai^{n-1} = ai^{n+1} = a_{n+1}$.

Theorem 4.2. *The coefficients* $a_n = aq^n$, $n = 0.1.2. \ldots$, *and the solutions* x_n, $n = 1, 2, \ldots$ *of the linear recurrence relation* (3.1) *are both generalized Fibonacci numbers of orders* α *and* β *if and only if*

$$\alpha = a + 2q, \ \beta = -q(a + q). \tag{4.2}$$

Proof. According to Theorem 3.1 and the above Lemma, the coefficients a_n and the solutions x_n of (3.1) are generalized Fibonacci numbers of orders α and β, if and only if

$$q^2 - \alpha q - \beta = 0, \ (a + q)^2 - \alpha(a + q) - \beta = 0, \tag{4.3}$$

hence the formula (4.2) holds. □

Example. If $a = q = i$, then $a_n = i^{n+1}$ and, according to Theorem 3.1, $x_n = \frac{x_0}{2}(2i)^n$. From Theorem 4.3 it results that both a_n and x_n are generalized Fibonacci numbers of orders $\alpha = a + 2q = 3i$ and $\beta = -q(a + q) = 2$. Indeed, we have

$$3ia_n + 2a_{n-1} = 3i^{n+2} + 2i^n = i^{n+2} = a_{n+1}$$

and

$$3ix_n + 2x_{n-1} = \frac{x_0}{2}[3i(2i)^n + 2(2i)^{n-1}] = \frac{x_0}{2}(2i)^{n+1} = x_{n+1}.$$

Corollary 4.3. *The coefficients* a_n *and the solutions* x_n *of the linear recurrence relation* (3.1) *are both Fibonacci type numbers if and only if*

$$a = \mp\sqrt{5}, \ q = \frac{1 \pm \sqrt{5}}{2}. \tag{4.4}$$

Proof. For $\alpha = \beta = 1$ it follows from Theorem 4.3 that $a+2q = 1$ and $-q(a+q) = 1$, from which we obtain (4.4). □

References

[1] Cîrnu, M. I., First order differential recurrence equations, *International Journal of Mathematics and Computation*, 4 (2009) 124–128

[2] Cîrnu M. I., Initial-value problems for first-order differential recurrence equations with auto-convolution, *Electronic Journal of Differential Equations*, 2011 (2011) no. 2, 1–13.

[3] Cîrnu M. I., Determinantal formulas for sum of generalized arithmetic-geometric series, *Boletin de la Associacion Matematica Venezolana*, 43 (2011) no. 1, 25–38.

[4] Cîrnu M. I., A certain integral-recurrence equation with discrete-continuous auto-convolution, *Archivum Mathematicum (Brno)*, 47 (2011) 267–272.

[5] Horadam A. F., A generalized Fibonacci sequence, *American Mathematical Monthly*, 68(5) (1961), 455–459.

[6] Horadam A. F., Basic properties of a certain generalized sequence of numbers, *Fibonacci Quarterly*, 3(1965) 161–176.

[7] Horadam A. F., Generating functions for powers of a certain generalized sequence of numbers, *Duke Math. J.*, 32(1965) 437–446.

[8] Jeske J. A., Linear recurrence relations, *Fibonacci Quarterly*, Part I, 1(1963), no. 2, 69-74, Part II, no. 4, 35–39.

[9] Koshy T., Fibonacci and Lucas numbers and applications, Wiley, 2001.

[10] Levesque C., On the m-th order linear recurrences, *Fibonacci Quarterly*, 23(1985) 290–295.

[11] Wilf H. S., Generatingfunctionology, Academic Press, Inc., 1994.

