INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

Year: 2006 Volume: I Number: 1

CCC Publications

www.journal.univagora.ro

EDITORIAL ORGANIZATION
Editor-in-Chief
Prof. Florin-Gheorghe Filip
Member of the Romanian Academy

Executive Editor
Dr. Ioan Dziţac

Managing Editor
Prof. Mişu-Jan Manolescu

Technical Editor & Desktop Publishing
Horea Oros

Publisher & Editorial Office
CCC Publications
Agora University
Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel: +40 259 427 398, Fax: +40 259 434 925, E-mail: ccc@univagora.ro
Website: www.journal.univagora.ro
ISSN 1841-9836 (print version)
ISSN 1841-9844 (online version)

Contents

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 5-6

Editorial

Florin Gheorghe Filip, Ioan Dziţac, Mişu-Jan Manolescu

Welcome to the International Journal of Computers, Communications & Control (IJCCC), a new scientific journal in Computer Science.

1 Why IJCCC?

Nowadays there are many prestigious journals in the field of Computer Science, so it is only natural to ask the question: "Is there any reason for a new journal in this area?" There is a saying "in the world of computers the only thing that stays the same is the changing".

We live in the most dynamic era of creation and circulation of information. The transition from the Information Society to Knowledge Society is accompanied by the globalization of information. More and more specialists, researchers from institutes, universities and industries of all over the world, write scientific papers in this field. These papers must be put in the scientific circuit as fast as possible and, the existing journals cannot publish the increased number of papers.

The increased interest of the participants from many countries for the *International Conference on Computers and Communications*, initiated and organized by Ioan Dziţac in 2004[1] (with Constantin Popescu and Horea Oros, http://iccc.rdsor.ro), under the guidance of acad. Florin-Gheorghe Filip and continued in 2006[2] under the name of *International Conference on Computers, Communications & Control* (www.iccc.univagora.ro) at Agora University of Oradea (president Mişu-Jan Manolescu), gave us the courage and confidence to start this new journal, entitled International Journal of Computers, Communications & Control (IJCCC).

To this new editorial project have subscribed as members of the Editorial Board, the following professors and specialists in Computer Science:

- Pierre Borne, Ecole Centrale de Lille, France;
- Antonio Di Nola, University of Salerno, Italy;
- Ioan Dziţac, Agora University, Oradea, Romania (Executive Editor);
- Ömer Egecioglu, University of California, Santa Barbara, USA;
- Florin Gheorghe Filip, Romania Academy, Romania (Editor in Chief);
- Constantin Gaindric, Institute of Mathematics and Computer Science of the Academy of Sciences of the Republic of Moldova;
- Kaoru Hirota, Tokyo Institute of Technology, Japan;
- Mişu-Jan Manolescu, Agora University, Oradea, Romania (Managing Editor);
- George Metakides, University of Patras, Greece;
- Shimon Y Nof, Purdue University, USA;
- Gheorghe Păun, IMAR, Romanian Academy, Romania;
- Mario J. Perez-Jimenez, University of Seville, Spain;
- Imre J. Rudas, Budapest Tech, Hungary;
- Athanasios D. Styliadis, Alexander Institute of Technology, Thessaloniki, Greece;
- Gheorghe Tecuci, George Mason University, USA;
- Horia-Nicolai Teodorescu, Institute of Computer Science, Romanian Academy, Iasi, Romania;
- Dan Tufiş, RACAI, Romanian Academy, Romania.

[1] In 2004 we received for review 112 papers (only 67 accepted) from authors of Australia, Austria, China, Egypt, Finland, France, Germany, Greece, India, Ireland, Italy, Japan, Republic of Moldova, Romania, Spain, United Kingdom and USA.

[2] In 2006 we received for reviews 142 papers (only 91 accepted) from authors of previous countries and from others countries: Algeria, Hungary, Iran, Serbia & Montenegro, Tunisia and Thailand.

2 About IJCCC

International Journal of Computers, Communications & Control (IJCCC) is published from 2006 and has 4 issues per year, edited by CCC Publications, powered by Agora University Editing House, Oradea, ROMANIA (www.journal.univagora.ro).

Every issue is published in online format (ISSN 1841-9844) and print format (ISSN 1841-9836). We offer free online access to the full content of the journal. The printed version of the journal should be ordered, by subscription, and will be delivered by regular mail.

IJCCC is directed to the international communities of scientific researchers from the universities, research units and industry.

IJCCC publishes original and recent scientific contributions in the following fields:

- Computing & Computational Mathematics;
- Information Technology & Communications;
- Computer -based Control.

To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (**C**omputing, **C**ommunication, **C**ontrol).

The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by two reviewers.

IJCCC also publishes:

- papers dedicated to the works and life of some remarkable personalities;
- reviews of some recent important published books.

Also, IJCCC will publish as supplementary issues the proceedings of some international conferences or symposiums on Computers, Communications and Control, scientific events that have reviewers and program committee.

The authors are kindly asked to observe the rules for typesetting and submitting described in Instructions for Authors, which are to be found at the end of the journal and on the journal's website www.journal.univagora.ro.

There are no fees for processing and publishing articles. The authors of the published articles will receive a hard copy of the journal.

topics of interest include, but are not limited to, the following: Applications of the Information Systems; Artificial Intelligence; Automata and Formal Languages; Collaborative Working Environments; Computational Mathematics; Cryptography and Security; E-Activities; Fuzzy Systems; Informatics in Control; Information Society - Knowledge Society; Natural Computing; Network Design & Internet Services; Multimedia & Communications; Parallel and Distributed Computing.

Florin-Gheorghe Filip
Editor in Chief of IJCCC
Romanian Academy
125, Calea Victoriei, 71102
Bucharest-1, Romania
E-mail: ffilip@acad.ro

Ioan Dziţac
Executive Editor of IJCCC
Agora University
8, P-ta Tineretului, 410526
Oradea, Romania
E-mail: idzitac@univagora.ro

Mişu-Jan Manolescu
Managing Editor
Agora University
8, P-ta Tineretului, 410526
Oradea, Romania
E-mail: rectorat@univagora.ro

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 7-8

Introduction to the first issue

Ioan Dziţac

In this inaugural issue we present:

- an editorial article, with a general presentation of IJCCC and this introduction with short description of the papers published in this first issue;
- one anniversary article dedicated to centenary of Kurt Gödel (1906-1978) and two anniversary articles dedicated to centenary of Grigore C. Moisil (1906-1973);
- some selected papers from the International Conference on Computers, Communications & Control, June 1-3, 2006, Băile Felix - Oradea, Romania, ICCCC 2006;
- two regular papers.

1 Introducing the papers

Daniel C. Doolan and Sabin Tabirca's paper[1] discusses how a Java 2 Micro Edition (J2ME) application may be developed to visualize a wide variety of differing fractal types on a mobile phone. An integrated tool has been developed by authors to generate a variety of two dimensional fractals (Mandelbrot Set, Julia Set, Prime Number Fractal and the Plasma Fractal).

Banshider Majhi, Y Santhosh Reddy, D Prasanna Babu's propose in their paper a new feature extraction scheme for offline signature verification. The method uses geometric center for feature extraction and Euclidean distance model for classification. This classifier is well suitable for features extracted and fast in computation. The method proposed in this paper leads to better results than existing offline signature verification methods.

Gheorghe Păun initiated *membrane computing*, most known as *P systems* (with "P" from "Păun"; for more details see http://psystems.disco.unimib.it) by paper "Computing with membranes", published in Journal of Computer and System Sciences, 61, 1 (2000), 108-143 (first circulated as TUCS Research Report No 208, November 1998, http://www.tucs.fi). In the paper published in this issue he continues the attempt to bridge brane calculi with membrane computing, following the investigation started in the paper L. Cardelli, Gh. Păun, An universality result for a (mem)brane calculus based on mate/drip operations, *Intern. J. Foundations of Computer Sci.*, 17, 1 (2006), 49-68.

Dana Petcu, Cosmin Bonchiş, Cornel Izbaşa present in their paper[2], a case study of a Web service-based Grid application for symbolic computations (known as Computer Algebra Systems (CAS)), using Maple2g (Maple-to-Grid).

Imre Rudas and Janos Fodor summarizes in their paper[3] the research results of the authors that have been carried out in recent years on generalization of conventional operators. Aggregating uncertain information is an important issue in the study of Intelligent Systems. In their contribution, the authors present an overview on generalized operators from the point of view of Information Aggregation. In this spirit, a comparative study of triangular norms and conorms, uninorms and nullnorms, generalized conjunctions and disjunctions and distance-based operations is presented.

Milan Stanojevic and Mirco Vujosevic's propose in their paper[4] a new original more efficient algorithm formulation and implementation for solving relatively low dimensions Steiner Tree problem on graph.

Athanasios D. Styliadis, Ioannis D. Karamitsos, Dimitrios I. Zachariou's paper[5] introduces the GIS e-learning implementation, based on a set of teaching (lecturing) rules according to the cognitive style of learning preferences of both the learners and the lecturers as well.

[1]Selected from ICCCC 2006
[2]selected from ICCCC 2006
[3]Plenary invited paper of ICCCC 2006
[4]Invited paper of ICCCC 2006
[5]Invited paper of ICCCC 2006

2 Introducing the anniversary articles

Gabriel Ciobanu's article presents some impressions about life, work, and foundational views of Kurt Gödel, after his participation to the international symposium celebrating the 100th birthday of Kurt Gödel, organized between 27-29 April 2006 by the Kurt Gödel Society and University of Vienna. Gabriel Ciobanu has been accepted to this symposium with a contribution regarding a new characterization of computable real numbers.

Solomon Marcus present in this first issue of IJCCC a very well documented article[6], devoted to 100 years since the birth of Great Romanian Mathematician and Computer Pioneer (IEEE-1996), Grigore C. Moisil (1906-1973).

In 1976, 1980 and 1992, acad. Solomon Marcus has edited and prefaced the following volumes:

- Gr. C. Moisil, Opera matematica, vol.I, Editura Academiei, Bucuresti, 1976 (preface, edition and introductory study).
- Gr. C. Moisil, Opera matematica, vol.II, Editura Academiei, Bucuresti, 1980 (preface, edition and introductory study).
- Gr. C. Moisil, Opera matematica, vol.III, Editura Academiei, Bucuresti, 1992 (edition and introductory study).

George Georgescu, Afrodita Iorgulescu, Sergiu Rudeanu, three remarkable Moisil's disciples, presents, in their technical paper[7], a very concise and updated survey emphasizing the research done by Grigore C. Moisil and his school in algebraic logic (n-valued Lukasiewicz-Moisil algebra, θ-valued Lukasiewicz-Moisil algebra, Post algebra etc.).

Ioan Dziţac
Executive Editor of IJCCC
Agora University
8, P-ta Tineretului, 410526
Oradea, Romania
E-mail: idzitac@univagora.ro

[6]Plenary invited talk of ICCCC 2006
[7]Invited paper of ICCCC 2006

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 9-15

Visualising Infinity on a Mobile Device

Daniel C. Doolan, Sabin Tabirca

Abstract: This paper discusses how a Java 2 Micro Edition (J2ME) application may be developed to visualise a wide variety of differing fractal types on a mobile phone. A limited number of J2ME applications are available that are capable of generating the Mandelbrot Set. At present there are no J2ME applications capable of generating a multitude of fractal image types.
Keywords: J2ME, Mobile Phone, Fractals

1 Introduction

It has been shown that mobile devices are capable of generating high quality images of infinite detail [2]. The generated images have generally been limited to the Mandelbrot Set. Since the late 19^{th} century fractals have been a favourite topic for mathematicians. Since the dawn of modern day microprocessor based computing the study of fractals has taken a radical leap, as it is within the computing domain that all the nuances of fractal type images can be visualised.

Benoit Mandelbrot made a huge contribution in the late 1970's with the discovery of the Mandelbrot Set (an index for all the possible Julia Sets). The dawn of the 21^{st} century has seen a radical changed in what we consider a computer to be. It has seen the widespread uptake of mobile phones throughout the world. Devices that started life in the latter years of the 20^{th} century as a mobile communications medium have evolved and mutated into mobile computing devices of considerable processing power. No longer are mobile phones used for just telecommunications but for just about any type of application that a standard desktop machine is capable of.

The 2D and 3D visualisation of fractal images is one interesting topic that is coming into the realm of reality within the mobile computing domain. Current high end phones have processing speeds in the region of 100 to 200Mhz [7] [1], typically running ARM9 type processors. The next evolution in processing power will see such devices fitted with the ARM11 processor cores with speeds as high as 500Mhz. Mobile devices clearly have a huge processing potential especially if the combined processing power of the many millions of phones around the world were put to task on a singular problem. Examples of such computation are already in existence for example "Distributed Fractal Generation Across a Piconet" [3] demonstrates that the combined processing power of several mobile devices may be used to distribute the processing load between several devices that are connected together over a Bluetooth network.

1.1 Mobile Phone Market Penetration

The uptake of mobile devices around the world is staggering. In September 2004 the market penetration stood at 89% in Ireland [10], by March 2005 it stood at 94% [11]. This is a huge increase when in 2001 penetration stood at only 67% [8]. Ireland achieved 100% penetration in September 2005, showing an increase of over 11% in a twelve month period. This allows Ireland to join Spain, Finland the Netherlands and Austria in having a 100% penetration rage. Luxembourg is currently on top with a rate of 156% [12]. It is expected that Western Europe will exceed 100% usage by 2007 [15]. The year 2015 should see half the world's population (Four billion people) using mobile phones as a communications medium [9].

1.2 Primary Aims

The primary purpose of this paper is the development of an application capable of running on a single mobile device that has the ability to generate a variety of two dimensional fractal images. One of the chief aims is that the application should be easy to use. This would allow it to be used as a teaching tool. To achieve this each section of the application has a easy to use Graphical User Interface (GUI) to allow the user to specify the parameters for the image generation process. Once the user is happy with the image parameters they can then press a button to begin the image generation process. The resultant output will be a fractal image based on the input parameters.

2 Mandelbrot & Julia Set Generation

The discovery of the Julia Set in 1918 by Gaston Julia described in his paper “Mémoire sur l’itération des fonctions rationnelles” proved to be a most important work at the time. It became almost forgotten until Benoit Mandelbrot brought it back to the forefront with his discovery of the Mandelbrot Set. This ensued a new field of research that became known as fractal geometry. Both the Julia and Mandelbrot Set images can be generated by the repeated iteration of a simple function (see Figure 2). The Mandelbrot Set is an Index into the Julia Set, in other words every possible Julia Set can be represented within the Mandelbrot Set (see Figure 1).

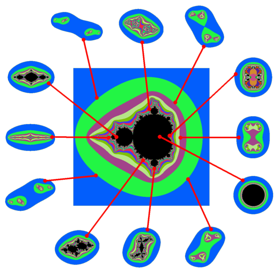


Figure 1: Index of the Julia Set (the Mandelbrot Set)

$$J_c = \left\{ Z_0 \in \mathbb{C} | \lim_{n\to\infty} Z_n \neq \infty \right\} where:$$

$$Z_0 = C, Z_{n+1} = f(Z_n), n \geq 0$$

$$M = \left\{ c \in \mathbb{C} | \lim_{n\to\infty} Z_n \neq \infty \right\} where:$$

$$Z_0 = 0, Z_{n+1} = f(Z_n), n \geq 0$$

Figure 2: Julia & Mandelbrot Set Definitions

Fractal images are usually obtained when the generating function $f(z)$ is non linear. The Mandelbrot Set is obtained by iterating the function $f(z) = z^2 + c$. When the generating function has the form of $f(z) = z^u + c^v$ many other Mandel-like Sets may be produced. Algorithms 1 and 2 show how both the Julia and Mandelbrot Sets can be generated.

Algorithm 1 The Julia Set Algorithm

for each (x,y) in $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$
  construct $z_0 = x + j \times y$;
  find the orbit of z_0 [first N_{iter} elements]
  if(all the orbit points are under the threshold)
    draw (x,y)

Algorithm 2 The Mandelbrot Set Algorithm

for each (x,y) in $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$
  $c = x + i \times y$;
  find the orbit of z_0 while under the threshold R
  if(all the orbit points are not under the threshold)
    draw (x,y)

2.1 Implementation

A simple to use Graphical User Interface (GUI) is provided within the application to allow the user to enter various parameters detailing the type of image to be generated (see Figure 3). The parameters dealing with the fractal image itself include: the image size, number of iterations, radius, cPower, zPower, formula type and image inversion. The other options are for the rate of fractal zoom, and the accuracy of the crosshair (seen in the Image output screen (Canvas)).

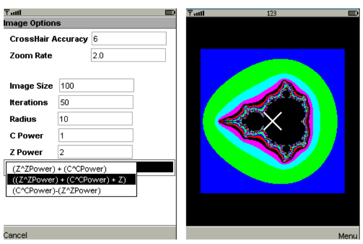


Figure 3: Mandelbrot Settings GUI and Output Image

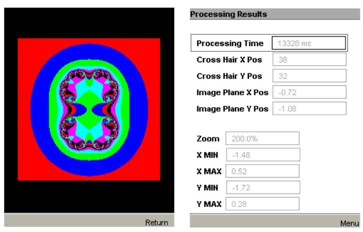


Figure 4: Julia Set Image & Results Output Screen

A thread is used for the generation of the fractal image to allow for user interaction with the image as it is being generated. The outputted image is redrawn at regular intervals so the user can see the progress of the image generation process. The fractal image itself is generated as an array of Integers (Listing 1). This array is then passed to the createRGBImage(...) method of the Image class to generate the Image object that is ultimately displayed within the onscreen Canvas. The Canvas has a crosshair present to allow the user to navigate around the image and select an area to zoom on. The crosshair is controlled by the directional keys of the mobile device. The user may also view the corresponding Julia Set (Figure 4) for any point that the crosshair is currently indicating, by selecting the “View Julia Set” option from the Mandelbrot Set Canvas Menu.

```
for(int i=0;i<SIZEX;i++)for(int j=0;j<SIZEY;j++){
   Complex c = new Complex((XMIN+i*STEPX),(YMIN+j*STEPY));
   Complex z = new Complex();
   for (k=0;k<NR_ITER;k++){
      z=f(z,c);
      if(z.getAbs()>R){
         r = c[k%l][0]; g = c[k%l][1]; b = c[k%l][2];
         color = b + (g<<8) + (r<<16) + alpha;
         pixels[(j*SIZEX) + i] = color;
         break;
      }
   }
}
```

Listing 1: Code listing of Mandelbrot Set Function

The application allows for a variety of Mandel-like images to be generated of the form $f(z) = z^u + c^v$, $f(z) = z^u - c^v$ and $f(z) = z^u + c^v + z$. The cPower and zPower parameters of the GUI dictate the values of u and v. The inverted representation of each form may also be generated by selecting the appropriate option from the GUI, producing images of the form $f(z) = z^u + inv(c^v)$, $f(z) = z^u - inv(c^v)$ and $f(z) = z^u + inv(c^v) + z$.

2.2 Processing Results

The application was tested using a number of image sizes as well as a varying the number of iterations (see Table 1). The 6680 was unable to generate an image of 600 pixels square. Testing with a Nokia 3220 it was unable to generate an image of 200^2 pixels. However it was capable of generating images of 100^2 and 150^2 pixels, but with considerable processing times, 56,503ms and 298,365ms for 100^2 at 50 and 500 iterations respectively.

Device	**Iter**	100×100	200×200	300×300	400×400	500×500	600×600
Nokia 6630	50	3,000ms	9,812ms	19,484ms	33,812ms	53,359ms	76,859ms
Nokia 6680	50	3,141ms	12,516ms	21,859ms	36,234ms	60,859ms	N/A
Nokia 6630	500	13,281ms	52,344ms	111,484ms	196,125ms	301,344ms	429,047ms
Nokia 6680	500	12,359ms	48,000ms	103,219ms	187,797ms	290,062ms	N/A

Table 1: Image Generation Times for the Mandelbrot Set (xmin,ymin -2.0, xmax,ymax 2.0).

3 Prime Number Fractal Generation

Mathematicians have been studying prime numbers for thousands of years, with it origins going all the way back to ancient Greece and the period of Euclid. Prime numbers have remained a mystery throughout the intervening centuries. Their is currently ongoing work to find a prime number with ten million digits or more [4]. The largest prime found to date was discovered 15^{th} December 2005 by Dr. Curtis Cooper and Dr. Steven Boone, the 43^{rd} Mersenne Prime $2^{30,402,457} - 1$ being $9,152,052$ digits in length. The second largest was discovered by Dr. Martin Nowak 18^{th} February 2005, the 42^{nd} Mersenne Prime, $2^{25,964,951} - 1$ containing 7,816,230 digits. The next largest was discovered 15^{th} May 2004 by Josh Findley (41^{st} Mersenne Prime, $2^{24,036,583} - 1$ containing 7,235,733 digits. These discoveries were part of the Great Internet Mersenne Prime Search (GIMPS) project. It is likely that within the next year or so a ten million digit prime number will be found.

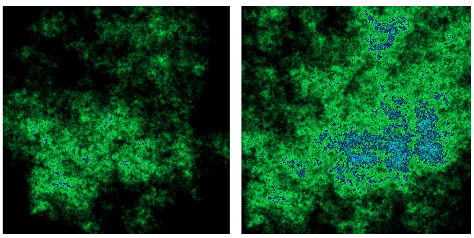


Figure 5: PNF Examples with 5 & 20 Million Primes

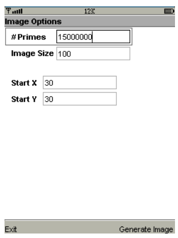


Figure 6: Prime Number Fractal User Interface

The visualisation of prime numbers is again of interest to many in the mathematical community. Probably the most well know visualisation was discovered by Stanislaw Ulam (the Ulam Spiral) in 1963 while doodling during a scientific meeting [14]. An alternate visualisation is the prime number fractal (Figure 5) where the resultant image has a central area of brightness and typically resembles a gaseous nebula or some other cosmic object. It is generally recognised that Adrian Leatherland from Monash University in Australia constructed the first prime number fractal [6].

The application has a very simple User Interface with only a few components (Figure 6). The options are: the Sieve Size, Image Size, and initial coordinates for x and y.

3.1 Theoretical Background

The most important and evident feature of a prime number fractal image is the central area of brightness. This results in the pixels within the vicinity of the central area are visited more often than pixels around the periphery of the image. The movements in the Up, Down, Left, Right directions occur randomly but each direction produces approximately the same number of moves (see Table 3). Hence the trajectory of moves is random, generally staying around the central area of brightness.

Theorem 1. *The number of Up, Down, Back, Forward movements are asymptotically equal.*

Proof. Dirichlet's theorem assures if a and b are relatively prime then there are an infinity of primes in the set $a \cdot k + b, k > 0$. This means that the random walk has an infinity of Up, Down, Left, Right moves. If $\pi_{a,b}(x)$ denotes the number of primes of the form $a \cdot k + b$ less than x then we know from a very recent result of Weisstein [16] that $\lim_{x\to\infty} \frac{\pi_{a,b}(x)}{li(x)} = \frac{1}{\varphi(a)}$ where $li(x)$ is the logarithmic integral function and $\varphi(a)$ the Euler totient function. The particular case $a = 5$ gives $\lim_{x\to\infty} \frac{\pi_{5,k}(x)}{li(x)} = \frac{1}{\varphi(5)} = \frac{1}{4}, \forall k \in 1,2,3,4$, which means that $\pi_{5,1} \approx \pi_{5,2} \approx \pi_{5,3} \approx \pi_{5,4}(x) \approx \frac{li(x)}{4}$ clearly the two dimensional prime number fractal algorithm has asymptotically the same number of Up, Down, Left and Right moves.

Even on Desktop systems the process of generating primes can take a significant amount of time for example 21,157ms on a AMD Athlon XP2600 system (Table 4). The generation of the primes examines all the numbers between 0 and n the input size. As the prime numbers become larger their distribution becomes far sparser (Table 2). Their should be at least one prime between n and $2(n)$.

# Primes	Sieve Size
1 Million	15,485,865
2 Million	32,452,850
5 Million	86,028,130
10 Million	179,424,680

Table 2: Sieve Size

#Primes	1 Million	5 Million	10 Million	20 Million
Left	249,934	1,249,832	2,499,755	4,999,690
Right	205,015	1,250,079	2,500,284	5,000,241
Up	250,110	1,250,195	2,500,209	5,000,270
Down	249,940	1,249,893	2,499,751	4,999,798

Table 3: Distribution of Moves for a two Dimensional Prime Number Fractal

#Primes	1 Million	5 Million	10 Million
Athlon XP2600	1,781ms	9,752ms	21,157ms
AMD 500Mhz System	4,084ms	23,946ms	51,552ms

Table 4: Processing Times to Generate Primes on Desktop Systems

3.2 The Fractal Algorithm

The generation of the fractal image requires the iteration through a loop (Algorithm 3) for all numbers from 1 to n. The first requirement of the algorithm is to determine if the number i is Prime or not.

The sieve of Eratosthenes is an time efficient method for determining primality of a sequence of numbers. The time complexity is $O(n \cdot log\, log\, n)$, however the space requirement is $O(n)$. The algorithm iterates through the sequence from 2 to n crossing off all numbers > 2 that are divisible by 2. It then moves on to the smallest remaining number, and removes all of its multiples. The process of moving on and removing multiples continues until all numbers up to $\sqrt{n}$ have been crossed off. In Java one may use an array of boolean values to indicate if the number at index i of the array is prime. A boolean in J2SE requires one byte for storage, however the BitSet class may be used. This reduces the storage requirements to $O(\frac{n}{8})$ bytes. As mobile devices have such a limited memory this approach cannot be used, hence the need for an alternate method. Determination of the primality is achieved by calling the method *isPrime*(i) (Algorithm 4). It returns a boolean value indicating if the number is prime or not.

If the number is prime then the process of plotting the prime takes place. Firstly the direction is calculated by carrying out modular division by 5 yielding $p \bmod 5 \in \{1,2,3,4\}$ this mapping can be clearly seen in Table 5. Next depending of the direction the current pixel location is updated to reflect the direction indicated by the prime. Lastly the colour of the pixel at the new cell location is incremented.

Algorithm 3 2D Prime Number Fractal Algorithm

```
for i = 0 to n -1 do begin
    if(isPrime(i)) then
        dir = p[i] mod 5;
        if dir = 1 then x–;
        if dir = 2 then x++
        if dir = 3 then y–
        if dir = 4 then y++
        pixels[x,y] ++;
    end if;
end for;
```

Algorithm 4 isPrime( ) Algorithm

```
boolean isPrime(long p)
begin
    long d;
    if(p=1) return false;
    if(p=2 || p=3) return true;
    if(p % 2=0 || p % 3=0) return false;
    for(d=3;d <= sqrt(p); d=d+2)
        if(p % d=0) return false;
    return true;
end
```

1. Left:	$p \bmod 5 = 1 \Rightarrow (x,y)$	goes in	$(x - inc, y)$
2. Right:	$p \bmod 5 = 2 \Rightarrow (x,y)$	goes in	$(x + inc, y)$
3. Up:	$p \bmod 5 = 3 \Rightarrow (x,y)$	goes in	$(x, y - inc)$
4. Down:	$p \bmod 5 = 4 \Rightarrow (x,y)$	goes in	$(x, y + inc)$

Table 5: Mapping of Direction Values to Pixel Movement

3.3 Processing Results

As is the case with Desktop systems the generation of the primes is the most computationally expensive operation. The results (Table 6) show that a significant amount of time is required to generate the prime numbers. This limits the computation of the image to just a few hundred thousand primes so the image may be generated and displayed to the user in a reasonable amount of time.

#Primes	20,000	40,000	60,000	80,000	100,000
Nokia 6630	44,797ms	129,235ms	240,313ms	373,437ms	524,703ms
Nokia 6680	44,984ms	129,000ms	240,078ms	372,875ms	524,046ms

Table 6: Processing Times to Generate PNF on Mobile Devices

4 Plasma Fractal Generation

Plasma Fractals are often referred to as “Fractal Clouds” and the resultant image typically has a cloud like appearance (Figure 7). The generation of this type of fractal uses a recursive algorithm known as Random Midpoint Displacement. Applying the exact same algorithm in the 3D universe to height values will result in the generation of fractal terrain. An example of this method being used for the generation of terrain in the film industry is Star Trek II “The Wrath of Kahn” where Random Midpoint Displacement was used to generate the terrain of a moon, the scene being called the “Geneses Sequence”.

The procedure for generating a “Fractal Cloud” (Algorithm 5) begins by assigning a colour to each of the four corners of a blank image. Executing the “divide(...)” algorithm will firstly find the colour for the central point of the image by calculating the average value of the four corners. The colour value at the central point is then randomly displaced. The image area is then divided into four smaller sections by recursively calling the “divide(...)” again for each of the four quadrants. This division process will continue until the image cannot be

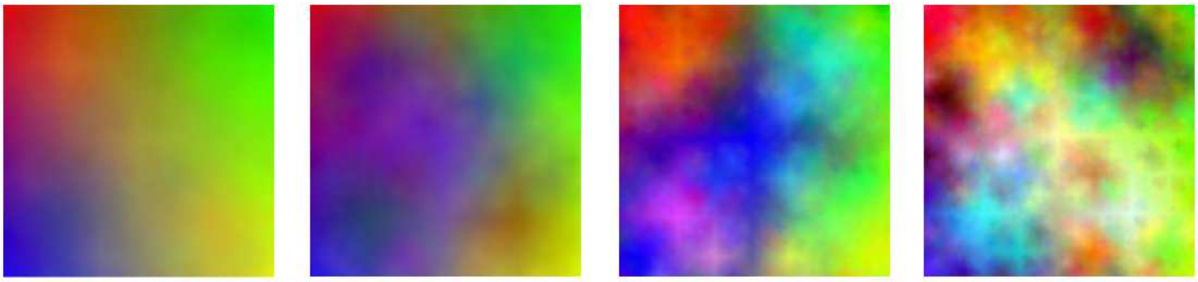


Figure 7: Plasma Fractal Examples (Grain 1.2 , 2.2, 5.8, 9.6)

further broken down. By this time the sub quadrants have reached the pixel level. Several example Applets that use this procedure may be found on the Internet [13] [5].

Algorithm 5 Plasma Fractal Algorithm

```
divide(x,y,w,h,tLC,tRC,bRC,bLC)
float nW = w /2, nH = h /2;
if(w > 1 || h > 1)
    int displace = displace(nW,nH);
    Color top = avgColors() + displace;
    Color right = avgColors() + displace;
    Color bottom = avgColors() + displace;
    Color left = avgColors() + displace;
    Color centre = avgColors() + displace;
    divide(x,y,nW,nH,tLC,top,centre,left);
    divide(x+nW,y,w,h,top,tRC,right,centre);
    divide(x+nW,y+nH,w,h,centre,right,bRC,bottom);
    divide(x,y+nH,w,h,left,centre,bottom,bLC);
else
    drawPixel(x,y)
```

4.1 Processing Results

The results show that mobile devices are capable of generating respectably sized Plasma Fractal Images in a reasonable amount of time (Table 7).

Image Size	100×100	200×200	300×300	400×400	500×500	600×600
Nokia 6630	672ms	2,219ms	6,672ms	6,328ms	6,156ms	23,453ms
Nokia 6680	671ms	2,328ms	7,468ms	7,344ms	6,828ms	25,329ms

Table 7: Processing Times to Generate Plasma Fractals

5 Further Work

The primary focus of this paper was on the generation of two dimensional fractal on Mobile Devices. The obvious progression from this application is to expand the generating functions into the third dimension. In the case of the Plasma Fractal this will yield a randomly generated terrain if the height values of a plane are randomly displaced instead of the colour values of a two dimensional image.

6 Conclusion

An integrated tool has been developed to generate a variety of two dimensional fractals. The fractals in question being the Mandelbrot Set, Julia Set, Prime Number Fractal and the Plasma Fractal. It has been shown that the Mandelbrot and Plasma Fractal Image can be generated in reasonable time. The generation of the PNF image however does involve significant computation resources, especially when the number of primes is greater than 100 thousand.

References

[1] A. Baker, "Mini Review - Enhancements in Nokia 6680," http://www.i-symbian.com/forum/images/articles/43/Mini_Review-Nokia_6680_Enhancements.pdf, 2005.

[2] D. Doolan, S. Tabirca, "Interactive Teaching Tool to Visualize Fractals on Mobile Devices," *Proceedings of Eurographics Ireland Chapter Workshop*,Dublin, Ireland pp. 7–12, 2005.

[3] D. Doolan, S. Tabirca, "Distributed Fractal Generation Across a Piconet," *Proceedings of SIGRAD05 - Mobile Computer Graphics Conference*,Lund, Sweden pp. 63–68, 2005.

[4] GIMPS, "Mersenne Prime Search," http://mersenne.org/.

[5] J. Lawlor, "Plasma Fractal Notes," http://charm.cs.uiuc.edu/users/olawlor/projects/2000/plasma/, 2001.

[6] A. Leatherland, "Pulchritudinous Primes: Visualizing the Distribution of Prime Numbers," http://yoyo.cc.monash.edu.au/~bunyip/primes.

[7] E. Murtazine, "Review GSM/UMTS Smartphone Nokia N90," http://www.mobile-review.com/review/nokia-n90-en.shtml, 2005.

[8] RTE, "Internet and Mobile Penetration Still Rising," http://www.rte.ie/business/2001/0308/odtr.html, 2001.

[9] RTE, "Half the world to have mobile phones by 2015," http://www.rte.ie/business/2004/0225/phones.html, 2004.

[10] RTE, "Mobile Penetration Now Stands at 89%," http://www.rte.ie/business/2004/0907/comreg.html, 2004.

[11] RTE, "Mobile Penetration Now Stands at 94%," http://www.rte.ie/business/2005/0318/comreg.html, 2005.

[12] RTE, "Mobile Penetration Now Stands at 100%," http://www.rte.ie/news/2005/1220/mobilephones.html, 2005.

[13] J. Seyster, "Plasma Fractals," http://www.ic.sunysb.edu/Stu/jseyster/plasma/, 2002.

[14] H. Systems, "Plasma Number Spiral," http://www.hermetic.ch/pns/pns.htm.

[15] C. Taylor, "Mobile penetration to hit 100% in Europe," http://www.enn.ie/news.html?code=9604990, 2005.

[16] E. Weisstein, "Arbitrarily Long Progressions of Primes," http://mathworld.wolfram.com/news/2004-04-12/primeprogressions/, 2004.

Daniel C. Doolan, Sabin Tabirca
University College Cork
Department of Computer Science
Cork, Ireland
E-mail: {d.doolan, tabirca }@cs.ucc.ie

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 17-24

Novel Features for Off-line Signature Verification

Banshider Majhi, Y Santhosh Reddy, D Prasanna Babu

Abstract: In this paper a novel feature extraction scheme has been suggested for offline signature verification. The proposed method used geometric center for feature extraction. Euclidean distance model was used for classification. This classifier is well suitable for features extracted and fast in computation. Method proposed in this paper leads to better results than existing offline signature verification methods. Threshold selection is based on statistical parameters like average and standard deviation (σ).
Keywords: Feature Extraction, Geometric Center, Euclidean Distance Model, Standard Deviation and Off-line Signature Verification.

1 Introduction

Signature verification is an important research area in the field of person authentication. We can generally distinguish between two different categories of verification systems: online, for which the signature signal is captured during the writing process, thus making the dynamic information available, and offline for which the signature is captured once the writing processing is over and, thus, only a static image is available[8]. The objective of the signature verification system is to discriminate between two classes: the original and the forgery, which are related to intra and interpersonal variability. The variation among signatures of same person is called Inrea Personal Variation. The vatiation between originals and forgeries is called Inter Personal Variation[7].
In this paper we concentrated on Offline Verification System. Upto now many signature verification methods proposed based on different strategies but no verification system classified near forgeries which were classified by this method. And the main advantage of this algorithm is efficiency and computational complexity. For general purpose applications like smart cards we want quick and efficient verification system[2]. This method is based on the Geometric Center and signature strokes distribution. Section 2 discusses the feature extraction from signature. This is a recursive method which applying on signature recursively. A lot of work has been done in the field of automatic off-line signature verification. While a large portion of work is focused on random forgery detection, more efforts are still needed to address the problem of skilled forgery detection[6]. Our method will be the first verification system which seperates some skilled forgeries from originals.
This paper organized in the following sections: Section 1.1 provides the different types of forgeries. Section 2 introduces new feature extraction method. Section 3 discusses classification based on Euclidean distance model. Section 4 discussed about threshold selection. Section 5 shows training, testing and results and Section 6 gives conclusion and furthure working directions.

1.1 Types of forgeries

There are three different types of forgeries to take into account. The first, known as *random forgery* which writtn by the person who don't know the shape of original signature. The second, called *simple forgery*, is represented by a signature sample which written by the person who know the shape of original signature without much practice. The last type is *skilled forgery*, represented by a suitable imitation of the genuine signature model[3]. Each type of forgery requires different types of verification approach[4]. Hybrid systems have also been developed[9] Fig. 1 shows the different types of forgeries and how much they are varies from original signature[5].

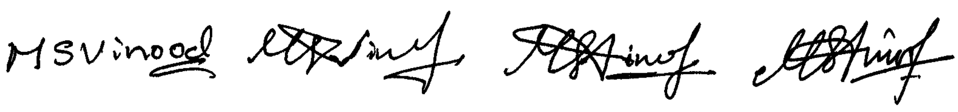


Figure 1: (a) Random Forgery (b) Simple Forgery (c) Skilled Forgery (d)Original Signature

By using this method we can easily eliminate random and simple forgeries. Some of the skilled forgeries also eliminated.

Figure 2: (a) Before adjustment of signature (b) After adjustment of signature

2 Feature Extraction

The geometric features proposed by this paper are based on two sets of points in two-dimentional plane. Each set having six feature points which represent the stroke distribution of signature pixels in image. These twelve feature points are calculated by Geometric Center[1]. Vertical Splitting and Horizontal Splitting are two main steps to retrieve these feature points. Vertical Splitting is discussed in Section 2.2 and Horizontal Splitting is discussed in Section 2.3.
Before finding feature points we have to do some adjustments to the signature image. That is moving signature strokes to the center of the image which discussed in Section 2.1.

2.1 Moving signature to the center of image

In this step signatures are moving to the center of image. Because of this we can reduce intra-personal variations. Here first we have to find out the geometric center of the image and move the signature pixels such that the geometric center should reside at center of image. Fig. 2 shows the signature images before moving and after moving.

2.2 Feature points based on vertical splitting

Six feature points are retrieving based on vertical splitting. Here feature points are nothing but geometric centers. The procedure for finding feature points by vertical splitting is mentioned in Algorithm.

Algorithm

This is the procedure for generating feature points based on verical splitting.
Input: Static signature image after moving the signature to center of image
Output: $v_1, v_2, v_3, v_4, v_5, v_6$ (feature points)
(a)Split image with vertical line at the center of image then we will get left and right parts of image.
(b)Find geometric centers v_1 and v_2 for left and right parts correspondingly.
(c)Split left part horizontal line at v_1 and find out geometric centers v_3 and v_4 for top and bottom parts of left part currespondingly.
(d)Split right part horizontal line at v_2 and find out geometric centers v_5 and v_6 for top and bottom parts of left part currespondingly.

