

MACHAZINE

Volume 22 - Issue 2
February 2018

Wiskunde
Informatica
Studievereniging



'Christiaan
Huygens'

TREATING LIVER CANCER AT TU DELFT USING

Microspheres

CHECK OUT OUR NEW

Website!

INTERNSHIP ABROAD AT

Google

HISTORICAL FIGURE

Ada Lovelace

CONTAINING:

CURRENT AFFAIRS | ASSOCIATION | COMPUTER SCIENCE | MATHEMATICS | MISCELLANEOUS



BUILDING ON EXPERIENCE

Leading in simulation & virtual solutions

We're Hiring!



Do you get enthusiastic about shaping the future of real-time simulation and virtual training? Or are you curious to learn how we built this amazing virtual environment? At VSTEP we are always eager to meet (graduate) students or Software Engineers!

Visit our career page to find out more!
<https://www.vstepsimulation.com/careers>



MACHAZINE

is a publication of

W.I.S.V. 'Christiaan Huygens'

Chief Editor

Arthur Breurkes, Rebecca Glans

Editorial staff

Marjolein Bouwmeester, Marc Corstanje, Daniël van Gelder, Wouter Versteegh, Eva Slingerland

Art Directors

Jamy Mahabier, Dieuwier Hondelink, Beryl Sicam, Marjolein Leegwater

Contact address

Mekelweg 4, 2628 CD Delft

E: machazine@ch.tudelft.nl

T: 015-2782532

Concept and design

G₂O Kesteren

Publisher

GildePrint drukkerijen

Cover

Physical Bitcoin stacked on laptop

January 28th, 2018 by David McBee

Cooperating on this issue:

Niek van der Laan, Wouter Versteegh, Hans Tonino, Fred Vermolen, Romi Kharisnawan, Sven van Hal, MeisCie 2017-2018, Fleur Rooijackers, Bram van Walraven, Robbert Krebbers, Cas Buijs, Jan van Neerven, Tim van der Lippe, Rebecca Glans, Daniël van Gelder, Eva Slingerland

Terms and Conditions

The MaCHazine-committee and the Board are responsible for the content within this MaCHazine, in such terms that the opinion of a writer is not (necessarily) a reflection of the opinion of the committee or association. All rights reserved. No part of this publication may be reproduced, stored in a (retrieval) system or transmitted in any form or any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of the committee or association.

Advertiser index

VStep	Inside front cover
De Delftse Bedrijvendagen	18-19

Marjolein Leegwater

Dear professors, alumni, fellow students and other interested people,

The past few months a lot of things have changed. Not only did half of EEMCS move to a different building, did a new year start and did the second semester begin, but also the members of the MaCHazine-committee have changed. In November Beryl, Daniël, Eva, Wouter and I were asked to join the committee that makes the MaCHazine and now, after a lot of spellchecking, writing articles, editing and working with InDesign, I can proudly say that the second issue of this year is finished.


When I was asked to write this editorial, I had no idea what to write about. I started asking people what they would actually like to read in the editorial. The only answers I got were answers like: "You should mention that the Dies/AnnuCie/... is the best committee ever."

I realised it is not my job to tell which committee I like the most.

It's up to you to decide which committees you like most, and to help you, this issue of the MaCHazine contains stories written by the W3Cie, LANcie, AkCie and MaPhyA. More committees will follow in the next editions of the MaCHazine. Also don't forget to check out the calendar at the end of this issue, so you know when all the upcoming activities will be. There is no better way to find out what the committees do than to attend their activities.

One of the things I like most about the MaCHazine is the amount of stories about professors, alumni and students and that all the MaCHazines. In this issue you can read, for example, about a director of studies, an international student that now studies in Delft, the BEPs of several students, a professor and many, many more things.

That I like those stories the most does not mean that you should not read the rest of the issue. I have seen that my fellow MaCHazine members were putting a lot of effort in making the articles for this issue. Eva and Beryl have provided a Computer Science and a Mathematical puzzle. Solving these puzzles is the first thing I will do once I receive this MaCHazine at home. Wouter has been working on finding out everything that has happened at the TU Delft within the last few months. Daniël has told us a lot about Ada Lovelace, the woman that published the first algorithm, specifically designed for a machine, ever. You can read about her in the section Historical Figure. If you want to know about the recent news in the field of Computer Science and Mathematics, I hope it makes you happy to know that Rebecca has made an overview of these innovations. This, you can find in the article named Science Trends, as you may have already guessed. And these are only the stories written by the MaCHazine editorial staff. As I have said before, you can read about many, many more things.

Well, I think you have wasted enough time reading this editorial and I think it is time for you to stop reading an editorial and actually start reading the interesting stuff in this MaCHazine. I hope you will enjoy reading this issue and I would like to mention that I really enjoyed making this Issue. I am really looking forward to keep on working on and trying to improve the MaCHazine, or as we would say: "Ik heb er MeCHa zin in!" 



AREA 51 LAN



Gaming Event

WHAT:

Unit Sports & Culture

WHERE:

WHEN:

2-4 March 2018

PRICE:

€ 40*

GET YOUR **TICKETS** AT
AreaFiftyLAN.nl

INFO:

- ALL MEALS INCLUDED
- 7 OFFICIAL TOURNAMENTS
- OFFLINE BOARDGAMES
- FOR CASUAL AND COMPETITIVE GAMERS

*CH members get a 5 euro discount



Table of Contents



Current Affairs

From the board	4
Tu Delft news	6
Spiderman	7
Is The Netherlands losing its winters?	8
What to take in the 3 rd and 4 th quarter?	11



Association

New website!	14
Committee kick-off 2017!	15
AkCie	17
Area FiftyLAN 2018	20



Computer Science

Formal verification	22
The adventure of a BEP: Support for human planners	24

Mathematics

The Hullbert-Pólya conjecture	28
-------------------------------	----

Miscellaneous

Internship at Google	34
Science trends	36
Historical figure: Ada Lovelace	37
Mathematical puzzle	38
Computer science puzzle	39
Last issue's puzzle solutions	40





From the Board

Niek van der Laan

“...great difficulties are felt at first and these cannot be overcome except by starting from experiments .. and then be conceiving certain hypotheses ... But even so, very much hard work remains to be done and one needs not only great perspicacity but often a degree of good fortune.”

A famous scientist once wrote that, and while obviously referring to his approach to research, it feels very applicable to the experience of my board year thus far. From starting off brainstorming how to improve the association that you are soon-to-be governing, to hopefully facing your first challenges and sometimes failing miserably, to finally being able to look back at a stressful, but educational, first few months. Using just this one side of paper, let me tell you about my journey thus far as the Chief Commissioner of Public Relations of Board 61.

Before our board year officially started on September 4th, me and my board members were full of ideas on improving the association. At the time, thinking of how to improve things as the Board was extremely difficult, considering we all did not have a single bit of experience being part of it, yet. There is a Dutch saying however that says ‘de beste stuurlied staan aan wal’ (‘the best pilots are ashore’), and in some way this may have been actually true. True change often requires giving up something you already have, making it that much easier when you actually don’t have anything yet.

Change was ahead, but September 4th passed and before I knew it, we were already six weeks in, on a Friday. Together with several other study association Boards from Delft, we visited the ‘McKinsey Boards Day’. We learned about our own and each other’s personalities through several tests, revealing once more our strengths and weaknesses as a group. During the afternoon we had a long chat with Gerardo de Geest, who worked there and was treasurer of Board 48. He told us about everything he and his board had been able to change, and


got us thinking again about what we were planning on changing. And honestly, apart from a few small things, we had been doing what all other boards had done.

And so as that famous scientist earlier wrote, we experimented. Instead of wasting paper, lunch lecture surveys became digital. Instead of individually, we asked the potential new committee members as a group. A group that, together with some veterans, would be faced with the challenge of producing this and several following editions of the MaCHazine you are reading right now. A group that is sure to be experimenting and thereby improving our MaCHazine in their own way.

The past three months have been amazing. The (Rubberd)AkCie brought students and teachers together at the ADSL drink, the WiFi got us all dancing in a packed Steck, but not after the MaPhyA made sure all double degree students played a couple of games of pool together. Several companies and lecturers have come by during lunch lectures and together with four other board members of different study associations here in Delft, and the ‘Bedrijven Informatie Team’ (BIT), we are well on our way to deliver you the biggest edition yet of ‘De Delftse Bedrijvendagen’.