Annales Mathematicae et Informaticae
42 (2013) pp. 129–134
http://ami.ektf.hu

A note on Golomb's method and the continued fraction method for Egyptian fractions

Eszter Gyimesi, Gábor Nyul*

Institute of Mathematics
University of Debrecen, Hungary
esztilevel@gmail.com
gnyul@science.unideb.hu

Submitted March 22, 2013 — Accepted June 1, 2013

Abstract

In this note we prove directly that Golomb's method and the continued fraction method are essentially the same, in the sense that they give the same Egyptian fraction expansions of positive rational numbers. Furthermore, we show their connection with the Farey sequence method.

Keywords: Egyptian fractions, Golomb's method, continued fraction method, Farey sequence method

MSC: 11D68

1. Introduction

It is well-known that every positive rational number can be expressed as a sum of distinct unit fractions (reciprocals of natural numbers). Ancient Egyptians already used such representations of rational numbers, for this reason we call a sum of distinct unit fractions an Egyptian fraction. We note that sometimes unit fractions themselves are called Egyptian fractions.

*Research was supported in part by Grants 75566 and 100339 from the Hungarian Scientific Research Fund.

Since the harmonic series $\sum_{n=1}^{\infty} \frac{1}{n}$ is divergent, it is enough to give an algorithm for finding an Egyptian fraction expansion of rational numbers between 0 and 1. There are several methods to do this, many of them are summarized in [6].

Probably the oldest such algorithm is the greedy method, which subtracts always the largest possible unit fraction from the current rational number. Sometimes it is referred to as Fibonacci method or Fibonacci-Sylvester method, because it was first described by Leonardo Pisano, better known as Fibonacci [5], and later it was rediscovered by J. J. Sylvester [10].

The splitting method is based on successive application of the identity $\frac{1}{n} = \frac{1}{n+1} + \frac{1}{n(n+1)}$. It was shown by L. Beeckmans [2] that this algorithm terminates after a finite number of steps, however it was stated previously without proof by P. J. Campbell [4].

We still mention here by name the method of S. W. Golomb [7] and the continued fraction method due to M. N. Bleicher [3], as they are the main subject of this paper. We should remark that there is a confusion in the literature, in [3] a further variant of the latter method is presented, and the author calls the modified version the continued fraction method. However, in [6] the original algorithm is called the continued fraction method, as will be in this note.

Writing her BSc thesis, the first author observed that surprisingly Golomb's method and the continued fraction method always give the same Egyptian fraction expansions. Before proving this statement directly, we discuss these methods briefly. After that, we present their connection with the Farey sequence method, and a possible usage of them in teaching basic number theory.

2. Golomb's method and the continued fraction method

Golomb's method Let $a < b$ be positive integers with $\gcd(a, b) = 1$, and consider the rational number $0 < \frac{a}{b} < 1$. If $a = 1$, then it is a unit fraction. Otherwise, since a and b are coprime, there exist a multiplicative inverse $0 < a' < b$ of a modulo b and a natural number r such that $aa' = br + 1$. Then

$$\frac{a}{b} = \frac{r}{a'} + \frac{1}{a'b}.$$

Now it follows from $aa' > br > ar$ and $aa' = br + 1$, that $0 < \frac{r}{a'} < 1$ and $\gcd(r, a') = 1$, and we can apply the above procedure for $\frac{r}{a'}$.

On the other hand, we have $aa' > br > a'r$, hence $r < a$, which guarantees the finiteness of the method. The algorithm is also correct, it gives distinct unit fractions in the Egyptian fraction expansion, which can be proved by induction showing that the unit fractions have denominators at most $b(b - 1)$.

Continued fraction method Let $0 < \frac{a}{b} < 1$ be again a rational number with coprime natural numbers a and b. Suppose that the finite simple continued fraction expansion of $\frac{a}{b}$ is $\langle c_0, c_1, \dots, c_n \rangle$, where $c_0 = 0$ and $c_1, \dots, c_n$ are positive integers. As it is well-known, $\frac{a}{b}$ can be represented by a finite simple continued fraction in exactly two ways, but it is indifferent which of them is used.