Fig. 3 shows the feature points retrieved from signature image and O is the center of image. These features we have to calculate for every signatrure image in both training and testing.

2.3 Feature points based on horizontal splitting

Six feature points are retrieving based on horizontal splitting. Here feature points are nothing but geometric centers. The procedure for finding feature points by horizontal splitting is mentioned in Algorithm.

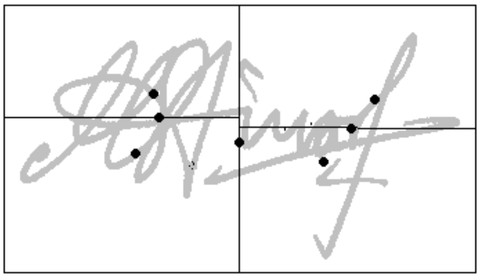


Figure 3: Feature points based on vertical splitting

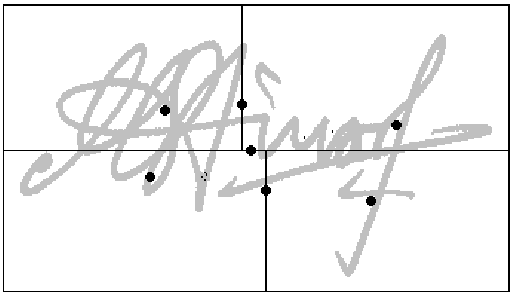


Figure 4: Feature points based on horizontal splitting

Algorithm

This is the procedure for generating feature points based on horizontal splitting.
Input: Static signature image after moving the signature to center of image
Output: $h_1, h_2, h_3, h_4, h_5, h_6$ (feature points)
(a)Split image with horizontal line at the center of image then we will get top and bottom parts of image.
(b)Find geometric centers h_1 and h_2 for top and bottom parts correspondingly.
(c)Split top part with vertical line at h_1 and find out geometric centers h_3 and h_4 for left and right parts of top part currespondingly.
(d)Split bottom part with vertical line at h_2 and find out geometric centers v_5 and h_6 for left and right parts of left part currespondingly.

Fog. 4 shows the feature points retrieved from signature image and O is the center of image. These features we have to calculate for every signatrure image in both training and testing. Now total twelve feature points ($v_1, ..., v_6$ and $h_1, ..., h_6$) are calculated by vertical and horizontal splittings. In Section 4 we will see how each feature point can classify.

3 Classification

In this paper features are based on geometric properties. So we used euclidean distance model for classification. This is the simple distance between a pair of vectors of size n. Here vectors are nothing but feature points, so the size of vector is 2. How to calculate distance using eucliden distance model is descibed in Section 3.1. In threshold calculation these distances are useful.

3.1 Euclidean distance model

Let $A(a_1, a_2, ..., a_n)$ and $B(b_1, b_2, ..., b_n)$ are two vectors of size n. We can calculate $distance(d)$ by using equarion 1.

$$distance(d) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{1}$$

In our application, vectors are points on plane. So d is the simple distance between two points.

4 Threshold

Individual thresholds for vertical splitting and horizontal splitting. Here we proposed one method for threshold selection which used in Section 5.1. Fig. 5 shows the variations in single curresponding feature points of training signatures. Let n is the number of training signatures and $x_1, x_2, ..., x_n$ are curresponding single feature points of training signatures(taking one curresponding feature point from each signature). x_{median} is the median of n features from n signatures. Let $d_1, ..., d_n$ are distances defined here,

$$
\begin{aligned}
d_1 &= distance(x_{median}, x_1) \\
d_2 &= distance(x_{median}, x_2) \\
&\vdots \\
d_n &= distance(x_{median}, x_n)
\end{aligned} \tag{2}
$$

Two main parameters we used in threshold calculation are d_{avg} and σ. Equations 3 and 4 shows the calculation

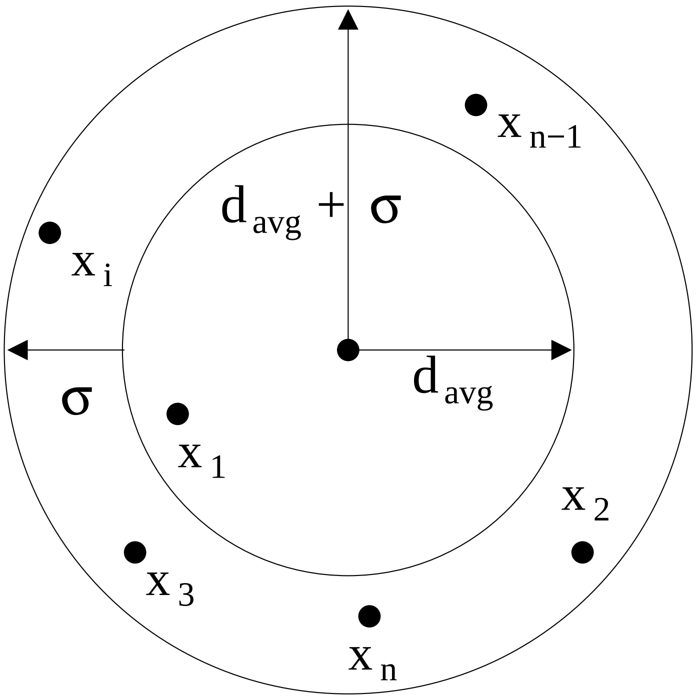


Figure 5: d_{avg} (average distance) and σ (standard deviation) derivation from distances

of these two parameters.

$$d_{avg} = average(d_1, d_2, ..., d_n) \tag{3}$$

$$\sigma = SD(d_1, d_2, ..., d_n) \tag{4}$$

Like this total six different feature points are there for both vertical and horizontal splitting based on average distance (d_{avg}) and standard deviation (σ). Equation 5 shows the main formula for threshold.

$$threshold(t) = \sqrt{\sum_{i=1}^{6} (d_{avg,i} + \sigma_i)^2} \tag{5}$$

5 Experiments & Results

For experiment we took 30 original signatures from each person and selected 9 for training. These original signatures are taken in different days. Forgeries taken by three persons and 10 from each. Total 21 originals and 30 forgeries for each person signature are going to be tested. There are two thresholds (one based on vertical splitting and another based on horizontal splitting) for each person signature.

5.1 Training

Let n signatures are taking for training from each person. There are 12 feature points from each original signature, 6 are taken by vertical splitting (Section2.2) and 6 are taken by horizontal splitting (Section2.3). Individual thresholds and patterns are calculating for vertical splitting and horizontal splitting. Pattern points based on vertical splitting are shown below.

$$\begin{aligned}
v_{pattern,1} &= median(v_{1,1}, v_{2,1}, \ldots, v_{n,1}) \\
v_{pattern,2} &= median(v_{1,2}, v_{2,2}, \ldots, v_{n,2}) \\
v_{pattern,3} &= median(v_{1,3}, v_{2,3}, \ldots, v_{n,3}) \\
v_{pattern,4} &= median(v_{1,4}, v_{2,4}, \ldots, v_{n,4}) \\
v_{pattern,5} &= median(v_{1,5}, v_{2,5}, \ldots, v_{n,5}) \\
v_{pattern,6} &= median(v_{1,6}, v_{2,6}, \ldots, v_{n,6})
\end{aligned} \tag{6}$$

Where $v_{i,1}, v_{i,2}, \ldots, v_{i,6}$ are vertical splitting features of i^{th} training signature sample. Threshold based on vertical splitting is shown below.

$$v_{threshold} = \sqrt{\sum_{i=1}^{6} (vd_{avg,i} + \sigma_{v,i})^2} \tag{7}$$

In equation 9 $vd_{avg,i}$ is same as average distance and $\sigma_{v,i}$ is same as standard deviation shown in Section 4. Pattern points based on horizontal splitting are shown below.

$$\begin{aligned}
h_{pattern,1} &= median(h_{1,1}, h_{2,1}, \ldots, h_{n,1}) \\
h_{pattern,2} &= median(h_{1,2}, h_{2,2}, \ldots, h_{n,2}) \\
h_{pattern,3} &= median(h_{1,3}, h_{2,3}, \ldots, h_{n,3}) \\
h_{pattern,4} &= median(h_{1,4}, h_{2,4}, \ldots, h_{n,4}) \\
h_{pattern,5} &= median(h_{1,5}, h_{2,5}, \ldots, h_{n,5}) \\
h_{pattern,6} &= median(h_{1,6}, h_{2,6}, \ldots, h_{n,6})
\end{aligned} \tag{8}$$

Where $h_{i,1}, h_{i,2}, \ldots, h_{i,6}$ are horizontal splitting features of i^{th} training signature sample. Threshold based on horizontal splitting is shown below.

$$h_{threshold} = \sqrt{\sum_{i=1}^{6} (hd_{avg,i} + \sigma_{h,i})^2} \tag{9}$$

We will store pattern points and thresholds of both horizontal splitting and vertical splitting. These values are useful in testing.

5.2 Testing

When new signature comes for testing we have to calculate features of vertical splitting and horizontal splitting. Feature points based vertical splitting are $v_{new,1}$, $v_{new,2}$, $v_{new,3}$, $v_{new,4}$, $v_{new,5}$, $v_{new,6}$. Distances between new signature features and pattern feature points based on vertical splitting are shown below.

$$\begin{aligned} vd_{new,1} &= distance(v_{pattern,1}, v_{new,1}) \\ vd_{new,2} &= distance(v_{pattern,2}, v_{new,2}) \\ vd_{new,3} &= distance(v_{pattern,3}, v_{new,3}) \\ vd_{new,4} &= distance(v_{pattern,4}, v_{new,4}) \\ vd_{new,5} &= distance(v_{pattern,5}, v_{new,5}) \\ vd_{new,6} &= distance(v_{pattern,6}, v_{new,6}) \end{aligned} \tag{10}$$

For classification of new signature we have to calculate $v_{distance}$ and compare this with $v_{threshold}$. If $v_{distance}$ is less than or equal to $v_{threshold}$ then new signature is acceptable by vertical splitting.

$$v_{distance} = \sqrt{\sum_{i=1}^{6} vd_{new,i}^2} \tag{11}$$

Feature points based vertical splitting are $h_{new,1}$, $h_{new,2}$, $h_{new,3}$, $h_{new,4}$, $h_{new,5}$, $h_{new,6}$. Distances between new signature features and pattern feature points based on vertical splitting are shown below.

$$\begin{aligned} hd_{new,1} &= distance(h_{pattern,1}, h_{new,1}) \\ hd_{new,2} &= distance(h_{pattern,2}, h_{new,2}) \\ hd_{new,3} &= distance(h_{pattern,3}, h_{new,3}) \\ hd_{new,4} &= distance(h_{pattern,4}, h_{new,4}) \\ hd_{new,5} &= distance(h_{pattern,5}, h_{new,5}) \\ hd_{new,6} &= distance(h_{pattern,6}, h_{new,6}) \end{aligned} \tag{12}$$

For classification of new signature we have to calculate $h_{distance}$ and compare this with $h_{threshold}$. If $h_{distance}$ is less than or equal to $h_{threshold}$ then new signature is acceptable by horizontal splitting.

$$h_{distance} = \sqrt{\sum_{i=1}^{6} hd_{new,i}^2} \tag{13}$$

New signature features have to satisfy both vertical splitting and horizontal splitting thresholds.

5.3 Results

False Acceptance Rate (FAR) and *False Rejection Rate (FRR)* are the two parameters using for measuring performance of any signature verification method. FAR is calculated by equation 14 and FRR is calculated by equation 15.

$$FAR = \frac{number\ of\ forgeries\ accepted}{number\ of\ forgeries\ tested} \times 100 \tag{14}$$

$$FRR = \frac{number\ of\ originals\ rejected}{number\ of\ originals\ tested} \times 100 \tag{15}$$

Table 1 shows the *False Acceptance Rate* of our method for different types of forgeries. Table 2 shows the *False Rejection Rate* for original sigature.

Table 1: False Acceptance Rate (FAR)

Forgery Type	FAR(%)
Random Forgeries	2.08
Simple Forgeries	9.75
Skilled Forgeries	16.36

In general there are different thresholds for different types of forgery detections. But here threshold is same for random, simple and skilled forgeries. Because this method is mainly eliminating random and simple forgeries.

Table 2: False Rejection Rate (FRR)

Signature	FRR(%)
Original Signatures	14.58

6 Conclusion

This method performs much better than any other off-line signature verification methods. Future direction in this is classifying the skilled forgeries correctly. For this we have to approach novel classification method.

Figure 6: Feature points based on vertical splitting of depth 2

Figure 7: Feature points based on horizontal splitting of depth 2

For better classification we can again split the sub-parts of Fig.3 using vertical splitting and Fig.4 using horizontal splitting. Then instead of six featire points we can get 24 feature points for each vertical and horizontal splittings. Fig.6 shows the vertical splitting of depth 2. Fig.7 shows the horizontal splitting of depth 2.

References

[1] J.J. Brault and R. Plamondon, "Segmanting Handwritten Signatures at Their Perceptually Important Points", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.15, no. 9, pp.953-957, Sept.1993.

[2] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "A comparison of SVM and HMM classifiers in the off-line signature verification", *Pattern Recognition Letters* 26, 1377-1385, 2005.

[3] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "An off-line signature verification using HMM for Random, Simple and Skilled Forgeries", *Sixth International Conference on Document Analysis and Recognition*, pp. 1031-1034, Sept.2001.

[4] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "The Interpersonal and Intrapersonal Variability Influences on Off-line Signature Verification Using HMM", *Proc. XV brazilian Symp. Computer Graphics and Image Processing*, 2002, pp. 197-202 Oct.2002.

[5] J Edson, R. Justino, A. El Yacoubi, F. Bortolozzi and R. Sabourin, "An off-line Signature Verification System Using HMM and Graphometric features", *DAS 2000*, pp. 211-222, Dec.2000.

[6] B. Fang, C.H. Leung, Y.Y. Tang, K.W. Tse, P.C.K. Kwok and Y.K. Wong, "Off-line signature verification by the tracking of feature and stroke positions", *Patter Recognition* 36, 2003, pp. 91–101.

[7] Migual A. Ferrer, Jesus B. Alonso and Carlos M. Travieso, "Off-line Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic", *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol.27, no.6, June 2005.

[8] R. Plamondon and S.N. Srihari, "Online and Offline Handwriting Recognition: A Comprehensive Survey", *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol.22 no.1, pp.63-84, Jan.2000.

[9] A. Zimmer and L.L. Ling, "A Hybrid On/Off Line Handwritten Signature Verification System", *Seventh International Conference on Document Analysis and Recognition*, vol.1, pp.424-428, Aug.2003.

Banshider Majhi, Y Santhosh Reddy, D Prasanna Babu
Department of CSEA
NIT Rourkela
India 769008
E-mail: ysantosh@rediffmail.com

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 25-32

One More Universality Result for P Systems with Objects on Membranes

Gheorghe Păun

Abstract: We continue here the attempt to bridge brane calculi with membrane computing, following the investigation started in [2]. Specifically, we consider P systems with objects placed *on* membranes, and processed by membrane operations. The operations used in this paper are membrane *creation* (*cre*), and membrane *dissolution* (*dis*), defined in a way which reminds the operations *pino, exo* from a brane calculus from [1]. For P systems based on these operations we prove the universality, for one of the two possible variants of the operations; for the other variant the problem remains open.
Keywords: Membrane computing, Brane calculi, Matrix grammar, Universality

1 Introduction

This paper is a direct continuation of [2], where a first step was made to bridge membrane computing [4], [5], [6] and brane calculi [1]. The main point of this effort is to define P systems which work with multisets of objects placed *on* the membranes rather than inside the compartments defined by membranes, and to process these multisets by means of operations with membranes rather than by multiset rewriting rules acting only on objects. The operations *pino, exo, mate, drip* were formalized in [2] as membrane computing rules, and used in defining P systems based on them. The universality of *mate, drip* operations was proved in [2] (for systems using simultaneously at any step of a computation at most eleven membranes). We give here an universality result for other two operations, membrane creation (*cre*), and membrane dissolution (*dis*), which have the same syntax as *pino, exo* operations, but a different interpretation in what concerns the contents of the handled membranes – details can be found in Section 3 below. Actually, as it was the case in [2] with *pino, exo*, we have two variants of each of the operations *cre, dis*. For one of these variants, we prove the Turing completeness, while the case of the other variant remains open (we believe that a similar result holds true).

2 Prerequisites

All notions of formal language theory we use are elementary and standard, and can be found in any basic monograph of formal language theory. For the sake of completeness, we introduce below only the notion of matrix grammars with appearance checking – after specifying that by *RE* we denote the family of recursively enumerable languages, and by *PsRE* the family of Parikh images of languages from *RE* (the Parikh mapping associated with an alphabet V is denoted by Ψ_V).

A matrix grammars with appearance checking [3] is a construct $G = (N, T, S, M, F)$, where N, T are disjoint alphabets (of non-terminals and terminals, respectively), $S \in N$ (axiom), M is a finite set of *matrices*, that is sequences of the form $(A_1 \to x_1, \ldots, A_n \to x_n)$, $n \geq 1$, of context-free rules over $N \cup T$, and F is a set of occurrences of rules in the matrices of M.

For $w, z \in (N \cup T)^*$ we write $w \Longrightarrow z$ if there is a matrix $(A_1 \to x_1, \ldots, A_n \to x_n)$ in M and the strings $w_i \in (N \cup T)^*, 1 \leq i \leq n+1$, such that $w = w_1, z = w_{n+1}$, and, for all $1 \leq i \leq n$, either (1) $w_i = w_i' A_i w_i'', w_{i+1} = w_i' x_i w_i''$, for some $w_i', w_i'' \in (N \cup T)^*$, or (2) $w_i = w_{i+1}$, A_i does not appear in w_i, and the rule $A_i \to x_i$ appears in F. (If applicable, the rules from F should be applied, but if they cannot be applied, then we may skip them. That is why the rules from F are said to be applied in the *appearance checking* mode.) If $F = \emptyset$, then the grammar is said to be without appearance checking.

The language generated by G is defined by $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$, where $\Longrightarrow^*$ is the reflexive and transitive closure of the relation $\Longrightarrow$.

The family of languages of this form is denoted by MAT_{ac}; it is known that $MAT_{ac} = RE$.

We say that a matrix grammar with appearance checking $G = (N, T, S, M, F)$ is in the *Z-binary normal form* if $N = N_1 \cup N_2 \cup \{S, Z, \#\}$, with these three sets mutually disjoint, and the matrices in M are in one of the following forms:

1. $(S \to XA)$, with $X \in N_1, A \in N_2$,

2. $(X \to Y, A \to w)$, with $X, Y \in N_1, A \in N_2, w \in (N_2 \cup T)^*, |w| \leq 2$,

3. $(X \to Y, A \to \#)$, with $X \in N_1, Y \in N_1 \cup \{Z\}, A \in N_2$,

4. $(Z \to \lambda)$.

Moreover, there is only one matrix of type 1, F consists exactly of all rules $A \to \#$ appearing in matrices of type 3, and, if a sentential form generated by G contains the symbol Z, then it is of the form Zw, for some $w \in (T \cup \{\#\})^*$ (that is, the appearance of Z makes sure that, except for Z, all symbols are either terminal or the trap-symbol #). The matrix of type 4 is used only once, in the last step of a derivation.

For each language $L \in RE$ there is a matrix grammar with appearance checking G in the Z-binary normal form such that $L = L(G)$.

As usual, we represent multisets over an alphabet V by strings over V, with the obvious observation that all permutations of a string represent the same multiset.

3 P Systems Using the Cre/Dis Operations

We start by recalling from [2] the formalization of the operations *pino, exo* in terms of membrane computing.

A membrane is represented, as usual, by a pair of square brackets, $[\ ]$, but we associate here with membranes *multisets of object* (corresponding to the proteins embedded in the real membranes). A membrane having associated a multiset u (represented by a string) is written in the form $[\ ]_u$; we also use to say that the membrane is *marked* with the multiset u.

The following four operations were defined in [2]:

$$\begin{aligned}
pino_i &: [\ ]_{uav} \to [[\ ]_{ux}]_v, && (1)\\
exo_i &: [[\ ]_{ua}]_v \to [\ ]_{uxv}, && (2)\\
pino_e &: [\ ]_{uav} \to [[\ ]_v]_{ux}, && (3)\\
exo_e &: [[\ ]_u]_{av} \to [\ ]_{uxv}. && (4)
\end{aligned}$$

in all cases with $a \in V$, $u, x \in V^*$, $v \in V^+$, with $ux \in V^+$ for *pino* rules, where V is a given alphabet of objects.

In each case, multisets of proteins are transferred from input membranes to output membranes as indicated in the rules, with protein a evolved into the multisets x (which can be empty). The subscripts i and e stand for "internal" and "external", respectively, pointing to the "main" membrane of the operation in each case.

It is important to note that the multisets u, v and the protein a marking the left hand membranes of these rules correspond to the multisets u, v, x from the right hand side of the rules; specifically, the multiset uxv resulting when applying the rule is precisely split into ux and v, with these two multisets assigned to the two new membranes.

The rules are applied as follows. Assume that we have a membrane $[\ ]_{zuav}$, for $a \in V, u, v, z \in V^*$. By a $pino_i$ rule as in (1), we obtain any one of the pairs of membranes $[\ [\ ]_{z_1ux}\ ]_{z_2v}$ such that $z = z_1z_2,\ z_1, z_2 \in V^*$, and by a $pino_e$ rule as in (3), we obtain any one of the pairs of membranes $[\ [\ ]_{z_1v}\ ]_{z_2ux}$ such that $z = z_1z_2,\ z_1, z_2 \in V^*$.

In the case of the two *exo* operations, the result is uniquely determined. From a pair of membranes $[\ [\ ]_{z_1ua}]_{z_2v}$, by an exo_i rule as in (2) we obtain the membrane $[\ ]_{z_1z_2uxv}$, and from $[\ [\ ]_{z_1u}]_{z_2av}$, by an exo_e rule as in (4) we obtain the same membrane $[\ ]_{z_1z_2uxv}$.

The contents of membranes involved in these operations is transferred from the input membranes to the output membranes in the same way as in brane calculi (**P,Q** represent here the possible contents of the respective membranes):

$$\begin{aligned}
pino_i &: [\ \mathbf{P}\,]_{uav} \to [[\ ]_{ux}\,\mathbf{P}\,]_v,\\
exo_i &: [[\,\mathbf{P}\,]_{ua}\,\mathbf{Q}\,]_v \to \mathbf{P}\,[\ \mathbf{Q}\,]_{uxv},\\
pino_e &: [\ \mathbf{P}\,]_{uav} \to [[\ ]_v\,\mathbf{P}\,]_{ux},\\
exo_e &: [[\,\mathbf{P}\,]_u\,\mathbf{Q}\,]_{av} \to \mathbf{P}\,[\ \mathbf{Q}\,]_{uxv}.
\end{aligned}$$

Here we change the interpretation of these rules, as suggested below (because the new semantics do not correspond to the operations *pino, exo*, we change the name of operations to *cre, dis*, for "membrane creation" and “membrane dissolution"):

$$\begin{array}{rcl} cre_i & : & [\ \mathbf{P}\ ]_{uav} \rightarrow [\,[\ \mathbf{P}\ ]_{ux}\ ]_v, \\ dis_i & : & [\,[\ \mathbf{P}\ ]_{ua}\ \mathbf{Q}\ ]_v \rightarrow [\ \mathbf{P}\ \mathbf{Q}\ ]_{uxv}, \\ cre_e & : & [\ \mathbf{P}\ ]_{uav} \rightarrow [\,[\ \mathbf{P}\ ]_v\ ]_{ux}, \\ dis_e & : & [\,[\ \mathbf{P}\ ]_u\ \mathbf{Q}\ ]_{av} \rightarrow [\ \mathbf{P}\ \mathbf{Q}\ ]_{uxv}. \end{array}$$

That is, when a membrane is created inside an existing membrane, the new membrane contains all previously existing membranes, and while dissolving a membrane, its contents remains inside the membrane where it was placed before the operation. The interpretation of the latter operation is rather similar to the usual dissolution operation in membrane computing, while the membrane creation is understood as doubling the existing membrane, with a distribution of the multiset marking the initial membrane to the two new membranes.

Using rules as defined above, we can define a P system as

$$\Pi = (A, \mu, u_1, \ldots, u_m, R),$$

where:

1. A is an alphabet (finite, non-empty) of objects;
2. μ is a membrane structure with $m \geq 2$ membranes;
3. $u_1, \ldots, u_m$ are multisets of objects (represented by strings over A) bound to the m membranes of μ at the beginning of the computation; the skin membrane is marked with $u_1 = \lambda$;
4. R is a finite set of *cre, dis* rules, of the forms specified above, with the objects from the set A.

For a rule of any type, with u, a, v as above, $|uav|$ is called the *weight* of the rule.

In what follows, the skin membrane plays no role in the computation, *no rule can be applied to it*. Also, we stress the fact that there is no object in the compartments of μ; a membrane can contain other membranes inside, but in-between membranes there is nothing.

When using any rule of any type, we say that the membranes from its left hand side are *involved* in the rule; they all are “consumed", and the membranes from the right hand side of the rule are produced instead. Similarly, the object a specified in the left hand side of rules is “consumed", and it is replaced by the multiset x.

The evolution of the system is defined in the standard way used in membrane computing, with the rules applied in the non-deterministic maximally parallel manner, with each membrane involved in at most one rule. Thus, the parallelism is maximal at the level of membranes – each membrane which can evolve has to do it – but each multiset of objects evolves in a sequential manner, as only one rule can act on any multiset in a transition step. More precise details can be found in [2]. A computation which starts from the initial configuration is *successful* if (i) it halts, that is, it reaches a configuration where no rule can be applied, and (ii) in the halting configuration there are only two membranes, the skin (marked with λ) and an inner one. The *result* of a successful computation is the vector of multiplicities of objects which mark the inner membrane in the halting configuration. The set of all vectors computed in this way by Π is denoted by $Ps(\Pi)$.

The family of all sets of vectors $Ps(\Pi)$ computed by P systems Π using at any moment during a computation at most m membranes, and cre_i, dis_i rules of weight at most p, q, respectively, is denoted by $PsOP_m(cre_p, dis_q)$. When one of the parameters m, p, q is not bounded we replace it with $*$.

We end this section by pointing out some relations which follow directly from the definitions (and from Turing-Church thesis).

Lemma 1. (i) $PsOP_m(cre_p, dis_q) \subseteq PsOP_{m'}(cre_{p'}, dis_{q'})$, *for all* $m \leq m', p \leq p', q \leq q'$.
(ii) $PsOP_*(cre_*, dis_*) \subseteq PsRE$.

We also recall the main result from [2]: $PsOP_{11}(mate_5, drip_5) = PsRE$ (the notation is self-explanatory).

4 Universality for the Cre/Dis Operations

In the case of *cre, dis* operations as defined above, we cannot generate vectors of norm 0 or 1: in each rule $[\ ]_{uav} \to [[\ ]_{ux}]_v$, $[[\ ]_{ua}]_v \to [\ ]_{uxv}$ (necessary in the last step of any computation in order to get only one internal membrane) we have imposed to have $|uxv| \geq 2$. That is why the universality below is obtained modulo vectors of the form $(0,\ldots,0)$ and $(0,\ldots,0,1,0,\ldots,0)$. We denote by $Ps'RE$ and $Ps'OP_m(cre_p,dis_q)$ the sets of vectors from $PsRE$ and $PsOP_m(cre_p,dis_q)$ having the sum of elements greater than or equal to 2.

Theorem 2. *$Ps'RE = Ps'OP_m(cre_p,dis_q)$ for all $m \geq 7, p \geq 4$, and $q \geq 4$.*

Proof. Let us consider a language $L \in RE = MAT_{ac}$, $L \subseteq V^2V^*$, for an alphabet V with n symbols. We write this language in the form

$$L = \bigcup_{a,b \in V} \{ab\}\partial^l_{ab}(L).$$

Let $G_{ab} = (N_{ab}, V, S_{ab}, M_{ab}, F_{ab})$ be a matrix grammar with appearance checking such that $L(G_{ab}) = \partial^l_{ab}(L)$, for $a,b \in V$. We consider these grammars G_{ab} in the Z-normal form, with the notations from Section 2 (hence $N_{ab} = N_{ab,1} \cup N_{ab,2} \cup \{S_{ab}, Z_{ab}, \#\}$), and we construct the matrix grammar $G = (N,V,S,M,F)$ with

$$\begin{aligned}
N &= N_1 \cup N_2 \cup \{Z_{ab} \mid a,b \in V\} \cup \{S,\#\},\\
N_1 &= \bigcup_{a,b \in V} N_{ab,1},\\
N_2 &= \bigcup_{a,b \in V} N_{ab,2},\\
M &= \{(S \to XA) \mid \text{for } (S_{ab} \to XA) \in M_{ab},\ a,b \in V\}\\
&\cup\ \{(X \to Y, A \to w) \mid \text{for } (X \to Y, A \to w) \in M_{ab}, a,b \in V\}\\
&\cup\ \{(Z_{ab} \to ab) \mid \text{for } (Z \to \lambda) \in M_{ab}, a,b \in V\}.
\end{aligned}$$

Obviously, $L(G) = L$.

We assume that all two-rules matrices from M are injectively labeled, in the form $m_l : (X \to Y, A \to x)$, $l \in Lab$, for a set of labels Lab.

Starting from the grammar G we now construct a P system

$$\Pi = (A, [\ [\ ]\ ], \lambda, S_1S_2, R),$$

with the alphabet

$$\begin{aligned}
A &= \{Y, Y', Y'', Y''', Y^{iv}, Y^{v}, Y^{vi}, Y^{vii}, Y^{viii}, Y^{ix}, Y^{x} \mid Y \in N_1\}\\
&\cup\ \{\alpha, \alpha', \alpha'' \mid \alpha \in N_2 \cup V\}\\
&\cup\ \{\bar{A} \mid A \in N_2\}\\
&\cup\ \{Z_{ab}, Z'_{ab}, Z''_{ab}, Z'''_{ab} \mid a,b \in V\}\\
&\cup\ \{E, H, H', S_1, S_2, S_3, c_1, \ldots, c_{11}, c_0, c'_0, c''_0, c'_3, c''_3, d_1, d_2, d'_1, d'_2, f', f'', \#\},
\end{aligned}$$

and the rules from the set R as constructed below.

Any computation starts from the configuration $[\ [\ ]_{S_1S_2}]_\lambda$, by using the following rules:

$$\begin{aligned}
&\text{Step 1}: && [\ ]_{S_1S_2} \to [\ [\ ]_X\ ]_{S_2},\\
&\text{Step 2}: && [\ [\ ]_X\ ]_{S_2} \to [\ ]_{Xc_0d_1S_2},\\
&\text{Step 3}: && [\ ]_{XS_2c_0d_1} \to [\ [\ ]_{XS_3}\ ]_{c_0d_1},\\
&\text{Step 4}: && [\ ]_{S_3X} \to [\ [\ ]_{E\bar{A}}\ ]_X,\ [\ ]_{c_0d_1} \to [\ [\ ]_{c'_0}\ ]_{d_1},\\
&\text{Step 5}: && [\ [\ ]_{E\bar{A}}\ ]_X \to [\ ]_{E\bar{A}X},\ [\ [\ ]_{c'_0}\ ]_{d_1} \to [\ ]_{c''_0d_1},\\
&\text{Step 6}: && [\ ]_{X\bar{A}E} \to [\ [\ ]_{XA}\ ]_E,\ [\ ]_{c''_0d_1} \to [\ [\ ]_{c_1}\ ]_{d_1},
\end{aligned}$$

for each matrix $(S_{ab} \to XA) \in M_{ab}$, for $a,b \in V$.

The rules are used as indicated in the table above, with two rules simultaneously applied in steps 4, 5, 6. The only possible branching is in step 3, when instead of the rule $[\ ]_{XS_2c_0d_1} \to [\ [\ ]_{XS_3}\ ]_{c_0d_1}$, we can also use the rule $[\ ]_{c_0d_1} \to [\ [\ ]_{c_0'}\ ]_{d_1}$. In this way we obtain the membranes $[\ [\ ]_{c_0'}\ ]_{d_1}$, with XS_2 distributed among them. Because S_3 will be never introduced, we continue only with rules which process membranes marked with c_i and d_1, namely, the rules from the third column of Table 1; in this way, the computation will never stop, both because we can return again and again to a pair of membranes of the form $[\ [\ ]_{c_1}\ ]_{d_1}$, and because pairs of membranes marked with c_3' will appear and introduce trap objects/membranes – see also below.

The evolution of the membrane structure is indicated in Figure 1.

Initial	$[\ [\ ]_{S_1S_2}\ ]_\lambda$
Step 1	$[\ [\ [\ ]_X\ ]_{S_2}\ ]_\lambda$
Step 2	$[\ [\ ]_{Xc_0d_1S_2}\ ]_\lambda$
Step 3	$[\ [\ [\ ]_{XS_3}\ ]_{c_0d_1}\ ]_\lambda$
Step 4	$[\ [\ [\ [\ [\ ]_{E\bar{A}}\ ]_X\ ]_{c_0'}\ ]_{d_1}\ ]_\lambda$
Step 5	$[\ [\ [\ ]_{E\bar{A}X}\ ]_{c_0''d_1}\ ]_\lambda$
Step 6	$[\ [\ [\ [\ [\ ]_{XA}\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$

Figure 1: The evolution of membranes at the beginning of computations.

Thus, we end with a configuration of the form $[\ [\ [\ [\ [\ ]_{XA}\ ]_E\ ]_{c_1}\ ]_{d_1}]_\lambda$.

The rules for simulating the two-rules matrices from M are indicated in Table 1; by w' we denote here the string obtained from w by priming one symbol; if $w = \lambda$, then $w' = f'$, hence $\alpha' = f', \alpha'' = f''$ and, in row 6, $\alpha = \lambda$.

Step	$m_l : (X \to Y, A \to w)$	$m_l : (X \to Y, B \to \#)$	
1	$[\ [\ ]_X\ ]_E \to [\ ]_{X_lE}$	$[\ [\ ]_X\ ]_E \to [\ ]_{X_lE}$	$[\ [\ ]_{c_1}\ ]_{d_1} \to [\ ]_{c_2c_3d_1}$
2	$[\ ]_{AEX_l} \to [\ [\ ]_{w'}\ ]_{EX_l}$	$[\ ]_{X_lBE} \to [[\ ]_{X_l\#\#}\ ]_E$	$[\ ]_{c_3c_2d_1} \to [[\ ]_{c_3'}]_{c_2d_1}$
3	$[\ [\ ]_{EX_l}\ ]_{c_3'} \to [\ ]_{EY'c_3'}$	$[\ [\ ]_{X_lE}\ ]_{c_3'} \to [\ ]_{Y^{vi}HEc_3'}$	$[\ ]_{c_2d_1} \to [\ [\ ]_{c_4}\ ]_{d_1}$
4	$[\ ]_{c_3'Y'E} \to [\ [\ ]_{c_3'Y''}\ ]_E$	$[\ ]_{Y^{vi}c_3'EH} \to [[\ ]_{c_3'Y^{vii}}\ ]_{EH}$	$[\ [\ ]_{c_4}\ ]_{d_1} \to [\ ]_{c_5d_1}$
5	$[[\ ]_{\alpha'}\ ]_{c_3'Y''} \to [\ ]_{\alpha''c_3'Y''}$	$[\ ]_{Y^{vii}c_3'} \to [\ [\ ]_{Y^{viii}}\ ]_{c_3'}$ $[\ ]_{HE} \to [\ [\ ]_{H'}\ ]_E$	$[\ ]_{c_5d_1} \to [\ [\ ]_{c_6}\ ]_{d_1}$
6	$[[\ ]_{\alpha''c_3'Y''}]_E \to [\ ]_{\alpha c_3'Y''E}$	$[[\ ]_{c_3'}\ ]_{H'} \to [\ ]_{c_3''H'}$	$[\ [\ ]_{c_6}\ ]_{d_1} \to [\ ]_{c_7d_1}$
7	$[\ ]_{c_3'Y''E} \to [\ [\ ]_{c_3'Y'''}\ ]_E$	$[[\ ]_{Y^{viii}}\ ]_{c_3''H'} \to [\ ]_{Y^{ix}c_3''H'}$	$[\ ]_{c_7d_1} \to [\ [\ ]_{c_8}\ ]_{d_1}$
8	$[\ [\ ]_{c_3'Y'''}\ ]_E \to [\ ]_{Y'''E}$	$[[\ ]_{Y^{ix}c_3''H'}]_E \to [\ ]_{Y^{ix}H'E}$	$[\ [\ ]_{c_8}\ ]_{d_1} \to [\ ]_{c_9d_1}$
9	$[\ ]_{Y'''E} \to [\ [\ ]_{Y^{iv}}\ ]_E$	$[\ ]_{Y^{ix}H'E} \to [\ ]_{Y^x}\ ]_E$	$[\ ]_{c_9d_1} \to [\ [\ ]_{c_{10}}\ ]_{d_1}$
10	$[\ [\ ]_{Y^{iv}}\ ]_E \to [\ ]_{Y^vE}$	$[\ [\ ]_{Y^x}\ ]_E \to [\ ]_{Y^xE}$	$[\ [\ ]_{c_{10}}\ ]_{d_1} \to [\ ]_{c_{11}d_1}$
11	$[\ ]_{Y^vE} \to [\ [\ ]_Y\ ]_E$	$[\ ]_{Y^xE} \to [\ [\ ]_Y\ ]_E$	$[\ ]_{c_{11}d_1} \to [\ [\ ]_{c_1}\ ]_{d_1}$

Table 1: Rules for simulating two-rules matrices.

We also consider the rules

$$[\ ]_{X_lE} \to [\ [\ ]_{\#\#}\ ]_E, \text{ for each matrix } m_l : (X \to Y, A \to w),$$
$$[\ [\ ]_{H'}\ ]_E \to [\ ]_{\#\#E},$$
$$[\ ]_{\#\#} \to [\ [\ ]_\#\ ]_\#,$$
$$[\ [\ ]_\#\ ]_\# \to [\ ]_{\#\#}.$$

The simulation of matrices in G is performed by modifying the marking of the central membranes, those emerging from the initial membranes with markings XA and E, with these operations being assisted by the two membranes with markings c_1 and d_1 and their successors, which are external to the central membranes where the sentential form of G is produced. Always during the computation, the membranes remain embedded one in another, in a linear manner, never having two membranes on the same level (here stands the essential difference between the interpretation of the *cre, dis* operations and the interpretation of the *pino, exo* operations from [1], [2]).

The evolution of the membranes and of their relevant markings can be followed in Figure 2. If in the second step the rule $[\ ]_{AEX_l} \to [\ [\ ]_{w'}\ ]_{EX_l}$ is not applicable (hence the matrix m_l cannot be applied), then the rule $[\ ]_{X_lE} \to [\ [\ ]_{\#\#}\ ]_E$ will be applied, introducing the trap-object #, and the computation will never halt.