Perhaps this is where my story deviates from that scientist I quoted earlier on. Whereas I am looking at all the hard work that remains to be done, just great individual perspicacity and some good fortune are not going to make that change happen. Together with you, all our committees, and the rest of my board, only together can we improve what we already have.

‘Change is ahead’, and with that thought I am very excited to see what the rest of this year has to offer. It’s been three months since September 4th as of writing this, and when you are reading this we are well over halfway through already. De Delftse Bedrijvendagen is about to open its doors to its Presentation Days, the SjaarCie is ready to get you dancing like a maniac and AreaFiftyLAN is set to once again be the most epic LAN-party ever. I hope to see you there, and I hope you enjoy reading this edition of the MaCHazine, just like always. I hope that may never change.

Oh right, that scientist, that was Christiaan Huygens. 

[illegible]



TU Delft news

Wouter Versteegh

The Delft University of Technology is the biggest and oldest public technical university in The Netherlands, established by King Willem II on January 8th, 1842. But what is currently happening in and around the TU Delft? This article will list the most important events of the recent months.

TU Delft researchers develop hybrid meta-biomaterial that can prolong lifespan of hip implants

Ten percent of the currently used hip prostheses will no longer be properly fixated in about ten years. This inspired some TU Delft researchers to develop a hybrid meta-biomaterial that promotes bone growth. Using a 3D printer and existing biomaterials, the researchers combined conventional meta-biomaterial with auxetic meta-biomaterial. Meta-biomaterials are the biomedical variant of metamaterials. These are materials that have characteristics which are not commonly found in nature, such as being extremely strong while being extremely light. Because of the auxetic material, the hip prostheses become much stronger fixated to the body, thus prolonging the lifespan of hip implants.

First liver cancer patient treated with microspheres in TU Delft irradiation facility

Medical isotopes play an important role in the medical world. They are used in diagnostics, imaging and radionuclide therapy which can be used to treat diseases like thyroid cancer or blood disorders.


The TU Delft have been working in collaboration with Quirem Medical and RadboudUMC to get medical isotopes with the required level of radioactivity to the patient. Recently in Italy, the first patient was successfully cured of liver cancer by using microspheres produced in Delft. This treatment uses spheres, about the thickness of a hair, filled with Holmium-165. Before those microspheres are

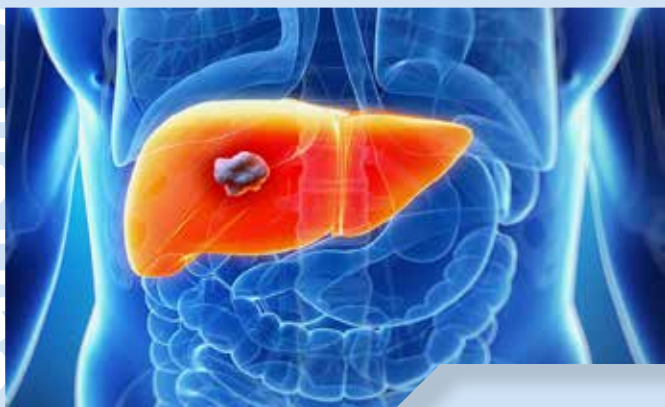
able to do their work, they first need to be activated. This activation happens in a new flexible irradiation facility that was recently developed by TU Delft's Reactor Institute (RID). Thanks to this TU Delft invention, patients around the world will now be able to benefit from this internal radiotherapy treatment.

Giulia Calabretta Best Lecturer TU Delft 2017

On the TU Delft Education Day, Giulia Calabretta was elected as the best lecturer of the TU Delft by a jury. Miss Calabretta is currently the course coordinator of the course Strategic Value of Design at the Industrial Design Engineering faculty. Every year, each faculty at the TU Delft elects the best lecturer of their faculty. For EEMCS, this was Mark Veraar. These lecturers are then nominated for the title Best Lecturer TU Delft. The jury, consisting of members of the Student Council, the Study Associations Council and previous winners then selects the winner of this TU Delft award. This winner is then automatically the TU Delft's nomination for the national ISO Teacher of the Year award.



According to the jury, what sets Giulia apart from the others is that she takes on many different roles that help her make a difference in her research, the Industrial Design Engineering faculty and the lives of her students. Her classrooms are defined by teaching based on discussion, where she stimulates discussion, debate and confrontation. 





Spiderman

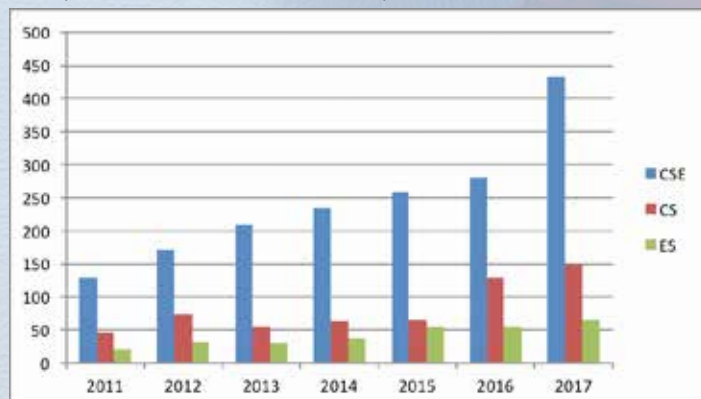
Hans Tonino

This is not a story about Spiderman nor about a red spider puppet (sorry...). Rather, in this article I will explain what is happening around the bachelor's program Computer Science & Engineering, I will spend a few words on the master's programs, and tell you something about two special projects that will be initiated. Finally, I will explain the title of this contribution.

As you all might know I finished my term as director of education of our faculty EEMCS and succeeded Emile Hendriks as director of studies in Computer Science just before Summer. Not quite a new job for me as I had been director of studies before Emile took over from me about six year ago. So, I knew or at least should have known what was expected from me. However, a lot of things have changed in the past 6 years.

The Bachelor's program

The most important factor has been the TU wide project called "Study Success". This project put a lot of emphasis on the quality of the bachelor's programs and their feasibility for students. Emile Hendriks did a very good job in transforming the bachelor program CSE to the requirements of Study Success. Last year 63% of the freshers got a positive BSA. And, as you know, we changed the official language of instruction to English after one pilot year in 2016-2017. The main reason has been the international character of CS itself and the fact that up to half of our lecturers does not speak Dutch.



Also, the program managed to attract more and more interested students (see figure). The intake of 2016-2017 already being around 300 that year, we were a bit shocked with 450 newcomers: a 50% increase. A real challenge with respect to teaching capacity. This year a number of courses are being doubly lectured in two groups. We also attracted two

new young lecturers to help with lecturing and organizing the enormous large numbers of teaching assistants. Both are doing a very good job. Is something else going to change? The answer is "yes". A curriculum committee, consisting of professors, lecturers, a study counsellor and two students, started preparing an update of the CSE program on request by Emile Hendriks one and a half year ago. The committee started doing a benchmark of the current program using the well-accepted ACM-IEEE-2013 curriculum guidelines. That means that our own curriculum was compared to these guidelines. It turned out that the current program lacks subjects like Cyber Security and Machine Learning. The committee also wants to strengthen the academic and research skills in the program as these turn out to be somewhat weaker when compared to other programs at TU Delft according to national student questionnaires like NSE. In the new curriculum, the organisation or contents of the Bachelor's Project might therefore be adapted. Finally, there will probably be more electives in the updated program. Of course, we will submit the new plans for advice to the Board of Studies and the Faculty Student Council, before implementing them.

The Master's programs

The intake of the master's program Computer Science is also rising significantly as can be concluded from the figure. This is also the case for the master's in Embedded Systems. In order to continue to be able to handle all master's thesis projects we will revisit the thesis procedures. In case of CS, we will also discuss the structure of the master's program and decide what we have to change to make the program more transparent to students. This project will start shortly.

Other news

Now something about the two special projects I mentioned above. First of these is called "Digital Skills". Recently the Executive Board of TU Delft decided that every bachelor program should have digital skills in their program objectives. We were asked to make a plan to this end. The idea is to develop a very flexible course consisting of several modules where students learn Python and learn to program solutions to assignments within their own discipline. The second project focusses on educational innovations in the bachelor's program. Some of the questions we will address are: How to implement academic skills in a scalable manner, i.e. for a large number of students? How do we stimulate the development of an academic attitude in students? And, how can we make assessments efficient, while retaining reliability, validity and transparency? We will invite Kristina Edström as external expert to help us carry out the project. Oh yeah, the title of this article... I got this Spiderman puppet from the EEMCS team of directors of studies at my farewell as director of education. They remembered that I once said that I wanted to be like a spider in its web, one who oversees the program and knows what to do. I hope I will be just that!