As usual, define two sequences $(a_k)_{k=-2}^{n}$ and $(b_k)_{k=-2}^{n}$ recursively:

$$a_{-2} = 0, \quad a_{-1} = 1, \quad a_k = c_k a_{k-1} + a_{k-2} \quad (k = 0, 1, \dots, n)$$

$$b_{-2} = 1, \quad b_{-1} = 0, \quad b_k = c_k b_{k-1} + b_{k-2} \quad (k = 0, 1, \dots, n)$$

Then $a_n = a$ and $b_n = b$.

Primary and secondary convergents satisfy equations

$$\frac{a_k}{b_k} - \frac{a_{k-1}}{b_{k-1}} = \frac{(-1)^{k+1}}{b_{k-1} b_k}$$

for $1 \le k \le n$, and

$$\frac{a_{k-2} + l a_{k-1}}{b_{k-2} + l b_{k-1}} - \frac{a_{k-2} + (l-1) a_{k-1}}{b_{k-2} + (l-1) b_{k-1}} = \frac{(-1)^k}{(b_{k-2} + (l-1) b_{k-1})(b_{k-2} + l b_{k-1})}$$

for $2 \le k \le n$, $1 \le l \le c_k$. Details about these and other properties of continued fractions can be found in [8, 9].

Using the above identities, we can describe the continued fraction method. If n is odd, then

$$\frac{a_n}{b_n} = \frac{a_{n-1}}{b_{n-1}} + \frac{1}{b_{n-1} b_n}, \tag{2.1}$$

and apply the method for $\frac{a_{n-1}}{b_{n-1}}$.

If n is even, then

$$\begin{aligned}\frac{a_n}{b_n} &= \frac{a_{n-2} + c_n a_{n-1}}{b_{n-2} + c_n b_{n-1}} = \frac{a_{n-2}}{b_{n-2}} + \sum_{l=1}^{c_n} \left(\frac{a_{n-2} + l a_{n-1}}{b_{n-2} + l b_{n-1}} - \frac{a_{n-2} + (l-1) a_{n-1}}{b_{n-2} + (l-1) b_{n-1}} \right) \\ &= \frac{a_{n-2}}{b_{n-2}} + \sum_{l=1}^{c_n} \frac{1}{(b_{n-2} + (l-1) b_{n-1})(b_{n-2} + l b_{n-1})}, \end{aligned} \tag{2.2}$$

and apply the method for $\frac{a_{n-2}}{b_{n-2}}$.

We note that the first case (odd subscript) is used at most once, while the correctness of the algorithm can be proved by induction on n showing that the denominators of the unit fractions do not exceed $b_n(b_n - 1)$.

Proof that these methods give the same Egyptian fraction expansions

If n is odd, then it follows from $a_n b_{n-1} - a_{n-1} b_n = 1$ that $0 < b_{n-1} < b_n$ is the multiplicative inverse of a_n modulo b_n, hence one step of Golomb's method gives (2.1), exactly the same sum as the continued fraction method.

If n is even, then $(a_{n-2} + l a_{n-1})(b_{n-2} + (l-1) b_{n-1}) - (a_{n-2} + (l-1) a_{n-1})(b_{n-2} + l b_{n-1}) = 1$ implies that $0 < b_{n-2} + (l-1) b_{n-1} < b_{n-2} + l b_{n-1}$ is the multiplicative inverse of $a_{n-2}+la_{n-1}$ modulo $b_{n-2}+lb_{n-1}$, hence applying Golomb's method for $\frac{a_{n-2}+la_{n-1}}{b_{n-2}+lb_{n-1}}$, it gives

$$\frac{a_{n-2}+la_{n-1}}{b_{n-2}+lb_{n-1}}=\frac{a_{n-2}+(l-1)a_{n-1}}{b_{n-2}+(l-1)b_{n-1}}+\frac{1}{(b_{n-2}+(l-1)b_{n-1})(b_{n-2}+lb_{n-1})}$$

$(l=c_n,c_n-1,\dots,1)$. It shows that after c_n steps of Golomb's method, we get (2.2) from $\frac{a_n}{b_n}$.