Starting	$[\ [\ [\ [\ [\ ]_X\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$
Step 1	$[\ [\ [\ ]_{X_lEA}\ ]_{c_2c_3d_1}\ ]_\lambda$
Step 2	$[\ [\ [\ [\ [\ ]_{w'}\ ]_{EX_l}\ ]_{c_3'}\ ]_{c_2d_1}\ ]_\lambda$
Step 3	$[\ [\ [\ [\ [\ ]_{\alpha'}\ ]_{EY'c_3'}\ ]_{c_4}\ ]_{d_1}\ ]_\lambda$
Step 4	$[\ [\ [\ [\ [\ ]_{\alpha'}\ ]_{c_3'Y''}\ ]_E\ ]_{c_5d_1}\ ]_\lambda$
Step 5	$[\ [\ [\ [\ [\ ]_{\alpha''c_3'Y''}\ ]_E\ ]_{c_6}\ ]_{d_1}\ ]_\lambda$
Step 6	$[\ [\ [\ ]_{\alpha c_3'Y''E}\ ]_{c_7d_1}\ ]_\lambda$
Step 7	$[\ [\ [\ [\ [\ ]_{c_3'Y'''}\ ]_E\ ]_{c_8}\ ]_{d_1}\ ]_\lambda$
Step 8	$[\ [\ [\ ]_{Y'''E}\ ]_{c_9d_1}\ ]_\lambda$
Step 9	$[\ [\ [\ [\ [\ ]_{Y^{iv}}\ ]_E\ ]_{c_{10}}\ ]_{d_1}\ ]_\lambda$
Step 10	$[\ [\ [\ ]_{Y^vE}\ ]_{c_{11}d_1}\ ]_\lambda$
Step 11	$[\ [\ [\ [\ [\ ]_Y\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$

Figure 2: The evolution of membranes when simulating $m_l : (X \to Y, A \to w)$.

The evolution of membranes in the case of the simulation of a matrix $m_l : (X \to Y, B \to \#)$ can be followed in Figure 3. This time, if B is present, in step 2 we have to use the rule $[\ ]_{X_lBE} \to [\ [\ ]_{X_i\#\#}\ ]_E$, and the computation will never halt. If no copy of B is present, then the central membrane does not evolve, waiting for the membrane marked with c_3' to be produced; this membrane can be used in the next step for evolving the central membrane.

Starting	$[\ [\ [\ [\ [\ ]_X\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$
Step 1	$[\ [\ [\ ]_{X_lE}\ ]_{c_2c_3d_1}\ ]_\lambda$
Step 2	$[\ [\ [\ [\ ]_{X_lE}\ ]_{c_3'}\ ]_{c_2d_1}\ ]_\lambda$
Step 3	$[\ [\ [\ [\ ]_{Y^{vi}HEc_3'}\ ]_{c_4}\ ]_{d_1}\ ]_\lambda$
Step 4	$[\ [\ [\ [\ ]_{c_3'Y^{vii}}\ ]_{EH}\ ]_{c_5d_1}\ ]_\lambda$
Step 5	$[\ [\ [\ [\ [\ [\ [\ ]_{Y^{viii}}\ ]_{c_3'}\ ]_{H'}\ ]_E\ ]_{c_6}\ ]_{d_1}\ ]_\lambda$
Step 6	$[\ [\ [\ [\ [\ ]_{Y^{viii}}\ ]_{c_3''H'}\ ]_E\ ]_{c_7d_1}\ ]_\lambda$
Step 7	$[\ [\ [\ [\ [\ ]_{Y^{ix}c_3''H'}\ ]_E\ ]_{c_8}\ ]_{d_1}\ ]_\lambda$
Step 8	$[\ [\ [\ ]_{Y^{ix}H'E}\ ]_{c_9d_1}\ ]_\lambda$
Step 9	$[\ [\ [\ [\ [\ ]_{Y^x}\ ]_E\ ]_{c_{10}}\ ]_{d_1}\ ]_\lambda$
Step 10	$[\ [\ [\ ]_{Y^xE}\ ]_{c_{11}d_1}\ ]_\lambda$
Step 11	$[\ [\ [\ [\ [\ ]_Y\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$

Figure 3: The evolution of membranes when simulating $m_l : (X \to Y, B \to \#)$.

Another step when we can apply a rule different from that indicated in Table 1 is step 4, when we can also use the rule $[\ ]_{HE} \to [\ [\ ]_{H'}\ ]_E$. In this way, we pass to the configuration of membranes $[\ [\ [\ [\ ]_{H'w_1}\ ]_{Ew_2}\ ]_{c_5d_1}\ ]_\lambda$, where $w_1w_2 = Y^{vi}c_3'$. No rule can be applied to the two inner membranes other than $[\ [\ ]_{H'}\ ]_E \to [\ ]_{\#\#E}$, and again the computation will never stop.

Therefore, the simulation of matrices in G should be done as above, and in this way we return to a configuration as that we have started with, with four membranes marked with X, E, c_1, d_1, respectively (the central membranes also having on them the symbols of the current sentential form of G which is simulated in Π).

Note that the rules used for simulating a matrix $m_l : (X \to Y, A \to w)$ cannot be mixed with the rules used for simulating a matrix $m_{l'} : (X' \to Y', A' \to \#)$, because of the injective labeling of matrices from M and because of the priming of symbols from N_1.

The process can be iterated, hence at some moment we introduce the symbol Z_{ab} identified by the symbols from N_1 used. The respective configuration is of the form: $[\ [\ [\ [\ [\ ]_{Z_{ab}}\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$. The central membrane will “swallow" all other membranes, also removing all auxiliary objects. To this aim, we use the following rules:

Step 1	$[\ [\ ]_{Z_{ab}}\ ]_E \to [\ ]_{Z'_{ab}E}$,
Step 2	$[\ [\ ]_{Z'_{ab}E}\ ]_{c_2c_3} \to [\ ]_{Z'_{ab}bc_2c_3}$,
Step 3	$[\ ]_{Z'_{ab}c_2c_3d_1} \to [\ [\ ]_{Z'_{ab}}\ ]_{c_3d_1}$,
Step 4	$[\ [\ ]_{Z'_{ab}}\ ]_{c_3d_1} \to [\ ]_{Z''_{ab}c_3d_1}$,
Step 5	$[\ ]_{Z''_{ab}c_3d_1} \to [\ [\ ]_{Z''_{ab}}\ ]_{d_1}$,
Step 6	$[\ [\ ]_{Z''_{ab}}\ ]_{d_1} \to [\ ]_{Z'''_{ab}d_1}$,
Step 7	$[\ ]_{Z'''_{ab}d_1b} \to [\ [\ ]_{Z'''_{ab}}\ ]_b$,
Step 8	$[\ [\ ]_{Z'''_{ab}}\ ]_b \to [\ ]_{ab}$,

for all $a,b \in V$. Furthermore, we consider the rules

$$[\ ]_{Z'_{ab}E} \to [\ [\ ]_{\#\#}\ ]_E,$$
$$[\ [\ ]_{c'_3}\ ]_{c'_3} \to [\ ]_{\#\#c'_3},$$
$$[\ ]_{\#a} \to [\ [\ ]_{\#\#}\ ]_a, \text{ for all } a \in V.$$

The first of these rules is used in step 2 if the rule $[\ [\ ]_{Z'_{ab}E}\ ]_{c_2c_3} \to [\ ]_{Z'_{ab}bc_2c_3}$ is not used – the objects $c_2c_3d_1$ might be used at that time by the rule $[\ ]_{c_3c_2d_1} \to [\ [\ ]_{c'_3}\ ]_{c_2d_1}$ from Table 1. Similarly, if this last rule is used in step 3 instead of the rule $[\ ]_{Z'_{ab}c_2c_3d_1} \to [\ [\ ]_{Z'_{ab}}\ ]_{c_3d_1}$, then a membrane marked with c'_3 is introduced, which will never be removed. In particular, after 11 steps, we introduce another membrane marked with c'_3, and then the rule $[\ [\ ]_{c'_3}\ ]_{c'_3} \to [\ ]_{\#\#c'_3}$ is used, preventing the termination of the computation. In conclusion, the evolution of the membranes in the final stage of the computation is as indicated in Figure 4.

Starting	$[\ [\ [\ [\ [\ ]_{Z_{ab}}\ ]_E\ ]_{c_1}\ ]_{d_1}\ ]_\lambda$
Step 1	$[\ [\ [\ ]_{Z'_{ab}E}\ ]_{c_2c_3d_1}\ ]_\lambda$
Step 2	$[\ [\ ]_{Z'_{ab}bc_2c_3d_1}\ ]_\lambda$
Step 3	$[\ [\ [\ ]_{Z'_{ab}}\ ]_{c_3d_1}\ ]_\lambda$
Step 4	$[\ [\ ]_{Z''_{ab}c_3d_1}\ ]_\lambda$
Step 5	$[\ [\ [\ ]_{Z''_{ab}}\ ]_{d_1}\ ]_\lambda$
Step 6	$[\ [\ ]_{Z'''_{ab}d_1}\ ]_\lambda$
Step 7	$[\ [\ [\ ]_{Z'''_{ab}}\ ]_b\ ]_\lambda$
Step 8	$[\ [\ ]_{ab}\ ]_\lambda$

Figure 4: The evolution of membranes in the end of computations.

The equality $\Psi_V(L(G)) = Ps(\Pi)$ follows from the previous explanations.

With the observation that the maximal number of membranes present in the system is seven, in step 5 from Figure 3 (during the simulation of matrices with a rule to be used in the appearance checking mode), and that the rules have the weight as specified in the theorem, we conclude the proof. □

5 Final Remarks

The case of using the operations cre_e, dis_e remains as a task for the reader, and the same with other operations from brane calculus – see also [2] for related problems. Improvements of the result in Theorem 2 are also plausible in what concerns the degree of context-sensitivity of the rules (and maybe also in what concerns the number of membranes). The same problems can be formulated for the result from [2].

As a general research topic, it remains to systematically investigate P systems with multisets of objects placed on membranes (maybe also in the compartments), processed by membrane handling operations like in brane calculi (maybe also by local multiset rewriting rules).

References

[1] L. Cardelli, Brane calculi. Interactions of biological membranes, *Proc. Computational Methods in Systems Biology, 2004*, Springer-Verlag, Berlin, to appear.

[2] L. Cardelli, Gh. Păun, An universality result for a (mem)brane calculus based on mate/drip operations, *Intern. J. Foundations of Computer Sci.*, 17, 1 (2006), 49–68.

[3] J. Dassow, Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, 1989.

[4] Gh. Păun, Computing with membranes, *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143 (and Turku Center for Computer Science–TUCS Report 208, November 1998, `www.tucs.fi`).

[5] Gh. Păun, *Membrane Computing. An Introduction*, Springer-Verlag, Berlin, 2002.

[6] The membrane computing web page: `http://psystems.disco.unimib.it`.

Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucureşti, Romania
and
Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: george.paun@imar.ro, gpaun@us.es

Editor's note about the author:

Gheorghe PĂUN (born on December 6, 1950) graduated the Faculty of Mathematics of the Bucharest University in 1974 and got his PhD at the same faculty in 1977. He has won many scholarships, in Germany, Finland, The Netherlands, Spain, etc. Presently he is a senior researcher at the Institute of Mathematics of the Romanian Academy, Bucharest, and a Ramon y Cajal research professor at Sevilla University, Spain. Since 1997 he is a Corresponding Member of the Romanian Academy, and since 2006 a member of Academia Europaea. His main research fields are formal language theory (regulated rewriting, contextual grammars, grammar systems), automata theory, combinatorics on words, computational linguistics, DNA computing, membrane computing (this last area was initiated by him in 1998). He has (co)authored and (co)edited more than fifty books in these areas, and he has (co)authored more than 400 research papers. In the last two decades he has visited many universities from Europe, USA, Canada, Japan, also participating to many international conferences, several times as an invited speaker. He is a member of the editorial board of numerous computer science journals and professional associations.

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 33-39

Symbolic Computations based on Grid Services

Dana Petcu, Cosmin Bonchiş, Cornel Izbaşa

Abstract: The widespread adoption of the current Grid technologies is still impeded by a number of problems, one of which is difficulty of developing and implementing Grid-enabled applications. In another dimension, symbolic computation, aiming to automatize the steps of mathematical problem solving, has become in the last years a basis for advanced applications in many areas of computer science.
In this context we have recently analyzed and developed grid-extensions of known tools for symbolic computations. We further present in this paper a case study of a Web service-based Grid application for symbolic computations.
Keywords: Grid computing, Web services, Mathematical software

1 Introduction

It is widely recognized that Grid computing is THE computer buzzword of the decade. Many of the greatest challenges for software systems lie in how to enable automated software components deployed by different organizations to dynamically discover one another, communicate and coordinate their actions and form sound, robust and effective compound applications and services. The Grid promises to solve these problems. The Grid underlying problems are multi-disciplinary and cover a wide range of issues - from service discovery, management, robust management of dependencies and system-system communication to security, legal or ethical frameworks, methodologies, verification, testing, and finally deployment of infrastructure in a shared and accessible environment. After the forerunner first-generation Grid systems, the second-generation proposed the vision of computing resources to be shared like content on the Web over the Internet. The associated layered architecture abstracts fundamental system components, their purpose, and their interaction with each other. Moreover the current, third-generation Grid is aligned with Web services. Comparing to the conventional distributed computing environments, the usual Grid environment focuses on the user: users working on their home machines see the illusion of a single computer, with access to all kinds of data and physical resources. Moreover, the specific machines that are used to execute an application are chosen from the user's point of view, maximizing the performance of that application, regardless of the effect on the system as a whole.

The two basic approaches to computational solution of mathematical problems are numerical and symbolic. For a long time, the numerical approach had an advantage of being capable of solving a substantially larger set of problems. Developments in symbolic computing are lagging relative to numerical computing, mainly due to the inadequacy of available computational resources: most importantly computer memory, but also processor power. Continuous growth in the capabilities of computer hardware led to an increasing interest in symbolic calculations. A transition from numerical modeling to analytical modeling can be observed today in various spheres of science and technology; a motivation consists in the desire to construct more accurate and faithful simulations. Currently software tools for symbolic computations, known as Computer Algebra Systems (CAS), allow users to study computational problems on the basis of their mathematical formulations and to focus on the problems themselves instead of spending time transforming the problems into forms that are numerically solvable.

There are three ways in which a CAS can interact with the Grid, that are detailed in the next section: the CAS uses a Grid service to improve its own services, the CAS uses the Grid infrastructure to improve its response time, or the CAS is presented to a client application as a Grid service. We have studied the first cases in [11] respectively [10]. In this paper we look to the third approach.

The paper is organized as follows. Section 2 presents a state-of-the-art in the field. Section 3 underlines the benefits of using Grid services for symbolic computations. In the context of the current Grid and CAS technologies, a case study is presented in Section 4. Further work directions are identified in Section 5.

2 The Grid and the CAS

2.1 Approaches for CAS-Grid interaction

A Grid-extension of a CAS system can tackle one or more of the following approaches:

Ability to accept services from the Grid: the CAS must be capable to augment its facilities with external modules, in particular it should be able to explore computational Grid facilities, to connect to a specific Grid service, to use it, and to translate its results for the CAS interface. This approach is taken into consideration by NetSolve/GridSolve [1], Geodise [4], MathGridLink [14], our Maple2g [9].

Ability to communicate and cooperate over the Grid: several kernels of CASs must be able cooperate within a Grid when solving problems; in order to have the same CAS on different computational nodes a Grid-version must be available; in the case of different CASs, appropriate interfaces between them must be developed and implemented or a common language for inter-communication must be adopted. This approach is taken into account by Maple2g [10].

Being a source of Grid or Web services: the CAS or some of its facilities must be reachable as grid or web services and allowed to be activated by remote users under appropriate security and licensing conditions; furthermore, deployment of the services must be done in an easy way from the inside of the CAS. This approach is taken into account by the several projects described bellow.

More details about the first two approaches can be found in [12]. Details about the third approach follows.

2.2 Web service-based CAS extensions

In number theory there exist a number of successful Internet projects [6] aiming, among others, at finding large prime numbers, factoring large numbers, computing digits of π, finding collisions on known encryption algorithms etc. A CAS web-wrapper component that can be used by multiple systems was reported in [15]. Another online system, OGB (for Gröbner basis computations), has been recently deployed [5].

MapleNet [7] offers a software platform to enhance mathematics and related courses over the Web. The client machine must be able to run Java applets. A publisher machine is responsible for creating and editing the content of the Web pages and, when complete, uploading them to the server. The server is the machine to which clients will connect to access Web pages and the applets associated with them. The server also respond to publishing requests from the publisher machine for the transfer of content between the publisher and the server. It manages concurrent Maple instances as required to serve client connections for mathematical computation and display services. The server can also provide some additional services including user authentication, logging information, and database access.

WebMathematica [16] offers access to Mathematica applications through a Web browser or other Web clients. Mathematica can be seen as a development environment for webMathematica sites. Standard Java technologies are used: Java Servlet and JavaServer Pages. WebMathematica allows a site to deliver HTML pages that are enhanced by the addition of Mathematica commands. When a request is made for one of these pages, the Mathematica commands are evaluated and the computed result is inserted into the page. Input can come from HTML forms, applets, JavaScript, and Web-enabled applications. It is also possible to send data files to a server for processing. Output can use different formats such as HTML, images, Mathematica notebooks, MathML, SVG, XML, PostScript, Pdf.

The Monet project [8], funded by the European Commission, was a two-year (April 2002-March 2004) investigation into mathematical Web services aiming to demonstrate the applicability of the Semantic Web to the world of mathematical software. The principal objective was to develop a framework for the description and provision of Web-based mathematical services. The key to such a framework is the ability to discover services dynamically based on published descriptions which describe both their mathematical and non-mathematical attributes. Such discovery and subsequent interaction are mediated by software agents capable of recognizing the criteria which should determine how particular kinds of problems are solved, and extracting them from the user's problem description. A symbolic solver wrapper tool architecture was designed to provide an environment that encapsulates CASs and expose their functionalities through symbolic services deployed. All symbolic services are running as independent Web services, each reachable at its own unique URL, all of them are enclosed within the symbolic server and they are managed by the wrapper tool symbolic solver environment. Each symbolic service is assigned to several instances, such as a service core Java class, a source code implementing the service with a mathematical solving software (a CAS), and a MSDL file. The principal information about each service is provided by the service configuration file that contains tree parts: service's MSDL, service interface to mathematical solving system and the actual service implementation. The technologies used for symbolic solver services implementation are Java, Axis, Tomcat, SOAP, WSDL, JSP, MSDL. Maple was chosen as an example of the solving engine for the first implementation and Axiom was used to validate the architecture and to demonstrate abilities to adopt different solving engine without performing major changes.

2.3 Grid service-based CAS extensions

There exist a number of grid-oriented projects that involve CASs.

The project Grid Enabled Numerical and Symbolic Services [3], GENSS (March 2004-February 2006), in the frame of UK e-Science programme, addressed the combination of Grid computing and mathematical Web services, their extension to deliver mathematical problem analysis, the code and the resources to compute the answers, using a common open agent-based framework. The main research focus lied on matchmaking techniques for advertisement and discovery of mathematical services. The project involved the design and implementation of an ontology for symbolic mathematical problems and used to support service specification and registration of services. The ontology has been extended based on work undertaken in Monet [8].

The Grid Execution Management for Legacy Code Architecture, GEMLCA [2] is a recent solution to deploy existing legacy code applications written in any programming language as a Grid service without modifying or even requiring access to the legacy code (source or binary). The access point for a client to the GEMLCA architecture is the front-end layer composed of a number of Grid services offering interfaces in order to deploy, query, submit, check the status of, and get the results back from computational jobs. The front-end layer is described in WSDL and can be used by any Grid service client to bind and make use of its functionality through SOAP. In order to access a legacy code program, the user executes the GEMLCA grid service client that creates a legacy code instance with the help of a legacy code factory. Following this, the system submits the job to the compute server through Globus Toolkit version 3 using a particular job manager. A specific XML format, LCID (Legacy Code Interface Description File) is necessary to be used.

3 Benefits of Using Symbolic-computing Services based on Globus-WSRF

The recent version 4 of Globus Toolkit, de-facto standard for Grid technologies, is based on standard Web Services technologies such as SOAP and WSDL. It is written according the WSRF specification (Web Services Resource Framework). The basic requirements addressed by WSRF is the ability to create, address, inspect, discover, and manage statefull resources. Grid services extends Web services (usually stateless services) by providing these extra functionalities.

WSRF approach is more flexible than the previous ones implemented in Globus Toolkit (e.g. OGSI implementation) allowing many-to-many mappings between Web services (the message processor) and any associated statefull resource (the statefull service instance).

3.1 Statefull services

The WSRF approach simplifies the development of Grid-service wrappers for CASs. The CAS can take now the role of the statefull service instance.

If we go back to the case of the Web-wrapper of a CAS, we can identify several problems solved by the Grid environment. Successive related requests to the service hiding the CAS will need the maintenance of the service instance in a command waiting cycle, without releasing the connections. When the connection is interrupted, but the client come back to the system it must start as a any new incomer.

If a statefull service is used, the latest state of the CAS can be registered. Despite the fact that the connection was closed, the client can come back and resume the computation at any time before the service instance destruction.

3.2 Service instances on remote Grid nodes

Using a appropriate scheduler the CAS service instance can run on a different Grid hardware resource than the one where the container for Web-based Grid services resides, primarily addressed by the service client. This approach solves the problem of the server overload of a classic client-server architecture.

3.3 Service discovery

The user is confronted with thousands of packages are available to perform all kinds of mathematical computations. A standard way to categorize, explore, discover, invoke and compose them is needed. Grid computing has awake high expectations for its potential as a discovery accelerator. Grid WSRF-based services are described in WSDL standard format that can be easily inspected by any potential client.

3.4 Security standards

Globus's GSI offers two message-level protection schemes, and one transport-level scheme [13]. GSI Secure Message scheme that provides message-level security can be used in the case on proprietary CAS usage.

Globus Toolkit implements three authentication methods: X.509 certificates, username-password, and anonymous authentication. The first two authentication methods are recommended in the case of a proprietary CAS usage.

GSI supports also authorization in both the server-side and the client-side. The server has six possible authorization modes: none, self, gridmap, identity-authorization, host authorization, SAML callout authorization. Depending on the authorization mode that if will be chosen, the server will decide if it accepts or declines an incoming request. Identity-authorization or SAML callout authorization are recommended in the case of a proprietary CAS usage.

4 A case study: Grid-based services using Maple

We proceed with a practical example of how a CAS can be made available as Grid-WSRF service and for this purpose Maple became our CAS of choice. The main reason is that, despite its robustness and ease of use, we were not able to locate efforts to link Maple with the Grid, accept ours, namely Maple2g. Furthermore, it is well known that Maple excels other CASs in solving selected classes of problems e.g. systems of nonlinear equations or inequalities. Finally, Maple has already a build-in socket library for communicating over the Internet, and a library for parsing XML. These capabilities match very well with our goal as they suffice to make Maple a client for an Grid computational service.

Maple2g (Maple-to-Grid) was build recently as a grid-wrapper for Maple. Maple2g consists of two parts a CAS-dependent and a Grid-dependent one. Therefore, any change in the CAS or in the Grid will be reflected only in one part of the proposed system. Furthermore, the CAS-dependent part is relatively simple and easy to be ported to support another CAS or legacy software.

Maple2g covers all three approaches described in Section 2.1. We describe here the newest component, the one that presents Maple as Grid service. It is based on the WSRF implementation from Globus Toolkit 4.

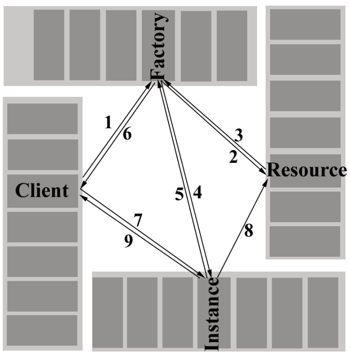


Figure 1: The four elements of the application, the client, the factory, the instance, and the resource. Their interaction in time

At the server side, the application has two main components: the factory service, the instance service. A resource is created by the factory service and used by the instance service. The interaction pushed by the client follows the steps (Figure 1):

1. the client request the service instance and resource creation;
2. the factory service creates the resource;
3. the key of the resource is identified by the factory;
4. the factory creates the service instance;

5. the service reference is returned to the factory;
6. the endpointreference (instance and resource identifiers) is send to the client;
7. the client contacts the service instance and request an operation involving the resource;
8. the service instance serves the request using the resource;
9. the service instance sends the result to the client.

The client can reiterate the steps (7-9) using the endpointreference.

The application components were written in Java. The factory service is activated once the Globus service container is activated on the resource were the service package was deployed. The instance service launches Maple as a concurrent thread on the same hardware resources on which it runs.

The communication between the Java-based instance service and Maple is done via sockets. At start Maple thread reads a temporary file that specify the socket channel that should be open and then the cycle in which it accepts any string coming on that communication channel, evaluate it as Maple command, and sends the result via the socket connection to the instance service.

For example, after the MapleFactoryService activation, the MapleClientPerform can send Maple commands to the instance service in form of strings:

```
$ MapleClientCreate http://194.102.62.15:8080/wsrf/services/MapleFactoryService >ref

$ MapleClientPerform ref "ifactor(1234567890567890987650987654321 00);"
(2)^2*(5)^2*(7)*(43)*(113)*(331)*(683)*(12067378391)*(65837)*(202087)

$ MapleClientPerform ref "P:=12*x^6+84*x^4-54*x^5-270*x^3+168*x^2-216*x+96:"

$ MapleClientPerform ref "solve(P);"
1/2, 4, I, -I, 2*I, -2*I
```

We have build also a Web interface that is depicted by Figure 2. We used AJAX (Asynchronous JavaScript And XML) for sending and receiving dates to the server. After the user authentification, the interface can be use in few simple step:

1. start the Globus container on the server (Figure 2.a);
2. choose the proper service (MathFactoryService) from the list of available services (Figure 2.b);
3. start the MathFactoryService which will launch the Maple instance (Figure 2.c);
4. fill and execute a Maple command (Figure 2.d and e).
5. the Maple instance executes the command and returns an MathML document as answer. The MathML document will be transform to a SVG file which is send back to the interface (Figure 2.f).

5 Conclusions and Further Work

At this stage Maple2g exists as a demonstrator system with some of the functionalities described above. In the near future it will be further developed to include facilities existing in other systems, in order for it to become more robust.

Currently if the access to Maple service is granted, any Maple commands can be used. Restricted access to a subset of commands (e.g. no access to shell commands revealing the host characteristics or establishing socket connections) should by implemented. Specialized services based on Maple should be developed and deployed as Grid services.

Experiments on the wide-area Grids will help guide further development of the system. Deployment of Grid services based on other CASs than Maple using the same codes must be take also into consideration.

Acknowledgment

This work was partially supported by the European project SCIEnce (FP6-2004-Infrastructure-5-026123) and the Romanian project CompGrid (CNCSIS-2004-949).

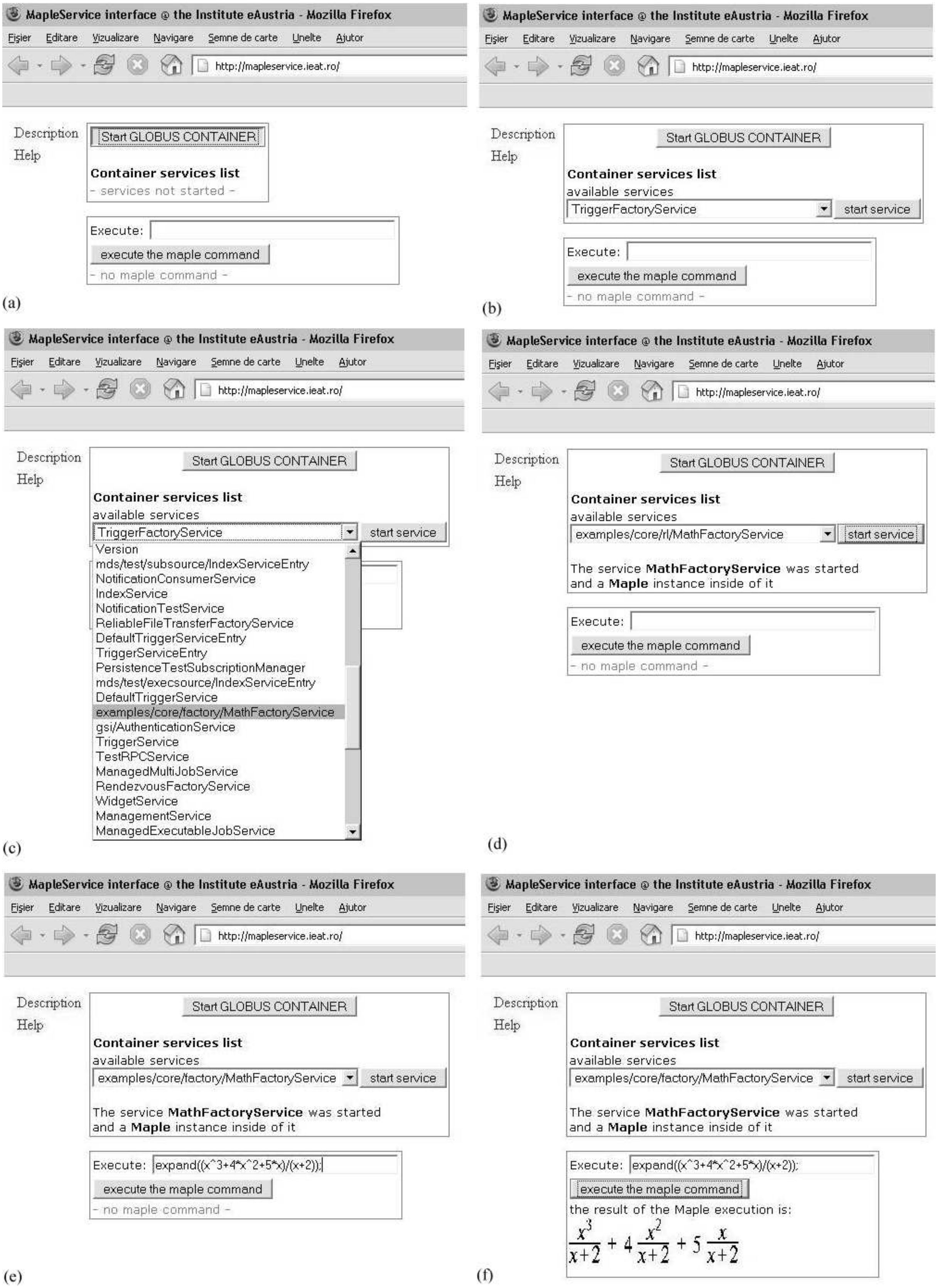


Figure 2: Using a Web interface to access symbolic computing services

References

[1] S. Agrawal, J. Dongarra, K. Seymour, S. Vadhiyar, “NetSolve: Past, Present, and Future – A Look at a Grid Enabled Server", in *Making the Global Infrastructure a Reality*, Wiley, Berman F. (Ed.), pp. 613-622, 2003.

[2] T. Delaitre, A. Goyeneche, P. Kacsuk, T. Kiss, G.Z. Terstyanszky, S.C. Winter, “GEMLCA: Grid Execution Management for Legacy Code Architecture Design”, in *Procs. of the 30th EUROMICRO conference*, Special Session on Advances in Web Computing, August 2004, Rennes, France, pp. 305-315, 2004.

[3] Genss, Available on-line at: http://genss.cs.bath.ac.uk/index.htm.

[4] Geodise, Available on-line at: http://www.geodise.org/.

[5] M. Gettrick, “OGB: Online Gröbner Bases”, http://grobner.it.nuigalway.ie, 2004.

[6] Internet-based Distributed Computing, Available on-line at: http://www.aspenleaf.com/ distributed/ap-math.html.

[7] MapleNet, Available on-line at: http://www.maplesoft.com/maplenet/.

[8] Monet, Available on-line at: http://monet.nag.co.uk.

[9] D. Petcu, D. Dubu, “An Extension of Maple for Grid and Cluster Computing”, *Procs. ICCC 2004*, I. Dziţac, T. Maghiar, C.Popescu (eds.), Oradea, Băile Felix SPA, May 27-29 2004, Ed. Metropolis, pp. 355-360, 2004.

[10] D.Petcu, D.Dubu, M.Paprzycki, “Grid-based Parallel Maple”, *LNCS* 3241, Procs. PVMMPI 2004, Budapest, Hungary, September 19-22, 2004, D. Kranzmüller, P. Kacsuk, J. Dongarra (eds.), Springer, pp. 215-223, 2004.

[11] D.Petcu, M.Paprycki, D.Dubu, “Design and Implementation of a Grid Extension of Maple”, *Scientific Programming*, vol. 13, no. 2, IOS Press, pp. 137-149, 2005.

[12] D. Petcu, D. Ţepeneu, M. Paprzycki, T. Ida, “Symbolic Computations on Grids”, Chapter 27, *Engineering the Grid: Status and Perspective*, B. di Martino, J. Dongarra, A. Hoisie, L. Yang, and H. Zima (eds.), 2006.

[13] B. Sotomayor, L. Childers, *Globus Toolkit 4 : Programming Java Services*, Morgan Kaufmann, 2005.

[14] D. Ţepeneu, T. Ida, “MathGridLink - Connecting Mathematica to the Grid”, in *Procs. IMS’04*, Banff, Alberta, Canada, 2004.

[15] A. Weber, W. Küchlin, B. Eggers, V. Simonis, “Parallel computer algebra software as a web component”, Available on-line at: http://www.cs.ucsb.edu/conferences/ java98/papers/algebra.pdf, 1998.

[16] webMathematica, Available on-line at: http://www.wolfram.com/products/webmathematica/.

Dana Petcu
Institute e-Austria Timişoara and
Western University of Timişoara
Computer Science Department
Address: B-dul Vasile Pârvan 4, 300223
Timişoara, Romania
E-mail: petcu@info.uvt.ro

Cosmin Bonchiş
Institute e-Austria Timişoara, and
Vasile Goldiş University of Arad
Computer Science Department

Cornel Izbaşa
Institute e-Austria Timişoara, and
Western University of Timişoara
Computer Science Department

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 41-46

An Exact Algorithm for Steiner Tree Problem on Graphs

Milan Stanojević, Mirko Vujošević

Abstract: The paper presents a new original algorithm for solving Steiner tree problem on graph. The algorithm is simple and intended for solving problems with relatively low dimensions. It is based on use of existing open source software for solving integer linear programming problems. The algorithm is tested and shown very efficient for different randomly generated problems on graphs of up to 50 nodes, up to 10 terminals and average node degree 7.
Keywords: Steiner tree problem on graph, branch and cut, algorithm, optimization

1 Introduction

The Steiner tree problem (STP) is met in telecommunication and energetic systems, VLSI technologies and in other network planning tasks. The problem is to find a minimal length tree which connects all terminal nodes of a given graph, and contains arbitrary number of other nodes. The problem is similar to the well known shortest spanning tree problem (SSTP), but unlike that it doesn't necessary contain all nodes of given graph. Also, a very important difference is that STP is much harder problem than SSTP.

The usual formulation of STP is the following: a connected undirected graph $G = (N,E)$, where $N = \{1,\ldots,n\}$ is a set of nodes and $E \subseteq \{\{i, j\} \mid i \in N, j \in N, i < j\}$ denotes a set of edges, is given. A positive value (length, weight, etc.) c_e is associated to every edge $e \in E$. Also, a set $T \subset N$ of so called terminal (Steiner) nodes is given.

Definition 1. Steiner tree for T in G is a subgraph $St = (N',E')$, $N' \subseteq N$, $E' \subset E$ which satisfies one of the following statements:

1. $T \subseteq N'$ and St is a tree;

2. $\forall s,t \in T$ in St exists exactly one path from s to t.

The length of Steiner tree is the sum of lengths of all edges which it consists of. The Steiner tree problem is to find the shortest Steiner tree. This is a NP-hard problem and its decision problem variant belongs to NP-complete problem class [6]. The problem is presented in details in surveys: [4], [5], [9] and [11]. In papers [2], [8] and [7] the problem was solved to optimality using sophisticated branch and cut methods.

In this paper an original exact algorithm, generally based on branch and cut procedure, for solving Steiner tree problems on graph is proposed. It uses code from an available open source project which develops software for solving linear and integer linear programming problems. An intention was to formulate an algorithm which can be implemented in relatively short time and which will be able to solve STP of "reasonable" dimensions in "reasonable" time. To be more precise, the implementation of proposed algorithm lasted about 3-4 days. The program could solve problems of dimensions which are larger of the most of real life problems in, at most, several minutes.

In this paper, in the next section, a mathematical model of the Steiner tree problem and its explanation are given. In Section 3, the algorithm and the process of implementation are explained in details. In Section 4 the experiments are described and some conclusions about the algorithm behavior are made. In Section 5, the conclusion of the whole paper is given.

2 Mathematical model

A Steiner tree can be represented by a vector of binary variables $x = (x_e)_{|E|}$. Each element of the vector is assigned to one edge from the set E. The value of a variable x_e indicates whether the corresponding edge e is in the Steiner tree ($x_e = 1$) or not ($x_e = 0$). A mathematical model of the STP on undirected graphs has the following form:

$$
\begin{array}{rlll}
\min & \sum\limits_{e \in E} c_e x_e & & \\
\text{subject to} & & & \\
(i) & \sum\limits_{e \in \delta(M)} x_e \geq 1 & \forall & M \subset N, \\
 & & & M \cap T \neq \emptyset, \\
 & & & (N \backslash M) \cap T \neq \emptyset \\
(ii) & x_e \in \{0,1\} & \forall & e \in E
\end{array}
\tag{1}
$$

where N, E and T are as in Definition 1, c_e is a positive value associated to each edge e and $\delta(M)$ denotes a graph cut defined by subset of nodes $M \subset N$, i.e. the set of edges with one end node in M and the second one in complement set $N \setminus M$, $\delta(M) = \{\{i, j\} \in E \mid i \in M, j \in N \setminus M\}$.

The mathematical model (1) is linear and integer. Constraints (i) ensure that in every cut, for which terminal nodes are on the both sides (M and $N \backslash M$), at least one edge exists. In other words, they ensure that between each two terminal nodes exists at least one path. A feasible solution of model (1) is not necessary a Steiner tree. But the optimal solution will be a Steiner tree because any edge constructing a contour would violate the optimality condition. So, although the formulation is rather comprehensive, it can be applied only for those problems where the goal is to minimize the length of the Steiner tree.

A disadvantage of model (1) is that the number of constraints grows exponentially in the problem size. On the other hand, in branch and cut methods, the relaxation of the formulation may give acceptable results.

3 Implementation

Two main challenges in solving the model (1) are: (*i*) the exponential number of constraints, and (*ii*) the exponential time needed to solve the integer program even with smaller number of constraints. To overcome the first challenge, the proposed algorithm uses a relaxation of model (1). The idea was inspired by the paper [7] and partly published at [10]. Namely, because of a big number of constraints, solving a model which includes all the constraints is very hard and in many cases impossible. On the other hand, in the most cases it is not necessary to include all the constraints in order to obtain an optimal solution. Only the constraints which are active in the optimal solution are really necessary. Of course, we cannot predict which of them will be active, but we can start with some smaller number of constraints, giving priority to those which are "more likely" to be necessary to obtain a feasible solution (Steiner tree). If we solve a model with smaller number of constraints and the solution is a Steiner tree of the given graph, then this solution will be the optimal for the starting problem, i.e. by adding more constraints we cannot improve the solution. Otherwise, the obtained solution will consist of two or more subtrees. Then, we iteratively add those constraints for which we find they are violated and which would probably lead to feasible solution until we finally obtain a Steiner tree.