Is the Netherlands losing its winters?

Fred Vermolen

The elderly in The Netherlands sometimes claim that the winters they experienced in their childhood were colder than recent ones. With the exception of that, everything else seemed better when they were young. Are their claims true?

Climate change is a hot topic in the news. Most people and scientists are convinced that globally the climate is changing and, in particular, the temperatures are rising. Next to global changes, there are also many local changes in climate such as the withdrawal of the arctic polar ice sheet, areas suffering extreme drought or extreme precipitation and flooding. Despite these very clear signals, some people keep denying climate changes. Some people even claim that Earth is cooling down instead of heating up. These people often refer to the Northeast of the USA and the East of Canada, which, indeed, have been reported to be cooling down in the sense that winters seem to become harsher. We can even find some more examples of areas that seem to get more chilly winters, such as the Middle East. To write this column, I performed a little research using the data from the Dutch Weather Forecast Service (KNMI, www.knmi.nl). This institute possesses a lot of data regarding the history of the climate and weather in The Netherlands.

Another reason for climate change denial is the (nowadays rare) coincidental occurrence of relatively low temperatures in The Netherlands. This immediately suggests that some people are a bit confused with the concepts weather and climate. Let us define these concepts:

Definition: *Weather* is the state of the atmosphere; *Climate* is the statistics of weather over a long, connected period (mostly 30 years is used).

Weather phenomena involve temperature, humidity, atmospheric pressure, wind, and precipitation. Climate describes the aforementioned variables over a long time. Since it is easy to access the weather data from the KNMI and since most Dutch people are fond of ice-skating, preferably over lakes, rivers and ponds, the current article will engage itself with an analysis of the Dutch winters. The amount of cold weather can be described by average temperature and the so-called Hellman number representing the accumulated amount of frost over a period. The Hellmann number is defined as follows:

Definition: Consider a period of n full days, with a sequence of average full day temperatures in Centigrade $\bar{T}_1, \dots, \bar{T}_n$, then the *Hellmann number* over this period is given by

$$H = \sum_{j=1}^n |\min(0, \bar{T}_j)|.$$

Example: Consider two days with average temperatures of $\bar{T}_1 = -5$ and $\bar{T}_2 = 10$, then $H = 5$ for these two days.

The use of this Hellmann number is motivated as follows. Suppose that the temperature is constant and given by $\bar{T} = 10^\circ\text{C}$ during an entire winter. This virtual winter does not contain any frost and hence ice-skating on natural ice is impossible and therefore many Dutch people will be frustrated. Whereas, if an other imaginary winter contains daily averages of $\bar{T} = -5^\circ\text{C}$ during one

third of the season, and $\bar{T} = +15^\circ\text{C}$ during the remainder of the season, then the average temperature is given by $\bar{T} = 8.33$. This imaginary winter would probably be appreciated much more by the Dutch ice skaters because it contains one month of good conditions for ice-skating despite the fact that the second winter is much warmer than the first one. For this reason, we use both the average temperature during the winter and the totally accumulated Hellmann number to describe the amount of cold weather and frost during a winter.

Using the data from the KNMI, we generated an estimate of the accumulated probability functions for both the total average winter temperature and the Hellmann number over the years in time-periods of 30 years in Figure 1. From Figure 1, we can see that over the period 1987–2017 the probability that the average winter temperature stays below 2°C and 4°C is, respectively, given by approximately 0.10 and 0.5. These numbers change respectively to 0.37 and 0.87 over the period 1701–2017. Next to the average winter temperatures, we plot the histograms of the accumulated Hellman numbers over two periods: 1957–1987 (blue bins) and 1987–2017 (green bins). Looking at the data, it can be seen that the distribution looks like a Pareto distribution, although the Characterisation Theorem for the period 1987–2017 does not hold since the accumulated Hellmann number of the winter of 2013–2014 was zero in De Bilt. One might consider exponential or Weibull distributions as well. However, from the histogram, it can be seen, in particular for the time period 1987–2017, that Sturgeon’s Law seems to be satisfied: “ninety percent of everything is crap”. We will explain why this law has a point.

The German physicist Hellmann qualifies a winter as mild if its accumulated Hellmann number remains below 50. From Figure 1, it can also be seen that the probability of having a mild winter is about 0.47 and 0.65 for the time periods of 1957–1987 and 1987–2017. The probability to have a ‘cold’ winter ($H > 100$) is about 0.09 and 0.30 over 1987–2017 and 1957–1987, respectively. This implies that approximately ninety percent of nowadays winters in the Netherlands are not cold, and hence ninety percent of the Dutch winters are crap nowadays, which also implies that Sturgeon’s Law is (approximately) satisfied. If you look at the time-frame between 2000–2017, then it looks like even 100 percent of the Dutch winters is crap. However, a period of seventeen years is not representative in the sense of the definition of climate, which usually amounts to a period of thirty years. Looking at these data sets and at the curves in Figure 1, I was mostly struck by the fact that the cumulative probability function from the recent period 1987–2017 looks quite different from the ones from all the other periods. With respect to the Hellmann number, the appearance of the probability density function is very different for the period 1987–2017.

I also evaluated the (moving) average temperatures over the time (not shown here) and from that data set, it can be seen that the moving average annual winter temperature also significantly increases over the last thirty years. This increase is in a way that has never been observed in the period 1701–2017. The confidence intervals point at significant increases of the average winter temperature (and a significant decrease of the Hellmann number). This suggests that the climate in The Netherlands is changing significantly at this very moment and very rapidly. Even in the life span of an individual, the change of climate (at least the winter) is significant. I am still in my forties, and I can

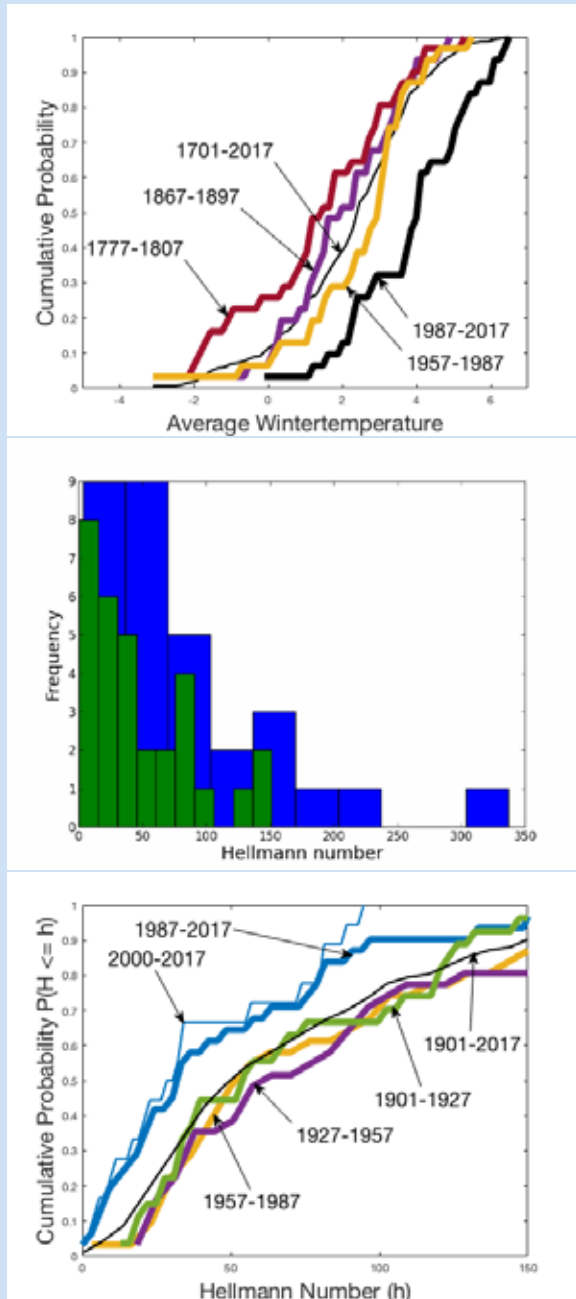


Figure 1: (1) The estimated accumulated probability that the average winter temperature (over the months December, January and February) is below a certain value for different time-intervals; (2) Histograms of the Hellmann number, green corresponds to the time span 1987–2017, blue corresponds to the time span 1957–1987; (3) The estimated accumulated probability that the accumulated Hellmann number is below a certain value for different time-intervals. All data have been extracted from the KNMI and are valid for De Bilt

already say that I experienced harsher winters during my childhood than the ones my children are experiencing currently.