3. Example

As an example, we calculate the Egyptian fraction expansions of the rational number $\frac{47}{64}$ both by Golomb's method and by the continued fraction method.

Golomb's method Golomb's method gives the result through the following steps:

The multiplicative inverse of 47 modulo 64 is 15, hence $\frac{47}{64}=\frac{11}{15}+\frac{1}{960}$.

The multiplicative inverse of 11 modulo 15 is 11, hence $\frac{11}{15}=\frac{8}{11}+\frac{1}{165}$.

The multiplicative inverse of 8 modulo 11 is 7, hence $\frac{8}{11}=\frac{5}{7}+\frac{1}{77}$.

The multiplicative inverse of 5 modulo 7 is 3, hence $\frac{5}{7}=\frac{2}{3}+\frac{1}{21}$.

Finally, the multiplicative inverse of 2 modulo 3 is 2, hence $\frac{2}{3}=\frac{1}{2}+\frac{1}{6}$.

Summarizing these equations, it follows that the Egyptian fraction expansion by Golomb's method is

$$\frac{47}{64}=\frac{1}{2}+\frac{1}{6}+\frac{1}{21}+\frac{1}{77}+\frac{1}{165}+\frac{1}{960}.$$

Continued fraction method The Euclidean algorithm gives the finite simple continued fraction expansion $\frac{47}{64}=\langle 0,1,2,1,3,4\rangle$ and the sequences $(a_k)_{k=0}^5=(0,1,2,3,11,47)$, $(b_k)_{k=0}^5=(1,1,3,4,15,64)$. Then the continued fraction method works as follows:
First, by application of the odd subscript case we obtain $\frac{47}{64}=\frac{11}{15}+\frac{1}{960}$.
Thereafter we apply the even subscript case twice to get $\frac{11}{15}=\frac{2}{3}+\frac{1}{21}+\frac{1}{77}+\frac{1}{165}$ and $\frac{2}{3}=\frac{1}{2}+\frac{1}{6}$.

Consequently, the Egyptian fraction expansion by this method is

$$\frac{47}{64}=\frac{1}{2}+\frac{1}{6}+\frac{1}{21}+\frac{1}{77}+\frac{1}{165}+\frac{1}{960},$$

which is the very same as above.

4. Connection with the Farey sequence method

Our observation could be verified also through Farey sequences. Denote by $\mathcal{F}_n$ the Farey sequence of order n, that is the list of all reduced rational numbers in $[0, 1]$, having denominators less than or equal to n, in increasing order. The main properties of Farey sequences can be found in [8, 9].

In [1], see also [3], the Farey sequence method is presented to obtain an Egyptian fraction expansion of a positive rational number. Let $0 < \frac{a}{b} < 1$ be a rational number, where a and b are positive integers with $\gcd(a, b) = 1$. If $\frac{c}{d}$ is the preceding fraction in $\mathcal{F}_b$, then

$$\frac{a}{b} = \frac{c}{d} + \frac{1}{db},$$

where $d < b$, and we can continue the method on $\frac{c}{d}$.

We have to notice that in practice this form of the Farey sequence method is only an algorithm in principle, because it says nothing about how to find the preceding fraction in $\mathcal{F}_b$.

Then it is straightforward that Golomb's method coincides with the Farey sequence method, since $ad = bc + 1$ and $0 < d < b$ is the multiplicative inverse of a modulo b.