The most common way to solve a binary linear programming problem is implementation of branch and bound method in combination with simplex method. According to the one of main features of the algorithm we intended to formulate – quick and easy implementation, development of simplex and branch and bound algorithms from a scratch wouldn't be appropriate. Although it would finally give better performance if they would be incorporated in the essence of complete procedure, the development of these procedures would last very long. An alternative was found among Open Source projects. The project used in this implementation was **lp_solve**.

Lp_solve [1] is an open source project that realize very robust procedures and techniques for solving linear programming problems. Beside that, it implements the branch and bound method for solving binary, integer and mixed integer linear problems. It has a lot of options by which it is possible to influence the branch and bound procedure changing its strategies, so it is possible to significantly improve its performance [13]. Lp_solve can be used as independent application when it can read problem files in *lp* and *mps* formats, and as a set of functions, when it can be incorporated into other programs and controlled from the host code.

The project itself doesn't have any restrictions of the problems dimensions. Some successful applications on mixed integer programming problems with several thousands variables were reported. The license of the project is GLGPL (GNU Lesser General Public License) [12] and it allows free download, using, changing and redistribution of the source code of lp_solve project.

The algorithm is formally formulated as follows:

Algorithm 1 Simplified branch and cut algorithm for STP

1. Formulation of initial integer linear mathematical model:

   (a) Goal function formulation: one variable is introduced for each edge and corresponding edge length is associated as a parameter to each variable.

   (b) For each terminal node, one constraint of type (i) is formulated, so the terminal node is a single node on one side of the cut and the rest of the nodes are on the other side, i.e. $\sum\limits_{e \in \delta(M)} x_e \geq 1 \; \forall M \in \{\{t\} \mid t \in T\}$.

   The number of constraints after Step 1 will be $|T|$.

2. Solve the current mathematical model.

3. Check if the obtained solution is a Steiner tree, i.e. if there is a path between all pairs of terminal nodes. If so, the optimal solution was found in Step 2; the end of the procedure. Otherwise, next step.

4. If solution is not a Steiner tree, then it represents two or more unconnected subtrees. For every subtree add one type (i) constraint defined by cut $\delta(M)$ where the nodes of that subtree belong to set M. Go to Step 2.

The proposed algorithm can be qualified as a simplified version of branch and cut method, i.e. a combination of branch and bound and cutting planes. The steps of the algorithm will be illustrated by an example. Suppose, we have to obtain the minimal Steiner tree for the given graph, illustrated in Figure 1, where four terminal nodes are marked with bigger circles.

The mathematical model created in Step 1 and updated in Step 4 will have one column for every edge. The constraints added in the first step are necessary to provide that every terminal is connected to, at least, one edge. In the example, four constraints will be added in the first step – one for every terminal. They will ensure that at least one edge is connected to each terminal. In Figure 2, the four cuts are marked as open curves surrounding each terminal node, and the edges marked by dashed lines are candidates to be in the first solution.

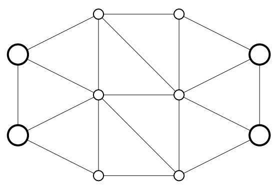


Figure 1: Initial graph

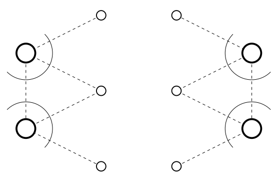


Figure 2: First step

The possible solution after Step 2 could be like one in Figure 3. As mentioned above, if the obtained solution is not a Steiner tree, it will consist of several subtrees. The number of subtrees generally can be between 2 and $|T|$. For determining if the obtained solution is a Steiner tree (in Step 3), Dijkstra's shortest path algorithm was used.

In Step 4 a new constraint for each subtree is added. In the example, three new constraints, corresponding to three subtrees shown in Figure 3, are added. On each image of Figure 4, one cut (represented by the curve surrounding the subtree) and edges (dashed lines) among which, at least one will be in the next solution are shown.

After the next optimization, a possible solution could be like the one shown in Figure 5. The solution satisfies all added constraints and the graph structure is a Steiner tree. Without further checking, we can claim that it is an

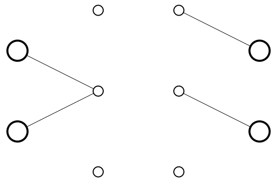


Figure 3: Possible solution after the first iteration

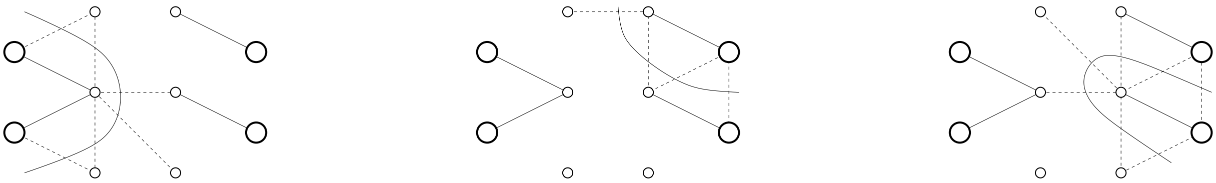


Figure 4: Three new constraints

optimal solution of the given Steiner tree problem.

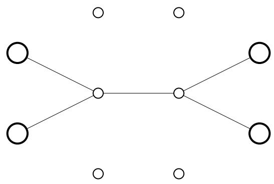


Figure 5: Final – feasible and optimal solution

The main criteria for determining data structures were access speed and simplicity of implementation. The amount of used memory was not considered because of relatively small graph dimensions of target instances. The logical choice were static structures (vectors and matrices). The realized structures enabled fast data access and mapping between graph structure (realized through neighborhood matrix) and vector of edges with all corresponding attributes.

The algorithm was implemented in C language and it is compiled and tested on Linux operation system (with gcc - GNU C Compiler) on PC with Pentium® III processor on 600 MHz and 384 MB of RAM. The implemented program supports so called *STP* format [15] of Steiner tree problems, and it is compatible with the library of standard Steiner tree problems [14].

4 Experimental results

The developed program was tested on different problem instances. A characteristic of the Steiner tree problem is that the complexity of a procedure for its solving depends on three attributes: number of nodes (n), number of edges and number of terminal nodes (t). The largest dimensions of instances successfully solved by the program were [n/t]: 20/10, 32/8 and 50/5, with the average node degree 7. The solving procedures lasted between 2 seconds and 2 minutes for the most of the instances. These dimensions may look modest in comparison with those with several thousands nodes which were successfully solved as it was reported in papers [2] and [7]. However, comparing the procedure complexity, simplicity of implementation and the fact that in many real-life telecommunication planning processes, even smaller size problems may appear, the proposed implementation could be very useful.

To get a more precise insight in the behavior of the algorithm, experiments were performed with instances where some parameters were varied. In the following table, results (mean values and standard deviation) obtained by experiments where every dimension was tested on 15 randomly generated instances are given. In the columns named “No. of rows” the number of constraints needed to obtain an optimal solution (the optimization in the last iteration) is given. The columns named “No. of iterations” represent a number of iterations of Algorithm 1 when passing through Steps 2-4, i.e. the total number of solved binary subproblems (in Step 2). The columns "Time" represent CPU time spent to solve the problem. The column "ANCAI" shows the average number of constraints added per iteration. It's obtained by formula: $\frac{No.rows-t}{No.iterations}$ where t represents the number of terminal nodes, i.e. the number of constraints added in initial mathematical model in Step 1.

Table 1: The complexity analysis of the algorithm

Instance dimensions		No. of rows		No. of iterations		Time [sec.]		AN-CAI
$[n/t]$		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	
1		2	3	4	5	6	7	8
1	20/5	25	8.9	7.5	3.0	0.2	0.3	2.7
2	32/5	38	22.1	12.3	8.0	3.0	9.7	2.7
3	50/5	60	28.3	20.0	11.3	7.2	13.3	2.8
4	20/8	65	20.3	19.5	8.5	11.5	27.5	2.9
5	32/8	88	31.9	26.3	12.0	34.5	46.7	3.0
6	20/10	83	15.5	23.6	5.2	13.4	9.9	3.1

On the basis of data given in Table 1, some conclusions can be made. The fact that complexity of STP grows with the number of nodes is obvious from the first three rows. Although the speed of the growth cannot be determined exactly on the basis of so small sample, it is obviously nonlinear – probably exponential. The more interesting conclusion is that the growth of complexity is faster by changing the number of terminal nodes than the total number of nodes in graph. Comparing rows 1, 4 and 6, a kind of "explosion" of complexity can be seen: when the number of terminals was increased two times, execution time was increased 67 times. Similar conclusion can be made observing rows 2 and 5: addition of three terminals resulted in execution time increase of more than 11 times.

The most important analysis here concerns the number of constraints needed to get an optimal solution. According the Column 2, that growth is almost linear in problem size. We cannot be certain if it is linear, but it is definitely not exponential. Finally, we can conclude that, although the number of constraints in model (1) grows exponentially, the number of constraints necessary to obtain an optimal solution grows much slowlier. The number of iterations grows even less. The parameter in Column 8 is also interesting. The average number of constraints added in each iteration also represents the average number of subtrees obtained in each sub solution. It seems that the relatively small values in Column 8 doesn't depend much on the number of terminal nodes. The explanation could be that the current solution of the solving procedure relatively quickly forms a structure which consists of a few subtrees each containing several terminals. This may contradict to the previous statement that complexity depends more on the number of terminals than on the total number of nodes, because, what influences the number of iterations is the number of unconnected subtrees, and not the number of terminals. One possible explanation is that the structure of subtrees changes, so subtrees contain different terminals in different iterations. It is also important to have in mind that every iteration lasts more than the previous one because every mathematical model has more rows (constraints) than the previous one.

Yet another interesting thing from Table 1 is relatively big dispersion (represented by standard deviation) of data obtained by different randomly generated instances with same characteristics. This is a consequence of a nature of the algorithm (which is nondeterministic polynomial). It is impossible to predict the number of iterations necessary to obtain final solution. In the worst case it can be exponential.

5 Conclusion

The first impressions and experiment conclusions indicate that the proposed algorithm can be efficiently used when there is a need for rapid development of an algorithm for solving smaller size Steiner tree problems. Although the worst case number of constraints needed to obtain a final solution remains exponential, the algorithm have shown a kind of "good behavior" – in all solved examples the number stayed relatively low. Although the exponential complexity of the branch and bound method (Step 2 of the algorithm) remains, instances with acceptable dimensions can be solved in real time.

The idea of successive adding violated constraints could be also applied to some other problems whose mathematical models have an exponential number of constraints. Concerning that, some new researches have been planed, where the idea would be applied to the traveling salesman problem (TSP). Namely, the so called DFJ formulation of TSP [3] also has exponential number of constraints, but it has shown a good behavior in relaxation based algorithms.

References

[1] M. Berkelaar, K. Eikland, P. Notebaert, Lp_solve, Files and Discussion Group, ftp://ftp.es.ele.tue.nl/pub/lp_solve, http://groups.yahoo.com/group/lp_solve/, 1994-2006.

[2] S. Chopra, E. Gorres, M. R. Rao, “Solving a Steiner tree problem on a graph using branch and cut”, *ORSA Journal on Computing*, Vol. 4, pp. 320-335, 1992.

[3] G. B. Dantzig, D. R. Fulkerson, S. M. Johnson, “Solution of a large-scale traveling-salesman problem”, *Operations Research*, Vol. 2, pp. 393-410, 1954.

[4] F. K. Hwang, D. S. Richards, “Steiner tree problems”, *Networks*, Vol. 22, pp. 55-89, 1992.

[5] F. K. Hwang, D. S. Richards, P. Winter, *The Steiner tree problem*, North-Holland, Amsterdam, 1992.

[6] R. M. Karp, “Reducibility among combinatorial problems”, R. E. Miller, J. W. Thatcher (Ed.), *Complexity of Computer Computations*, pp. 85-103, Plenum Press, New York, 1972.

[7] T. Koch, A. Martin, “Solving Steiner tree problems in graphs to optimality”, *Networks*, Vol. 32, pp. 207-232, 1998.

[8] A. Lucena, J.E. Beasley, “A branch and cut algorithm for the Steiner problem in graphs”, *Networks*, Vol. 31, pp. 39-59, 1998.

[9] N. Maculan, “The Steiner tree problem in graphs”, Surveys in Combinatorial Optimization, S. Martello, G. Laporte, M. Minoux, C. C. Ribeiro (Ed.), *Annals of Discrete Mathematics*, Vol. 31, pp. 185-212, 1987.

[10] M. Stanojević, M. Vujošević, “A new algorithm for solving Steiner tree problem on graph” (in Serbian), 12th Telecommunications Forum TELFOR 2004, Belgrade, http://www.telfor.org.yu/telfor2004/e-index.html (http://www.telfor.org.yu/telfor2004/radovi/TM-2-4.pdf), 2004.

[11] P. Winter, “Steiner problem in networks: A survey”, *Networks*, Vol. 17, pp. 129-167, 1987.

[12] GNU Lesser General Public License, http://www.gnu.org/copyleft/lesser.html

[13] Lp_solve Reference Guide, http://www.geocities.com/lpsolve/

[14] SteinLib TestSets – The Library of Standard Steiner Problems, http://elib.zib.de/steinlib/testset.php

[15] STP – Description of the STP Data Format, http://elib.zib.de/steinlib/format.php

Milan Stanojević, Mirko Vujošević
University of Belgrade
Faculty of Organizational Sciences
Address: Jove Ilića 154, Belgrade, Serbia and Montenegro
E-mail: {milans,mirkov}@fon.bg.ac.yu

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 47-57

Information Aggregation in Intelligent Systems Using Generalized Operators

Imre J. Rudas, János Fodor

Abstract: Aggregation of information represented by membership functions is a central matter in intelligent systems where fuzzy rule base and reasoning mechanism are applied. Typical examples of such systems consist of, but not limited to, fuzzy control, decision support and expert systems. Since the advent of fuzzy sets a great number of fuzzy connectives, aggregation operators have been introduced. Some families of such operators (like t-norms) have become standard in the field. Nevertheless, it also became clear that these operators do not always follow the real phenomena. Therefore, there is a natural need for finding new operators to develop more sophisticated intelligent systems. This paper summarizes the research results of the authors that have been carried out in recent years on generalization of conventional operators.
Keywords: t-norm, t-conorm, uninorm, entropy- and distance-based conjunctions and disjunctions.

1 Introduction

Information aggregation is one of the key issues in development of intelligent systems. Although fuzzy set theory provides a host of attractive aggregation operators for integrating the membership values representing uncertain information, the results do not always follow the modeled real phenomena and it has been shown that in some situations some operations may work better than others.

Since the pioneering work of Zadeh the basic research was oriented towards the investigation of the properties of t-norms and t-conorms and also to find new ones satisfying the axiom system. As a result of this a great number (of various type) of t-operators have been introduced accepting the axiom system as a fixed, unchangeable skeleton.

Until the last few years no strong efforts were devoted to generalize t-operators by modifying "weakening" this axiom system. On one hand, the sound theoretical foundation as well as their wide variety have given t-norms and t-conorms almost an exclusive role in different theoretical investigations and practical applications. On the other hand, people are inclined to use them also as a matter of routine. The following observations support this statement and .

When one works with binary conjunctions and there is no need to extend them for three or more arguments, associativity is an unnecessarily restrictive condition. The same is valid for commutativity if the two arguments have different semantical backgrounds and it has no sense to interchange one with the other.

These observations, which are very often left out of consideration, advocate our study and have urged us to revise definitions and properties of operations for information aggregation and reasoning.

2 Traditional Operations

The original fuzzy set theory was formulated in terms of Zadeh's standard operations of intersection, union and complement. The axiomatic skeleton used for characterizing fuzzy intersection and fuzzy union are known as *triangular norms (t-norms)* and *triangular conorms (t-conorms)*, respectively. For more details we refer to the book [9].

2.1 Triangular Norms and Conorms

Definition 1. A non-increasing function $N : [0,1] \to [0,1]$ satisfying $N(0) = 1$, $N(1) = 0$ is called a *negation*. A negation N is called *strict* if N is strictly decreasing and continuous. A strict negation N is said to be a *strong negation* if N is also involutive: $N(N(x)) = x$ for all $x \in [0,1]$.

The standard negation is simply $N_s(x) = 1 - x, \quad x \in [0,1]$. Clearly, this negation is strong. It plays a key role in the representation of strong negations.

We call a continuous, strictly increasing function $\varphi : [0,1] \to [0,1]$ with $\varphi(0) = 0$, $\varphi(1) = 1$ an *automorphism* of the unit interval.

Definition 2. A *triangular norm* (shortly: a t-norm) is a function $T:[0,1]^2 \to [0,1]$ which is associative, increasing and commutative, and satisfies the boundary condition $T(1,x) = x$ for all $x \in [0,1]$.

Definition 3. A *triangular conorm* (shortly: a t-conorm) is an associative, commutative, increasing $S:[0,1]^2 \to [0,1]$ function, with boundary condition $S(0,x) = x$ for all $x \in [0,1]$.

Notice that continuity of a t-norm and a t-conorm is not taken for granted.

In what follows we assume that T is a t-norm, S is a t-conorm and N is a strict negation.

Clearly, for every t-norm T and strong negation N, the operation S defined by

$$S(x,y) = N(T(N(x),N(y))), \quad x,y \in [0,1] \tag{1}$$

is a t-conorm. In addition, $T(x,y) = N(S(N(x),N(y)))$ $(x,y \in [0,1])$. In this case S and T are called N-*duals*. In case of the standard negation (i.e., when $N(x) = 1-x$ for $x \in [0,1]$) we simply speak about duals. Obviously, equality (1) expresses the De Morgan's law in the fuzzy case.

Generally, for any t-norm T and t-conorm S we have

$$T_W(x,y) \le T(x,y) \le T_M(x,y) \quad \text{and} \quad S_M(x,y) \le S(x,y) \le S_S(x,y),$$

where $T_M(x,y) = \min(x,y)$, $S_M(x,y) = \max(x,y)$, T_W is the weakest t-norm, and S_S is the strongest t-conorm.

These inequalities are important from practical point of view as they establish the boundaries of the possible range of mappings T and S.

2.2 Uninorms and Nullnorms

Uninorms

Uninorms were introduced by Yager and Rybalov [19] as a generalization of t-norms and t-conorms. For uninorms, the neutral element is not forced to be either 0 or 1, but can be any value in the unit interval.

Definition 4. [19] A uninorm U is a commutative, associative and increasing binary operator with a neutral element $e \in [0,1]$, i.e., for all $x \in [0,1]$ we have $U(x,e) = x$.

T-norms do not allow low values to be compensated by high values, while t-conorms do not allow high values to be compensated by low values. Uninorms may allow values separated by their neutral element to be aggregated in a compensating way. The structure of uninorms was studied by Fodor *et al.* [11]. For a uninorm U with neutral element $e \in]0,1]$, the binary operator T_U defined by

$$T_U(x,y) = \frac{U(ex,ey)}{e}$$

is a t-norm; for a uninorm U with neutral element $e \in [0,1[$, the binary operator S_U defined by

$$S_U(x,y) = \frac{U(e+(1-e)x, e+(1-e)y) - e}{1-e}$$

is a t-conorm. The structure of a uninorm with neutral element $e \in]0,1[$ on the squares $[0,e]^2$ and $[e,1]^2$ is therefore closely related to t-norms and t-conorms. For $e \in]0,1[$, we denote by ϕ_e and ψ_e the linear transformations defined by $\phi_e(x) = \frac{x}{e}$ and $\psi_e(x) = \frac{x-e}{1-e}$. To any uninorm U with neutral element $e \in]0,1[$, there corresponds a t-norm T and a t-conorm S such that:

(i) for any $(x,y) \in [0,e]^2$: $U(x,y) = \phi_e^{-1}(T(\phi_e(x),\phi_e(y)))$;

(ii) for any $(x,y) \in [e,1]^2$: $U(x,y) = \psi_e^{-1}(S(\psi_e(x),\psi_e(y)))$.

On the remaining part of the unit square, i.e. on $E = [0,e[\times]e,1] \cup]e,1] \times [0,e[$, it satisfies

$$\min(x,y) \le U(x,y) \le \max(x,y),$$

and could therefore partially show a compensating behaviour, i.e. take values strictly between minimum and maximum. Note that any uninorm U is either *conjunctive*, i.e. $U(0,1) = U(1,0) = 0$, or *disjunctive*, i.e. $U(0,1) = U(1,0) = 1$.

Representation of Uninorms

In analogy to the representation of continuous Archimedean t-norms and t-conorms in terms of additive generators, Fodor *et al.* [11] have investigated the existence of uninorms with a similar representation in terms of a single-variable function. This search leads back to Dombi's class of *aggregative operators* [7]. This work is also closely related to that of Klement *et al.* on associative compensatory operators [15]. Consider $e \in]0,1[$ and a strictly increasing continuous $[0,1] \to \overline{\mathbb{R}}$ mapping h with $h(0) = -\infty$, $h(e) = 0$ and $h(1) = +\infty$. The binary operator U defined by

$$U(x,y) = h^{-1}(h(x) + h(y))$$

for any $(x,y) \in [0,1]^2 \setminus \{(0,1),(1,0)\}$, and either $U(0,1) = U(1,0) = 0$ or $U(0,1) = U(1,0) = 1$, is a uninorm with neutral element e. The class of uninorms that can be constructed in this way has been characterized [11].

Consider a uninorm U with neutral element $e \in]0,1[$, then there exists a strictly increasing continuous $[0,1] \to \overline{\mathbb{R}}$ mapping h with $h(0) = -\infty$, $h(e) = 0$ and $h(1) = +\infty$ such that

$$U(x,y) = h^{-1}(h(x) + h(y))$$

for any $(x,y) \in [0,1]^2 \setminus \{(0,1),(1,0)\}$ if and only if

(i) U is strictly increasing and continuous on $]0,1[^2$;

(ii) there exists an involutive negator N with fixpoint e such that

$$U(x,y) = N(U(N(x),N(y))))$$

for any $(x,y) \in [0,1]^2 \setminus \{(0,1),(1,0)\}$.

The uninorms characterized above are called *representable* uninorms. The mapping h is called an *additive generator* of U. The involutive negator corresponding to a representable uninorm U with additive generator h, as mentioned in condition (ii) above, is denoted N_U and is given by

$$N_U(x) = h^{-1}(-h(x)). \tag{2}$$

Clearly, any representable uninorm comes in a conjunctive and a disjunctive version, i.e. there always exist two representable uninorms that only differ in the points $(0,1)$ and $(1,0)$. Representable uninorms are almost continuous, i.e. continuous except in $(0,1)$ and $(1,0)$, and Archimedean, in the sense that $(\forall x \in]0,e[)(U(x,x) < x)$ and $(\forall x \in]e,1[)(U(x,x) > x)$. Clearly, representable uninorms are not idempotent. The classes $\mathcal{U}_{\min}$ and $\mathcal{U}_{\max}$ do not contain representable uninorms. A very important fact is that the underlying t-norm and t-conorm of a representable uninorm must be strict and cannot be nilpotent. Moreover, given a strict t-norm T with decreasing additive generator f and a strict t-conorm S with increasing additive generator g, we can always construct a representable uninorm U with desired neutral element $e \in]0,1[$ that has T and S as underlying t-norm and t-conorm. It suffices to consider as additive generator the mapping h defined by

$$h(x) = \begin{cases} -f\left(\dfrac{x}{e}\right) & \text{, if } x \le e \\ g\left(\dfrac{x-e}{1-e}\right) & \text{, if } x \ge e \end{cases}. \tag{3}$$

On the other hand, the following property indicates that representable uninorms are in some sense also generalizations of nilpotent t-norms and nilpotent t-conorms: $(\forall x \in [0,1])(U(x,N_U(x)) = N_U(e))$. This claim is further supported by studying the residual operators of representable uninorms in [6].

As an example of the representable case, consider the additive generator h defined by $h(x) = \log \frac{x}{1-x}$, then the corresponding conjunctive representable uninorm $\mathcal{U}$ is given by $U(x,y) = 0$ if $(x,y) \in \{(1,0),(0,1)\}$, and

$$U(x,y) = \frac{xy}{(1-x)(1-y)+xy}$$

otherwise, and has as neutral element $\frac{1}{2}$. Note that N_U is the standard negator: $N_U(x) = 1-x$.

The class of representable uninorms contains famous operators, such as the functions for combining certainty factors in the expert systems MYCIN (see [18, 5]) and PROSPECTOR [5]. The MYCIN expert system was one of the first systems capable of reasoning under uncertainty [2]. To that end, certainty factors were introduced as numbers in the interval $[-1,1]$. Essential in the processing of these certainty factors is the modified combining function C proposed by van Melle [2]. The $[-1,1]^2 \to [-1,1]$ mapping C is defined by

$$C(x,y) = \begin{cases} x+y(1-x) & \text{, if } \min(x,y) \geq 0 \\ x+y(1+x) & \text{, if } \max(x,y) \leq 0 \\ \dfrac{x+y}{1-\min(|x|,|y|)} & \text{, otherwise} \end{cases} .$$

The definition of C is not clear in the points $(-1,1)$ and $(1,-1)$, though it is understood that $C(-1,1) = C(1,-1) = -1$. Rescaling the function C to a binary operator on $[0,1]$, we obtain a representable uninorm with neutral element $\frac{1}{2}$ and as underlying t-norm and t-conorm the product and the probabilistic sum. Implicitly, these results are contained in the book of Hájek *et al.* [14], in the context of ordered Abelian groups.

Nullnorms

Definition 5. [3] A nullnorm V is a commutative, associative and increasing binary operator with an absorbing element $a \in [0,1]$, i.e. $(\forall x \in [0,1])(V(x,a) = a)$, and that satisfies

$$(\forall x \in [0,a])(V(x,0) = x) \tag{4}$$

$$(\forall x \in [a,1])(V(x,1) = x) \tag{5}$$

The absorbing element a corresponding to a nullnorm V is clearly unique. By definition, the case $a = 0$ leads back to t-norms, while the case $a = 1$ leads back to t-conorms. In the following proposition, we show that the structure of a nullnorm is similar to that of a uninorm. In particular, it can be shown that it is built up from a t-norm, a t-conorm and the absorbing element [3].

Theorem 6. *Consider $a \in [0,1]$. A binary operator V is a nullnorm with absorbing element a if and only if*

(i) *if $a = 0$: V is a t-norm;*

(ii) *if $0 < a < 1$: there exists a t-norm T_V and a t-conorm S_V such that $V(x,y)$ is given by*

$$\begin{cases} \phi_a^{-1}(S_V(\phi_a(x),\phi_a(y))) & \text{, if } (x,y) \in [0,a]^2 \\ \psi_a^{-1}(T_V(\psi_a(x),\psi_a(y))) & \text{, if } (x,y) \in [a,1]^2 \\ a & \text{, elsewhere} \end{cases} ; \tag{6}$$

(iii) *if $a = 1$: V is a t-conorm.*

Recall that for any t-norm T and t-conorm S it holds that $T(x,y) \leq \min(x,y) \leq \max(x,y) \leq S(x,y)$, for any $(x,y) \in [0,1]^2$. Hence, for a nullnorm V with absorbing element a it holds that $(\forall (x,y) \in [0,a]^2)$ $(V(x,y) \geq \max(x,y))$ and $(\forall (x,y) \in [a,1]^2)$ $(V(x,y) \leq \min(x,y))$. Clearly, for any nullnorm V with absorbing element a it holds for all $x \in [0,1]$ that

$$V(x,0) = \min(x,a) \quad \text{and} \quad V(x,1) = \max(x,a). \tag{7}$$

Notice that, without the additional conditions (4) and (5), it cannot be shown that a commutative, associative and increasing binary operator V with absorbing element a behaves as a t-conorm and t-norm on the squares $[0,a]^2$ and $[a,1]^2$.

Nullnorms are a generalization of the well-known *median* studied by Fung and Fu [13], which corresponds to the case $T = \min$ and $S = \max$. For a more general treatment of this operator, we refer to [10]. We recall here the characterization of that median as given by Czogala and Drewniak [4]. Firstly, they observe that an idempotent, associative and increasing binary operator O has as absorbing element $a \in [0,1]$ if and only if $O(0,1) = O(1,0) = a$. Then the following theorem can be proven.

Theorem 7. [4] *Consider $a \in [0,1]$. A continuous, idempotent, associative and increasing binary operator O satisfies $O(0,1) = O(1,0) = a$ if and only if it is given by*

$$O(x,y) = \begin{cases} \max(x,y) & \text{, if } (x,y) \in [0,a]^2 \\ \min(x,y) & \text{, if } (x,y) \in [a,1]^2 \\ a & \text{, elsewhere} \end{cases} .$$

Nullnorms are also a special case of the class of T-S aggregation functions introduced and studied by Fodor and Calvo [12].

Definition 8. Consider a continuous t-norm T and a continuous t-conorm S. A binary operator F is called a T-S aggregation function if it is increasing and commutative, and satisfies the boundary conditions

$$(\forall x \in [0,1])(F(x,0) = T(F(1,0),x))$$
$$(\forall x \in [0,1])(F(x,1) = S(F(1,0),x)).$$

When T is the algebraic product and S is the probabilistic sum, we recover the class of aggregation functions studied by Mayor and Trillas [17]. Rephrasing a result of Fodor and Calvo, we can state that the class of associative T-S aggregation functions coincides with the class of nullnorms with underlying t-norm T and t-conorm S.

2.3 The Role of Commutativity and Associativity

One possible way of simplification of axiom skeletons of t-norms and t-conorms may be not requiring that these operations to have the commutative and the associative properties. Non-commutative and non-associative operations are widely used in mathematics, so, why do we restrict our investigations by keeping these axioms? What are the requirements of the most typical applications?

From theoretical point of view the commutative law is not required, while the associative law is necessary to extend the operation to more than two variables. In applications, like fuzzy logic control, fuzzy expert systems and fuzzy systems modeling fuzzy rule base and fuzzy inference mechanism are used, where the information aggregation is performed by operations. The inference procedures do not always require commutative and associative laws of the operations used in these procedures. These properties are not necessary for conjunction operations used in the simplest fuzzy controllers with two inputs and one output. For rules with greater amount of inputs and outputs these properties are also not required if the sequence of variables in the rules are fixed.

Moreover, the non-commutativity of conjunction may in fact be desirable for rules because it can reflect different influences of the input variables on the output of the system. For example, in fuzzy control, the positions of the input variables the “error” and the “change in error” in rules are usually fixed and these variables have different influences on the output of the system. In the application areas of fuzzy models when the sequence of operands is not fixed, the property of non-commutativity may not be desirable. Later some examples will be given for parametric non-commutative and non-associative operations.

3 Generalized Conjunctions and Disjunctions

The axiom systems of t-norms and t-conorms are very similar to each other except the neutral element, i.e. the type is characterized by the neutral element. If the neutral element is equal to 1 then the operation is a conjunction type, while if the neutral element is zero the disjunction operation is obtained. By using these properties we introduce the concepts of conjunction and disjunction operations [1].

Definition 9. Let T be a mapping $T : [0,1] \times [0,1] \to [0,1]$. T is a *conjunction operation* if $T(x,1) = x$ for all $x \in [0,1]$.

Definition 10. Let S be a mapping $S : [0,1] \times [0,1] \to [0,1]$. S is a *conjunction operation* if $S(x,0) = x$ for all $x \in [0,1]$.

Conjunction and disjunction operations may also be obtained one from another by means of an involutive negation N: $S(x,y) = N(T(N(x),N(y)))$, and $T(x,y) = N(S(N(x),N(y)))$.

It can be seen easily that conjunction and disjunction operations satisfy the following boundary conditions: $T(1,1) = 1$, $T(0,x) = T(x,0) = 0$, $S(0,0) = 0$, $S(1,x) = S(x,1) = 1$. By fixing these conditions, new types of generalized operations are introduced.

Definition 11. Let T be a mapping $T : [0,1] \times [0,1] \to [0,1]$. T is a *quasi-conjunction operation* if $T(0,0) = T(0,1) = T(1,0) = 0$, and $T(1,1) = 1$.

Definition 12. Let S be a mapping $S : [0,1] \times [0,1] \to [0,1]$. S is a *quasi-disjunction operation* if $S(0,1) = S(1,0) = S(1,1) = 1$, and $S(0,0) = 0$.

It is easy to see that conjunction and disjunction operations are quasi-conjunctions and quasi-disjunctions, respectively, but the converse is not true.

Omitting $T(1,1)=1$ and $S(0,0)=0$ from the definitions further generalization can be obtained.

Definition 13. Let T be a mapping $T:[0,1]\times[0,1]\to[0,1]$. T is a *pseudo-conjunction operation* if $T(0,0)=T(0,1)=T(1,0)=0$.

Definition 14. Let S be a mapping $S:[0,1]\times[0,1]\to[0,1]$. S is a *pseudo-disjunction operation* if $S(0,1)=S(1,0)=S(1,1)=1$.

Theorem 15. *Assume that T and S are non-decreasing pseudo-conjunctions and pseudo-disjunctions, respectively. Then there exist the absorbing elements 0 and 1 such as $T(x,0)=T(0,x)=0$ and $S(x,1)=S(1,x)=1$.*

3.1 Entropy-based Conjunction and Disjunction Operators

The question of how fuzzy is a fuzzy set has been one of the issues associated with the development of the fuzzy set theory. In accordance with a current terminological trend in the literature, measure of uncertainty is being referred as *measure of fuzziness*, or *fuzzy entropy* [16].

Throughout this part the following notations will be used; X is the universal set, $\boldsymbol{F}(X)$ is the class of all fuzzy subsets of X , $\mathfrak{R}^+$ is the set of non negative real numbers, $\overline{A}$ is the fuzzy complement of $A\in\boldsymbol{F}(X)$ and $|A|$ is the cardinality of A.

Definition 16. Let Xbe a universal set and A is a fuzzy subset of X with membership function μ_A. The *fuzzy entropy* is a mapping e: $\boldsymbol{F}(X)\to\mathfrak{R}^+$ which satisfies the following axioms:

AE 1 $e(A)$ = 0 if A is a crisp set.

AE 2 If $A\prec B$ then $e(A) \quad\leq\quad e(B)$; where $A\prec B$ means that *A is sharper than B.*

AE 3 $e(A)$ assumes its maximum value if and only if A is *maximally fuzzy*.

AE 4 $e(A)=e\left(\overline{A}\right),\forall A\in X$**.**

Let e_p be equilibrium of the fuzzy complement C and specify **AE 2** and **AE 3** as follows:

AES 2 *A is sharper than B* in the following sense:

$\mu_A(x)\leq\mu_B(x)$ for $\mu_B(x)\leq e_p$ and $\mu_A(x)\geq\mu_B(x)$ for $\mu_B(x)\geq e_p$, for all $x\in X$.

AES 3 A is defined maximally fuzzy when $\mu_A(x)=e_p\ \forall x\in X$.

Let A be a fuzzy subset of X and define the following function $f_A:X\to[0,1]$ by

$$f_A:x\mapsto\begin{cases}\mu_A(x) & \text{if } \mu_A(x)\leq e_p\\ C(\mu_A(x)) & \text{if } \mu_A(x)>e_p\end{cases}\tag{8}$$

Denote Φ_A the fuzzy set generated by f_A as its membership function.

Theorem 17. *The $g(|\Phi_A|)$ is an entropy, where g:$\mathfrak{R}\to\mathfrak{R}$ is a monotonically increasing real function and $g(0)=0$.*

Definition 18. Let A be a fuzzy subset of X. f_A is said to be an *elementary fuzzy entropy function* if the cardinality of the fuzzy set $\Phi_A=\{(x,f_A(x))\,|x\in X,\ f_A(x)\in[0,1]\}$ is an entropy of A.

It is obvious that f_A is an elementary entropy function.

Now we introduce some operations based on entropy. For more details we refer to [1].

Definition 19. Let A and B be two fuzzy subsets of the universe of discourse X and denote φ_A and φ_B their elementary entropy functions, respectively. The *minimum entropy conjunction operations* is defined as $I^*_\varphi=I^*_\varphi(A,B)=\left\{\left(x,\mu_{I^*_\varphi}(x)\right)|x\in X,\ \mu_{I^*_\varphi}(x)\in[0,1]\right\}$, where

$$\mu_{I^*_\varphi}:x\mapsto\begin{cases}\mu_A(x), & \text{if } \varphi_A(x)<\varphi_B(x)\\ \mu_B(x), & \text{if } \varphi_B(x)<\varphi_A(x)\\ \min(\mu_A(x),\mu_B(x)), & \text{if } \varphi_A(x)=\varphi_B(x)\end{cases}.\tag{9}$$

Definition 20. Let A and Bbe two fuzzy subsets of the universe of discourse X and denote φ_A and φ_B their elementary entropy functions, respectively. The *maximum entropy disjunction operation* is defined as $U^*_\varphi=U^*_\varphi(A,B)=\left\{\left(x,\mu_{U^*_\varphi}(x)\right)|x\in X,\mu_{U^*_\varphi}(x)\in[0,1]\right\}$, where

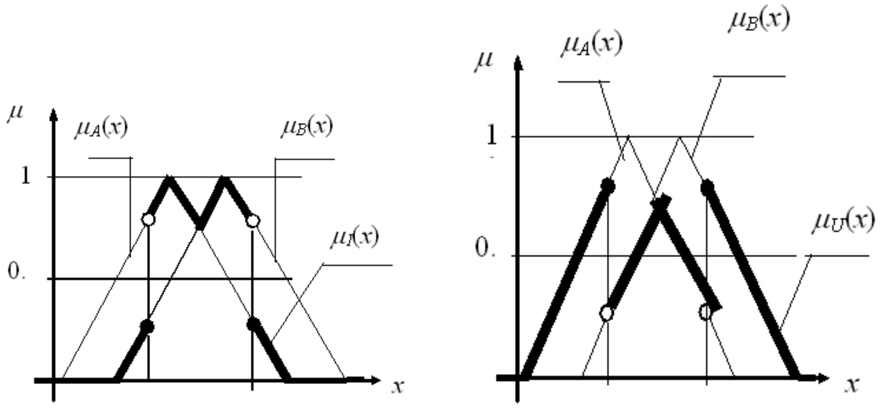


Figure 1: Entropy based conjunction operator (left) and entropy based disjunction operator (right)

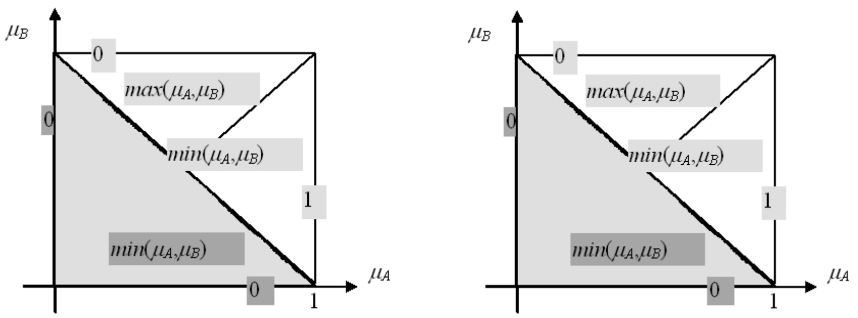


Figure 2: The construction of I_{φ}^{*} (left) and the construction of U_{φ}^{*} (right).

$$\mu_{U_{\varphi}^{*}} : x \mapsto \begin{cases} \mu_A(x), & \text{if } \varphi_A(x) > \varphi_B(x) \\ \mu_B(x), & \text{if } \varphi_B(x) > \varphi_A(x) \\ \max(\mu_A(x), \mu_B(x)), & \text{if } \varphi_A(x) = \varphi_B(x) \end{cases} . \tag{10}$$

The geometrical representation of the minimum fuzziness conjunction and the maximum fuzziness disjunction operators can be seen in Fig. 1.