From this, I could conclude that either I am already old, or that there is really something going on regarding the winters in The Netherlands. Probably, a change of patterns regarding the distribution of high-and low pressure areas in Europe is going on. It seems that high pressure above Scandinavia (which gives low winter temperatures in The Netherlands) has become rare nowadays. This observation is also consistent with the higher warming rate in regions close to the Arctic (such as Scandinavia). Even Svalbard (far above the Polar Circle), had rain with $+3^{\circ}\text{C}$ in February 2017 (The Netherlands had a small frosty period then by the way), instead of its usual deep-freeze temperatures.

A critical remark regarding this piece of text is that weather systems represent a pattern of the globe. This pattern of high and low pressures, the polar vortex could change with the composition and temperature of the atmosphere. It seems that during summer, the North West passage over Northern Canada and the strait between Canada and Greenland can be sailed. This was not possible earlier and this is a consequence of the ice having vanished on the North coast of Canada. The reason why this ice has disappeared is actually attributed to the absence of North Westerly winds on the northern coast of Canada over the past years. This wind used to blow the sea ice from the North Pole to the Canadian northern coast. Why has this wind disappeared? Why did the high pressure areas over Scandinavia during the winter season disappear? This is not known right now. Could it be the atmospheric composition like carbon dioxide or methane? Or, referring to the Scandinavian high pressure areas during the current winters, could this be caused by the huge numbers of turbines in the North Sea region? I don't know. Possibly if the global temperature increases more, then the vortex system and high and low pressure patterns could again become more favourable for the Dutch winters.

Although I seem to be pessimistic about this since in the past when global temperatures were much lower than currently, the winters had a lot of build up of high pressure above Scandinavia. This also means that next to global temperature rise, the Netherlands also experience a change of the wind pattern with even more wind from the mild Atlantic than before, and this means probably that the so-called eleven cities tour (Elfstedentocht) in Frisia will never be skated over ice again. One of the main things is that climate change should really worry us. I don't know whether we are able to prevent climate changes, however, taking good care of our environment seems a good idea. Further, we should adapt to a world with a different climate. This requires a lot of work for technology and policy makers. Hence, this poses nice opportunities and challenges for us!

Sorry for all my useless information; Thanks, for reading my nonsense; Time for *ALCOHOL*; Skål!!





What to take in the 3rd and 4th quarter?

Romi Kharisnawan

Imperceptibly, half a year has passed since the new academic year began.

For second year master students, I wish you the best of luck in doing your thesis and for first year master students, I hope that you've already gotten used to the rhythm of masters at the TU Delft and are ready for the next quarter. In this edition, I would like to share which Computer Science courses in Q3 and Q4 are interesting to follow based on my experiences.

In the 3rd quarter, I took all courses without an exam. It sounded as a great deal since I didn't have to study for written exams and I could enjoy the whole exam week as a holiday. But, that was not entirely true. There were still final deadlines in the exam week, which took the same amount of time as studying for exams. Moreover, there were no chill weeks during the quarter because deadlines usually came up on a weekly basis. Courses that I took in the 3rd quarter:

1. Information retrieval:

The course was interesting with several topics: web crawling, natural language processing, searching mechanism, crowdsourcing, etc. During the course you receive lectures by the professor, PhD students and a company. In term of load, it was pretty demanding because of many components contributing to the final grade: weekly assignments, final project, and an individual survey paper. The final project was the most interesting part of the course, yet took the most time. The highlight of the assignment for me was group work, where you have your responsibility to make the project successful, and also involvement of PhD students who helped my team to shape the project. Not only to have a working product in the end but also reasoning why the project has meaningful value.

2. Seminar Research Methodology for Data Science

The course is in seminar setup, which mainly divides into 3 parts: analysis of experimental research data, exploration and validation of a real world dataset, linear and nonlinear problem solving in data processing. Each part consists of a written paper and presentation. I found this course interesting because it helps you a lot to deal with analyzing data. Moreover, I love the setup of this course which pushed students to learn three popular programming languages nowadays, namely R, Python, and Matlab.

3. Software Architecture

This is also one of my favorite subjects because of the setup of the course, energetic lecturers, and its practicability. As a spoiler, this course requires a lot of time and effort because there were deadlines every week. If you are an aspiring software developer or architect, you should take this course! You will work a lot with github, which I found really helpful as I had limited experience with it. Don't be ashamed not to know much about it; the lecturers had a special github session and you will learn more with each week's deadline. During this course we only had lectures where you meet with companies, such as Google, ThoughtWorks, Bunq, Neo4J. I found it really helpful and inspiring because you get insight from someone who already has experiences in real world industry.

4. Intelligent User Experience Engineering

This course runs for 2 quarters with a 6 ECTS load. There were two main topics: children with cancer-robot interaction (work with NAO) and elder people with dementia-robot interaction (work with Pepper). This course helps in building framework of interaction needed between human and intelligent agent. The highlight of this course is the project where I dived in on the real experiment. My group conducted experiments with real children as participants. We analyzed the effect of nuanced and explicit response by a robot to children.

In the 4th quarter, I added 3 more courses:

1. Multimedia Search and Recommendation

This course is also interesting to follow. It has two different tracks: analytics and system. I took the latter which focuses on the technical side of the multimedia domain, such as compression, color channel, etc. At the end of the course, we had to develop a system (you can choose your own topic) related to this domain. My team worked on classifying genres of songs.

2. Cyber Data Analytics

You should take this course if you are interested in cyber security. This course elaborates the techniques of identifying attacks using different kinds of approach, such as fingerprinting. Besides that, it might pull your interest in data analysis as well since digital attack datasets are usually imbalanced.

3. Web Data Management

This course is interesting to follow if you are keen to know more about database technology from relational to unstructured databases. It helps you to understand the concept and implementation of different kinds of databases, such as MongoDB, Cassandra, etc. You get to choose between giving a presentation about a specific topic or implementing an interface for using different kinds of database. So, I hope my experience might help you in choosing your courses during these quarters. Good luck with your study!

Studytrip to Milan!

Fill in the interestlist: wisv.ch/icom
and we will keep you updated!

12th June - 16th June

Location: Milan, Italy

- ✓ Educational
- ✓ Career
- ✓ Social

[illegible]



New website!

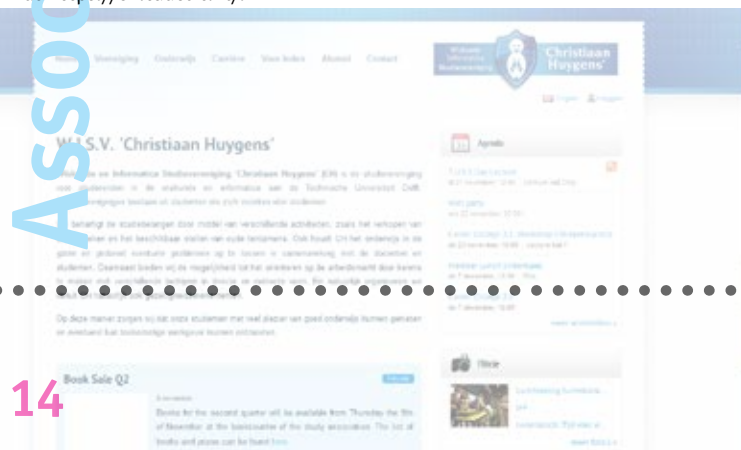
Sven van Hal

The online presence of W.I.S.V. 'Christiaan Huygens' has never been better: for a couple of months now, the association has a brand new website. The old site was starting to show its age, which prompted us to build a new one. With a fresh redesign and a variety of new features, we're ready for the future.

Let's take a step back in time. Board 52 introduced a completely revamped corporate identity for the association. The logo was redesigned, and with it the posters, envelopes, letterhead, business cards, and: the website. This means that the old design dates back all the way to 2009. There have been lots of developments in the field of web design in the meantime. However, we only marginally improved our site.