On the other hand, the Farey sequence method gives the same result as the continued fraction method, which can be deduced from the following fact: For odd n, the preceding fraction of $\frac{a}{b} = \langle c_0, c_1, \ldots, c_n \rangle$ $(c_0 = 0)$ in $\mathcal{F}_b$ is $\langle c_0, c_1, \ldots, c_{n-1} \rangle$. While for even n, the preceding fraction is $\langle c_0, c_1, \ldots, c_{n-2} \rangle$ if $c_n = 1$, furthermore $\langle c_0, c_1, \ldots, c_{n-1}, c_n - 1 \rangle$ if $c_n \geq 2$. Thus the preceding fraction is a primary or secondary convergent, which is already mentioned in a half sentence in [3].

5. Teaching possibilities

Elementary number theory textbooks, lecture notes (see e.g. [8, 9]) and undergraduate courses often deal with Farey sequences and continued fractions. Our experiences show that these topics are rather popular among university students. Because of their interesting properties, they are also suitable to be the subject of popular science lectures or mathematics study circles for advanced secondary school students. At a higher level, in the theory of diophantine approximation, both Farey sequences and continued fractions are used to give alternative proofs of Hurwitz's theorem. Nevertheless, these topics are always handled in separate chapters, we can hardly find any sources about their connection.

Thanks to the simplicity of the necessary notions and the historical background, we think that Egyptian fractions also give a rewarding topic to popularize mathematics. On the other hand, at university level, as the lecturer's material or as the subject of students' project work, it can be an unordinary base to introduce both Farey sequences and continued fractions, as well as their properties. And it allows us not only to study them separately, but one can find out their close connection, as we have done above.

References

[1] A. Beck, M. N. Bleicher and D. W. Crowe, Excursions into Mathematics, A K Peters, 2000.

[2] L. Beeckmans, The splitting algorithm for Egyptian fractions, *Journal of Number Theory* **43** (1993), 173–185.

[3] M. N. Bleicher, A new algorithm for the expansion of Egyptian fractions, *Journal of Number Theory* **4** (1972), 342–382.

[4] P. J. Campbell, A „practical" approach to Egyptian fractions, *Journal of Recreational Mathematics* **10** (1977-78), 81–86.

[5] M. Dunton and R. E. Grimm, Fibonacci on Egyptian fractions, *Fibonacci Quarterly* **4** (1966), 339–354.

[6] D. Eppstein, Ten algorithms for Egyptian fractions, *Mathematica in Education and Research* **4** (1995), 5–15. (An extended version is available at the author's web page: `http://www.ics.uci.edu/~eppstein/numth/egypt/intro.html`)

[7] S. W. Golomb, An algebraic algorithm for the representation problems of the Ahmes papyrus, *American Mathematical Monthly* **69** (1962), 785–786.

[8] G. H. Hardy and E. M. Wright, An Introduction to the Theory of Numbers, Oxford University Press, 2008.

[9] I. Niven, H. S. Zuckerman and H. L. Montgomery, An Introduction to the Theory of Numbers, John Wiley & Sons, 1991.

[10] J. J. Sylvester, On a point in the theory of vulgar fractions, *American Journal of Mathematics* **3** (1880), 332–335.

Annales Mathematicae et Informaticae
42 (2013) pp. 135–143
http://ami.ektf.hu

Inserting RFID systems into the Software Information Technology course*

Tibor Radványi, Emőd Kovács

Eszterházy Károly Collage
Institute of Mathematics and Informatics
radvanyi.tibor@ektf.hu emod@aries.ektf.hu

Submitted November 4, 2013 — Accepted December 19, 2013

Abstract

In this article we would like to introduce you to the educational structure of the program design and informatics course at the Eszterházy Károly College. We will also mention the new fields of RFID system usage and the importance of this system in winning a spectacular tender. Furthermore you will get some details about the extra competencies our students gain by taking part in the RFID project. In order to familiarize them with the appliances and their features used in the RFID technology we have established a lab which serves more functions. First of all we find essential for students to take part in tender-related projects. Moreover, it is also important for our students to get the chance of learning the technology, using the appliances in practice, and examining them while they are working.