Several important properties of these operations as well as their construction can be found in [1]. Now we present only two figures about the construction.

Notice also that I_{φ}^{*} is a quasi-conjunction, U_{φ}^{*} is a quasi-disjunction operation, and U_{φ}^{*} is a commutative semi-group operation on $[0,1]$ [1].

3.2 A Parametric Family of Quasi-Conjunctions

Let us cite the following result, which is the base of the forthcoming parametric construction, from [1].

Theorem 21. *Suppose T_1, T_2 are quasi-conjunctions, S_1 and S_2 are pseudo disjunctions and h, $g_1, g_2 :$ $[0,1] \to [0,1]$ are non-decreasing functions such that $g_1(1) = g_2(1) = 1$. Then the following functions*

$$T(x,y) = T_2(T_1(x,y), S_1(g_1(x), g_2(y))) \tag{11}$$

$$T(x,y) = T_2(T_1(x,y), g_1 S_1(x,y)) \tag{12}$$

$$T(x,y) = T_2(T_1(x,y), S_2(h(x), S_1(x,y))) \tag{13}$$

are quasi-conjunctions.

By the use of this Theorem the simplest parametric *quasi*-conjunction operations can be obtained as follows:

$$T(x,y) = x^p y^q, \tag{14}$$

$$T(x,y) = \min(x^p, y^q), \tag{15}$$

$$T(x,y) = (xy)^p (x+y-xy)^q \tag{16}$$

where $p,q \geq 0$.

4 Distance-based Operations

Let e be an arbitrary element of the closed unit interval $[0,1]$ and denote by $d(x,y)$ the distance of two elements x and $y$of $[0,1]$. The idea of definitions of distance-based operators is generated from the reformulation of the definition of the min and max operators as follows

$$\min(x,y) = \begin{cases} x, & \text{if } d(x,0) \leq d(y,0) \\ y, & \text{if } d(x,0) > d(y,0) \end{cases}, \quad \max(x,y) = \begin{cases} x, & \text{if } d(x,0) \geq d(y,0) \\ y, & \text{if } d(x,0) < d(y,0) \end{cases}$$

Based on this observation the following definitions can be introduced, see [1].

Definition 22. The *maximum distance minimum operator* with respect to $e \in [0,1]$ is defined as

$$\max_e^{\min}(x,y) = \begin{cases} x, & \text{if } d(x,e) > d(y,e) \\ y, & \text{if } d(x,e) < d(y,e) \\ \min(x,y), & \text{if } d(x,e) = d(y,e) \end{cases}. \tag{17}$$

Definition 23. The *maximum distance maximum operator* with respect to $e \in [0,1]$ is defined as

$$\max_e^{\max}(x,y) = \begin{cases} x, & \text{if } d(x,e) > d(y,e) \\ y, & \text{if } d(x,e) < d(y,e) \\ \max(x,y), & \text{if } d(x,e) = d(y,e) \end{cases}. \tag{18}$$

Definition 24. The *minimum distance minimum operator* with respect to $e \in [0,1]$ is defined as

$$\min_e^{\min}(x,y) = \begin{cases} x, & \text{if } d(x,e) < d(y,e) \\ y, & \text{if } d(x,e) > d(y,e) \\ \min(x,y), & \text{if } d(x,e) = d(y,e) \end{cases}. \tag{19}$$

Definition 25. The *minimum distance maximum operator* with respect to $e \in [0,1]$ is defined as

$$\min_e^{\max}(x,y) = \begin{cases} x, & \text{if } d(x,e) < d(y,e) \\ y, & \text{if } d(x,e) > d(y,e) \\ \max(x,y), & \text{if } d(x,e) = d(y,e) \end{cases}. \tag{20}$$

4.1 The Structure of Distance-based Operators

It can be proved by simple computation that if the distance of x and $y$is defined as $d(x,y) = |x-y|$ then the distance-based operators can be expressed by means of the min and max operators as follows.

$$\max_e^{\min} = \begin{cases} \max(x,y), & \text{if } y > 2e-x \\ \min(x,y), & \text{if } y < 2e-x \\ \min(x,y), & \text{if } y = 2e-x \end{cases}, \quad \min_e^{\min} = \begin{cases} \min(x,y), & \text{if } y > 2e-x \\ \max(x,y), & \text{if } y < 2e-x \\ \min(x,y), & \text{if } y = 2e-x \end{cases} \tag{21}$$

$$\max_e^{\max} = \begin{cases} \max(x,y), & \text{if } y > 2e-x \\ \min(x,y), & \text{if } y < 2e-x \\ \max(x,y), & \text{if } y = 2e-x \end{cases}, \quad \min_e^{\max} = \begin{cases} \min(x,y), & \text{if } y > 2e-x \\ \max(x,y), & \text{if } y < 2e-x \\ \max(x,y), & \text{if } y = 2e-x \end{cases} \tag{22}$$

The structures of the $\max_e^{\min}$ and the $\min_e^{\min}$ operators are illustrated in Fig. 3.

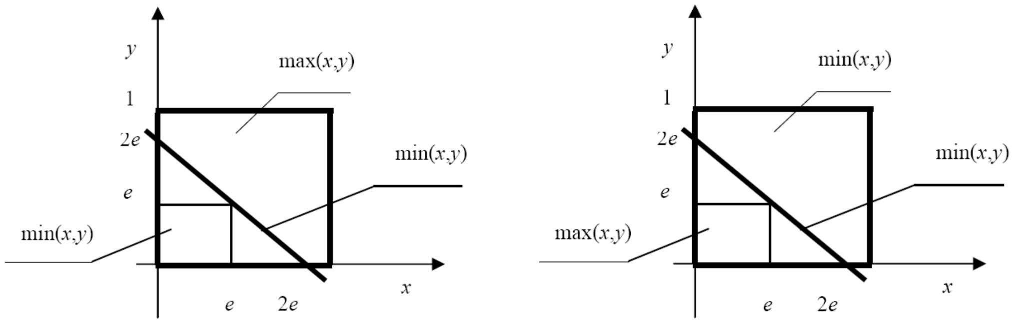


Figure 3: Maximum distance minimum operator (left) and minimum distance minimum operator (right).

5 Summary and Conclusions

In this paper we summarized some of our contributions to the theory of non-conventional aggregation operators. Further details and another classes of aggregation operators can be found in [1].

References

[1] I. Batyrshin, O. Kaynak, and I. Rudas, Fuzzy Modeling Based on Generalized Conjunction Operations, *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 5 (2002), pp. 678–683.

[2] B. Buchanan and E. Shortliffe, *Rule-Based Expert Systems, The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA, 1984.

[3] T. Calvo, B. De Baets and J. Fodor, The functional equations of Frank and Alsina for uninorms and nullnorms, *Fuzzy Sets and Systems* **120** (2001) 385–394.

[4] E. Czogala and J. Drewniak, *Associative monotonic operations in fuzzy set theory*, Fuzzy Sets and Systems **12** (1984), 249–269.

[5] B. De Baets and J. Fodor, Van Melle's combining function in MYCIN is a representable uninorm: An alternative proof. *Fuzzy Sets and Systems* **104** (1999) 133-136.

[6] B. De Baets and J. Fodor, Residual operators of uninorms, *Soft Computing* **3** (1999), 89–100.

[7] J. Dombi, Basic concepts for the theory of evaluation: the aggregative operator, *European J. Oper. Res.* **10** (1982), 282–293.

[8] J.C. Fodor, A new look at fuzzy connectives, *Fuzzy Sets and Systems* **57** (1993) 141–148.

[9] J.C. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support,* (Kluwer, Dordrecht, 1994).

[10] J. Fodor, *An extension of Fung–Fu's theorem*, Internat. J. Uncertain. Fuzziness Knowledge-Based Systems **4** (1996), 235–243.

[11] J. Fodor, R. Yager and A. Rybalov, Structure of uninorms, *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems* **5** (1997) 411–427.

[12] J. Fodor and T. Calvo, *Aggregation functions defined by t-norms and t-conorms*, Aggregation and Fusion of Imperfect Information (B. Bouchon-Meunier, ed.), Physica-Verlag, 1998, pp. 36–48.

[13] L. Fung and K. Fu, *An axiomatic approach to rational decision-making in a fuzzy environment*, Fuzzy Sets and their Applications to Cognitive and Decision Processes (K. Tanaka, L. Zadeh, K. Fu and M. Shimura, eds.), Academic Press, New York, San Francisco, London, 1975, pp. 227–256.

[14] P. Hájek, T. Havránek and R. Jiroušek, *Uncertain Information Processing in Expert Systems* (CRC Press, 1992).

[15] E.-P. Klement, R. Mesiar and E. Pap, On the relationship of associative compensatory operators to triangular norms and conorms, *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems* **4** (1996), 129–144.

[16] J.G. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information* (Prentice-Hall International Editions, USA, 1988).

[17] G. Mayor and E. Trillas, *On the representation of some aggregation functions*, Proc. Internat. Symposium on Multiple-Valued Logic, 1986, pp. 110–114.

[18] A. Tsadiras and K. Margaritis, The MYCIN certainty factor handling function as uninorm operator and its use as a threshold function in artificial neurons, *Fuzzy Sets and Systems* **93** (1998), 263–274.

[19] R. Yager and A. Rybalov, Uninorm aggregation operators, *Fuzzy Sets and Systems* **80** (1996) 111–120.

Imre J. Rudas, János Fodor
Budapest Tech
Institute of Intelligent Engineering Systems
Address: Bécsi út 96/b, H-1034 Budapest, Hungary
E-mail: {rudas,fodor}@bmf.hu

The research has been supported in part by OTKA T046762.

Editor's note about the authors:

Imre J. Rudas graduated from Bánki Donát Polytechnic, Budapest in 1971, received the Master Degree in Mathematics from the Eötvös Loránd University, Budapest, the Ph.D. in Robotics from the Hungarian Academy of Sciences in 1987, while the Doctor of Science degree from the Hungarian Academy of Sciences. He received his first Doctor Honoris Causa degree from the Technical University of Kosice, Slovakia and his second Honorary Doctorate from University Polytechnica Timisoara, Romania. He is active as a full university professor and Head of Department of Intelligent Engineering Systems. He serves as the Rector of Budapest Tech from August 1, 2003 for a period of four years. He is a Fellow of IEEE, Administrative Committee member of IEEE Industrial Electronics Society, Chairman of the Hungarian Chapters of IEEE Computational Intelligence and IEEE Systems, Man and Cybernetics Societies. He is also a registered expert of the United Nations Industrial Development Organization and the EU. He is the President of the Hungarian Fuzzy Association and Steering Committee Member of the Hungarian Robotics Association and the John von Neumann Computer Society. He is member of the editorial boards of several leading international scientific journals and founder of the IEEE International Conference Series on Intelligent Engineering Systems and IEEE International Conference on Computational Cybernetics, and some regional symposia. His present areas of research activity are: Robotics with special emphasis on Robot Control, Soft Computing, Computed Aided Process Planning, Fuzzy Control and Fuzzy Sets. He has published one book, more than 300 papers in various journals and international conference proceedings.

János Fodor obtained his master degree in Mathematics in 1981, his university doctorate in 1984. In 2000 he habilitated at the Roland Eötvös University, Budapest, Hungary. He defended his CSc and DSc degrees in 1991 and 2004, respectively, at the Hungarian Academy of Sciences. In 1997 he won the "Széchenyi Professorship" in the first phase of this award. At the beginning of his researcher's activity he was involved in the stochastic problems of operations research, later on in the field of fuzzy logics, preference modeling, decision making, computational intelligence, intelligent control, and mathematical foundations of the management of uncertainties. He is author of two monographs (issued by Kluwer and Springer), and about 200 other scientific publications. He served as the editor of two scientific books, and serves as the editor of various journal issues. His works obtained more than 700 independent citations. He is member of the editorial boards of various leading international scientific journals. János Fodor is the Co-Chairman of the Hungarian Fuzzy Association, and the EUROFUSE EURO Working Group on Fuzzy Sets. He is member of the Operation Research Committee of the Hungarian Academy of Sciences, and the Jury of Electronics and Electrical Sciences of the Hungarian Research Fund (OTKA). He regularly leads various research projects supported by diverse funds and resources. Many times he served as invited professor in various universities in Italy, Spain, Belgium, and France.

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 59-67

Personalized e-Learning Implementation - The GIS Case

Athanasios D. Styliadis, Ioannis D. Karamitsos, Dimitrios I. Zachariou

Abstract: Personalized e-learning implementation is recognized as among one of the most interesting research areas in the distance learning Web-based education. In particular, the GIS e-learning initiatives that incorporate —by default— a number of sequencing spatial techniques (i.e. spatial objects selection and sequencing), will well benefit from a well-defined personalized e-learning implementation with embedded spatial functionality. This is the case addressed in this paper.
The GIS e-learning implementation introduced in the current paper is based on a set of teaching (lecturing) rules according to the cognitive style of learning preferences of both the learners and the lecturers as well. It is important to note that, in spite of the fact that most of these teaching rules are *generic* (i.e. domain, view and user independent), there are no so far well-defined and commonly accepted rules on how the learning spatial GIS objects and techniques should be selected and how they should be sequenced to make "*instructional sense*" in a Web-based GIS course.
Keywords: e-Learning, Distance Learning, GIS, LIS, AM/FM, Spatial Sciences.

1 Introduction

GIS learning object selection is the first step to adaptive space navigation and adaptive course sequencing with GIS (space) functionality. Adaptive navigation seeks to present the GIS learning objects associated with an on-line course in an optimized order, where the optimisation criteria takes into consideration the learner's background and performance in related learning objects [2], whereas adaptive course sequencing is defined as the process that selects learning objects from a digital repository and sequence them in a way which is appropriate for the targeted GIS learning community or individuals [16, 11, 17]. Selection and sequencing is recognized as among the most interesting research questions in intelligent web-based GIS education [13, 6].

Although many types of intelligent learning systems with no GIS functionality are available, in the proposed GIS case five key components could be identified, which are common in most GIS systems, namely: *data acquisition*, *retrieval and analysis*, *preliminary data processing*, *database construction*, and *communication and visualization*. Figure 1 provides a view of the interactions between these five GIS modules [18].

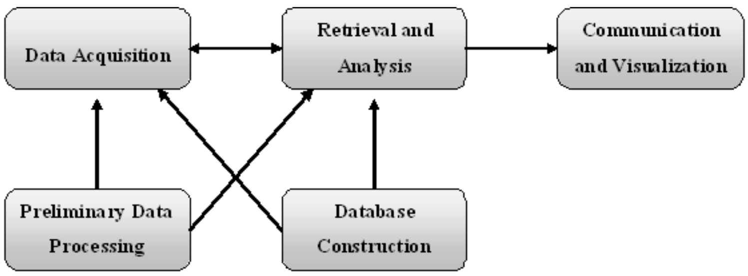


Figure 1: The Main Components of a GIS Intelligent Learning Systems

In most intelligent learning systems that incorporate course sequencing techniques, the pedagogical module is responsible for setting the principles of content selection and instructional planning. The selection of content (in our case, the GIS learning objects) is based on a set of teaching rules according to the topology-based cognitive style or learning preferences of the learners [3, 15]. In spite of the fact that most of these rules are *generic* (i.e. domain independent), there are no well defined and commonly accepted rules on how the GIS learning objects should be selected and how they should be sequenced to make "*instructional sense*" with GIS functionality [14]. Moreover, in order to design highly adaptive learning systems a huge set of topology-based rules is required, since dependencies between educational characteristics of GIS learning objects and learners are rather complex [8, 12, 9, 10].

In this paper, the learning object selection problem in GIS intelligent learning systems is addressed by proposing a methodology that instead of “forcing” an instructional designer to manually define the set of selection rules; produces a decision model that mimics the way the designer decides, based on the observation of the designer’s reaction over a small-scale learning object selection problem.

In the next (second) section the GIS learning object selection process is discussed as a part of a spatial-course sequencing. The third section discusses a topology-based filtering process of GIS learning objects used for reduction of learning objects searching space and proposes GIS metadata elements that can be used for learning object filtering according to the Open GIS Consortium guidelines (standards) for GIS functionality.

2 GIS Learning Object Selection in Spatial-Course Sequencing

In automatic and semi-automatic course sequencing, the main idea is to generate a course suited to the needs of the learners. As described in the literature, two main approaches for automatic and semi-automatic course sequencing have been identified: *Adaptive Courseware Generation (ACG)* and *Dynamic Courseware Generation (DCG)* [3].

In ACG the goal is to generate an individualized course taking into account specific learning goals, as well as, the initial level of the student’s knowledge. The entire course is adaptively generated before presenting it to the learner, instead of generating a course incrementally, as in a traditional sequencing context. In DCG on the other hand, the system observes the student progress during his interaction with the course and dynamically adapts the course according to the specific student needs and requirements. If the student’s performance does not meet the expectations, the course is dynamically re-planned. The benefit of this approach is that it applies as much adaptivity to an individual student as possible.

Both, the above mentioned techniques, in the case of GIS lecturing employ a pre-filtering topology-based mechanism to generate a group (“pool”) of GIS learning objects that match the general content requirements. This pool can be generated from both distributed and local GIS learning object repositories, provided that the appropriate access controls have been granted. The filtering process is based on general requirements such as topology, GIS functionality, characteristics of the teaching language, the media of the targeted GIS learning objects, as well as, the use of ontologies and topology for the domain in question (known as the *Spatial Knowledge module*). The result of the filtering process falls in a virtual pool of GIS learning objects that will act as an input space for the content selector (GIS functionality).

After the creation of the initial *pool* of the GIS learning objects, the content selection process and the underlined topology is applied based on learner characteristics, such as accessibility and competency characteristics or even historical information about related learning activities, included in the *Student Model* module. In the next section some filtering elements based on the Open GIS Consortium (`O-GIS.C`) Learning Object Metadata (LOM) standard are presented and the methodology is analyzed for the proposed content selection phase of a semi-automatic GIS course sequencing with spatial functionality.

3 GIS Learning Object Filtering

Generally, the main goal of a filtering process is the reduction of the searching space. GIS Learning Object Repositories (G.LOR) often contain hundreds of thousands of GIS learning objects, thus the selection process may require a significant computational time and effort. In most intelligent learning systems, learning object filtering is based, either on the knowledge domain they cover, or on the media type characteristics they contain [11].

In the Open GIS Consortium’s (`O-GIS.C`) spatial metadata model, there exist a number of elements covering requirements such as GIS functionality, topology, geometry, subject, teaching language, media type of the targeted GIS learning object, etc. Figure 2 presents the `O-GIS.C` elements identified for each one of the above mentioned topology-based filtering categories and the conditions required.

Alternatively, filtering can be based on integration of the `O-GIS.C` metadata model elements and ontologies [19, 16]. Those approaches assume that both the domain model and the learning objects themselves use the same ontology [14] and limit the filtering only to knowledge domain filtering with GIS functionality.

Filters	O-GIS.C Function	Explanation	Usage
Surveying & Monitoring	Topographic and Land Survey	A keyword or phrase describing the topic of a GIS Learning Object	O-GIS.C/Classification/ Purpose = "Geometry, Topology"
	Hydrology, Marine Survey and Geodesy	The time and geography or region to which a GIS Learning Object applies.	O-GIS.C/Classification/ Purpose = "Sea Level, Datum"
	Soil & Geological Surveys	This category describes where a Learning Object falls within a particular Classification system.	O-GIS.C/Classification/ Purpose = "Land, Material"
Navigation	City Modelling & Biosphere	The primary human language/s used within a Learning Object.	O-GIS.C/Classification/ Purpose = "3-D Model"
	Archaeology & Historical Sites	The human language and culture used by the typical intended user of a GIS Learning Object	O-GIS.C/Classification/ Purpose = "History, Culture"
Networks & Utilities Maintenance (AM/FM)	Asset Management	Technical data type/s of all the components of a GIS Learning Object	O-GIS.C/Classification/ Purpose = "Profit, Money"
	Marketing Market	The Marketing GIS Learning Object.	O-GIS.C/Classification/ Purpose = "Profit, Sale"
	Oil & Gas Network	AM/FM Learning Objects	O-GIS.C/Classification/ Purpose = "Automated Mapping (AM)"
	Water Supply Network	The completion status or condition of a AM/FM GIS Learning Object	O-GIS.C/Lifecycle/Status ! = "Water"
	Electricity Network	Whether use of a GIS Learning Object requires some kind of power.	O-GIS.C/Classification/ Purpose = "Facilities Management (FM)"

Figure 2: Elements for GIS Learning Object Filtering

The GIS Learning Object Selection Procedure

Typically, the design of highly adaptive learning systems requires a huge set of rules, since dependencies between educational characteristics of learning objects and learners are rather complex. This complexity introduces several problems on the definition of the rules required [20, 4], namely:

Inconsistency, when two or more rules are conf1icting.

Conf1uence, when two or more rules are equivalent.

Insufficiency, when one or more rules required have not been defined.

The proposed methodology is based on an intelligent mechanism that tries to mimic an instructional designer's decision model on the selection of the GIS learning objects. For this purpose, a framework that attempts to construct a suitability function that maps GIS learning object characteristics over learner characteristics and vice versa is designed.

The main advantage of this method is that it requires less effort by the instructional designer, since instead of identifying a huge set of rules related to space and topology, only the designer's selection from a small set of GIS learning objects over a reference set of learners is needed. The machine learning technique will try then to discover the dependence between GIS learning object and learner characteristics that produce the same selection of GIS learning objects per learner as the instructional designer did [1, 13, 20].

The proposed methodology does not depend on the characteristics used for learning objects and learner modelling, thus can be used for extraction of even complex pedagogy-related dependencies. It is obvious that since characteristics/requirements like the domain are used for filtering, the dependencies produced are quite generic, depending only on the educational characteristics of the content and the cognitive characteristics of the learner GIS student [5, 2, 16].

Figure 2 presents a graphical representation of the proposed selection model extraction framework, which it is consisting of the following three main steps:

1) The 3-D Modelling and Selection of GIS Criteria

The selection methodology is generic, independent of the learning object and the learner characteristics used for the selection. In the proposed method experiment, GIS learning object characteristics was used derived from

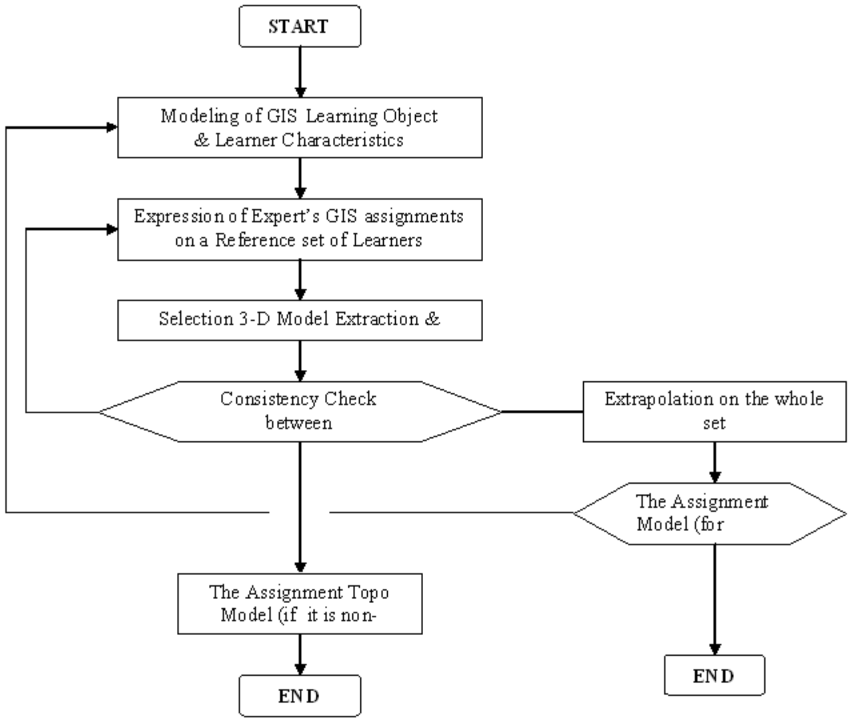


Figure 3: Selection Model Extraction Framework

the O-GIS.C standard, and learner characteristics derived from the IMS Global Learning Consortium Inc. Learner Information Package (LIP) specification.

There exist many criteria affecting the decision of GIS learning objects selection. Those criteria that lead to a straightforward exclusion of learning objects, such as the topology, the subject, the language and the media type and the GIS functionality are used for filtering. The rest set of criteria such as the educational characteristics of GIS learning objects are used for selection model extraction, since the dependencies of those criteria can model the pedagogy applied by the instructional designer, when selecting learning objects.

Those criteria, due to the complexity of interdependencies between them, are the ones that cannot be directly mapped to rules from the instructional designer. Thus an automatic or semi-automatic extraction method, like the proposed one, is needed.

2) The Selection Model Extraction

After identifying the set of the characteristics and the criteria (see: step 1) that will be used as the input space of the I/O selector, the extract procedure for each GIS learning object characteristic and the expert's suitability evaluation model over a reference set of LIP-based characterized learners is presented.

The input to this phase is the O-GIS.C characteristics of a reference set of learning objects, the IMS LIP characteristics of a reference set of learners and the suitability preference of an expert for each of the GIS learning objects over the whole reference set of learners.

The model extraction methodology has the following formulation:

Let us consider a set of learning objects, called A, which is valued by a set of criteria $g = (g_1, \ldots, g_n)$. The assessment model of the suitability of each GIS learning object for a specific learner, leads to the aggregation of all criteria into a unique criterion that we call a suitability function $S(g) = S(g_1, \ldots, g_n)$.

We define the suitability function as an additive function of the form

$$S(g) = \sum_{i=1}^{n} S_i(g_i)$$

with the following additional notation:

- $S_i(g_i)$: Marginal suitability of the ith selection criterion valued g_i,
- $S(g)$: Global suitability of a learning object.

The marginal suitability evaluation for the criterion **gi** is calculated using the formula

$$S_i(x) = a_i + b_i x \exp(-c_i x^2),$$

where x is the corresponding value of the **gi** GIS learning object selection criterion. This formula produces, according to parameters a, b and c as well as the value space of each criterion, the main criteria forms, we have identified:

- Monotonic form: when the marginal suitability of a criterion is a monotonic function.
- Non monotonic form: when the marginal suitability of a criterion is a non-monotonic function.

The calculation of the optimal values of parameters a, b and c for each selection criterion is the subject of the Knowledge Model Extraction step.

Let us call N the strict preference relation and E the indifference relation. If S_{O_1} is the global suitability of a learning object O_1 and S_{O_2} is the global suitability of a learning object O_2, then the fol1owing properties generally hold for the suitability function S:

$$S_{O_1} > S_{O_2} \iff (O_1)P(O_2),$$

and the relation $R = P \cup I$ is a week order relation

$$S_{O_1} = S_{O_2} \iff (O_1)I(O_2).$$

The expert's requested information then consists of the weak order R defined on A for several learner instances. Using the provided weak order relation R and based on the form definition of each learning object characteristic we can define the suitability differences $\Delta = (\Delta_1, \ldots, \Delta_{m-1})$, where m is the number of learning objects in the reference set A and $\Delta_k = S_{O_k} - S_{O_{k+1}} \geq 0$, depending on the suitability relation of (k) and $(k+1)$ preferred learning object for a specific learner of the reference set.

We can introduce an error function e for each suitability difference:

$$\Delta_k = S_{O_k} - S_{O_{k+1}} + e_k \geq 0.$$

Using constrained optimization techniques, we can then solve the non-linear problem:

$$\text{Minimize} \sum_{j=1}^{m-1} \left(e_j\right)^2$$

subject to the constraints:

$$\left.\begin{array}{lll} \Delta_j > 0 & \text{if} & O_j P O_{j+1} \\ \Delta_j = 0 & \text{if} & O_j I O_{j+1} \end{array}\right\}$$

for each one of the learners of the reference set.

This optimisation problem will lead to the calculation of the optimal values of the parameter a, b and c for each GIS learning object selection criteria over the reference set of learners.

3) The Extrapolation

The purpose of this third phase is to generalize the resulted marginal suitability model from the reference set of learners to all learners, by calculating the corresponding marginal suitability values for every combination of learner characteristics. This calculation is based on the interpolation of the marginal suitability values between the two closest instances of the reference set of learners.

Suppose that we have calculated the marginal suitability $S_i^{L_1}$ and $S_i^{L_2}$ of a criterion g_i matching the characteristics of learners L_1 and L_2 respectively. We can then calculate the corresponding marginal suitability value for another learner L using interpolation if the characteristics of learner L are mapped inside the polyhedron that the characteristics of learners L_1 and L_2 define, using the formula:

$$S_i(g_i^L) = S_i(g_i^{L_1}) + \frac{g_i^L - g_i^{L_1}}{g_i^{L_2} - g_i^{L_1}} \left[S_i(g_i^{L_2}) - S_i(g_i^{L_1}) \right], \text{ if } S_i(g_i^{L_2}) > S_i(g_i^{L_1})$$

Let $C_i = \lfloor c_i, c_i^{\star} \rfloor$, $i = 1, 2, \ldots, n$ be the intervals in which the values of each criterion — for both learning object and learners — are found, then we call global suitability surface the space $C = \times_{i=1}^{n} C_i$.

The calculation of the global suitability over the above mentioned space is the addition of the marginal suitability surfaces for each of the learning object characteristics over the whole combination set of learner characteristics.

Some Experimental Results and Discussion

In order to evaluate the total efficiency of the proposed methodology both ïn calculating the suitability on the training set of GIS learning objects and on estimating the suitability of GIS learning objects external from the reference set, we have designed an evaluation criterion, defined by:

$$\text{Success}(\%) = 100 \cdot \frac{\text{Correct Learning Objects Selected}}{n},$$

where n is the number of the desired GIS learning objects from the virtual *pool* that will act as input to the instructional planner.

It is assumed that the number of desired GIS learning objects is less than the total number of the GIS learning objects in the input space (the learning objects *pool*) and that both the GIS learning object metadata and the learner information metadata have normal distribution over the value space of each criterion.

Additionally, the GIS learning objects are classified, for both testing and estimation set, in two classes according to their functionality and aggregation level, since granularity is a parameter affecting the capability of an instructional designer to select learning content for a specific learner. The classification is based on the value space of the "*General/Aggregation Leíel*" element of the `O-GIS.C` standard.

In the rest of the paper experimental results are presented of the proposed methodology by comparing the resulting selected GIS learning objects with those selected by GIS experts. Also, the success of the proposed method has been evaluated on both the training set of learning objects (the *Training Success*) and on the suitability estimation of learning objects external from the reference set (*the Estimation Success*). So, Figure 4 present average experimental results for GIS learning objects with aggregation level 1 and 2 respectively.

If it is considered that for one learner instance, the different combinations of GIS learning objects, calculated as the multiplication of the value instances of characteristics with GIS functionality, lead to more than 1,000 GIS learning objects, it is evident that it is almost un-realistic to assume that an instructional designer can manually define the full set of selection rules which correspond to the dependencies extracted by the proposal method and at the same time to avoid the inconsistencies, conf1uence and insufficiency of the produced selection rules.

The proposed topology-based methodology is capable of effectively extracting dependencies between GIS learning object and learner characteristics affecting the decision of an instructional designer on the GIS learning object selection problem.

The analysis on the results, presented in Figure 3, shows that when the desired number of GIS learning objects (i.e. n) is relatively small (less than 100), the selected learning objects by the extracted decision model are almost similar to those the instructional designer would select. On the other hand, when the desired number of GIS learning objects is relatively large (say, about 500) the success of the selection is affected, but remains at acceptable level (i.e. about 90%).

Another parameter affecting the selection success is proved to be the granularity of GIS learning objects. Granularity mainly affects the capability of an instructional designer to express selection preferences over learning

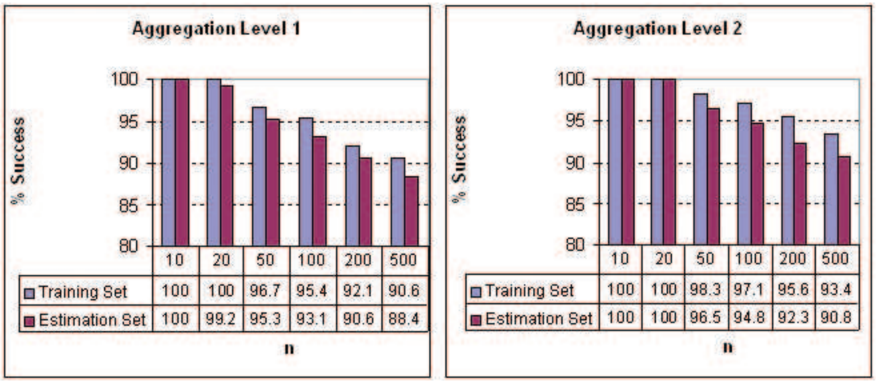


Figure 4: Average Experimental Results for GIS Learning Objects

objects. Learning objects with small aggregation level have bigger possibility of producing “gray” decision areas, where the instructional designer cannot decide which GIS learning object matches most the cognitive style or learning preferences of a learner [1, 5].

In paper’s experiments, GIS learning objects with aggregation level 2, which can be small or even bigger collections of GIS learning objects with aggregation level 1, appear to have less possibility of producing indifference relations, enabling to make secure decisions even for bigger desired number of learning objects ($n = 1,000$).

4 Summary and Conclusions

In this paper the GIS learning object selection problem is addressed as an intelligent topology-based GIS learning systems, by proposing a methodology that instead of “forcing” an instructional designer to manual1y define the set of selection rules; produces a decision model that mimics the way the designer decides, based on the observation of the designer’s reaction over a small-scale GIS learning object selection problem.

Hence, the proposed personalized e-learning method, describes a new methodology about selecting and sequencing learning objects with GIS functionality in an *instructional-sense* and *user-friendly* way.

Acknowledgements

The current paper is supported by the EPEAEK II - ARCHIMEDES I research project (Action 2.2.17), “Personalized e-Learning in a Reusable Way”, of the Alexandrium Technological Educational Institute of Thessaloniki (ATEI), Greece. This project is co-funded by the Greek Ministry of Education & Religious Affairs and the European Union.

References

[1] L. B. Baruque & R.N. Melo, “Learning Theory and Instructional Design using Learning Object”, in *Learning Objects 2003 Symposium: Lessons Learned, Questions Asked.* Honolulu, Hawaii, USA. Retrieved in September 2005 from the World Wide Web: `http://www.cs.kuleuven.ac.be/erikd/PRES/2003/LO2003/Baruque.pdf`

[2] P. Brusilovsky, “Adaptive and intelligent technologies for web-based education”, *Kunstliche Intelligenz Journal*, Vol. 4, pp.19-25, 1999.

[3] P. Brusilovsky and J. Vassileva, “Course Sequencing Techniques for Large-Scale Web-based Education", *International Journal of Continuing Engineering Education and Life-long Learning*, Vol.13 (1/2), pp. 75-94, 2003.

[4] L. Calvi & A. Cristea. “Towards Generic Adaptive Systems: Analysis of a Case Study”, *In Proc, of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, Spain, 2002.

[5] K. Chitwood, et al., “Battle Stories from the Field: Wisconsin Online Resource Centre Learning Objects Project", in *The Instructional Use of Learning Objects: Online Version.* D.A. Wiley, ed., 2000. Retrieved in September 2005 from the World Wide Web: `http://reusability.org/read/chapters/chitwood.doc`

[6] P. Dolog, W. Nejdl, “Challenges and Benefits of the Semantic Web for User Modelling”, *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, In Proc. of the 12th International World Wide Web Conference*, Budapest, Hungary, 2003

[7] P. Dolog, N. Henze, W. Nejdl & M. Sintek, “Personalization in Distributed eLearning Environments”, *In Proc. of the 13th International World Wide Web Conference*, New York, USA, 2004

[8] EPEAEK II - Common European Union and Greek Government Project (2001-2006). World Wide Web page: `http://www.epeaek.gr/epeaek/en/home.html`

[9] IEEE, “Draft Standard for Learning Object Metadata”, IEEE 1484.12.1-2002, 2002. Retrieved in September 2005 from the World Wide Web: `http://ieeeltsc.org/wg12LOM/1484.12.1`

[10] IMS – Global Learning Consortium. QTI Lite Specification. IMS Question & Test Interoperability, 2002. Retrieved in September 2005 from the World Wide Web: `http://www.imsglobal.org/question/qtilite03.html`

[11] Kinshuk, R. Oppermann, A. Patel & A. Kashihara, “Multiple Representation Approach in Multimedia based Intelligent Educational Systems”, *Artificial Intelligence in Education Journal*, Amsterdam: IOS Press. pp. 259-266, 1999.

[12] LSAL, SCORM Best Practices Guide for Content Developers, 2003. Cernegie Mellon Learning Systems Architecture Lab. Retrieved in September 2005 from the World Wide Web: *http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide*

[13] G. McCalla, “The Fragmentation of Culture, Learning, Teaching and Technology: Implications for the Artificial Intelligence in Education Research Agenda in 2010”, *International Journal of Artificial Intelligence in Education*, Vol. 11, pp. 177-196, 2000.

[14] P. Mohan, J. Greer & G. McGalla, “Instructional Planning with Learning Objects”, *Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems. 2003. In Proc. Of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.

[15] P. R. Polsani, “Use and Abuse of Reusable Learning Objects”. *Journal of Digital information*, (2003). Retrieved in September 2005 from the World Wide Web: `http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani.`

[16] A. Styliadis, K. Pehlivanis & D. Zahariou, “Personalized e-Learning for GIS Lecturing”. Hawaii International Conference on Education. Honolulu, Hawaii, USA, January 6th-9th, 2006.

[17] A. Styliadis, K. Pehlivanis & D. Zahariou, “E-Learning with Re-Usable GIS Functionality”. Global Universities In Distance Education (GUIDE) 2006 – *Developing a Common Platform for Global Co-Operation*. Rome IT, February 13th-14th, 2006.

[18] A. Styliadis & K. Pehlivanis, “Personalized e-Learning in a Re-Usable Way: A Proposed GIS System Design – The Architecture”. *International Journal on Engineering and Applied Sciences (JEAS)*, 2005.

[19] M. S. Urban & E. G. Barriocanal, “On the Integration of IEEE-LOM Metadata Instances and Ontologies”, *Learning Technology Newsletter*, Vol. 5 (1), 2003.

[20] H. Wu & P. De Bra, “Sufficient Conditions for Well-behaved Adaptive Hypermedia Systems” *In Proc. of the First Asia-Pacific Conference on Web Intelligence: Research and Development*, Maebashi City, Japan, 2001.