With that in mind, we have had a new website designed in line with contemporary standards and requirements. We needed the website to look fresh and clean, be accessible on a wide range of screen dimensions and better provide our members with the information they're looking for. A dedicated committee called the W3Cie then built the new website from the ground up, using the latest web technologies. Now we are finally able to share the result of our work at <https://ch.tudelft.nl/>.




The main new feature of the new website is the possibility to manage your own profile online. From now on, you're able to view and change your personal details via the website yourself, whereas you previously had to send an e-mail to the secretary of the association. Furthermore, activities are now categorized online like we have been doing 'offline' for a couple of years now, so you can more easily understand whether an activity is for example social or educational. All of this in a redesigned, responsive package.



This milestone, the launch of the new website, is not the end of the road (at all). We will continue to develop and add new features in the future. On the roadmap are currently the possibility to sign up for activities online, including purchasing tickets for any paid events, and adding more functionality to the members-only area. In the meantime, we are monitoring and resolving any issues users might have with the website.

Would you like to contribute to the website? You are welcome to submit any feedback or suggestions using the Feedback Tool (<https://wisv.ch/feedback/>) or at our GitHub-repository at <https://github.com/WISVCH/website>.

We hope you'll enjoy the new site! 





Comittee kick-off 2017!

MeisCie 2017-2018

A new year has started, so that means that new committees will be formed! One of these committees is called 'MeisCie'. The members of this group are only girls and we organize three activities for girls during the coming year. To meet the members of the other committees, a Committee Kick-Off was organized. During this Kick-Off we could show the other committees how incredibly awesome the MeisCie is.


The Committee Kick-Off was also a pleasant way to meet each other. On Monday the 13th of November 2017 we assembled at EWI at 5:30 PM. Then we went on our bicycles to our destination, which was the Delftse Hout. It was about 15 minutes cycling. When we arrived at our destination, we were split into two groups of 50 members. One group went fencing and the other group had to play a game called "Levend Stratego".

MeisCie started with the game Levend Stratego. For this game they also had to split us up in two groups. Every participant of the game got a card which contained a number, a name and a list of players that you could eliminate. The goal was to find the flag of the opponent's team to win the game. After the game started, we didn't play it very seriously, because it was cold, dark and wet; Because we were outside.

First, our group ran away and the others just stood there waiting. Then we just stood there with a group of people and were asking each other which number we had on our cards. Our team then found the flag of the opponent's team and therefore we won.

Fifteen minutes later we decided to visit the other group, who were fencing. Most of us were looking forward to fence as well and yes, it was great to do! We learned some new techniques with the sword. Most of us never did that before, so we really liked it :) In the end we could fence against each other, so in pairs we fenced against each other and this was exciting to do.

After hitting each other with a sword and a lot of fun later, we were not done yet, because it was time for dinner!

Again we went on our bikes, this time to 'Proeflokaal Sijgje'. The whole restaurant was booked for us. Here we got a three-course menu. The food was delicious and the dessert was even more delicious ofcourse :P. The dessert consisted of mini desserts like a mini brownie. Yammie! We sat at the table with members of the 'SjaarCie' and we socialized with them. When we all got our main course, Willemijn of the board gave a speech. After dessert it was time to go home and say goodbye to everyone. After all, the Committee Kick-Off was really sociable. 



Association



Do you want to keep receiving the MaCHazine?

Fill in the form at wisv.ch/machazine after the third edition of this year.

Otherwise, you will be automatically unsubscribed!

The next edition (Volume 22, Issue 3) will still be send to you automatically. If you want to keep receiving the following issues, you'll have to opt-in via the link. Also help us by filling in the survey at the end of the form! This is optional, but will only take a few minutes. Thank you!



AkCie

Fleur Rooijackers

We are the RubberdAkCie! The activities committee (AkCie) of the year

2017-2018. Our committee consists of twelve members. These members

are: Ines Roebroek (chairman), Diederik Heijbroeck (treasurer), Bram

Dikker (secretary), Lieke van der Linden (commissioner buy-in), Moos

Castelijm (commissioner buy-in), Sterre Hoek van Dijke (commissioner

furnishing), Ivor Zagorac (commissioner music), Tristan Cotino (commis-

sioner promotion), Fleur Rooijackers (commissioner Promotion), Arkin

Zoodsma (commissioner AkCievity) and of course our two QQ'ers Daphne

van Teteringen and Niek van der Laan.

Our job as a committee is to organize a number of fun activities throughout the year, with the goal to have fun with your fellow students outside of the lectures at the university. These activities, amongst others, include member lunches and having drinks. We also organize a big party in the theme of our committee. So far we have already organized two activities: the ADSL drinks and the SinterklaaslunCH.

The beginning

If you want to become part of a committee, you have to express interest in the committee. After letting the board know you want to become a part of a certain committee, you will get asked by the QQ'ers (supervisor and board member) of that committee to join. In our case, each of us got a text message saying we had to go to the 20th floor of the EWI building, and we had to bring something which expressed our love for owls. There, our two QQ'ers Daphne and Niek were waiting for us and asked if we wanted to join.

After we all said yes, it was time for the committee kick-off! This is where all the committees are revealed, and do a fun activity together. This year we all went fencing, which was great! Afterwards, we all went to grab a bite to eat and to have some drinks. Overall, the committee kick-off was a great success!

Then it was time for our first meeting as a committee. In the first meeting we determined who got which role in the committee. We also had to come up with a theme for our committee. After a lot of different ideas (tAkCie, kat in het bAkCie, kernreAkCie, etc.) we came up with RubberdAkcie (rubber duck). For our committee color we chose maCHenta, because we like pink and CH. In order to sufficiently represent the RubberdAkCie, we chose a maCHenta sweater with a rubber duck and maCHenta suspenders to tie the look together.

ADSL

After we established ourselves as a committee, it was time for our first activity: the ADSL. This is an activity in the /pub (the pub below EWI) with professors from Applied Mathematics and Computer Science. The fun thing about this activity is that the professors get free drinks, and the students don't. This has the aim to make students integrate (or differentiate, haha) with professors in order to get free drinks. The turnout was pretty big, and everyone seemed to enjoy themselves. After loads of free drinks and integrating, the activity had to come to an end. This activity was not too difficult to organize, since the only necessity was free drinks, which were already available in the /pub. We did buy some extra snacks and some fun decorations (with rubber ducks cut out) to brighten up the pub. Our commissioner promotion Tristan also designed a cool poster to promote the activity. Overall, organizing the first AkCievity went pretty smoothly.

SinterklaaslunCH

After the ADSL, it was time for us to organize our first member lunCH. This lunCH took place on December 7th, and the theme was Sinterklaas. At the member lunCHes you get a big lunCH for only one euro. These lunCHes usually take place in the /pub below the faculty. That way, you are able to quickly go back to your lectures after the lunCH has finished.

Organizing the lunCH was a bit more of a hassle compared to organizing the ADSL. The biggest task was to buy enough food. This is where our commissioners buy-in Moos and Lieke came in and bought loads of groceries. Of course they also bought loads of pepernoten. Aside from the groceries, our commissioner promotion Tristan again designed a cool poster, our secretary Diederik printed the tickets and the commissioner music Ivor composed a special Sinterklaas playlist. The tickets sold out pretty quickly, and the /pub filled up with students and pepernoten. Sinterklaas and Zwarte Piet even came back from Spain to attend the lunCH.

After Sinterklaas and Zwarte Piet left, it was time for the RubberdAkCie to clean the /pub. Thanks to Zwarte Piet, it took quite a while to get rid of pepernoten, but thankfully we had the commissioner music's amazing Sinterklaas playlist to listen to whilst cleaning.

We learned a lot whilst organizing these activities, and we hope to see you at the next one! 🐥

Association



KICK START YOUR CAREER!

Every year 'De Delftse Bedrijvendagen' (DDB) leads over 3000 students to the start of their career. If you want to hit the ground running this year, make sure you do not miss this event. Whether you are job hunting or in need of an internship or graduation project, DDB offers a unique opportunity to get to know a wide variety of high-profile companies.

DDB is the largest technology oriented career event in the Netherlands, and takes place each year in the Aula of the TU Delft. This career event is specifically aimed at the students of Delft University of Technology and offers them a wide range of national and international companies and institutions.