The lab is continuously being updated with gadgets, this way students get the opportunity of taking parts in group or individual projects beside their "guided tour" and the theoretical class.

Keywords: RFID, education, program design and informatics, project work

MSC: 68U35, 97Qxx

*This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of the TÁMOP-4.2.2.C-11/1/KONV-2012-0014, title: The developing possibilities of RFID/NFC technology by the conception of "Internet of Things".

1. Introduction

The aim of the program is training informatics experts capable of elaboration, introducing, operating, maintaining, developing, and applying software-oriented information technology equipments and systems either in groups or individually. Furthermore, the acquired theoretical knowledge will ensure graduates to continue their studies in second cycle M.A. programs.[7] Their education is separated to the following fields.

1.1. Basic classes

Mathematical and Scientific basics (Introduction to informatics, Discrete Mathematics, Calculus, Numeral Mathematics, Searching Operation, Combinatory and Probability Theory, Computer Statistics) Theory of Computing (Logical basics of Informatics, Theory of Computing, Automata and Formal Languages, Data Structures and Algorithms, Design and Examine algorithms, Basics of the Artificial Intelligence, Introduction to Computer Graphics)

1.2. Profession-related classes

Software technology module: (Advanced programming languages, Translation programmes, Programming technologies, Programming environments, Assembly languages) Information systems module: (Database systems, Managing database systems, System design, Technology of system design)

1.3. Facultative profession-related classes

Computerised word and publication processing, Spreadsheet systems, History of Informatics, Descriptive Geometry

1.4. Majors, specializations

Our students are capable of choosing specialization in the second half of the course. The theses written about the following subjects have to be justified at the final examination. Data models (Managing database systems 2, Advanced DBMS) Networks (Examining the effectiveness of networks, Server administration, Dynamical WEB programming) Computer graphics and geometry (Computer graphics, Graphical systems, Geometry modelling, Multimedia) Mathematical methods within Informatics (Neural networks, Computer Statistics 2, Searching Operation 2, Cryptography, Computer algebraic systems)

1.5. Inserting Research & Development and Talent Development into the course

We put a high pressure on seeking students, who are looking for a great academic career, succeed at TDK competitions, or seem to be the future generation of lecturers and researchers. Students who turn out to be curious or talented in the first term are attending lectures for two months held by the college's researchers to pique their interest. The common work gets started at the end of the second term and might bring fruitful results in the second and third year. The college runs three RFID and a Robotics lab, where students can learn some teamwork. Problems they solve in the labs are partly everyday programming issues, but sometimes they work on tasks that might occur in the industrial sphere.

2. The RFID lab and the connecting projects

The lab was established in the Institute of Mathematics and Informatics in one room in 2009. The inventory was partly ensured by the Institution like the furniture, three workstations, and a server. The appliances of the lab were donated by an industrial company. These appliances include a power-type industrial tag printer, an RFID writer, and mobile RFID reader, installed with a compatible PDA. These gadgets were expanded with a fix installed gate, and an RFID reader. The mobile reader provides us to be able to take through examinations out of the lab. We have expanded the tools with numerous passive RFID tags. They include the paper-based tag used by the printer, and the industrially capable hardtag as well. After the difficulties at the beginning, three enthusiastic students helped with starting to run the lab.

We achieved the following results in the period 2009-2013:

Figure 1: Intermec mobile reader, PDA Windows CE

Figure 2: industrial hardtags

2.1. Thesis

All together 14 thesis have been written based on the topic

- Usage of RFID in the library
- Temperature allowance of RFID tags
- Examining efficiency issues
- Attack possibilities and cryptography
- Stock registration with the help of RFID
- Possibilities of following the products manufacturing

2.2. Academic results

- 2 articles
- 2 conference articles
- 7 conference lectures and posters
- 1 submitted article

We began the examinations with using UHF passive tags.[6] Its reason roots in the fact that the projects' specification starting or running now requires it. Moreover, they are absolutely capable of learning, taking examinations, coming to know the system, and their price is more affordable, than active tags'.[3]

To sum up in this period approximately 15-18 students got in touch with the RFID technology. Furthermore, with the help of projects and theses we have managed to acquire the basics and pique the interest.