Athanasios D. Styliadis, Ioannis D. Karamitsos, Dimitrios I. Zachariou
The ATEI of Thessaloniki, Greece
EPEAEK II, The Arximedes 2-2-17 Project
E-mail: styl@it.teithe.gr

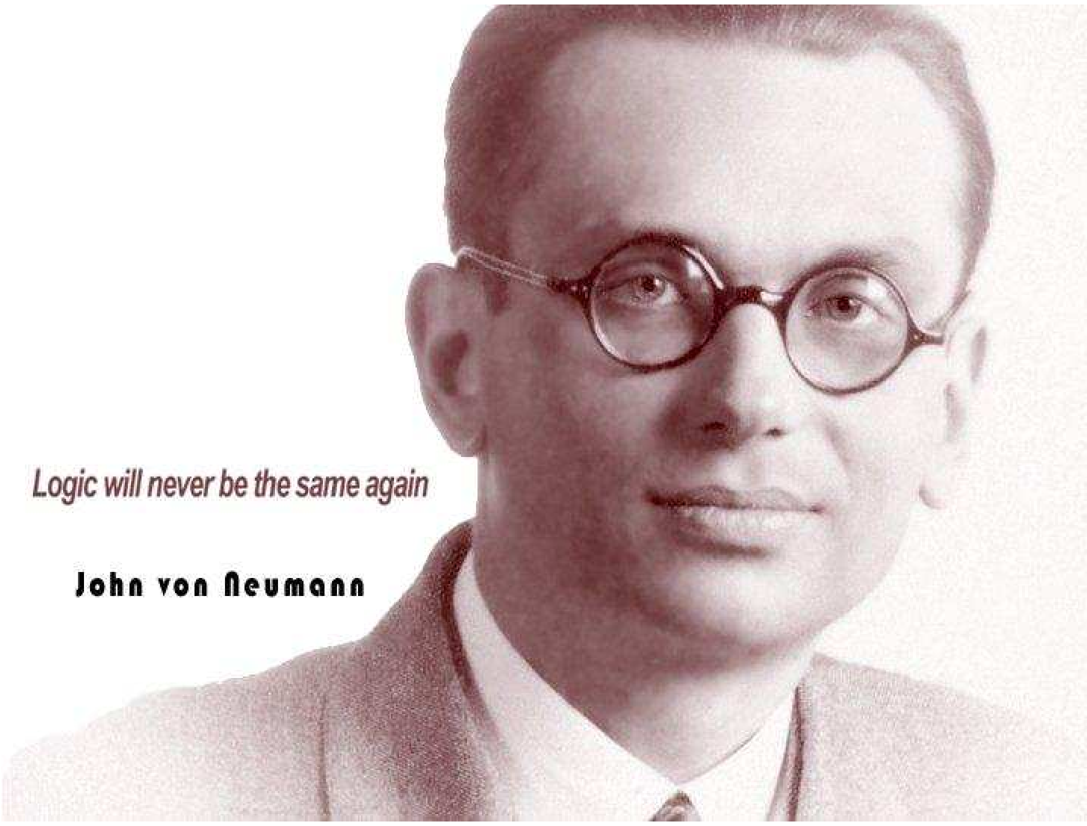
Editor's note about the author:

Athanasios D. Styliadis is a Professor at the Department of Information Technology at the Alexander Institute of Technology (ATEI) of Thessaloniki, Greece. He was born in 1956 in Florina, Greece and he received a Diploma in Surveying Engineering (Aristotle University of Thessaloniki, Greece, 1980), an M.Sc. in Computer Science (Dundee University, Scotland, 1987) and a Ph.D. in CAD/GIS/Computer Modeling (Aristotle University of Thessaloniki, Greece, 1997). He was a Fellow Research Scholar at the Department of Geomatics, University of Melbourne and at the Center for GIS and Modeling (CGISM), Australia. Prof. Styliadis is the writer of three books in Computer Graphics, Human-Computer Interaction (HCI) Programming and GIS-Geomatics Engineering, and he has over 70 journal and conference proceedings publications. Also, has been worked at the Hellenic Army Geographical Service (HAGS, Athens) for three years as GIS system analyst and programmer. His current research interests include: e-learning & geographic information, computer modeling, web-based CAAD, temporal GIS systems, concurrent engineering & 3-D geometry, and digital documentation of monuments and sites.

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 69-71

“Logic will never be the same again” - Kurt Gödel Centenary

Gabriel Ciobanu



Kurt Gödel (1906 - 1978)

Kurt Gödel was born on 28th of April 1906 in Brno, which was at that time a city of the Austrian-Hungarian Monarchy. This year we celebrate the 100th birthday of Kurt Gödel, perhaps the greatest logician of the twentieth century. We mark this event by a short article on his life and scientific results.

Kurt Gödel enrolled at the University of Vienna in 1923, and his scientific evolution is strongly related to the cultural environment of Vienna. As a student, Gödel became interested in mathematical logic, the field to which he made his major contributions. During that period, under the strong influence of David Hilbert, the mathematicians were concerned about the consistency and completeness of the formal systems used in mathematics. Hilbert has noticed the strong relationship between logic and mathematics; he argued in favor of an axiomatic approach, believing that all of mathematics can be formulated based on a logical foundation, and each result in mathematics follows from a system of axioms. Moreover, such an axiom system can be proved to be consistent. A formal system S is consistent whenever no contradiction is provable from S, and it is complete whenever every sentence A is decided by S in the sense that either S proves A, or S proves A. If neither A nor A is provable in S, then A is undecidable by S, and S is said to be incomplete.

In his doctoral dissertation, Gödel proved the completeness of first-order predicate logic, that states that any logically valid formula is provable, i.e., any sentence that holds in every model of the first-order predicate logic is derivable in the logic. The completeness theorem is an important property of first-order predicate logic. It does not hold for all logics; for instance, second-order predicate logic does not have a completeness theorem. This result was published in 1930, in “Die Vollstandigkeit der Axiome des logischen Funktionenkalkuls”, Monatshefte f.Math. 37, p.349-360. In 1930 Gödel became a member of the University of Vienna (where he remains until 1938, the year when Austria became part of nazi Germany).

In 1931, at only 25 years old, Gödel proved his most famous results, the incompleteness theorems. The first incompleteness theorem is perhaps the most surprising result in mathematical logic. It states that for any consistent formal theory that proves basic arithmetical truths, it is possible to construct an arithmetical statement that is true, but not provable in the theory. That is, any consistent theory of a certain expressive strength cannot prove everything which is true, i.e. such theories are necessarily incomplete. The second incompleteness theorem states that for any formal theory S including basic arithmetical truths and certain truths about formal provability, S includes a statement of its own consistency if and only if S is inconsistent. These results showed that Hilbert's program is impossible. They were first announced by Gödel at a meeting in 1930 at Koenigsberg, where also Hilbert and von Neumann were present, and it was published in 1931 as "Uber formal unentscheidbare Satze der Principia Mathematica und verwandter Systeme", in I.Monatsh.Math.Phys. 38, p.173-198. This paper is elegantly organized and clearly presented, progressing efficiently with no wasted energy devoted to collateral aspects.

Godel's incompleteness results were simply unexpected, and their proofs, though involving new techniques, are not very difficult. Their relevance to mathematical logic and in the theory of computation is preeminent and dominant. Their relevance to philosophy is important, even though it is not clear yet in what way and how the things will evolve. In the philosophy of mind, Lucas-Penrose arguments that the human mind does not work on mechanical principles in mathematics use the incompleteness theorem. The second incompleteness theorem has prompted theologians and postmodernists to reflect why mathematics cannot prove its own consistency. Gödel's results has also interesting interpretations in the language of computer science. On the other hand, the impact of Gödel's incompleteness results among working mathematicians is not impressive. The mathematicians, although generally aware of the theoretical possibility that a problem they are working on may be unsolvable in the current axiomatic framework of mathematics, do not have difficulties in using their field results in order to find a solution. In mathematics, Gödel's incompleteness results show that in many cases, such as in number theory or real analysis, it is not possible to create a complete and consistent finite list of axioms, or even an infinite list that can be produced by a computer program. Each time you add a statement as an axiom, there are other true statements which still cannot be proved as true, even with the new axiom. Furthermore if the system can prove that it is consistent, then it is inconsistent. In his book "From Here to Infinity", Ian Stewart expresses this by saying that "Gödel showed that if anyone finds a proof that arithmetic is consistent, then it isn't!".

In 1935 Gödel established the relative consistency of the axiom of choice, and in 1938 that of the generalized continuum hypothesis; these results are published in "The consistency of the axiom of choice and of the generalized continuum-hypothesis", Proc.Nat.Acad.Sci.USA 24, p.556-557, 1938.

In 1940 Gödel took a position at the Institute for Advanced Study in Princeton. At Princeton University Press he published "Consistency of the axiom of choice and of the generalized continuum-hypothesis with the axioms of set theory" where he introduced the constructible universe, a model of the set theory in which the only sets which exist are those which can be constructed from simpler sets. In such a constructible universe both the axiom of choice and the generalized continuum hypothesis are true, and therefore the model should be consistent.

During his many years at Princeton, Gödel's interests turned to philosophy and physics. At Princeton Godel had a legendary friendship with Einstein. Regarding their walks to and from the Institute for Advanced Studies, Einstein said that "his own work no longer meant much; he came to the Institute merely to have the privilege of walking home with Gödel". Gödel demonstrated the existence of some paradoxical solutions to Einstein's field equations in general relativity. Gödel's "rotating universes" allow time travel, and caused Einstein to have doubts about his own theory.

In the early 1970s Gödel wrote an ontological proof of God's existence.

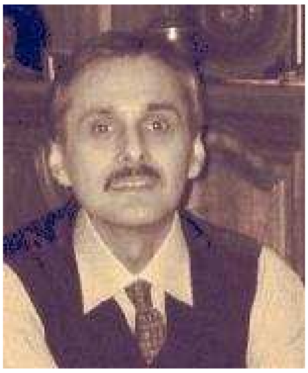
Due to his psychological disorder, in late 1977 Gödel refused to eat, and thus he died of starvation in January 1978.

An extensive biography of Gödel can be found at the MacTutor History of Mathematics archive at http://www-gap.dcs.st-and.ac.uk/ history/Mathematicians/ Gödel.html, which also provides a list of papers. "Kurt Gödel Collected Works" were published in five volumes at Oxford University Press (S. Feferman et al., editors).

An international symposium celebrating the 100th birthday of Kurt Godel was organized between 27 and 29 April 2006 by the Kurt Gödel Society and University of Vienna. This symposium commemorated the life, work, and foundational views of Kurt Gödel, exploring also the current research and ideas in the fields of the logic, mathematics, and computer science. The symposium has attracted important scientific personalities: P.Cohen (Fields Medal), S.Feferman (main editor of "Collected Works"), H.Putnam (Harvard), H.Woodin (Berkeley), G.Kreisel (FRS), D.Scott (Turing Award), A.Wigderson (Nevanlinna Prize), C.Papadimitriou (Knuth Prize), J.D.Barrow (Kelvin Medal), R.Penrose (Wolf Prize). The talks presented various aspects related to logic, mathematics, computer science, artificial intelligence, cosmology, philosophy, and theology. There were a lot of discussions, including difficult questions, debatable answers, and polemics. According to a selection process, two submissions of Romanian authors were accepted to be presented at this symposium: G.Ciobanu (Iasi) with a paper regarding a new characterization of computable real numbers, and L.Leustean (Darmstadt) with a paper on "proof mining", a technique of getting new knowledge from proofs in mathematics. The volumes devoted to this symposium will be published in the series Collegium Logicum of the Kurt Gödel Society, and by Cambridge University Press.

Gabriel Ciobanu
http://thor.info.uaic.ro/ gabriel
E-mail: gabriel@iit.tuiasi.ro

Editor's note about the author:



Gabriel Ciobanu
Institute of Computer Science
Romanian Academy, Iaşi

Born: 5 July 1957, Piatra Neamţ. Dr. Gabriel Ciobanu has wide ranging interests in computing including Distributed Systems, Theory of Programming and Computational Aspects in Biology (molecular interaction, membrane computing). He has edited/authored 6 books and over 100 papers on these topics. He enjoy developing new ideas, both the practical ones and the more abstract ones, drawing on both his computer science and mathematical background. Over the years he has received public recognition for his research, including the 2004 "Octav Mayer" Award, and the 2000 "Grigore Moisil" Award of the Romanian Academy of Sciences. He was a visiting academic to Edinburgh University, Paris XI, Tohoku University, and National University of Singapore, among others. He is a member of some professional associations and societies (EATCS, EAPLS, AMS, ISCB), and he have been in the program committees for many international conferences.

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 73-79

Grigore C. Moisil: A Life Becoming a Myth

Solomon Marcus

"All what is correct thinking is
either mathematics or feasible to be
transposed in a mathematical model."
Grigore C. Moisil
(1906 - 1973)

Grigore C. Mosil, Romanian mathematician, Professor at the University of Bucharest - Faculty of Mathematics and Computer Science, member of the Romanian Academy, member of the Academy of Sciences in Bologna and of the International Institute of Philosophy, Computer Pioneer Award of IEEE Computer Society (IEEE - 1996).

Grigore C. Moisil belongs to the fifth generation of Romanian mathematicians. The first generation includes the founders Spiru Haret and David Emmanuel, both with doctorates at Sorbonne (Paris). They were born at the middle of the XIXth century and have the merit to initiate the high level mathematical research in Romania. (We leave aside Transylvania, with Janos Bolyai, who already in the first half of the XIXth century discovered non-Euclidean geometry). The second generation includes the first Romanian mathematicians with a long career of scientific research: Gh. Ţiţeica, D. Pompeiu (both with doctorates at Sorbonne), Al. Myller and Vera Myller (both with doctorates at Gottingen, Germany). They had already an important impact on Moisil's formation as a mathematician.

The third generation includes Victor Valcovici, Traian Lalescu and Simion Stoilow (considered as the most important Romanian mathematician until the sixties of the XXth century), born in the eighties of the XIXth century. Lalescu died very young (in 1929), but Valcovici and Stoilow became great professors not only before, but also after the second world war. We have then a fourth generation of mathematicians born in the last decade of the XIXth century: Octav Onicescu (the initiator of the Romanian school of Probability theory), Petre Sergescu (the organizer of the first two congresses of Romanian mathematicians, before the second world war), Dan Barbilian (the same as the poet Ion Barbu), Alexandru Froda and Gheorghe Vranceanu (the greatest Romanian geometer after Ţiţeica).

Directly or indirectly, all these mathematicians had an impact on Moisil's personality. Born in Tulcea (Dobrogea), with a long genealogic tree of high intellectuals in the North-Western part(Bistriţa-Năsăud) of Romania, with school training partly in Bucharest, partly in Iasi (Moldova), Moisil's childhood and adolescence are well known from his own diary published in the recent years by Moisil's widow Viorica Moisil, who took great care of the whole scientific and human heritage of her husband.

The main teachers of the child Moisil were his parents; his father was an important historian, while his mother was just an educator for elementary school. The child Grigore enjoyed to look around, to give free expression to his curiosity and wonder, to ask questions, to read books of a large diversity, from science to literature and from practical jobs to philosophy. He felt always the need to react to what he was seeing, listening and reading. His diary is an excellent mirror of this fact. His main pleasure was always of an intellectual nature, he was not attracted to play with other children or to practice various sport games. His mother taught him to count and to make calculations and only in a second step to read and to write. The most important part of the learning process took place at his home, with his parents and some time with his brothers and his sister, all of them becoming intellectuals. Moisil's diary should be known by the children of the new generations.

The attitude of the child Moisil towards learning remained his attitude towards life. He enjoyed to interact with people, to read, to write and to learn. He did all these things with pleasure and with humor, he was able to contaminate people with his capacity to discover something new where most people see nothing new; to invent questions where most people believe that everything was already answered. His attraction for mathematics did not diminish his interest for the other school disciplines. His curiosity was total and remained total during his whole life. But his passion for mathematics and his way to understand mathematics lead him to see the world through the glasses of mathematics; so mathematics was not for him only a profession, a job, it was a way to look at the life and at the universe.

When Moisil had to become a university student, choosing math was equivalent for most people with choosing engineering, so his parents pressed him to choose Polytechnical School, where he became a student in 1924; but one year before he became also a student of the Faculty of Science, Mathematical Section. He did not like engineering, so he never finished Polytechnical School, but the irony of life decided that Moisil had to return to engineering much later, when he discovered that logic and engineering share some very important features. On the other hand, the university student Moisil attended many classes of philosophy, of history and of art. His teachers were Ţiţeica, Lalescu, Davidoglu and others, but the university teacher who impressed him the most and became his spiritual model was Dimitrie Pompeiu. His PhD thesis (1929) was about the analytic mechanics of continuous systems.

As a characteristic feature of his works published in the twenties and the thirties of the past century, we observe that most of them were at the interference of differential equations (with partial derivatives), differential geometry, function theory and mechanics. The main reason of this situation was that Moisil was devoted to the topics most considered by his predecessors and his colleagues (Pompeiu, with his areolar derivative, Vrânceanu, with his neolonomic varieties, Nicolae Teodorescu with his papers on Finsler spaces etc.), but he also paid attention to the work of some great Western mathematicians of his time: Eli Cartan, W. Blaschke, and mainly Vito Volterra and Jacques Hadamard. Having the opportunity to work with Volterra (1931-1932) in Rome, Moisil became a pioneer of the functional methods in differential geometry and mechanics (let us recall that Volterra is one of the initiators of Functional Analysis). But before this, Moisil was in Paris, working with great French mathematicians such as Hadamard (1930-1931); he spent a second period in Paris after returninf from Rome. So, in this atmosphere Moisil obtained a generalization of Volterra's conjugate functions and he also generalized Hadamard's total geodesic varieties. From the same period let us recall the joint paper by Moisil and Teodorescu on holomorphic functions in the space (1931), the first joint work in Romanian mathematics. Joint works remained a rare phenomenon until the middle of the past century.

At Nevember 1, 1932, Moisil is named a provisional associate professor of Algebra at the University of Iasi. At that moment, Moisil was already the author of an important number of scientific papers, but no of them was in the field of Algebra. His first algebraic paper (in the field of non-associative algebra) was published only in 1934. So, this fact came as a surprise. But for a good observer, nothing surprising was in this fact. Behind the diversity of topics, Moisil was, in most of his mathematical papers, mainly an algebraist, his genuine thinking was almost always of an algebraic nature. In everything he did, he projected an algebraic spirit. Within the framework of his traditional preoccupations of analysis, geometry and mechanics (monogeneity, function theory, geodesics in some Riemann spaces, partial differential equations etc.), he works with algebraic tools such as functional groups, parametric groups, monogenic quaternions, hypercomplex numbers, ideals of polynomials, areolar polynomials, polynomials associated to some bilinear differential expressions with constant coefficients, interpretation of the fundamental group of a differential variety etc.

To the above fact, we have to add an event having an important role in Moisil's life: the publication, in the first part of the fourth decade of the past century, of the book of Van der Waerden, "Moderne Algebra". It was an important sign of the move of algebra from the quantitative to the qualitative, from the algorithmic to the structural phase of its evolution. It was a turning moment in Moisil's mathematical life, an event having a huge impact on Moisil's further evolution. In this order of ideas, we should observe that Moisil is the first to introduce Bourbaki's ideas in Romania, at a moment when Bourbaki's mathematical structuralism, strongly influenced by the German school of structural algebra, was only a project. In his course of general analysis (Analiza Genarala) published towards the end of the thirties of the past century, Moisil gives an account of Henri Cartan's theory of filters, while uniform spaces were also presented there. But Moisil was not the first to teach in Iaşi modern structural algebra; he was preceded in this respect by Vera Myller. On the other hand, Emmanuel and Lalescu had also, at some moments, the opportunity to teach some ideas on groups and fields. However, it seems that Moisil was the first to give to this structural style the whole amplitude. Only Barbilian will go further in this respect, but this will happen in the second half of the fourties of the past century.

Moisil became full associate professor of Algebra at January 1, 1935, but at November 1, 1936 he got a position of Professor of Differential and Integral Calculus and then full Professor of Calculus in November 1939, all of them at Iaşi University.

Moisil spent ten years at Iaşi University (1931-1941). It was a period in which he alternated his old interests in continuous mathematics, with applications to mechanics and physics, with his new interests in discrete mathematics, mainly in algebra and logic. Moisil was very impressed by an article due to the Polish logician Jan Lukasiewicz, on the logic with three values (then the interest moved to several values) and by the analogy proposed by Lukasiewicz between the non-classical logics, on the one hand, and the non-Euclidean geometries, on the other hand. As the initiator of the logic of three values, Lukasiewicz was considered as the Lobatchevski of logic. It was one more reason for Moisil to be attracted by logic: his philosophical interests lead him to pay great attention to the philosophical consequences of quantum mechanics, where Aristotle logic is no longer valid. Indeed, the principle of excluded middle is here replced by a principle of included middle. In the thirties of the past century, important authors, such as John von Neumann, paid attention to the logic of quantum mechanics. It was also the principle of universal determinism which was under question; certainty is replaced sometimes by probability. Moisil and some of his colleagues (Onicescu, Procopiu, Barbilian etc.) organized some debates on this topic and reading his writings in this respect one can understand how this interest motivated him to orient his attention towards mathematical logic, a field where he published his first paper in 1936, in a volume devoted to the 75th birthday of Vitto Volterra. He will never leave this field, but concomitantly he will remain also stable in his old interests. Only in the last decade of his life he will be devoted exclusively to discrete mathematics. But, being first of all an algebraist, Moisil will project in his studies in logic the same algebraic spirit. His main project was to build for Lukasiewicz's logic of several values an algebraic framework in a way similar to the way George Boole has proposed in the XIXth century an algebraic model for Aristotle's logic (based on the principles of identity, non-contradiction and excluded middle). Moisil called this framework "Lukasiewicz algebras", but ultimately these algebras received the more appropriate name "Lukasiewicz-Moisil algebras"; this name became the title of a monograph published at North Holland Publishing House by a team lead by Professor Sergiu Rudeanu.

His move from Iaşi to Bucharest University, at the end of the year 1941. was a dramatic one. Not only Moisil, but also Vrânceanu, Barbilian and Miron Nicolescu were competing for the same position of professor. Among them, Moisil was the youngest and with the smallest chance to win. The winner was Vrânceanu. Then, Moisil had the happy idea to convince the ministry of education, Ion Petrovici, to create three different chairs and so all of them became professors; it was the great chance for the next generation (that of the author of these lines) to benefit of such great professors.

After the second world war, Moisil had many interesting initiatives. One of them was related to an idea proposed by Shannon in his PhD thesis and independently by some Russian engineers, to associate electric circuits with binary logic, because each of them works with two values: yes or no in logic, while the circuit is open or closed. Moisil succeeded to develop this idea in many variants, stimulating a whole team of researchers to articulate engineering, classical and non-classical logics and various types of algebraic structures with some ideas from number theory. Another fruitful idea was to associate some matrices to some systems of linear partial differential equations.

But perhaps more important was the way Moisil understood, in the early fifties, that the emergence of the new paradigm of information, communication and computation could change to a large extent the social, cultural and scientific life of the next decades. In 1949, he initiates a whole school in the field of the algebraic theory of automatic mechanisms As a professor of the Bucharest University, he was the first to teach there mathematical logic. Articulating logic and automata, Moisil was well prepared to organize the Romanian development in the emergent field of Computer Science. He monitorized the building of he first Romanian computer, by Victor Toma, at the Institute of Atomic Physics, in 1957, and, on the other hand, he organized courses in the field of computation at the Faculty of Mathematics, University of Bucharest. He also directed the first promotion of students in Mathematics to work with the team of Victor Toma, at the Institute of Atomic Physics; they were trained to learn programming at the new computers CIFA (Calculatorul Institutului de Fizica Atomica). The first Romanian team of mathematicians included Dragos Vaida, I..Moldovanu, Gh. Zamfirescu, G. Klarsfeld. So, we can say that 1957 is the date of birth of Romanian Computer Science, under the guidance of Professor Moisil and by the collaboration between engineers and mathematicians.

In 1962, Moisil initiates a new section "Computing Machines" at the Faculty of Mathematics and Physics of the Univ of Bucharest and, associated with this section, the Computing Center of the Univ of Bucharest (CCUB); as a matter of fact, CCUB was under the guidance of the Chair of Algebra, whose chief was Moisil. In 1963, CCUB is endowed with the computer CIFA 3, the third version of the first Romanian electronic computer, and with an analogic computer of MEDA type. Moisil is very active in preparing the corresponding mathematical background: learning of ALGOL 60, organization of a seminar of algebraic theory of automatic mechanisms (started in 1954), organization of a course of logic applied to electric circuits and of a seminar of mathematical logic (started in 1966); numerical analysis and combinatorics are also stimulated. In a further step, mathematical biology and mathematical linguistics as well as perspectives of computation in various fields of the humanities: history, archeology, musical composition etc. In 1968, CCUB is endowed with a computer IBM 360/30 of the third generation and the learning of FORTRAN and COBOL is introduced. CCUB became a place where people from all cultural horizons came to learn from Professor Moisil how could they take profit in using in their own field the mathematical and computational thinking. Lawyers and musicians (among them, Aurel Stroe), engineers and economists, linguists and philosophers, biologists and medical doctors, painters and writers were visiting CCUB and the main reason of this fact was the presence there of Professor Moisil, who had the gift to leave the mathematical jargon and to address non-mathematicians in the simplest possible language; and in this simple language he was able to explain the mathematical and computational way of thinking.

The capacity of Moisil to seduce and to fascinate the auditory became very fast an element of attraction for mass media. Newspapers, radio and television began to invite him and Moisil became a star whose fame was in competition with that of the most popular singers and actors. His unique voice, his way to transform the speech in a song, his spontaneity, his humor, his permanent state of joy made Moisil so popular, that even today, 33 years after his death in May 1973, his statements are still in the attention of the public. "New ideas appear first as paradoxes, then they become common truth and ultimately they die as prejudices"; "Is logic a practical science ? Yes, because you learn from it how to take decisions" ; "You lose a lot of time when you believe that you know what in fact you don't know". Many of his jokes have a mathematical structure. Here is a joke illustrating the recursive thinking: "Every man has right to a glass of wine; but when you drink a glass of wine, you become another man" (the corollary: every man has right to infinitely many glasses of wine). A joke illustrating self-reference: a child asked him: "Professor Moisil, do you like dreams ?" – "Yes, I had once a dream in which I was sleeping during a session and when I waked up I was really in a session". Jokes having the same pattern, being so able to be produced algorithmically: "The water is bad, even in the shoes"; "everything can be proved; even the truth"; "you can fall in love with any women; even with your wife"; "every joke can make you to be in the best mood; even the above ones". All these jokes have as a common denominator the confusion between normality and exceptionality: the water in the shoes, the proof of a truth; to be in love with your wife. As soon as you understood the pattern, you can produce infinitely many jokes of similar type.

At 11 february 1971, Moisil sends a letter to the Rector of the Bucharest University, proposing a whole program of organization of education in the field of computers and their mathematics. In January 1973, he sends another letter to the dean of the Faculty of Mathematics, where he explains that this Faculty has a great responsibility concerning the formation of the teachers of computer science with a solid mathematical background. Moisil explains that the main job in this respect is to assure the computational literacy of the coming generations, because in the emergent period of the information and computation all professions will need in some way familiarity with computing and programming. Very few people were aware at that moment of this truth which today is obvious. In this respect, we can consider Moisil as a kind of Spiru Haret of the second half of the past century: Haret was an important fighter against illiteracy, while Moisil was an important fighter against computational illiteracy.

Due to his multiple interests, Moisil succeeded to form a lot of disciples in various directions: in mechanics of solids (Nicolae Cristescu, P.P.Teodorescu, M. Predeleanu, George Dinca), in logic (Sergiu Rudeanu, George Georgescu, Afrodita Iorgulescu), in computer science (Dragoş Vaida, Constantin Popovici, Paul Constantinescu), in logic of electric circuits (L. Livovschi), in algebra, in analysis, in differential geometry.

If the child Moisil revealed a total curiosity, the same totality characterizes the creative work of the adult Moisil. Within mathematics, he interrelates all its domains; beyond mathematics, he is looking for the way mathematics may have an impact on natural and social sciences; beyond science, he is questioning the relation between math and philosophy, between math and art; beyond culture, he is interested in the impact of math in the everyday life.

Scholars are of two types: the ant type, looking for what happens in a specific area of knowledge and trying to deepen more and more the respective segment of investigation; but there is also the bee type, going from flower to flower and changing frequently the area of investigation. Obviously, Moisil was of the second type.

But, looking with more attention at his behavior, we realize that he was sometimes of a mixed type, because he liked to go back to flowers already visited. I remember his renewed interest in the sixties in the problem of mechanics he discussed long time ago in his PhD thesis. After the second world war, his growing interest in discrete mathematics was concomitant with the continuation of his work in continuous mathematics. His general strategy was to trust the unity of mathematics and the potential solidarity between its different parts, including the case when these parts seem to be completely away each other. For this reason, he used to oblige his PhD students to pass examinations on some chapters of math which apparently were very far from the object of direct interest of the respective student. This is the reason why most of his papers combine different branches of math.

Another interesting feature of Moisil's works is their strong link with the works of his colleagues and of his professors. A typical example are his papers of geometry. Arrived in Iaşi at the end of the year 1931, when he was only 26, but with his thesis published by Gauthier-Villars (Paris, 1929), Moisil found an adequate atmosphere in the Seminar lead by Al. Myller, predominantly concerning differential geometry. At that moment, he was considered, in view of his already published works, the founder of the theory of infinite dimensional Riemann spaces. To this, he added the study of infinite Lie groups and of mechanical systems with infinitely many degrees of freedom. But the geometric methods used by Moisil were for him a tool to investigate the mechanics of systems of material points with infinitely many degrees of freedom. For him, the respective geometric model consists of infinite dimensional subvarieties in an adequate Hilbert space. In other situations, he is oriented towards the geometrization of systems of equations with partial derivatives. So, analysis, geometry, mechanics form an organic mathematical entity. See, for more, in this respect, the article by Acad. Radu Miron in "Academica" (forthcoming, 2006), from which we have borrowed some elements.

In order to illustrate the style of work done by Moisil in the field of math and humanities, we will indicate some ideas he developed in the field of what he called "the mechanical grammar of Romanian". He proposed some new classifications of Romanian nouns and verbs. Two ideas deserve to be mentioned. The first one concerns the possibility to use, in the declension of nouns and conjugation of verbs, of what he calls the method of variable letters, by means of which he copes, in a very elegant way, with the phenomenon called in linguistics "morphological alternances". The variable letters are like the functions defined by means of two or several analytic expressions, each of them for a specific part of the domain of definition of the function. Another idea proposed by Moisil concerns the conjugation of verbs, where a classification is made according to the behavior of what is called "the more than the perfect" (mai mult ca perfectul). Another paper concerns a comparative analysis between the linguistic conjunction "and" and the mathematical conjunction "and", in the case of Romanian language, but things remain valid, to a large extent, for other languages too. For instance, it is shown the contrast between the possibility to iterate indefinitely the use of "and" in logic, and the impossibility of doing the same in linguistics.

To his own works in this respect, Moisil added his capcity to stimulate and to guide the first steps in the development of mathematical and of computational linguistics in Romania. He guided the first algorithm of automatic translation (English-Romanian) by Erica Nistor, the similar work by Minerva Bocsa in Timisoara, the work done by the team lead by P. Schveiger in Cluj-Napoca and the work done by Eliza Roman in the field of automatic abstracts and automatic documentation. The author of this lines remains indebted to Moisil for his major help jointly with the linguist Alexandru Rosetti, concerning the first steps of mathematical and computational linguistics in Romania. Moisil and Rosetti made possible the organization of the first university courses with this profile; they founded "Cahiers de Linguistique Theorique et Appliquee", a journal of an interdisciplinary nature, devoted to the interferences among linguistics, mathematics, computer science and poetics.

The remarkable fact, in these articles, is the capacity of Moisil to develop the mathematical way of thinking in absence of the usual mathematical jargon, consisting of formulas, equations, calculations etc. He never leaves in these texts the natural language.

Another aspect of Moisil's personality can be seen in his philosophical writings. Mathematics and philosophy were for him two faces of the same coin, each of them requiring the other. Already during his childhood and adolescence, his interrogative nature and his readings prepared the way towards his philosophical personality. He reads Poincare and selects in his diary statements such as : "Science deserves to be studied for the glory of human spirit" and "for the enormous pleasure offered by the knowledge of truth", more than "for its practical utility". Somewhere he notes: "Life is a work of art. It is a pleasure to think to what happened sometimes in the past and will never happen again !". In another place he notes: "Pleasure is more attractive when you are looking for it than when you feel it". A lot of remarks related to his readings in the field of history, of literature, of natural sciences, of religion etc. All of them, when he was 7, 8, 9, 10, 11, 12, 13, 14. Here is a comment about the cause of wars: "If we cancel the fights having no rational motivation, only a few of wars remain in the memory of the history". A word of wisdom: "You don't have to exagerate in love, because you risk to end by exagerating in hate".

The articles in the field of philosophy deserve a special attention. The first one, chronologically, was published in 1937 and concerns the successive steps in the development of the mathematical knowledge. Moisil was at that time under the strong influence of the ideas of Vito Volterra, who pays attention to the way mathematical knowledge is born from the pre-mathematical knowledge. In a first step, qualitative descriptions are converted in quantitative ones, by means of measurements and counting. Then the mathematics of quantity are developed, until the moment when qualitative aspects are again in attention. The development of physics and of mathematical analysis are followed concomitantly. The mechanical stage, the energy stage and the Einsteinian stage of the generalized relativity theory are analyzed. Then he directs his attention towards the development of qualitative mathematics and the main example is here the notion of a group, observed on the particular cases of the group of rotations and the group of permutations. We are lead in this way to what Moisil will call structural mathematics and the way is prepared to connect it to the general emergence of structuralism (in psychology, in linguistics, in anthropology and before them in chemistry, with the idea of isomerism). Moisil will be among the first to observe the big change brought by the new fields of topology, functional analysis, combinatorics, graph theory, mathematical logic, abstract algebra. In the same way, Moisil stresses the importance of the theory of complex numbers and of their structural aspects. Moisil analyzes the way classical infinitesimal calculus lead to general topology. The idea of a differential is both quantitative and structural (having the structure of a polynomial). Logistics is a clear example of reduction of quantity to structure and this is the task of Russell's and Whitehead's "Principia Mathematica". Starting from the contrast observed by other authors between the quantitative aspect of mathematics and the qualitative aspect of the acts of thinking, Moisil observes that the acts of thinking have a structural rather than a qualitative aspect and so the mathematician's job is to investigate the algebraic structure of the acts of thinking. So, concludes Moisil, "ce n'est pas a l'ancienne logique de la qualite qu'on devra s'adresser, mais a la nouvelle algebre de la structure. Ce n'est pas en effet trop tot si on essaie de construire une theorie coherente de la vie spirituelle".

In his "La logique formelle et son probleme actuel" (1939), Moisil investigates the principles of classical logic (identity, contradiction, excluded middle) and their modifications in Brouwer-Heyting's intuitionistic logic, in Kolmogorov's intuitionistic logic, in Lukasiewicz's ternary logic; he also discusses the very nature of axiomatic deductive systems, with special attention to Hilbert, Russell and Poincare. This investigation is continued in "Sur l'autonomie des mathematiques" (1941), where he is using the term "panmathematisme" understood as the process of approaching mathematically the natural sciences and those of the human spirit (sciences de l'esprit)". In this order of ideas, he characterizes the mathematical activity as being irreducible, i.e., independent of empirical investigations and of any previous rational development. This is for him the autonomy of mathematics. Moisil meets, in this respect, the way old Greeks (Pythagoras, Platon), then Kant and Goethe, conceived mathematics (for Kant, mathematics is what is called in German Geisteswissenschaft). The utilitarian function of mathematics is in most cases a consequence of its cognitive function, but the temporal distance between the cognitive moment and the utilitarian one is usually imprevisible. We stress this fact, because in his writings after 1950, in view of the ideological constraints, he will no longer state explicitly the autonomy of mathematics, but he will defend strongly the need to develop pure mathematics, mathematics for its own sake, as a condition to reach applied mathematics. Moisil considers that the human spirit can reach the center of a deductive discipline; mathematics can bridge the self and the non-self. He is consistent with the idea emergent much later, according to which the subject-object distinction will no longer be considered as sharp as in the classical science; consequently, the distinction between natural and human sciences is under question. As a matter of fact, mathematics is for Moisil by excellence a human discipline.

The last philosophical article he published before the communist regime was his "Closure lecture at the University of Iaşi" (16 January 1942): "The perspectives of axiomatic philosophy". We learn from this lecture that Romanian scientists paid a great attention to the philosophical problems of their science, mainly those related to non-Euclidean geometries and to relativity and to quantum mechanics. Philosophers like Ion Petrovici, physicists like St. Procopiu, mathematicians like Victor Valcovici, Simion Stoilow, Octav Onicescu, Dan Barbilian and Gr. C. Moisil were involved in hot discussions on these topics. Ultimately, Moisil stresses the human side of mathematics. "The axiomatic freedom of mathematics does not fit with something similar in the real life" observes Moisil and this premonitory statement is followed by other similar statements: "Human creative work is by excellence one of expression. The deep human desire is to be understood by other peoples. This expression which succeeds to be communicated is just what we call culture". Will this communication remain possible ? Such thoughts expressed in the hot year 1942, when the war knew dramatic changes and the future became increasingly unsure are just the thoughts of Moisil at the moment when his stage in Iasi ends and a new university life will begin in Bucharest.

At some moment, Moisil makes reference to "Goedel paradox". Clearly, he has in view the famous 1931 Goedel incompleteness theorem, but it is clear that this theorem had to wait until it will be understood in its deep meaning and huge consequences for the whole mathematics. This was not only Moisil's shortcoming, but a shortcoming of the quasi totality of the mathematical community. During the summer of the year 1942, Stoilow and Barbilian had a daily correspondence about foundations of mathematics, but the reference to Goedel 1931 theorem did not exist in their dialogue.

Sometimes, Moisil is very near to some ideas emergent much later. For instance, in "Determinism si inlantuire" (1940), he refers to Georges Bouligand in connection with the fact that small changes in the initial data may have a big impact on the further development of a phenomenon. "Throwing a small stone may have some influence on the movement of the sun", observes Moisil. Does not anticipate it the modern "butterfly effect"? In another place, he refers to the involvement of continuous nowhere differentiable functions in the study of Brownian motion. So, chaotic systems and fractals are suggested to a contemporary reader. Let us recall that several decades later Brownian motion will be recognized as a strange fractal (whose Hausdorff dimension is an integer).

In one respect only, Moisil failed. He did not succeed to organize his life in order to make it more efficient. Moreover, he did not try this. Ho died at 67, when his head was still full of ideas and projects. He did not know how to alternate work and rest, how to pay attention to his health.