DDB KICK-STARTS YOUR CAREER

- 1 Receive tips and tricks about job interviews during the Application Trainings
- 2 Get to know over 150 companies during the Presentation Days
- 3 Go to the companies during the In-house Days to get an even better idea of each company and experience their work environment
- 4 Have personal conversations with recruiters during the Interview Days

Contact

De Delftse Bedrijvendagen
Secretariat
Mekelweg 2, 2628 CD Delft
Tel: (015) 27881 900
Email: ddb@ddb.tudelft.nl

PARTICIPATION

All activities described above are included in a single price when participating in DDB. On top of this, you also receive a full-color magazine with tips and tricks for a successful start of your career. You can participate by signing up via our website, ddb.tudelft.nl, or by coming to the Aula Congress Centre on the **23rd, 24th or 25th of January**. Until **January 25th** it is possible to register for a reduced fee. Hence we advise you to register before **January 25th**.

All personal information will be treated strictly confidentially. So if you are looking for a job, an internship or a graduation project, subscribe now at ddb.tudelft.nl.

STARTUP AREA

After the success from last year, the startup area will again be present in 2018 at 'De Delftse Bedrijvendagen', in cooperation with YES!Delft. During the presentation days 24 startups will be present spread over 3 days. The startup area can be found on the floor in between the auditorium and the 1st floor. New this year is that startups will get the opportunity to present themselves in a short pitch given twice a day.

In short, find your job, internship or thesis project via DDB!



WHAT DO **DE DELFTSE** **BEDRIJVENDAGEN** OFFER YOU?



Application Trainings

Prepare yourself for your job interview and the fair

On the **13th, 14th and 15th of February** DDB offers Application Trainings. The companies will provide you with both general tips and tricks as well as detailed personal advice. During the Application Trainings you can also have your résumé checked.



Presentation Days

Get to know over 150 national and international companies

On the **19th, 20th and 21st of February** DDB hosts its most well-known event, the Presentation Days in the Aula of the TU Delft. This technical career fair gives you the opportunity to get to know over 150 different national and international companies!



In-house Days

Take a look behind the scenes at the companies

From the **14th** until the **29th of March** the In-house Days offer an opportunity to form a better, more complete idea of the companies you are interested in. Your résumé will be forwarded to the companies of your choice. Based on the résumés the companies will select the participants. You can sign up for the In-house Days until the **25th of February**, the Sunday after the Presentation Days.



Interview Days

Take part in one-on-one interviews with the companies

From the **14th** until the **25th of May** DDB will be completed with the Interview Days. During this period, there are two different kind of interviews possible: orienting interviews or even the first step in an application process. For the Interview Days your résumé will be sent to companies you are interested in. You will receive an overview of which companies are interested in meeting you, and from these you can make a selection. You can sign up for the Interview Days until the 4th of April.

IMPORTANT DATES

8 January	Registration open
25 January	Deadline reduced (entree fee)
23 February	Deadline résumé upload
25 February	Deadline registration In-house Days
4 April	Deadline registration Interview Days

13, 14 & 15 February	Application Trainings
19, 20 & 21 February	Presentation Days
14 - 29 March	In-house Days
14 - 25 May	Interview Days

Organization

DDB is organized by five study societies, that form 'The Pentagon' together:

Vereniging voor Technische Physica Applied Physics
Gezelschap Leeghwater Mechanical Engineering
Technologisch Gezelschap Chemical Engineering
VSV 'Leonardo da Vinci' Aerospace Engineering
W.I.S.V. 'Christiaan Huygens' Applied Mathematics and Computer Science



Area FiftyLAN 2018

Bram van Walraven

As we all know, a large part of the computer science students are excited for computer games. The amount of screens showing Hearthstone is nowhere as big as in first year computer science lectures. It's not surprising a student association for these students organizes one of the biggest gaming events for all students in Delft: the Area FiftyLAN.

Since the first edition of the Area FiftyLAN in 2015, the event has grown to a weekend long event for around 220 people. For a whole weekend they will participate in tournaments for games like League of Legends, Rocket League and Hearthstone. For the winners of each of these tournaments there are some very nice prizes. In addition to these official tournaments, every year there have been tournaments in almost any game organized by the participants themselves. Nothing limits the ability to play the games one likes.

However, as our study association is not only for the group of gaming computer science students, there will be a lot of other possibilities for people who do not enjoy (hardcore) gaming as much. You could think of the LAUNCH area with consoles, where there are games like Just Dance or other more softcore games and drinks! There will even be a poker tournament, for everyone who likes games which do not need any screens as well. This makes the Area FiftyLAN fun for every student in Delft. And the best thing is: because everyone who is busy gaming won't have the time to take care of their food needs, all meals during the weekend will be provided for.

This whole event is organized by the LANcie. The committee, consisting of six CH members, has been busy since the start of September to make this year's edition even more exciting than the past three. In numerous meetings the possibilities for new additions and improvements have been discussed: the possibility to be able to order snacks at your table or to play on old-school arcade machines. Although the whole committee is excited by all these interesting new ideas, it is not possible to promise everything will make it to the final event.

Right now the committee has finished working out the basics for the whole weekend. Tables, chairs, layouts and carpeting has been taken care of and now it is a matter of working out the fun stuff: finding companies for collaborating to get the most awesome prizes and, as said before, scraping away the too many ideas there are to make this event the most fun for everyone. All, of course, under tight supervision of the board of CH, and especially the amazing Francis Behnen.

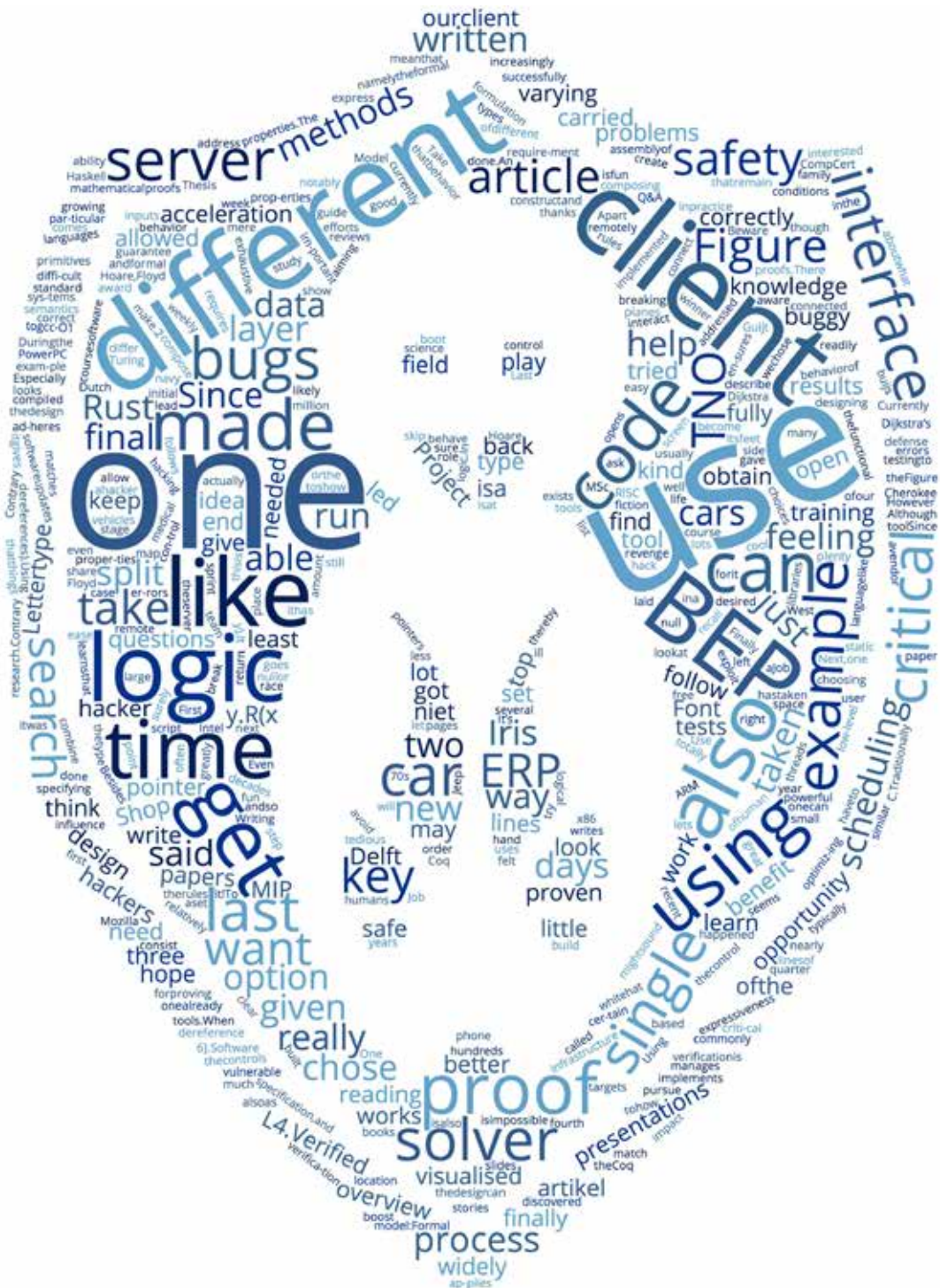
After the holidays it is most likely you will see some nice banners and flyers being handed out to make sure as many people as possible will know about the Area FiftyLAN. So make sure to like our Facebook page to be informed about any giveaways and to keep 2nd, 3rd and 4th of March free to be able to participate in one of the biggest gaming events in Delft, especially for students. For any questions you can always contact members of the LANcie or the CH board.