The TÁMOP-4.2.2.C-11/1/KONV-2012-0014 The developing possibilities of RFID/NFC technology by the conception of "Internet of Things" tender has started on January 1st 2013. This tender is providing appreciable financial sources, and the support of the engineers of the Bay Zoltan Institute also means a lot of help for us. The number of the RFID labs has grown to three. The former lab contains the appliances, antennas, and readers which are capable of examining the UHF frequency interval. These tools have been expanded with more mobile readers and industrial readers, ready to be installed.[5, 6]

Another examining room has been set up with tools to examine the features of the electro-magnetic space and the active (33 MHz) technology.

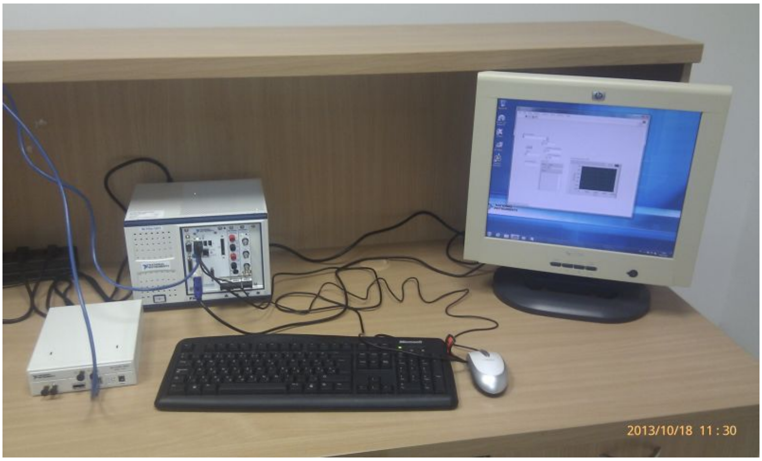

Figure 3: National Instruments examining set

The most amazing item of the third lab is the plotting board which is controlled with RFID tags and readers working in the HF interval (13,56 MHz). Beside it the lab contains readers, cards and tags which are necessary to learn about and work with the HF/NFC technology.

The teachers of the Institute of Mathematics and Informatics are already working in the set-up labs, and under their control students can work on this technology. In the first three years one teacher and 15-18 students were dealing with the RFID technology. Winning the tender has given us such a boost that today 16 teachers control 20 students' job in the labs.

3. Part-tasks connected to the tender

The tender project is separated into part-tasks, which are joined by our students as well. These are the following:

- Digital proprietary rights
- Reliability

- Unionisation
- Localisation, tasks of the softwares and the middle layer
- Sensors, energy harvesting, data safety
- Hybrid technologies

Meanwhile the project was processing the basic research fields were expanded with applied research and development projects. More projects are also joined by enthusiastic and wholehearted students:

- Entering and work-time registration systems supported with RFID
- Automatic library
- Selective waste collection using RFID
- Registering living animals and their health condition
- The effect of micro climate changes on the fauns, collecting physical data automatically with the help of sensor networks
- Using RFID in food safety projects

The Institute of Mathematics and Informatics creates the RFID research institution in order to continue the arrangement and coordination of the common work of students and teachers in the above mentioned widespread basic and applied research fields.

Managing to insert the Automatic identification subject into the course is considered to be a great success. This subject includes two lectures and two practices a week. During the lecture the following topics are discussed:

- Review of the automatic identification, standards, and organisations
- The function, structure, mathematical background, and types of the 1D and 2D barcodes
- Security issues in the barcode technology, typical appliances and solutions, steps of development
- Physical bases of the RFID system, dynamic EM spaces, and their description
- Basics of the RFID, standards, frequencies, physical bases
- Items of the system, tags, readers and their types
- Describing the RFID communication algorithm
- RFID system design, opened and closed systems, developing RFID softwares

- Data security and cryptographic opportunities in the field of automatic identification
- Smartcards, chip cards, and their usage
- Social and judicial environment, expectations

Furthermore advantages can be made from the opportunities provided by the labs and the programming competencies students have gained during the former terms.