I remember my last meeting with him.It was in an evening of the spring of the year 1973. We left together the University and we were walking in the direction of his home, Armeneasca street, 14. When we arrived at his house, he told me: "You know that Sudan is the real author of the first recursive function that is not primitive recursive ?" (All treatises of mathematical logic claim that the author of such an example is G. Ackermann). Very interesting, I said. Where did you find this ? Then, Moisil said: "It is too late now, I will tell you this at a next occasion". This "next occasion" never arrived. Moisil left next days for Canada, where he died on 21 May 1973. Today is May 18. In three days we can say: Moisil died just 33 years ago. But my curiosity to find out what is behind the mysterious message left to me by Moisil obliged me to accomplish the respective research. Together with two young students, Cristian Calude and Ionel Tevy, we took piece by piece all the published papers of Gabriel Sudan. No of them had in its title or in its introduction something suggesting the presence of such an example. It is perhaps a rule of the nature that more interesting is something, more hidden it is and more effort we need to discover it. This was the reality. The respective example was hidden in the last part of an article which was explicitly concerned with a problem having nothing to do with a recursive function which is not primitive recursive. Clearly, Sudan was not aware of the fact, like Moliere's hero, Jourdain, who remained a symbol of such situations.

I had the happy opportunity to edit a part of the scientific work of Moisil, in three volumes published by the Publishing House of the Romanian Academy. Other writings of Moisil were published also after his death, some of them under the care of Viorica Moisil. I edited his "Lectii depre logica rationamentului nuantat" and his articles in the newspaper "Contemporanul", under the label "Stiinta si umanism". The articles published in "Viata economica" were edited in a small book "Indoieli si certitudini". Now, we have in front of us the duty to publish his papers of "mechanical grammar of Romanian" and his philosophical papers.

Moisil's heritage belongs to the Romanian culture and the new generations deserve to know this unusual personality.

Solomon Marcus
Romanian Academy
E-mail: Solomon.Marcus@imar.ro

Editor's note about the author:

Solomon Marcus
Member of the
Romanian Academy

Born: March 1, 1925, Bacău, Romania. Elementary and high school in Bacau. Diploma of merit in Mathematics, University of Bucharest. Asistant Professor(1950), Lecturer, Associate Professor, Professor (1966) at the Faculty of Mathematics and Computer Science, University of Bucharest. PhD in Mathematics 1956, Doctor in Science 1967, Corresponding member of the Romanian Academy 1993, Full Member of the Romanian Academy 2001. Research and Teaching in Mathematical Analysis, Theoretical Computer Science, Linguistics, Semiotics, Poetics, History and Philosophy of Science, fields where he published about 50 books in Romanian, English, French, German, Italian, Spanish,Russian, Greek, Hungarian, Czech, Serbo- Croatian and about 400 research articles in specialized journals in almost all European countries, in USA, Canada, South America, Japan, India, New Zealand etc. More than 1000 authors quoted his works. He is recognized as one of the initiators of mathematical linguistics and of mathematical poetics. Hundreds invited lectures at various international scientific meetings. Member of the editorial board of several tens of international scientific journals.

International Journal of Computers, Communications & Control
Vol. I (2006), No. 1, pp. 81-99

Grigore C. Moisil (1906 - 1973) and his School in Algebraic Logic

George Georgescu, Afrodita Iorgulescu, Sergiu Rudeanu

Abstract: We present in the paper a very concise but updated survey emphasizing the research done by Gr. C. Moisil and his school in algebraic logic.
Keywords: n-valued Lukasiewicz-Moisil algebra, θ-valued Lukasiewicz-Moisil algebra, Post algebra

The mathematical logic is one of the domain in which the creative spirit of Gr.C. Moisil manifested plenary. His work in logic stands out by the the novelty, the variety and the depth of treated subjects. His first works are connected to the top results of the time and wear an algebraic seal. The young professor from Jassy came after a rich experience in mechanics and differential equations. Van der Waerden treatise of algebra has decisively influenced his entry in logic by the algebraic gate. In the same time, these works have a powerful philosophical imprint.

From this vast creation, the contributions in multiple-valued logics represent the part with the most intense impact on today researches.

The first system of multiple-valued logic was introduced by J. Łukasiewicz in 1920. Independently, E. Post introduced in 1921 a different multiple-valued logic. For Łukasiewicz, the motivation was of philosophical nature - he was looking for an interpretation of the concepts of possibility and necessity - while for Post, the research was intended as a natural mathematical generalization of bivalent logic.

In 1930, Łukasiewicz and Tarski studied a logic whose truth values are the real numbers from the interval $[0,1]$.

1 Łukasiewicz-Moisil algebras

In 1940, Gr. C. Moisil has defined the 3**-valued** and the 4**-valued Łukasiewicz algebras** and in 1942, the n-**valued Łukasiewicz algebras** ($n \geq 2$). His goal was to algebrize Łukasiewicz's logic. Boolean algebras, algebraic models of classical logic, are particular cases of that new structures.

In the description of a logical system, the implication was traditionally the principal connector. The n-valent system of Łukasiewicz had as truth values the elements of the set

$$L_n = \left\{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, \frac{n-2}{n-1}, 1\right\}$$

and was built around a new concept of implication, on which are based the definitions of the other connectors.

For Moisil, the basic structure is that of lattice, to which he adds a negation (getting the so called "De Morgan algebra") and also some unary operations (called by Moisil "chryssipian endomorphisms"), representing the "nuancing". The Łukasiewicz implication was let on a secondary plane and, in the case of an arbitrary valence, was completely lost.

Further axiomatizations were suggested by A. Monteiro, R. Cignoli, C. Sicoe, S. Rudeanu and others.

An example of A. Rose from 1956 established that for $n \geq 5$ the Łukasiewicz implication can no more be defined on a Łukasiewicz algebra. Consequently, only for $n = 3$ and $n = 4$ the structures introduced by Moisil are models for Łukasiewicz logic. The lost of implication has lead to another type of logic, called today "Moisil logic", distinct from Łukasiewicz system; the logic corresponding to n-valued Łukasiewicz-Moisil algebras was created by Moisil in 1964. The fundamental concept of Moisil logic is the **nuancing**.

Nowadays we feel it appropiate to call these algebras **Łukasiewicz-Moisil algebras** or **LM algebras** for short.

For complete information and references on Łukasiewicz-Moisil algebras see [25]

The work of Moisil on LM algebras covers two periods of time: a first period, during 1940-1942, when he introduces the n-valued LM algebras with negation and studies special classes of these structures, as **centered** and **axed** LM algebras and a second one, during 1954-1973, when he introduces the θ-valued LM algebras without negation, applies multiple-valued logics to swiching theory and study algebraic properties of LM algebras (representation, ideals, reziduation).

Moisil's works traced research directions for many Romanian and foreign mathematicians. In Argentina, at Bahia Blanca, Antonio Monteiro and his school (Roberto Cignoli, Luiz Monteiro, Luiza Iturrioz, Maurice Abad etc.) have contributed decisively to consolidate LM agebras as a domain of algebra of logic and to disseminate them in the mathematical world.

In his PhD thesis from 1969 [29], R. Cignoli makes a very deep study of n-valued Moisil algebras (the name he first gives to the n-valued Łukasiewicz algebras introduced by Moisil).

1.1 n-valued Łukasiewicz-Moisil algebras

The structure called "De Morgan algebra" was first studied by Moisil; the name was given by Antonio Monteiro [142]; a duplicate name is "quasi-Boolean algebra" given by A. Bialynicki-Birula and H. Rasiowa.

Definition 1.1. *A* **De Morgan algebra** *is a structure*

$$(A, \vee, \wedge, ^{-}, 0, 1)$$

such that $(A, \vee, \wedge, 0, 1)$ *is a distributive lattice with 0 and 1 and the unary operation* $^{-}$*, called negation, verifies:*

(DM0) $1 = 0^{-}$,
(DM1) $(x^{-})^{-} = x$,
(DM2) $(x \wedge y)^{-} = x^{-} \vee y^{-}$.

Remark 1.2. *In a De Morgan algebra we also have:*
(DM3) $(x \vee y)^{-} = x^{-} \wedge y^{-}$.

Definition 1.3. *Let* $J = \{1, 2, \ldots, n-1\}$.
An n**-valued Łukasiewicz-Moisil algebra** $(n \geq 2)$ *or an* LM_n *bf algebra for short is an algebra*

$$\mathcal{A} = (A, \vee, \wedge, ^{-}, (r_j)_{j \in J}, 0, 1)$$

of type $(2, 2, 1, (1)_{j \in J}, 0, 0)$ *such that:*

(i) $(A, \vee, \wedge, ^{-}, 0, 1)$ *is a De Morgan algebra.*

(ii) the unary operations $r_1, r_2, \ldots, r_{n-1}$ *fulfil the following axioms: for every* $x, y \in A$ *and every* $i, j \in J$,

(L1) $r_j(x \vee y) = r_j x \vee r_j y$,

(L2) $r_j x \vee (r_j x)^{-} = 1$,

(L3) $r_j \circ r_i = r_i$,

(L4) $r_j(x^{-}) = (r_{n-j} x)^{-}$,

(L5) $r_1 x \leq r_2 x \leq \cdots \leq r_{n-1} x$,

(L6) if $r_j x = r_j y$ *for every* $j \in J$*, then* $x = y$*; this is* **the determination principle.**

If $\mathcal{A}$ *fulfils (i) and only (L1)–(L5) we shall say that* $\mathcal{A}$ *is an* LM_n **pre-algebra***.*

Proposition 1.4. In every LM_n algebra $\mathcal{A}$, the following properties are verified: for every $x, y \in A$ and every $j \in J$,
(L7) $r_j(x \wedge y) = r_j x \wedge r_j y$;
(L8) $r_j x \wedge (r_j x)^{-} = 0$;
(L9) $x \leq y$ *if and only if* ($r_j x \leq r_j y$*, for every* $j \in J$);
(L10) $r_1 x \leq x \leq r_{n-1} x$;
(L11) $r_j 0 = 0$, $r_j 1 = 1$;
(L12) *Let* $C(A)$ *be the set of complemented elements of* A*, i.e.*

$$C(A) = \{x \in A \mid \exists x' \in A,\ x \vee x' = 1,\ x \wedge x' = 0\}.$$

Let K_j be the set of all elements of A left invariant by r_j, $j \in J$, i.e.

$$K_j = \{x \in A \mid r_j x = x\}.$$

Then:
(i) $r_j x \in C(A)$, for every $j \in J$, $x \in A$ and
(ii) $C(A) = K_j$, for every $j \in J$;
(L12') $(C(A), \vee, \wedge, ^-, 0, 1)$ is a Boolean algebra, where $x^- = x'$;
(L12") If $z \in C(A)$, then for every $x \in A$:
$x \wedge z = 0 \Longleftrightarrow x \leq z^-$,
$z \vee x = 1 \Longleftrightarrow z^- \leq x$;
(L13) $x^- \vee r_{n-1}x = 1$;
(L14) $x \wedge (r_{n-1}x)^- = 0$.

Example 1.5. The algebra

$$\mathcal{L}_n = \mathcal{L}_n^{(LM_n)} = (L_n, \vee, \wedge, ^-, (r_j)_{j\in J}, 0, 1),$$

where

$$L_n = \left\{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, \frac{n-2}{n-1}, 1\right\}$$

and

$$\begin{cases} x \vee y = \max(x,y), \quad x \wedge y = \min(x,y), \quad x^- = 1 - x, \\ r_j\left(\frac{i}{n-1}\right) = \begin{cases} 0, & \text{if } j+i < n, \\ 1, & \text{if } j+i \geq n, \end{cases} \quad i \in \{0\} \cup J,\ j \in J, \end{cases}$$

is an LM_n algebra, that we shall call **the canonical LM_n algebra**.
The proper subalgebras of $\mathcal{L}_n$ have the form:

$$S = L_n - \bigcup_{x \in L_n - \{0\}} \{x, x^-\}.$$

They are LM_n algebras.
The smallest subalgebra of $\mathcal{L}_n$ (with respect to $\subseteq$) is $C(L_n) = \{0, 1\}$, which is also a Boolean algebra, cf. (L12').
For instance, the subalgebras of
- L_3 are L_2 and L_3,
- L_4 are L_2 and L_4 and
- L_5 are L_2, L_3, $\{0, 1/4, 3/4, 1\}$ and L_5.

Remark 1.6. *LM_2 algebras coincide with Boolean algebras.*

Proposition 1.7. In every LM_n pre-algebra, the determination principle (L6) is equivalent to each of the following conditions: for every $x, y \in L$,

(a) $x \wedge (r_j x)^- \wedge r_{j+1} y \leq y$, *for every* $j \in J - \{n-1\}$;

(b) $x \wedge \bigwedge_{j=1}^{n-1} ((r_j x)^- \vee r_j y) \leq y$.

þ (Representation theorem of Moisil)
Every LM_n algebra can be embedded in a direct product of copies of the canonical LM_n algebra $\mathcal{L}_n$.

Corollary 1.1. Every LM_n algebra is a subdirect product of subalgebras of the canonical LM_n algebra $\mathcal{L}_n$.

***

In 1968, Gr. C.Moisil introduced the **θ-valued Łukasiewicz algebras or LM$_\theta$ algebras for short (without negation)**, where θ is the order type of a chain with first and last element. The concept of θ-valued Łukasiewicz algebra is obtained from that of n-valued, on the one hand, by dropping the negation $^-$ and on the other hand, by replacing the set L_n by a totally ordered set I with first and least elements and by adapting the axioms to this case; the Determination Principle is preserved. These structures were thought by Moisil as models of a logic with an infinity of nuances. According to a confession done by Moisil, he imagined LM$_\theta$ algebras (without negation) long time ago, but the care of finding a strong motivation for them delayed the announcement; the motivation was found when Moisil met Zadeh's fuzzy set theory, in which he saw a confirmation of his old ideas.

In 1969, Marek and Traczyk [110] introduced the notion of **generalized Łukasewicz algebra (with negation)**, in an attempt to generalize to the infinite case the LM_n algebras; but their generalization is not a natural one.

In his PhD thesis from 1972 [64], G. Georgescu studied duality theory for Moisil's LM_θ algebras (without negation), the injective objects (and their characterization), monadic and poliadic algebras.

In his PhD thesis from 1981 [53], A. Filipoiu studied the LM_θ algebras (without negation) and their associated logic. He gives a representation theorem for LM_θ algebras by aids of θ-valent Moisil field.

In his Master thesis from 1981 [12] also, L. Beznea studies a generalization of LM_θ algebras (without negation) obtained by eliminating the Determination Principle.

Later on, in his PhD theses from 1984 [21], V. Boicescu introduced and studied the ***n*-valued LM algebras without negation**, as a particular case of Moisil's LM_θ algebras (without negation).

Following the inverse way, A. Iorgulescu, in her PhD thesis from 1984 [90] also, introduced and studied a natural generalization of Moisil's LM_n algebras to the infinite case, called **θ-valued LM algebras with negation** or **LM_θ algebras with negation** for short; any LM_θ algebra with negation is a Moisil's LM_θ algebra without negation.

2 Connection with logic

Gr. C. Moisil invented LM algebras in order to create an algebraic structure playing the same role with respect to the multiple-valued logic as Boolean algebras play with respect to classical, bivalent logic. However, as shown by the example of A. Rose, this only happens for the cases $n = 3$ and $n = 4$.

The algebraic structures adequate to the infinite-valued logic of Łukasiewicz (truth valued in the real interval $[0,1]$) are the MV-algebras introduced by C.C. Chang in 1958 or, equivalently, the Wajsberg algebras introduced by Font, Rodriguez and Torrens in 1984; D. Mundici proved in in 1986 that MV algebras are categorically equivalent to lattice-ordered Abelian groups with strong unit.

R. Grigolia's MV_n algebras, introduced in 1977, and Cignoli's proper Łukasiewicz algebras, introduced in 1982, are algebraic structures corresponding to n-valued logic of Łukasiewicz.

The logic corresponding to LM_n algebras was created by Moisil himself in 1964. Łukasiewicz logic has implication as its primary connector, while Moisil logic is based on the idea of nuance, expressed algebraically by the Chrysippian endomorphisms. The “engine" of the latter logic is Moisil's Determination Principle, according to which an n-valued sentence is determined by its Boolean nuances. The Determination Principle realizes a transfer from the multiple-valued logic to the classical logic. This determination brings Moisil logic much closer to classical logic than Łukasiewicz logic. One could say that Moisil logic is **derived** from classical logic by the idea of nuancing. Algebraically, this tight relationship is expressed by the fundamental adjunction between the categories of Boolean and Łukasiewicz algebras.

V. Boicescu in 1971 and A. Filipoiu in 1981 introduced and studied logics appropiate to $LM\theta$ algebras without negation. (i.e. infinite-valued LM algebras).

A. Filipoiu generalized Smullyan's method of analytic tableaux to θ-valued logic without negation and studied the θ-valued predicate calculus as well, with applications to systems of recording and retrieval of information.

Łukasiewicz logic, Post logic and Moisil logic consitute the three directions in the classical theory of multiple-valued logic. Their corresponding algebraic models are MV algebras, Post algebras and LM algebras.

3 Connections with other structures of algebraic logic

Moisil introduced in 1941 the **centered** LM_3 algebras.

Post algebras (cf. P. Rosenbloom (1942), G. Epstein (1960), T. Traczyk (1963)) turn out to be centered LM algebras, cf. R. Cignoli (1969) and G. Georgescu - C. Vraciu (1969). The θ-valued Post algebras were studied by T. Traczyk (1967) and G. Georgescu (1971).

Gr. C. Moisil, R. Cignoli, L. Iturrioz, A. Monteiro and V. Boicescu studied LM algebras as particular cases of Heyting algebras. V. Boicescu also studied LM algebras as Stone algebras.

LM_3 algebras and LM_4 algebras are polynomially equivalent to MV_3 algebras and MV_4 algebras, respectively, since they are the algebraic counterpart of the 3-valued Łukasiewicz logic and the 4-valued Łukasiewicz logic, respectively. D. Mundici was first to point out the equivalence between LM_3 algebras and MV_3 algebras, in 1989. Then A. Iorgulescu, in 1998-2000 [91] - [94], pointed out the isomorphism between the categories of LM_k algebras and of MV_k algebras, for $k = 3,4$ and also studied the categories $\mathbf{LM}_n$ and $\mathbf{MV}_n$ for $n \geq 5$, showing that every MV_n can be made into an LM_n algebra. She then studied those LM_n algebras that can be viewed as MV_n algebras:

3.1 Connections between LM_n algebras and MV_n algebras

MV algebras were introduced by C.C. Chang, in 1958 [26]. A simplified list of axioms of MV algebras was given by Mangani [109], as follows:

Definition 3.1. *An MV* **algebra** *is an algebra*

$$\mathcal{A} = (A, \oplus, ^-, 0)$$

of type $(2,1,0)$, *where the following axioms are verified: for every* $x,y,z \in A$,
(MV1) $(A, \oplus, 0)$ *is an Abelian monoid,*
(MV2) $x \oplus 0^- = 0^-$,
(MV3) $(x^-)^- = x$,
(MV4) $(x^- \oplus y)^- \oplus y = (y^- \oplus x)^- \oplus x$,
where $x \cdot y = (x^- \oplus y^-)^-$.

Definition 3.2. *For any* $m \in \mathbb{N}$, *we have:*

(i) $0x = 0$ *and* $(m+1)x = mx \oplus x$,

(ii) $x^0 = 1$ *and* $x^{m+1} = x^m \cdot x$.

The MV_n algebras were introduced by Revaz Grigolia in 1977 [87], as follows.

Definition 3.3. *An* MV_n **algebra** $(n \geq 2)$ *is an MV algebra* $\mathcal{A} = (A, \oplus, ^-, 0)$, *whose operations fulfil the additional axioms:*
(M1) $(n-1)x \oplus x = (n-1)x$,
(M1') $x^{n-1} \cdot x = x^{n-1}$
and, if $n \geq 4$, *the axioms:*
(M2) $[(jx) \cdot (x^- \oplus [(j-1)x]^-)]^{n-1} = 0$,
(M2') $(n-1)[x^j \oplus (x^- \cdot [x^{j-1}]^-)] = 1$,
where $1 < j < n-1$ *and* j *does not divide* $n-1$.

Corollary 3.1. MV_2 algebras coincide with Boolean algebras.

Example 3.4. The MV algebra $\mathcal{L}_n = \mathcal{L}_n^{(MV_n)} = (L_n, \oplus, ^-, 0)$, where

$$L_n = \left\{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, \frac{n-2}{n-1}, 1\right\}$$

and for any $x,y \in L_n$:

$$x \oplus y = \min(1, x+y), \quad x \cdot y = \max(0, x+y-1), \quad x^- = 1-x$$

and

$$x \vee y = \max(x,y), \quad x \wedge y = \min(x,y),$$

is an MV_n algebra. We shall call it **the canonical** MV_n **algebra**.
Note that $B(L_n) = \{0,1\}$.
The subalgebras of $\mathcal{L}_n$ are of the form:

$$S_m = \left\{0, \frac{K}{n-1}, \ldots, \frac{(m-2)K}{n-1}, 1\right\},$$

where $K = \frac{n-1}{m-1}$, if $m-1$ divides $n-1$.

The subalgebras S_m of $\mathcal{L}_n$ are isomorphic to $L_m = \left\{0, \frac{1}{m-1}, \ldots, \frac{m-2}{m-1}, 1\right\}$, if $m-1$ divides $n-1$, and they are MV_n algebras.

Hence $\mathcal{L}_m = (L_m, \oplus, \cdot, ^-, 0, 1)$ $(m \leq n)$ is an MV_n algebra if and only if $m-1$ divides $n-1$.

For instance, the subalgebras of:

- L_3 are L_2 and L_3,
- L_4 are L_2 and L_4 and
- L_5 are L_2, L_3 and L_5.

þ

Every MV_n algebra is a subdirect product of subalgebras of the canonical MV_n algebra $\mathcal{L}_n$.

D. Mundici was the first to prove in 1989 [152] that MV_3 algebras coincide with LM_3 algebras.

A. Iorgulescu has proved in 1998-2000 [91] - [94] that:

1 - MV_4 algebras coincide with LM_4 algebras,
2 - the canonical MV_n algebra coincides with the canonical LM_n algebra ($n \geq 2$),
3 - for $n \geq 5$, any MV_n algebra is a LM_n algebra,
4 - those LM_n algebras which are MV_n algebras, for every $n \geq 5$, are exactly Cignoli's proper n-valued Łukasiewicz algebras.

Here are for short the results 1-3:

***

To obtain the transformation of an MV_n algebra into an LM_n algebra, for any $n \geq 3$, Iorgulescu used Suchoń's transformation [174]:

Suchoń defines Moisil operators $(\sigma_j)_{j\in J}$ ($\sigma_j = r_{n-j}$) of the canonical LM_n algebra $(n \geq 3)$ starting from the Łukasiewiczian implication $\rightarrow$ and from the negation $^-$. He puts

$$B_3(x) = (x^-) \rightarrow x \text{ and } B_{j+1}(x) = (x^-) \rightarrow B_j(x),\ j \geq 3. \tag{1}$$

Then he defines:

$$\sigma_1 x = B_n(x) \tag{2}$$

$$\text{and for } 1 < j \leq [n/2], \qquad \sigma_j x = \begin{cases} \sigma_{n-1}(B_{l+1}(x)),\ lj \geq n-1 \\ \sigma_{lj}(B_{l+1}(x)),\ lj < n-1, \end{cases} \tag{3}$$

where $l = \max\{m \mid m(j-1) < n-1\}$,

$$\text{while } \sigma_{n-j}(x) = (\sigma_j(x^-))^-, \text{ for } \quad 1 \leq j \leq [n/2]. \tag{4}$$

Suchoń's Moisil operators verify: $\quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{n-1}$.

Remark 3.5. *If we want to use Suchoń's construction, it is convenient to consider not the MV algebra $(A, \oplus, ^-, 0)$, but the Wajsberg algebra, $(A, \rightarrow, ^-, 1)$, introduced by J. M. Font, A. J. Rodriguez and A. Torrens in 1984; MV algebras and Wajsberg algebras are isomorphic structures:*

- *if $\mathcal{A} = (A, \rightarrow, ^-, 1)$ is a Wajsberg algebra and if we define*

$$\alpha(\mathcal{A}) = (A, \oplus, ^-, 0)$$

by

$$x \oplus y = x^- \rightarrow y, \quad 0 = 1^-, \tag{5}$$

then $\alpha(\mathcal{A})$ is an MV algebra.

- *Conversely, if $\mathcal{A} = (A, \oplus, ^-, 0)$ is an MV algebra and if we define*

$$\beta(\mathcal{A}) = (A, \rightarrow, ^-, 1)$$

by

$$x \rightarrow y = x^- \oplus y, \tag{6}$$
$$1 = 0^-,$$

then $\beta(\mathcal{A})$ is a Wajsberg algebra.

- *The maps α, β are mutually inverse.*

It follows immediately by (1) that

$$B_3(x) = x \oplus x = 2x \quad \text{and} \quad B_{j+1}(x) = x \oplus B_j(x) = jx,\ j \geq 3. \tag{7}$$

By using Suchoń's construction, Iorgulescu then gave the following

Definition 3.6. *Let $\mathcal{A} = (A, \oplus, ^-, 0)$ be an MV_n algebra ($n \geq 3$). Define*

$$\Phi^S(\mathcal{A}) = (A, \vee, \wedge, ^-, (r_j)_{j\in J}, 0, 1)$$

by

$$x \vee y = x \cdot y^- \oplus y, \qquad x \wedge y = (x^- \vee y^-)^-,$$

$$r_{n-1}x = (n-1)x, \tag{8}$$

$$r_{n-j}x = \begin{cases} r_1(lx), & lj \geq n-1 \\ r_{n-lj}(lx), & lj < n-1, \end{cases} \tag{9}$$

for $1 < j \leq [n/2], \qquad l = max\{m \mid m(j-1) < n-1\}$,

$$r_j x = (r_{n-j}(x^-))^-, \qquad 1 \leq j \leq [n/2]. \tag{10}$$

Proposition 3.7. If $\mathcal{L}_n$ is the canonical MV_n algebra ($n \geq 3$), then $\Phi^S(\mathcal{L}_n)$ is the canonical LM_n algebra.

þ If $\mathcal{A}$ is an MV_n algebra ($n \geq 3$), then $\Phi^S(\mathcal{A})$ is an LM_n algebra.

***

Proposition 3.8. 1) Given the canonical LM_n algebra ($n \geq 3$)

$$\mathcal{L}_n = (L_n, \vee, \wedge, ^-, (r_j)_{j\in J}, 0, 1),$$

$$\text{define} \quad \Psi(\mathcal{L}_n) = (L_n, \oplus^n, ^-, 0) \quad \text{by :}$$

if $n = 2k+1$,

$$\begin{aligned} x \oplus^{2k+1} y \quad = \quad & (x \vee r_{2k}y) \wedge (y \vee r_{2k}x) \\ \wedge \quad & (x^* \vee r_{2k-1}y) \wedge (y^* \vee r_{2k-1}x) \\ & \vdots \\ \wedge \quad & (x^{(k-1)*} \vee r_{k+1}y) \wedge (y^{(k-1)*} \vee r_{k+1}x), \end{aligned} \tag{11}$$

if $n = 2k$,

$$\begin{aligned} x \oplus^{2k} y \quad = \quad & (x \vee r_{2k-1}y) \wedge (y \vee r_{2k-1}x) \\ \wedge \quad & (x^* \vee r_{2k-2}y) \wedge (y^* \vee r_{2k-2}x) \\ & \vdots \\ \wedge \quad & (x^{(k-1)*} \vee r_k y) \wedge (y^{(k-1)*} \vee r_k x), \end{aligned} \tag{12}$$

where x^* is the **successor** of x and

$$x^{2*} = (x^*)^* \quad , \quad x^{m*} = \left(x^{(m-1)*}\right)^* \tag{13}$$

Then $\Psi(\mathcal{L}_n)$ is the canonical MV_n algebra.

2) The maps Φ^S, from Proposition 3.7, and Ψ are mutually inverse.

Since for $n = 3$, in the canonical LM_3 algebra $\mathcal{L}_3$, the operation $\oplus$ is:

$$x \oplus y = (x \vee r_2 y) \wedge (y \vee r_2 x)$$

and for $n = 4$, in the canonical LM_4 algebra $\mathcal{L}_4$, the operation $\oplus$ is defined by:

$$x \oplus y = (x \vee r_3 y) \wedge (y \vee r_3 x) \wedge (x^* \vee r_2 y) \wedge (y^* \vee r_2 x) =$$

$$= (x \vee r_3 y) \wedge (y \vee r_3 x) \wedge (x^- \vee y^- \vee r_2 x \vee r_2 y),$$

it follows that the transformation Ψ is not polynomial for $n \geq 5$).

Those LM_n algebras which are MV_n algebras (i.e. for which the transformation Ψ is defined), for every $n \geq 5$, are exactly Cignoli's proper n-valued Łukasiewicz algebras [34], but the proof is very technical [94].

4 Representation theorems

Numerous representation theorems have been given for LM algebras. The first is due to Moisil himself and is reminiscent of the representation theorem for Boolean algebras: every LM_n algebra can be embadded into a Cartesian power of L_n.

In a modern vision [25], every LM algebra is a subdirect product of subalgebras of the algebra $In(I, L_2)$ of incresing functions from I to L_2. In particular every LM_n algebra is a subdirect product of subalgebras of $\mathcal{L}_n$ (Cignoli). $\mathcal{L}$ is a direct product of subalgebras of $\mathcal{L}_n$ if and only if it is complete and atomic (Boicescu 1984).

The representation by continuous functions studied by Cignoli, Boicescu and Filipoiu, means that for every LM_θ algebra without negation $\mathcal{L}$ there is a unique Boolean space X such that $\mathcal{L}$ is isomorphic to the algebra of all continuous functions $f : X \to In(I, L_2)$, where $In(I, L_2)$ is endowed with the topology having as basis the principal ideals and the principal filters generated by the characteristic functions of the sets $\{k \mid k > \alpha\}$, $\alpha \in \theta$.

The representation of LM_θ algebras without or with negation by Moisil fields of sets is due to Filipoiu. The Stone duality was extended from Boolean algebras to LM_θ algebras without or with negation with the aid of a suitable concept called LM_θ-valued Stone space (Cignoli, Georgescu, Iorgulescu), while the Priestley duality is based on a suitable adaptation of the concept of Priestley space (Filipoiu).

The representation of LM algebras as algebras of fuzzy sets was studied by D. Ponasse, J.L. Coulon and J. Coulon, S. Ribeyre and S. Rudeanu.

The representation of LM_n algebras by LM_3 algebras is legitimated by the "good" properties of the latter and was studied by A. Monteiro, L. Monteiro, F. Coppola, V. Boicescu and A. Iorgulescu.

5 Categorial aspects

The Stone and Priestley dualities are in fact equivalences of categories. Other categorial properties of LM algebras were studied. Here are a few samples.

The association of L with the Boolean alegebra $C(L)$ of complemented elements of L is extended to a functor $C : \mathbf{LM}\theta \to \mathbf{B}$, while the association of a Boolean algebra B with the algebra $In(I, B)$ is extended to a functor $T : \mathbf{B} \to \mathbf{LM}\theta$. Then C and T are adjoint functors, C is faithful and T is fully faithful. This yields in particular the representation theorem of Moisil. The construction of the functors C and T was given by Moisil himself.

The injective and projective objects have also been studied, for instance, an LM_θ algebra is injective if and only if it is a complete Post algebra (whose center is a complete Boolean algebra), cf. L. Monteiro, R. Cignoli, G. Georgescu and C. Vraciu, V. Boicescu.

6 Ideals and congruences

The study of the appropiate ideal and congruence theory for LM algebras was undertaken by Gr. C. Moisil, A. Monteiro, R. Cignoli, C. Sicoe. V. Boicescu introduced the concepts of θ-ideal and θ-congruence, the prime spectre. For instance, in the case of LM_n algebras without negation, the congruence lattice of $\mathcal{L}$ is a Boolean algebra (a Stone algebra) if and only if $\mathcal{L}$ is finite ($C(L)$ is a complete Boolean algebra).

7 Monadic and polyadic algebras

L. Monteiro and G. Georgescu studied the generalization to LM algebras of the monadic and polyadic Boolean algebras introduced by P.R. Halmos. Sample results: the representation of monadic LM algebras by functional monadic LM algebras and the semantic completeness for polyadic LM algebras. A paper of G. Geogescu, A. Iorgulescu and I. Leuştean investigates monadic MV_n algebras and closed MV_n algebras.

8 Miscellanea

Various other topics have also been studied. Thus:

V. Boicescu proved that the lattice of equational subclass of LM_n is a finite Heyting algebra. The study of atomic algebras and the characterization of simple algebras as subalgebras of $In(I,L_2)$ and the property that LM_n algebras without negation form an equational class, are also due to Boicescu. The study of irredundant algebras and of exactly n-valued algebras is due to Boicescu as well.

A. Iorgulescu introduced and studied m-complete LM_θ algebras with negations, generalizing many of the properties of m-complete Boolean algebras.

G. Georgescu and I. Leuştean studied probabilities on LM algebras.

L. Beznea studied a generalization of LM algebras, obtained by dropping the determination principle.

Let us also mention M. Sularia's theory of D algebras. These structures are subdirect products between a Heyting algebra and a Brouwer algebra and represent the algebraic counterpart of a logic of problem solving.

In [44], [48], the authors study Łukasiewicz BCK algebras endowed with Moisil oparators.

In [163], [164], C. Sanza introduced and studied (monadic) $n \times m$-valued Łukasiewicz algebras with negation.

In a very recent paper [103], I. Leuştean proposes a unifying framework for LM_n algebras, MV algebras and Post algebras; essentially, an LM_{n+1} algebra is charcterized by a string of n Boolean ideals of his Boolean center. The necessary and sufficient conditions are given that such a string must satisfy to define a MV_{n+1} algebra or a Post algebra of order $n+1$. This result could be seen as a generalization of Moisil's Determination Principle. As an application, in paper [75], some special Cauchy completions of MV_{n+1} algebras are characterized by using the properties of corresponding strings of Boolean ideals.

In another very recent paper [75], G.Georgescu and A. Popescu introduced the notion of ***n*-nuanced MV algebra**, by performing a Łukasiewicz-Moisil nuancing construction on top of MV-algebras. These structures extend both MV-algebras and Łukasiewicz-Moisil algebras, thus unifying two important types of structures in the algebra of logic. On a logical level, n-nuanced MV algebras amalgamate two distinct approaches to many-valuedness: that of the infinitely valued Łukasiewicz logic, more related in spirit to the fuzzy approach, and that of Moisil n-nuanced logic, which is more concerned with **nuances of truth** rather than **truth degrees**. They study n-nuanced MV algebras mainly from the algebraic and categorial points of view and also consider some basic model-theoretic aspects. The relationship with a suitable notion of n-nuanced ordered group via an extension of the Γ construction is also analyzed:

8.1 *n*-nuanced MV algebras

Usually, MV algebras are defined only in terms of $\oplus$, $^-$ and 0. However, in order to point out the symmetry of these structures, the authors prefered the following slightly redundant definition:

Definition 8.1. *An* **MV algebra** *is a structure* $(A,\oplus,\odot,^-,0,1)$, *satisfying the following axioms:*

(MV1') $(A,\oplus,0)$ *and* $(A,\odot,1)$ *are commutative monoids,*
(MV2') $x\odot 0 = 0$ *and* $x\oplus 1 = 1$,
(MV3') $(x^-)^- = x$,
(MV4') $(x\oplus y)^- = x^-\odot y^-$,
(MV5') $(x\odot y^-)\oplus y = (y\odot x^-)\oplus x$.

Definition 8.2. *A* **generalized De Morgan algebra** *is a structure* $\mathcal{L} = (L,\oplus,\odot,^-,0,1)$, *where* $\oplus$, $\odot$ *are binary operations,* $^-$ *is a unary operation, and* $0,1$ *are constants such that the following conditions hold:*
(i) $(L,\oplus,0)$, $(L,\odot,1)$ *are commutative monoids;*
(ii) $(x\oplus y)^- = x^-\odot y^-$ *and* $(x^-)^- = x$ *for all* $x,y\in L$*; .*

Remark 8.3. *If* $\mathcal{L}$ *is a generalized De Morgan algebra, then* $(x\odot y)^- = x^-\oplus y^-$ *for all* $x,y\in L$.

Definition 8.4. *An n*-**nuanced MV-algebra (**NMV_n **algebra** *for short) is a structure*

$$\mathcal{L} = (L,\oplus,\odot,^-,r_1,\ldots,r_{n-1},0,1)$$

such that $(L,\oplus,\odot,^-,0,1)$ *is a generalized De Morgan algebra and* $r_1,\ldots,r_{n-1}$ *satisfy the following axioms:*
(A0) $r_ix\oplus((r_ix)^-\odot r_iy) = r_iy\oplus((r_iy)^-\odot r_ix)$, *for* $i\in\{1,\ldots,n-1\}$,

(A1) $r_i(x \oplus y) = r_i x \oplus r_i y$, $r_i(x \odot y) = r_i x \odot r_i y$, $r_i(0) = 0$, $r_i(1) = 1$, *for* $i \in \{1, \ldots, n-1\}$,
(A2) $r_i x \oplus (r_i x)^- = 1$, $r_i x \odot (r_i x)^- = 0$, *for* $i \in \{1, \ldots, n-1\}$,
(A3) $r_i \circ r_j = r_j$, *for* $i, j \in \{1, \ldots, n-1\}$,
(A4) $r_i(x^-) = (r_{n-i}x)^-$, *for* $i \in \{1, \ldots, n-1\}$,
(A5) (Determination Principle:) if $r_i x = r_i y$ *for each* $i \in \{1, \ldots, n-1\}$, *then* $x = y$,
(A6) $r_1 x \leq r_2 x \leq \ldots \leq r_{n-1} x$.

Remark 8.5. *NMV_n algebras provide a common generalization of MV- and Łukasiwicz-Moisil algebras. Indeed,*
- if $n = 2$, then, because of the Determination Principle, r_1 is the identity, thus an NMV_n algebra can be identified with an MV-algebra;
- if $(L, \oplus, \odot, ^-, 0, 1)$ is a De Morgan algebra, then $(L, \oplus, \odot, ^-, r_1, \ldots, r_{n-1}, 0, 1)$ becomes an LM_n algebra.

Example 8.6. *Let $\mathcal{A} = (A, \oplus, \odot, ^-, 0, 1)$ be an MV-algebra. Consider the set*

$$T(A) = \{(x_1, \ldots, x_{n-1}) \in A^{n-1} \mid x_1 \leq \ldots \leq x_{n-1}\}.$$

Since A^{n-1} is an MV-algebra (with operations taken component-wise from A) and $T(A)$ is closed under the operations $\mathbf{0}$, $\mathbf{1}$, $\oplus$, $\odot$ (where $\mathbf{0}$ and $\mathbf{1}$ are the constant vectors), then we can consider these operations on $T(A)$. We furthermore define $^-$, $r_1, \ldots, r_{n-1}$ by:
$(x_1, \ldots, x_{n-1})^- = (x_{n-1}^-, \ldots, x_1^-)$,
$r_i(x_1, \ldots, x_{n-1}) = (x_i, \ldots, x_i)$, *for* $i \in \{1, \ldots, n-1\}$.
Then $(T(A), \oplus, \odot, ^-, r_1, \ldots, r_{n-1}, \mathbf{0}, \mathbf{1})$ is an NMV_n algebra.