See you all at the 4th edition of the Area FiftyLAN,

The LANcie,

Jip, Bram, Jurriaan, Maaïke, Laetitia, Dieuwer and Francis 

Computer Science





Formal verification

Ensure the absence of bugs using mathematical proofs

Robbert Krebbers

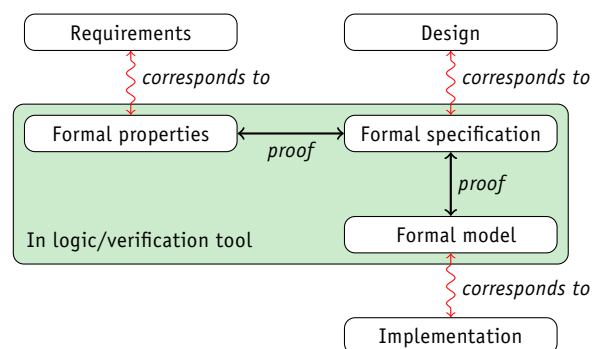
The field of *formal verification* makes it possible to use mathematical proofs to ensure that software does not crash and behaves as it should. This is especially useful for safety critical applications where ill behaved software can lead to dangerous, or even life threatening, situations.

Since software is used more and more in security sensitive and safety critical applications, software errors can have serious consequences. For instance, the controlling (steering, braking, and acceleration) of a modern car is carried out by software instead of mere mechanics. At the same time, there is a tendency to connect cars to the internet, which opens up the opportunity for hackers to take over control of a car while it is in a totally different place than the hacker itself. This might sound like science fiction, but it actually happened: white hat hackers have been able to remotely take over the control of actual cars by hacking into the entertainment system. Fiat-Chrysler had to recall 1.5 million vehicles for software updates because of this hack [?]. Software problems do not just concern cars, but also planes, defense systems, critical infrastructure, medical systems, and so on. In this article I will describe how logic and mathematics can help to avoid these problems.

Formal verification

Formal verification is about specifying the desired behavior of software using logic, and using mathematical proofs to show that the software satisfies these specifications in logic. In order to get an idea of how this works, let us take a look at the way software is commonly developed. When designing software, one usually starts with the *design*: an overview of the different components, with descriptions of how these components should behave and interact. Next, one writes down the *requirements*. For a car, a key requirement is that the entertainment system cannot influence the controls

(steering, breaking, acceleration). That way, if a hacker manages to break into the entertainment system, or the phone connected to it, the hacker cannot take over the control of the car. Finally, there is of course the *implementation*, which for a car, is likely to be written in a language like C. Traditionally, one uses methods like code reviews and testing to establish that the implementation correctly follows the specification, and the specification satisfies the requirements. In formal verification, there are formal counterparts to the design, requirements, and implementation in logic, namely the *formal specification*, *formal properties*, and *formal model*:



Since these formal counterparts are all written in logic, one can use mathematical proofs to formally establish that the formal model correctly implements the formal specification, and that the formal specification satisfies the formal properties. By composing these mathematical proofs, one learns that the formal model satisfies the formal properties.

The idea of formal verification goes back to at least Hoare, Floyd and Dijkstra in the 70s. Although they laid groundbreaking work, formal verification did not really get off its feet back then: to verify relatively small programs, one already needed several hundreds of pages of mathematical proofs. Writing such proofs by hand is tedious, and it is impossible for humans to verify that these proofs follow the rules of logic. During the last decades, formal verification has made its revenge. Since computers became more powerful, it has become possible to implement formal verification methods as tools, and thereby use computers to help us construct and verify said proofs.

There exist a number of different formal verification methods, and for all of these methods these are different tools. When choosing a method (and a corresponding tool) there is a trade-off between logical expressiveness and the amount of human guidance that is required. For proving safety properties (for example, the program does not dereference any null or dangling pointers) less human guidance is needed than for proving functional correctness (for example, the program adheres to a detailed specification of a protocol). Figure 2 gives an overview of the trade-offs between popular methods:

- **Static analysis** is limited to proving safety properties, but is very effective at that: it is fully automatic and requires little human guidance. It is used in many large software projects, not just for critical software, but also as an effective way of finding bugs, for example: by Facebook [?].



Figure 1: This Jeep Cherokee was vulnerable to a remote exploit.

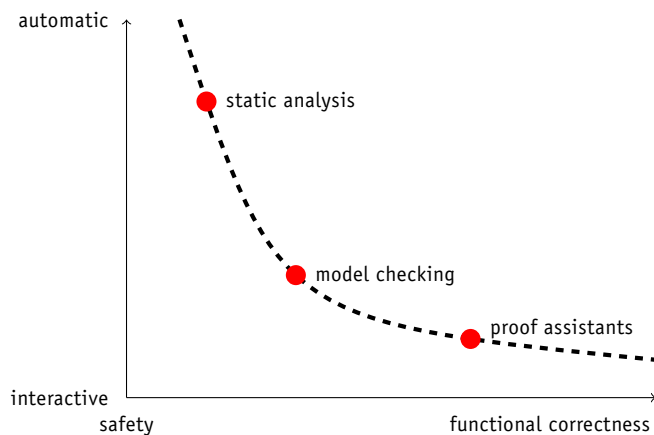


Figure 2: The trade-off between formal verification methods.

- **Model checking** is a method to establish that a model satisfies a specification for all possible inputs by exhaustive search. Contrary to static analysis, model checking is not limited to safety properties, but requires humans to write down specifications and properties. Since model checking works by exhaustive search, it typically only applies to programs that can be given a model with a finite state space. Model checking is widely used for hardware verification, for example, by Intel [?].
- **Proof assistants** allow one to write down specifications, properties, and models in very expressive (higher-order) logics. Since they allow one to express nearly any property one could think of, human guidance is necessary for carrying out proofs. As I will show in this article, proof assistants have been successfully used to verify the functional correctness of complex software such as operating systems and compilers.

Once a program has been formally verified, does this mean that it is surely free of bugs? As Turing award winner Knuth famously said “Beware of bugs in the above code; I have only proved it correct, not tried it.” The answer is thus ‘no’, one has to be very careful that the formal specification and properties correspond to the informal design and requirements, and that the formal model matches up with the actual implementation. For example, in the specification one may have written $\forall x.\exists y.R(x,y)$ instead of $\exists x.\forall y.R(x,y)$. It is therefore important to combine formal verification with traditional testing methods to make sure that the specification and model match up with reality.

Success stories in critical software

An important concern when using formal verification is the accuracy of the model: does it correspond to the implementation that is used in the actual system? Even though it is possible these days to use actual programming code (in languages like C) as a formal model, we need to think about what happens when we compile said code with a buggy compiler, run it on a buggy operating system, or run it on buggy hardware.

Two large verification projects have addressed this concern: they have used proof assistants to formally verify an operating system and compiler. These efforts were groundbreaking for formal verification.