- Using the plotting board
- Programming the plotting board. Signals.
- Programming the plotting board. Switches.
- Programming the plotting board. Mixed access.
- Mobile readers, type Intermec and Alien, passive tags.
- Chance of reading and programming on the mobile readers. Passive reading and writing.
- Usage of active tags and readers

Figure 4: Plotting board with trains and HF tags

The work students do in the projects insensate and specialise their knowledge.

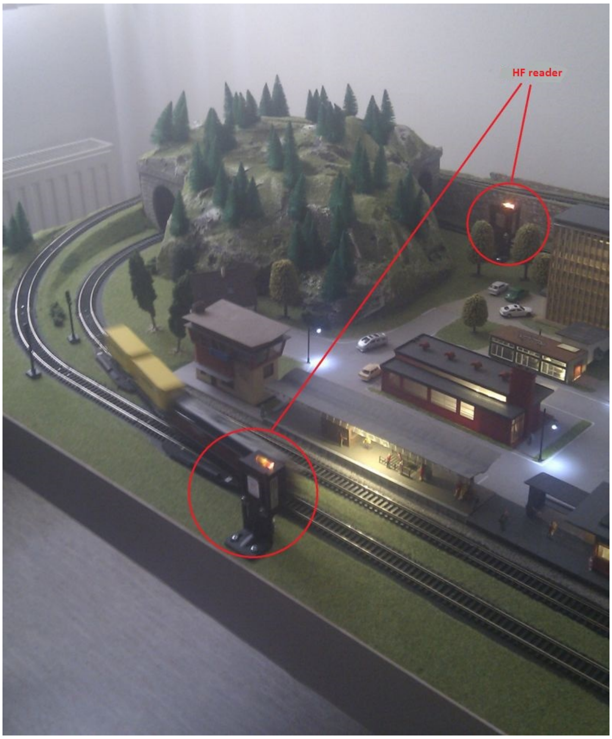

Figure 5: Plotting board with HF readers

4. Summarize

All in all increasing the role of practice in the course is inevitable. This field could be strengthened with projects generated by tenders. Students are unquestionably eager to join the projects, work on them diligently, and profit from these experiences during their latter work. Another highlighted aspect is that these students own a vantage with their new and notable competency within the labour power. According to our former experiences our graduated students are pleased to use this vantage.

References

[1] Kovács Emőd, Kusper Gábor, Web-programozó felsőfokú szakképzés kialakítása az Eszterházy Károly Főiskolán, *Informatika a felsőoktatásban*(2008).

[2] Szűcs László, Schwirg Laszló, Deploying an automaton warehouse system with RFID technology, *GINNT Budapest*, (2009).

[3] Az oktatási miniszter 15/2006. (IV. 3.) OM rendelete az alap- és mesterképzési szakok képzési és kimeneti követelményeiről.

[4] Klaus Finkenzeller, RFID HANDBOOK Fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication, third edition, *WILEY, ISBN: 978-0-470-69506-7.*

[5] Qinghan Xiao1 Thomas Gibbons Herve Lebrun, RFID Technology, Security Vulnerabilities, and Countermeasures.

[6] EPCglobal Inc., Class-1 Generation-2 UHF RFID Conformance Requirements Specification v. 1.0.2.(2005).

[7] Tibor Radványi, Emőd Kovács, Infiltration of RFID technological knowledge in teaching of informatics teacher MA, *Romanian Journal of Education Kolozsvar, ISSN: 2067-8347* Vol.1 (2010) 49–54.