Define

$$M(L) = \{x \in L \mid r_i x = x \text{ for all } i \in \{1, \ldots, n-1\}\}.$$

Then $M(L)$, together with the operations $\oplus, \odot, ^-, 0, 1$ induced by $\mathcal{L}$, is an MV algebra, called **the MV-center** of $\mathcal{L}$.

In the MV-algebra $M(L)$ we have a canonical order $\leq$. Let us define an extension of this order to L by:

$x \leq y$ iff, for each $i \in \{1, \ldots, n-1\}$, $r_i x \leq r_i y$.

Because of the Determination Principle, this is indeed an order and because of (A3), it is indeed an extension of the order on $M(L)$. Moreover, the compatibility properties listed in the following lemma are obvious:

Proposition 8.7. *The following properties are true in a $\mathcal{L}$:*
(1) 0 *is the greatest and* 1 *is the least element in L w.r.t.* $\leq$*;*
(2) for each $x, y \in L$, $x \leq y$ *iff* $y^- \leq x^-$*;*
(3) for each $x, x', y, y' \in L$, *if* $x \leq x'$ *and* $y \leq y'$, *then* $x \oplus y \leq x' \oplus y'$ *and* $x \odot y \leq x' \odot y'$,
(4) $r_1 x \leq x \leq r_{n-1} x$, *for any* $x \in L$,
(5) for $x, y \in L$, *if* $x \oplus y = 1$ *and* $x \odot y = 0$, *then* $x, y \in M(L)$ *and* $y = x^-$,
(6) $M(L) = \{x \in L \mid x \oplus x^- = 1', \qquad x \odot x^- = 0\}$.

9 Applications to switching theory

Whereas Boolean algebra is a suitable tool for the study of networks made up of binary devices, the study of networks involving multi-positional devices and the so-called hazard and race phenomena have imposed the use of other algebraic tools, namely Galois fields, Łukasiewicz-Moisil algebras and the theory of discrete functions.

Moisil investigated circuits involving devices such as polarized relays with unstable neutral, ordinary relays under low self-maintaining current, valves, resistances, multi-positional relays, as well as transistors and other electronic devices. See also [125]. Moisil has described the operation of such devices by characteristic equations of the form $x_{k+1} = \varphi(\xi_k, x_k)$, where the variable x associated with the relay contact takes values in L_n, where $n \leq 5$

depends on the type of the relay, $\xi \in L_2$ is a variable associated with the current and the index k or $k+1$ indicates the value of the corresponing variable at time $t = k$ or $t = k+1$, respectively.

The **synthesis problem** consists in designing a circuit made up of several relays and whose operation be described by a given equation of the form

$$X_{k+1} = F(A_k, X_k), \tag{14}$$

where X is the vector of the variables x associated with the relays of the circuit, A is the input vector and the meaning of the index k or $k+1$ is the same as above. To solve this problem, Moisil notices the crucial point that the structure of such a circuit is determined by the expression of a function G which satisfies the identity

$$\Xi = G(A, X), \tag{15}$$

where Ξ is the vector of the variables ξ associated with the relay of the circuit. So if

$$X_{k+1} = Phi(\Xi_k, X_k) \tag{16}$$

is the vector form of the characteristic equations of the relays of the circuit, it follows from (14) and (15) that

$$F(A_k, X_k) = \Phi(\Xi_k, X_k), \tag{17}$$

for any k. Therefore (15) transforms (17) into the identity

$$F(A, X) = \Phi(G(A, X), X), \tag{18}$$

and Moisil's method for solving the synthesis problem consists in solving the functional equation (18) with respect to G.

References

[1] A. Amroune, Representation des algèbres de Łukasiewicz θ-valentes involutives par des structures floues, BUSEFAL (Institut de Recherche en Informatique de Toulouse), 43, 1990, 5-11.

[2] M. Abad, Three valued Łukasiewicz algebras with an additional operation, Rev. Union Mat. Argentina, 32, 1985, 107-117.

[3] M. Abad, A. Figallo, Characterization of three-valued Łukasiewicz algebras, Rep. Math. Logic, 18, 1984, 47-59.

[4] M. Abad, L. Monteiro, On three-valued Moisil algebras, Logique et Analyse, 27, 1984, 407-414.

[5] I. Bădele, V. Boicescu, Sur les extensions des algèbres de Łukasiewicz, C.R. Acad. Sci. Paris, 269, 1969, 313-315.

[6] D. Becchio, Axiomatisation d'une logique trivalente Łukasiewiczienne, 1971, Manuscript.

[7] D. Becchio, Nouvelle démonstration de la complétude du système de Wajsberg axiomatisant la logique trivalente de Łukasiewicz, C.R. Acad. Sci. Paris, 275, 1972, 679-681.

[8] D. Becchio, Sur les définitions des algèbres trivalentes de Łukasiewicz donées par A. Monteiro, Logique et Analyse, 63-64, 1973, 339-344.

[9] D. Becchio, Algèbres de Heyting, algèbres de Brouwer et algèbres trivalentes de Łukasiewicz, Logique et Analyse, 21, 1978, 237-248.

[10] D. Becchio, Logique trivalente de Łukasiewicz, Ann. Sci., Univ. Clermont-Ferrand, 16, 1978, 38-89.

[11] D. Becchio, L. Iturrioz, Sur une définition des algèbres de Łukasiewicz et de Post d'ordre n, Demonstratio Math., 11, 1978, 1083-1094.

[12] L. Beznea, θ-valued Moisil algebras and dual categories (Romanian), Master Thesis, University of Bucarest, 1981.

[13] V. Boicescu, Sur la représentation des algèbres de Łukasiewicz θ-valentes, C.R. Acad. Sci. Paris, 270, 1970, 4-7.

[14] V. Boicescu, Sur les algèbres de Łukasiewicz, Logique,Automatique, Informatique, Editions de l'Academie de la R.S.Roumanie, 1971, 71-89.

[15] V. Boicescu, Sur les systèmes déductifs dans la logique θ-valente, Publ. Dép. Math. Lyon, 8, 1971, 123-133.

[16] V. Boicescu, On Łukasiewicz algebras (Romanian), In: Probleme de logică, vol.IV, Ed. Academiei R.S.Romania, Bucharest, 1972, 245-276.

[17] V. Boicescu, On θ-valued logics (Romanian), In: Probleme de Logică, vol. V, 1973, Edit. Acad. R.S.Romania, 241-255.

[18] V. Boicescu, Sur une logique polivalente, Rev. Roum. Sci. Soc., sér. Philos. et Logique, 17, 1973, 393-405.

[19] V. Boicescu, Researches in Łukasiewicz algebras, Rev. Roum. Sci. Soc., sér. Philos. et Logique, 20, 1976, 197-200.

[20] V. Boicescu, Extensions of homomorphisms of Lulasiewicz algebras with bounding semimorphisms, Rev. Roum. Sci. Soc., sér. Philos et Logique, 23, 1979, 367-370.

[21] V. Boicescu, Contributions to the study of Łukasiewicz algebras (Romanian), Ph.D. Thesis, University of Bucharest, 1984.

[22] V. Boicescu, Irredundant *n*-valued Moisil algebras, Discrete Math., 71, 1988, 197-204.

[23] V. Boicescu, G. Georgescu, Les algèbres de Łukasiewicz centrées et axées, Rev. Roum. Math. Pures et Appl., 15, 1970, 675-681.

[24] V. Boicescu, A. Iorgulescu, Current research in the field of Łukasiewicz-Moisil algebras (Romanian), Studii Cerc. Mat., 39, 1987, 97-106.

[25] V. Boicescu, A. Filipoiu, G. Georgescu, S. Rudeanu, Łukasiewicz-Moisil algebras, Annals of Discrete Mathematics, 49, 1991, North-Holland.

[26] C.C. Chang, Algebraic analysis of many valued logics, Trans. Amer. Math. Soc., 1958, 88, 467-490.

[27] R. Cignoli, Boolean elements in Łukasiewicz algebras. I. Proc. Japan Acad., 41, 1965, 670-675.

[28] R. Cignoli, Un teorema de representation para algebras de Łukasiewicz trivalentes, Rev. Union Mat. Argentina, 23, 1966, 41.

[29] R. Cignoli, Algebras de Moisil de orden *n*, Ph.D. Thesis, Universidad Nacional del Sur, Bahia Blanca, 1969.

[30] R. Cignoli, Moisil algebras, Notas de Logica Matematica, Inst. Mat., Univ. National del Sur, Bahia-Blanca, No. 27, 1970.

[31] R. Cignoli, Representation of Łukasiewicz algebras and Post algebras by continuous functions, Colloq. Math. 24, 1972, 127-138.

[32] R. Cignoli, Topological representation of Łukasiewicz and Post algebras, Notas de Logica Matematica, Inst. Mat. Univ. National del Sur, Bahia-Blanca, No. 33, 1974.

[33] R. Cignoli, Coproducts in the categories of Kleene and three-valued Łukasiewicz algebras, Studia Logica, 39, 1979, 237-245.

[34] R. Cignoli, Proper *n*-Valued Łukasiewicz Algebras as S-Algebras of Łukasiewicz n-Valued Propositional Calculi, Studia Logica, 41, 1982, 3-16.

[35] R. Cignoli, An algebraic approach to elementary theories based on *n*-valued Łukasiewicz logics, Z. Math. Logik u. Grundl. Math., 30, 1984, 87-96.

[36] R. Cignoli, M.S. De Gallego, The lattice structure of 4-valued Łukasiewicz algebras, J. Symbolic Logic, 46, No. 1, 1981, 185.

[37] R. Cignoli, M.S. De Gallego, The lattice structure of some Łukasiewicz algebras, Algebra Universalis, 13, 1981, 315-328.

[38] R. Cignoli, A. Monteiro, Boolean elements in Łukasiewicz algebras. II., Proc. Japan Acad., 41, 1965, 676-680.

[39] R. Cignoli and D. Mundici, An elementary proof of Chang's completeness theorem for the infinite-valued calculus of Łukasiewicz, Studia Logica, to appear.

[40] R. Cignoli, I.M.L. D'Ottaviano, D. Mundici, Algebraic Foundations of many-valued Reasoning, Kluwer 2000, Volume 7.

[41] J. Coulon, J.L. Coulon, A propos de la représentation des algèbres de Łukasiewicz et des algèbres booléiennes floues, Rev. Roum. Math. Pures et Appl., 34, 1989, 403-411.

[42] J. Coulon, J.L. Coulon, Un nouveau resultat concernant la representation d'une algèbre de Łukasiewicz involutive dans l'algèbre des parties floues d'une structure floue involutive, Rev. Roumaine Math. Pures et Appl., 38, 1993, 319-326.

[43] A. Figallo, J. Tolosa, Algebras de Łukasiewicz trivalente, Univ. Nac. de San Juan, 1982.

[44] A. Figallo Jr., M. Figallo and A. Ziliani, Free $(n+1)$-valued Łukasiewicz BCK algebras, Demonstratio Mathematica, XXXVII, 2, 2004, 245-254.

[45] A.V. Figallo, I. Pascual, A. Ziliani, Notes on monadic n-valued Łukasiewicz algebras, Math. Bohem., 3, 129, 2004, 255-271.

[46] A.V. Figallo, C. Sanza , A. Ziliani, Functional monadic n-valued Łukasiewicz algebras, accepted by Mathematica Bohemica.

[47] A.V. Figallo, I. Pascual, A. Ziliani, Subdirectly irreducible monadic Łukasiewicz-Moisil algebras, Manuscript.

[48] A.V. Figallo, Łukasiewicz BCK-algebras endowed with Moisil operators, Manuscript.

[49] A.V. Figallo, A. Figallo Jr., M. Figallo, A. Ziliani, Łukasiewicz residuation algebras with infimum, Manuscript.

[50] A. Filipoiu, Analytic tableaux for θ-valued propositional logic, Math. Seminar Notes, 6, 1978, 517-526.

[51] A. Filipoiu, Representation theorems for Łukasiewicz algebras, Discrete Math., 27, 1979, 107-110.

[52] A. Filipoiu, Representation of Łukasiewicz algebras by means of ordered Stone spaces, Discrete Math., 30, 1980, 111-116.

[53] A. Filipoiu, θ-valued Łukasiewicz-Moisil algebras and logics (Romanian), Ph.D. Thesis, Univ. of Bucharest, 1981.

[54] A. Filipoiu, Representation theorems for θ-valued Łukasiewicz algebras, Discrete Math., 33, 1981, 21-27.

[55] A. Filipoiu, Some remarks on the representation theorem of Moisil, Discrete Math., 33, 1981, 163-170.

[56] J. M. Font, A. J. Rodriguez, A. Torrens, Wajsberg Algebras, Stochastica, VIII, 1, 5-31, 1984.

[57] G. Georgescu, Algébres de Łukasiewicz complètes, C.R. Acad. Sci. Paris, 269, 1969, 1181-1184.

[58] G. Georgescu, The centered epimorphisms and the construction of the tensor product in Łuk, Rev. Roum. Math. Pures Appl., 15, 1970, 693-709.

[59] G. Georgescu, n-valued complete Łukasiewicz algebras, Rev. Roum. Math. Pures Appl., 16, 1971, 41-50.

[60] G. Georgescu, The θ-valued Łukasiewicz algebras. I., Rev. Roum. Math. Pures Appl., 16, 1971, 195-209.

[61] G. Georgescu, Algebras de Łukasiewicz de orden θ . II., Rev. Roum. Math. Pures Appl., 16, 1971, 363-369.

[62] G. Georgescu, Les algèbres de Łukasiewicz θ-valentes, In: Logique, Automatique, Informatique, Edit. Acad. R.S.Romania, Bucharest, 1971, 99-169.

[63] G. Georgescu, The θ-valued Łukasiewicz algebras. III. Duality theory., Rev. Roum. Math. Pures Appl., 16, 1971, 1365-1390.

[64] G. Georgescu, Algebre Łukasiewicz θ-valente, Ph.D. Thesis, Math. Institute, Bucharest, mai 1972.

[65] G. Georgescu, Représentation des algèbres de Łukasiewicz θ-valentes polyadiques, C. R. Acad. Sci. Paris, A-B, 274, 1972, A944-A946.

[66] G. Georgescu, Reprezentarea algebrelor Łukasiewicz poliadice local finite de grad infinit, Studii Cerc. Mat., 24, 1972, 1107-1117.

[67] G. Georgescu, Some remarks on the polyadic Łukasiewicz algebras, Rev. Roum. Mat. Pures Appl., 22, 1977, 641-648.

[68] G. Georgescu, On the homogeneous-universal Łukasiewicz algebras, Rev. Roum. Mat. Pures Appl., 23, 1978, 29-32.

[69] G. Georgescu, A. Iorgulescu, Pseudo-MV Algebras: a Noncommutative Extension of MV Algebras, The Proceedings of the Fourth International Symposium on Economic Informatics, Bucharest, Romania, May 1999, 961-968.

[70] G. Georgescu, I. Leuştean, Towards a probability theory based on Moisil logic, Soft Computing, 2000, 4, No.1, 19-26.

[71] G. Georgescu, A. Iorgulescu, Pseudo-BL algebras: A noncommutative extension of BL algebras, Abstracts of The Fifth International Conference FSTA 2000, Slovakia, February 2000, 90-92.

[72] G. Georgescu, A. Iorgulescu, Pseudo-BCK algebras: An extension of BCK algebras, Proceedings of DMTCS'01: Combinatorics, Computability and Logic, Springer, London, 2001, 97-114.

[73] G. Georgescu, I. Leuştean, Probabilities on Łukasiewicz-Moisil algebras, International Journal of Approximate Reasoning, 1998, 18, No.3-4, 201-215.

[74] G. Georgescu, I. Leuştean, Conditional probabilities on Łukasiewicz-Moisil algebras, Analele Universitatii Bucuresti, 1998, 47, 55-64.

[75] G. Georgescu., I. Leuştean, A. Popescu, Order convergence and distance on Łukasiewicz-Moisil algebras, Multiple Valued Logic, to appear.

[76] G. Georgescu, A. Popescu, A common generalization for algebras and Łukasiewicz-Moisil algebras, submitted.

[77] G. Georgescu, C. Vraciu, Le spectre maximal dŠune algèbre de Łukasiewicz, C.R. Acad. Sci. Paris, 268, 1969, 928-929.

[78] G. Georgescu, C. Vraciu, *n*-valued centered Łukasiewicz algebras, Rev. Roum. Math. Pures Appl., 14, 1969, 712-723.

[79] G. Georgescu, C. Vraciu, Sur les algèbres de Łukasiewicz centrées, C.R. Acad. Sci. Paris, 268, 1969, 998-1000.

[80] G. Georgescu, C. Vraciu, Sur le spectre maximal d'une algèbre de Łukasiewicz, Publ. Dép. Math. Lyon, 6, 1969, 42-54.

[81] G. Georgescu, C. Vraciu, Sur les épimorphismes centrées des algèbres de Łukasiewicz, C.R. Acad. Sci. Paris,269, 1969, 4-6.

[82] G. Georgescu, C. Vraciu, On the characterization of centered Łukasiewicz algebras, J. Algebra, 16, 1970, 486-495.

[83] G. Georgescu, C. Vraciu, Monadic Boolean algebras and monadic Łukasiewicz algebras (Romanian), Studii Cerc. Mat., 23, 1971, 1025-1048.

[84] G. Georgescu, C. Vraciu, La dualité des algèbres de Post θ-valentes, J. Algebra, 21, 1972, 74-86.

[85] H. Goldberg, H. Leblanc, G. Weaver, A strong completeness theorem for 3-valued logic, Notre Dame J. Formal Logic, 15,1974, 325-332.

[86] M. Greniewski, Using three valued logics in the theory of swiching theory (Romanian). I. Realizarea cu circuite a funcţiilor fundamentale, Comunic. Acad. R.P.R., 6, 1956, 225-229.

[87] R. Grigolia, Algebraic analysis of Łukasiewicz-Tarski's n-valued logical systems, in: Selected Papers on Łukasiewicz Sentential Calculi (R. Wójcicki and G. Malinowski, Eds.), Polish Acad. Of Sciences, Ossolineum, Wroclaw, 1977, 81-92.

[88] A. Iorgulescu, On the construction of three-valued Łukasiewicz-Moisil algebras, Discrete Math., 47, 1984, 213-227.

[89] A. Iorgulescu, Functors between categories of three-valued Łukasiewicz-Moisil algebras, Discrete Math., 49, 1984, 121-131.

[90] A. Iorgulescu, $(1+\theta)$-valued Łukasiewicz-Moisil algebras with negation (Romanian), Ph.D. Thesis, Univ. of Bucharest, 1984.

[91] A. Iorgulescu, Connections between MV_n algebras and n-valued Łukasiewicz-Moisil algebras - I, Discrete Mathematics, 181 (1-3), 1998, 155-177.

[92] A. Iorgulescu, Connections between MV_n algebras and n-valued Łukasiewicz-Moisil algebras - II, Discrete Mathematics, 202, 1999, 113-134.

[93] A. Iorgulescu, Connections between MV_n algebras and n-valued Łukasiewicz-Moisil algebras -III, Manuscript.

[94] A. Iorgulescu, Connections between MV_n algebras and n-valued Łukasiewicz-Moisil algebras - IV, Journal of Universal Computer Science, vol. 6, no I , 2000, 139-154.

[95] L. Iturrioz, Axiomas para el calculo proposicional trivalente de Łukasiewicz, Rev. Union Mat. Argentina, 22, 1965, 150.

[96] L. Iturrioz, Sur une classe particulière d'algèbres de Moisil, C. R. Acad. Sci. Paris, 267, 1968, 585-588.

[97] L. Iturrioz, Les algèbres de Heyting-Brouwer et de Łukasiewicz trivalentes, Notre Dame J. Formal Logic, 17, 1976, 119-126.

[98] L. Iturrioz, Algèbres de Łukasiewicz symétriques, Publ. Dép. Math. Lyon, 13, 1976, 73-96.

[99] L. Iturrioz, Łukasiewicz and symmetrical Heyting algebras, Z. Math. Logik u. Grund.Math., 23, 1977, 131-136.

[100] L. Iturrioz, Two characteristic properties of monadic three-valued Łukasiewicz algebras, Rep. Math. Logic, 8, 1977, 63-68.

[101] L. Iturrioz, An axiom system for three-valued Łukasiewicz propositional calculus, Notre Dame J. Formal Logic, 18, 1977, 616-620.

[102] L. Iturrioz, O. Rueda, Algèbres implicatives trivalentes de Łukasiewicz libres, Discrete Math., 18, 1977, 35-44.

[103] I. Leuştean, A unifying framework for Łukasiewicz-Moisil algebras, MV-algebras and Post algebras, submitted.

[104] J. Łukasiewicz, On three-valued logic (Polish), Ruch Filozoficzny, 5, 1920, 160-171.

[105] J. Łukasiewicz, Philosophische Bemerkungen zur mehrwertigen Systemen des Aussagenkalküls, C.R. Séances Soc. Sci. Lettres Varsovie, Cl. III, 23, 1930, 51-77. Romanian translation in: Logică şi Filozofie, Ed. Politică, Bucureşti, 1966, 295-320.

[106] J. Łukasiewicz, Die Logik und das Grundlagenproblem, Les entretiens de Zurich sur les fondements et la méthode des sciences mathématiques, 1941, 88-100.

[107] J. Łukasiewicz, A. Tarski, Untersuchungen über den Aussagenkalkül, C.R. Séances Soc. Sci. Lettres Varsovie, Cl. III, 23, 1930, 30-50.

[108] M.G. Malinowski, n-valued Łukasiewicz algebras and their connection to Post algebras of order (Polish), Zeszyty Naukowe u.t., Filozofia, 1972.

[109] P. Mangani, On certain algebras related to many-valued logics (Italian), Boll. Un. Mat. Ital. (4) 8, 68-78, 1973.

[110] W. Marek, T. Traczyk, Generalized Łukasiewicz algebras, Bull. Acad. Polonaise Sci. Sér. Math. Astronom. Phys., 17, 1969, 789-792.

[111] Gr. C. Moisil, Recherches sur l'algèbre de la logique, Ann. Sci. Univ. Jassy, 22, 1935, 1-117.

[112] Gr. C. Moisil, Sur le mode problématique, C.R. Séances Acad. Sci. Roumanie, 2, No. 2, 1938, 101-103.

[113] Gr. C. Moisil, Recherches sur les logiques non-chrysippiennes, Ann. Sci. Univ. Jassy, 26, 1940, 431-466.

[114] Gr.C. Moisil, Notes sur les logiques non-chrysippiennes, Ann. Sci, Univ. Jassy, 27, 1941, 86-98.

[115] Gr. C. Moisil, Sur les anneaux de caractéristique 2 ou 3 et leurs applications, Bul. Politechn. Bucharest, 12, 1941, 66-90.

[116] Gr. C. Moisil, Contributions à l'étude des logiques non-chrysippiennes. I. Un nouveau système d'axiomes pour les algèbres Łukasiewicziennes tétravalentes, C.R. Acad. Sci. Roumanie, 5, 1941, 289-293.

[117] Gr. C. Moisil, Contributions à l'étude des logiques non-chrysippiennes. II. Anneaux engendrés par les algèbres Łukasiewicziennes centrées, C.R. Acad. Sci. Roumanie, 6, 1942, 9-14.

[118] Gr. C. Moisil, Contributions à l'étude des logiques non-chrysippiennes. III. Anneaux engendrés par les algèbres Łukasiewicziennes tétravalentes axées, C.R. Acad. Sci. Roumanie, 7, 1942, 14-18.

[119] Gr. C. Moisil, Logique modale, Disquis. Math. Phys., 2, 1942, 3-98.

[120] Gr. C. Moisil, The algebra of networks with rectifiers (Romanian), Rev. Univ. C.I. Parhon şi a Politehnicii Bucureşti, 4-5, 1954, 9-41.

[121] Gr. C. Moisil, Utilization of three-valued logics in the theory of switching circuits. II. The characteristic equation of a relay. III. Actual-contact circuits. IV. Realization of the working functions in actual operation (Romanian). Comun. Acad. R.P. Române, 6, 1956, 231-239, 385-386, 971-973.

[122] Gr. C. Moisil, Applications of three-valued logics to the study of actual operation of relay-contact circuits (Romanian), Bul. Mat. Soc. St. Fiz. R.P. Române, 1(49), 1957, 147-191.

[123] Gr. C. Moisil, Utilization of three-valued logics to the theory of switching circuits. V. P-I circuits (Romanian), Comunic. Acad. R.P. Române, 8, 1958, 1127-1128.

[124] Gr. C. Moisil, Utilization of three-valued logics to the theory of switching circuits. VI. Polarized relays with unstable neutral. VII. Operation of ordinary relays under low self-maintaining current. VIII. -two-terminals with contacts and resistances. IX. -two-terminals with contacts, valves and resistances. X. Physical interpretation of the characteristic function of a multiterminal (Romanian), Comunic. Acad. R.P. Române, 9, 1959, 411-413, 531-532, 533-535, 665-666, 667-669.

[125] Gr. C. Moisil, Sur l'application de la logique à trois valeurs à l'étude des circuits électriques à contacts, redresseurs et résistances, Rev. Math. Pures Appl. , 4, 1959, 173-183.

[126] Gr. C. Moisil, The algebraic Theory of Switching Circuits (Romanian), Ed. Tehnică Bucureşti. English translation 1969, Pergamon Press, Oxford, and Editura Tehnică Bucureşti.

[127] Gr.C. Moisil, Sur les idéaux des algèbres Łukasiewicziennes trivalentes, Analele Univ. C.I. Parhon, Seria Acta Logica, 3, 1960, 83-95.

[128] Gr.C. Moisil, On predicate calculus in three-valued logics (Russian), An. Univ. C.I. Parhon, Acta Logica, 4, 1961, 103-112.

[129] Gr.C. Moisil, Sur la logique à trois valeurs de Łukasiewicz, An. Univ. C.I. Parhon, Acta Logica, 5, 1962, 103-117.

[130] Gr.C. Moisil, Les logiques non-chrysippiennes et leurs applications, Acta Philos. Fennica, 16, 1963, 137-152.

[131] Gr.C. Moisil, Le algèbre di Łukasiewicz, An. Univ. C.I. Parhon, Acta Logica 6, 1963, 97-135.

[132] Gr.C. Moisil, Applicazioni dell'algebra alle calcolatrici moderne, Atti Reunione del Groupement des Math. D'Expression Latine, 26.IX-3.X. 1961, Ed. Cremonese, Roma, 1963.

[133] Gr.C. Moisil, The interest of the actual operation of switching circuits for the logician, An. Univ. Bucureşti, Acta Logica, 7-8, 1964, 131-139.

[134] Gr.C. Moisil, Sur les logiques de Łukasiewicz à un nombre fini de valeurs, Rev. Roum. Math. Pures Appl., 9, 1964, 905-920.

[135] Gr.C. Moisil, Încercări vechi şi noi de logică neclasică (Old and New Essays on Non-Classical Logics), Edit. Ştiinţifică, Bucharest, 1965.

[136] Gr.C. Moisil, Théorie structurelle des automats finis, Gauthiers-Villars, Paris, 1967.

[137] Gr.C. Moisil, Łukasiewiczian algebras, Computing Center, University of Bucharest (preprint), 311-324, 1968.

[138] Gr.C. Moisil, Essais sur les logiques non-chrysippiennnes, Ed. Academiei R.S.R., Bucharest, 1972.

[139] Gr.C. Moisil, Ensembles flous et logiques à plusieurs valeurs, Centr. Rech. Math., Université de Montréal, mai, 1973 (preprint).

[140] Gr.C. Moisil, Lecţii despre logica raţionamentului nuanţat (Lectures on the Logic of Fuzzy Reasoning), Ed. Ştiinţifică şi Enciclopedică, Bucharest, 1975.

[141] Gr.C. Moisil, Sur l'emploi des mathématiques dans les sciences de l'homme, Accad. Naz. Lincei, Contributi del Centro Linceo Interdisciplinare di Sci. Mat. E loro Appl., No. 17, 1976.

[142] A. Monteiro, Matrices de Morgan caracteristiques pour le calcul propositional classique, Ann. Acad. Brasil, 52, 1960, 1-7.

[143] A. Monteiro, Sur la définition des algèbres de Łukasiewicz trivalentes, Notas de Logica matematica, 21, 1964.

[144] A. Monteiro, Construction des algèbres de Łukasiewicz trivalentes dans les algèbres de Boole monadiques, Notas de Logica Matematica, 11, 1964.

[145] A. Monteiro, R. Cignoli, Construccion geometrica de las algebras de Łukasiewicz trivalentes libres, Rev. Union Mat. Argentina, 22 , 1965, 152-153.

[146] L. Monteiro, Axiomes indépendentes pour les algèbres de Łukasiewicz trivalentes, Notas de Logica Matematica, 32, 1974.

[147] L. Monteiro, Algebras de Łukasiewicz trivalentes monadicas, Notas de Logica Matematica, 32, 1974.

[148] L. Monteiro, Sur la construction L des algèbres de Łukasiewicz trivalentes, Rev. Roum. Math. Pures Appl., Tome XXIII, No. 1, 1978, 77-83.

[149] L. Monteiro, L.G. Coppola, Sur une construction des algèbres de Łukasiewicz trivalentes, Notas de Logica Matematica, 17, 1964.

[150] D. Mundici, MV-algebras are categorically equivalent to bounded commutative BCK-algebras, Math. Japonica, 31, No. 6, 1986, 889-894.

[151] D. Mundici, Interpretation of $AF\ C^{\star}$-algebras in Łukasiewicz sentential calculus, J. Funct. Anal., 65, 1986, 15-63.

[152] D. Mundici, The C^{*}-algebras of three-valued logic, Logic Colloquium'88, Ferro, Bonotto, Valentini and Zanardo (Editors), Elsevier Science Publishers B.V. (North-Holland), 1989, 61-77.

[153] Gh. Nadiu, On a method for the construction of three-valued Łukasiewicz algebras (Romanian), Studii Cerc. Mat., 19, 1967, 1063-1070.

[154] A. Petcu, The definition of the trivalent Łukasiewicz algebras by three equations, Rev. Roumaine Math. Pures Appl., 13, 1968, 247-250.

[155] I. Petrescu (I. Voiculescu), Injective objects in the category of De Morgan algebras, Rev. Roumaine Math. Pures Appl., 16, 1971, 921-926.

[156] D. Ponasse, Algèbres floues et algèbres de Łukasiewicz, Rev. Roumaine Math. Pures Appl., XXIII, 1, 1978, 103-111.

[157] E. Post, Introduction to a general theory of elementary propositions, Amer. J. Math., 43, 1921, 163-185.

[158] E. Radu, L'oeuvre de Gr. C. Moisil en logique mathématique, I., II., Rev. Roumaine Math. Pures Appl., 23, 1978, 463-477, 605-610.

[159] J. Rodriguez, A. Torrens, Wajsberg Algebras and Post Algebras, Studia logica, 53, 1994, 1-19.

[160] P. Rosenbloom, Post algebras. I. Postulates and general theory, Amer. J. Math., 64, 1942, 167-183.

[161] S. Rudeanu, On Łukasiewicz-Moisil algebras of fuzzy sets, Studia Logica, 52, 1993, 95-111.

[162] A. Sade, Algèbres de Łukasiewicz dans la logique trivalente, Univ. Beograd, Publ. Elektrotehn. Fak. Ser. Mat. Fiz., No. 247-273, 1969, 123-130.

[163] C. Sanza, Notes on $n \times m$-valued Łukasiewicz algebras with negation, L. J. of the IGPL, 6, 12, 2004, 499-507 (<http://jigpal.oupjournals.org/current.dtl>)

[164] C. Sanza, Monadic $n \times m$-valued Łukasiewicz algebras with negation, Manuscript.

[165] C. Sicoe, Strictly chrysippian elements in many-valued Łukasiewicz algebra (Romanian), An. Univ. of Bucharest, 15, 1966, 123-126.

[166] C. Sicoe, Sur les ideaux des algèbres Łukasiewicziennes polivalentes, Rev. Roum. Math. Pures Appl., 12, 1967, 391-401.

[167] C. Sicoe, Note asupra algebrelor Łukasiewicziene polivalente, Stud. şi Cerc. Mat., 19, 1967, 1203-1207.

[168] C. Sicoe, On many-valued Łukasiewicz algebras, Proc. Japan Acad., 43, 1967, 725-728.

[169] C. Sicoe, A characterization of Łukasiewicz algebras. I. II. Proc. Japan Acad., 43, 1967, 729-732, 733-736.

[170] C. Sicoe, Sur la définition des algèbres Łukasiewicziennes polyvalentes, Rev. Roumaine Math. Pures Appl., 13, 1968, 1027-1030.

[171] W. Suchon, On the non-equivalence of two definitions of the algebra of Łukasiewicz, P Polish Acad. Sci. Inst. Philos. Sociol. Bull. Sect. Logic, 1, No. 1, 1972, 35-37.

[172] W. Suchon, Inéquivalence de certaines définitions des algèbres infinites de Łukasiewicz, Rep. Math. Logic, 1, 1973, 21-26.

[173] W. Suchon, On defining Moisil's functors in *n*-valued Łukasiewicz propositional logic, Polish Acad. Sci. Inst. Philos. Sociol. Bull. Sect. Logic, 2, 1973, 195-196.

[174] W. Suchoń, Définition des foncteurs modaux de Moisil dans le calcul *n*-valent des propositions de Łukasiewicz avec implication et négation, Reports on Mathematical Logic 2, 43-48, 1974.

[175] W. Suchon, Matrix Łukasiewicz algebras, Rep. Math. Logic, 4, 1975, 91-104.

[176] W. Suchon, Définition des foncteurs modaux de Moisil dans le calcul *n*-valent des propositions de Łukasiewicz avec implication et négation, Reports on Mathematical Logic, 2, 1974, 43-48.

[177] J. Varlet, Algèbres de Łukasiewicz trivalentes, Bull. Soc. Roy. Sci. Liège, 36, 1968, 399-408.

[178] J. Varlet, Considérations sur les algèbres de Łukasiewicz trivalentes, Bull. Soc. Roy. Sci. Liège,38, 1969, 462-469.

[179] M. Wajsberg, Axiomatization of three-valued propositional calculus (Polish), C.R. Séances Soc. Sci. Lettres Varsovie, Cl. III, 24, 1931, 126-145.

[180] R. Wójcicki, A theorem on the finiteness of the degree of maximality of the *n*-valued Łukasiewicz logic, Polish Acad. Sci. Inst. Philos. Sociol. Bull. Sect. Logic, 4, 1975, 19-25.

[181] R. Wójcicki, G. Malinowski (editors), Selected Papers on Łukasiewicz sentencial calculi, Ossolineaum, Wroclaw and Warsaw, 1977.

[182] L.A. Zadeh, Fuzzy Sets, Inform. And Control, 8, 1965, 338-353.

George Georgescu
University of Bucharest
Address: Str. Academiei No. 14, Romania
E-mail: georgescu@funinf.cs.unibuc.ro

Afrodita Iorgulescu
Academy of Economic Studies
Address: Piaţa Romană Nr. 6 - R 70167,
Oficiul Poştal 22, Bucharest, Romania
E-mail: afrodita@ase.ro

Sergiu Rudeanu
University of Bucharest
Address: Str. Academiei No. 14, Romania
E-mail: rud@funinf.cs.unibuc.ro

Author index

Instructions for authors

Papers submitted to the International Journal of Computers, Communications & Control must be prepared using a LaTeX typesetting system. A template for preparing the papers is available on the journal website http://journal.univagora.ro. In the template file you will find instructions that will help you prepare the source file. Please, read carefully those instructions.

Any graphics or pictures must be saved in Encapsulated PostScript (.eps) format.

Papers must be submitted electronically to the following address: ccc@univagora.ro.

The papers must be written in English. The first page of the paper must contain title of the paper, name of author(s), an abstract of about 300 words and 3-5 keywords. The name, affiliation (institution and department), regular mailing address and email of the author(s) should be filled in at the end of the paper. Manuscripts must be accompanied by a signed copyright transfer form. The copyright transfer form is available on the journal website.

Order

If you are interested in having a subscription to "Journal of Computers, Communications and Control", please fill in and send us the order form below:

ORDER FORM		
I wish to receive a subscription to "Journal of Computers, Communications and Control"		
NAME AND SURNAME:		
Company:		
Number of subscription:	Price Euro	for issues yearly (4 number/year)
ADDRESS: City: Zip code: Country: Fax: Telephone: E-mail:		
Notes for Editors (optional)		

1. **Standard Subscription Rates (2006)**
   - Individual: One Year (4 issues) 70 Euro
   - Institutional: One Year (4 issues) 90 Euro
2. **Single Issue Rates**
   - Individual: 20 Euro
   - Institutional: 25 Euro

For payment subscription rates please use following data:
HOLDER: Fundatia AGORA, CUI: 12613360
BANK: EUROM BANK ORADEA
BANK ADDRESS: Piata Unirii nr. 2-4, Oradea, ROMANIA
IBAN ACCOUNT for EURO: RO02DAFB1041041A4767EU01
IBAN ACCOUNT for LEI/ RON: RO45DAFB1041041A4767RO01,
Mention, please, on the payment form that the fee is "for IJCCC".

EDITORIAL ADDRESS:
CCC Publications
Piata Tineretului nr. 8
ORADEA, jud. BIHOR
ROMANIA
Zip Code 410526
Tel.: +40 259 427 398
Fax: +40 259 434 925
E-mail: ccc@univagora.ro
Website: www.journal.univagora.ro

Copyright Transfer Form

To The Publisher of the International Journal of Computers, Communications & Control

This form refers to the manuscript of the paper having the title and the authors as below:

The Title of Paper (hereinafter, "Paper"):

. . . .

The Author(s):

. . . .

. . . .

. . . .

. . . .

The undersigned Author(s) of the above mentioned Paper here by transfer any and all copyright-rights in and to The Paper to The Publisher. The Author(s) warrants that The Paper is based on their original work and that the undersigned has the power and authority to make and execute this assignment. It is the author's responsibility to obtain written permission to quote material that has been previously published in any form. The Publisher recognizes the retained rights noted below and grants to the above authors and employers for whom the work performed royalty-free permission to reuse their materials below. Authors may reuse all or portions of the above Paper in other works, excepting the publication of the paper in the same form. Authors may reproduce or authorize others to reproduce the above Paper for the Author's personal use or for internal company use, provided that the source and The Publisher copyright notice are mentioned, that the copies are not used in any way that implies The Publisher endorsement of a product or service of an employer, and that the copies are not offered for sale as such. Authors are permitted to grant third party requests for reprinting, republishing or other types of reuse. The Authors may make limited distribution of all or portions of the above Paper prior to publication if they inform The Publisher of the nature and extent of such limited distribution prior there to. Authors retain all proprietary rights in any process, procedure, or article of manufacture described in The Paper. This agreement becomes null and void if and only if the above paper is not accepted and published by The Publisher, or is withdrawn by the author(s) before acceptance by the Publisher.

Authorized Signature (or representative, for ALL AUTHORS): . . . .

Signature of the Employer for whom work was done, if any: . . . .

Date: . . . .

Third Party(ies) Signature(s) (if necessary): . . . .