L4.Verified (OS). L4.Verified [?] is a high-performance microkernel operating system of the L4 family. It consists of 8.700 lines of C code, of which 7.500 lines have been formally verified (it excludes the boot code). The specification is given as a executable prototype in the functional programming language Haskell, on top of which a collection of security properties has been proven

(notably integrity and confidentiality). These proofs consist of about 200,000 lines of proof script in the proof assistant HOL/Isabelle. During the verification process, 160 bugs were discovered in the C implementation.

CompCert (C compiler). CompCert [?] is an optimizing compiler that targets x86, ARM, PowerPC, and RISC-V assembly. CompCert’s performance is comparable to `gcc -O1`, and has been implemented and formally verified in the Coq proof assistant. The specification of CompCert involves a *formal* semantics of both the C programming language and assembly. It has been proven that if a program has a certain behavior according to the C semantics, it also has that behavior according to the assembly semantics.

Next challenges


About 50 years after Hoare, Floyd and Dijkstra’s seminal work on formal verification, it finally starts to be used in practice thanks to tool support. Especially in safety critical systems (planes, defense systems, critical infrastructures, ...), formal verification starts to play an increasingly important role. However, there are still plenty of things that remains to be done.

An important challenge is to be able to compose the proofs of different systems that have been formally verified. For example, one may want to compile L4.Verified using the CompCert compiler to obtain the guarantee that the compiled assembly of L4.Verified also satisfies the specification. Currently this is not possible, both projects have been carried out in different proof assistants and use a different formulation of the C semantics. As such, the correctness proofs of both projects cannot readily be composed.

Another challenge that I pursue in my own research is (fine-grained) concurrency. Concurrent algorithms are very difficult to get right, especially when they use low-level concurrency primitives to implement sophisticated synchronization, and as such, they could benefit greatly from formal verification. To address this problem, together with a growing network of collaborators, I have developed Iris [?], a logic for verifying concurrent programs. This logic is build on top of the widely used Coq proof assistant.

One of our recent applications of Iris is the verification of the type system and standard libraries of Rust [?], a new safe systems programming language developed by Mozilla research. Contrary to conventional programming languages, Rust ensures the absence of race conditions (two threads accessing the same memory location at the same time) and memory errors (such as null pointer and dangling pointer dereferences). Using Iris we have proved that any well-typed Rust program indeed satisfies these properties [?].

Interested?

If you are interested in formal verification, you should consider the *programming languages* master program. In particular, you should follow my MSc course *software verification* (CS4135) in the fourth quarter. 

The adventure of a BEP: Support for human planners

Cas Buijs

The last chance to give everything you got and to get something in return during your bachelor is with the Bachelor Thesis Project (BEP).

It's a great opportunity to make some impact, so use the experience of your predecessors to help you get the most out of it!

To obtain your bachelor degree, you have to overcome the last challenge: the BEP. It tests the student's ability to work in a team, independently from the university, to solve a real-world problem. Last year, together with Arthur Guijt and Lars Stegman, I worked on a project called 'Support for Human Planners'. Through this article, I'd like to share my own experiences with you.

The Problem

The company providing our project, West IT Solutions, is a vendor of an enterprise resource planning (ERP) system that allows companies to centralise business data and use it for different applications existing in the same system. Our challenge was to create a product that could be used to demonstrate the possibilities of supporting human planners, using the data available in the ERP system. The product should visualise a planning made in the ERP system, it should allow manual adjustments being made to this visual planning and be able to propose an automatically generated improvement of the planning. As a planning often adheres to a set of rules, we used a case study of TNO concerning the scheduling for battleships of the Dutch navy to guide us here.

Preparations

A project starts with preparations to ease whatever comes afterwards. The preparations that are required differ for each project and are not limited to the process of programming. Meetings, research and planning are key components of this stage, so take your time to figure out what you really need.

The need of both a set of requirements for the product and a set of rules for the planning led us to have meetings with our client but also with TNO employees. These meetings demanded a lot of initiative from our own side, allowing us to experience how things are in the real world. Being fully prepared for the meetings was not enough, having the thought that you got what you wanted only to realise afterwards that things were missing. A lesson to remember is to always keep an option open to ask for more information, a single e-mail saved us from a lot of headaches. Apart from the new experiences these meetings brought, they also allowed us to look inside two cool companies during a single project!

The product required a visualised planning that could be manually adjusted. Besides reading academic papers to come up with different approaches that could be taken, there was also some research into how similar products answered our research questions. Together with our client, we then used all the knowledge gained from the research to define a list of requirements for the

product. Knowledge about competing products was very helpful to defend particular choices, as referring to how things are currently done in the industry seems to be very effective!

The initial design of the system was based on personal experiences and decisions made in the process of creating the requirements list. The product was split into two components, the interface and the back-end, and had to follow the single responsibility principle. For the interface, drawings were made, like in Figure 1. Using paper for this gave us a few advantages; it was not only easy to adjust the drawing during the meeting with the client, but a better atmosphere emerged as well without a screen in between. After agreement by the client, the next step was to 'draw' it on a computer to give the client a better feeling of what things would look like.



Figure 1: Drawing of the visualised planning

The last preparation to be taken also required most of our time. As a scheduling problem was also involved, we created a mathematical model of the problem and proved its complexity to be NP-complete. Having this knowledge allowed a more structural search for approaches that could be taken to solve our problem. Using advice of both our supervisor and client, who both have lots of knowledge in the algorithmic field, we tried to map our problem to existing problems, like Resource Constrained Project Scheduling, a Job Shop variant and Mixed-Integer Programming (MIP). All of these approaches, besides Job Shop, are not taught in the bachelor and it felt like following a mini-course during the BEP. First, reading in books and papers about the different problems for a better understanding, and then analysing the possibility of a mapping from our mathematical model to this problem's model. In the end, we chose to develop MIP models to solve the scheduling problem.

Implementation Phase

Having finished all preparations it was finally time to start the development of the product in weekly sprints in accordance to the SCRUM methodology. Each sprint contained a feedback session with our supervisor and client regarding



the process and the research, and at the end of the week there was a feedback session with our client to discuss the product.

The product was split in the interface and the back-end (server) and we wanted to keep it as independent from other software as possible. The server had to communicate with both a database, containing the data in the ERP system, and a solver, to solve our problem model. To overcome the dependence on a specific solver, an abstraction layer between our system and the solver was built. This allowed multiple solvers to be used and provided the client with a larger degree of freedom in this matter. The interface was extended to support multiple projects and multiple functionalities were added to the visualisation of the planning; figure 2 shows the final look. For the communication between the interface and the server we chose to make use of REST, allowing the interface to make requests to the server for information. As the interface and server were split, it is possible to run the server on a dedicated machine to improve the solver's performance, and thereby the performance of the complete product. We chose to use frameworks for this project which we had little to no experience with and that really made the challenge more interesting. Remember that you only get a single opportunity, so do not try to always play it safe and make the BEP worthy of being a life experience!

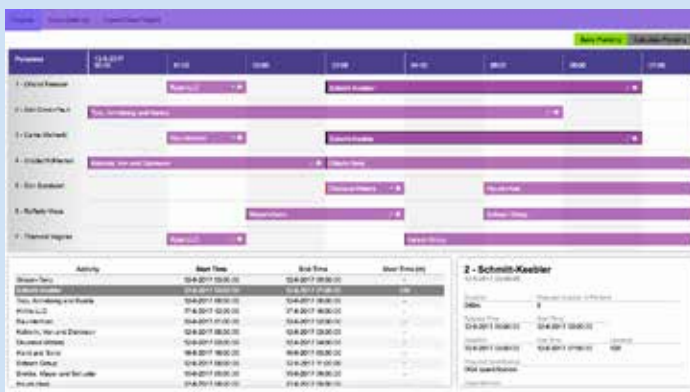


Figure 2: Interface for creating a planning

In order not to be stuck with a single choice, we created three different MIP models that could be used to solve the scheduling problem. As we chose to use an abstraction layer between the server and the solver, it became necessary to analyse its influence on the performance of the product. Experiments with varying parameters, like the number of activities in the planning and the number of days the planning covers, were performed and both the time necessary for building the model and solving the model were recorded. There were two types of tests used: for the first type, all instances were given to the solver through the abstraction layer, and for the second type, they were given directly through the solver's interface. The results were then converted into diagrams, like in figure 3, and resulted into a recommendation to our client for making use of two models at once. One to find a solution and another to discover infeasibility of the model.

Presentations

After finishing the product only one thing was left: presenting the new product as a supreme masterpiece! There were three presentations scheduled, one for

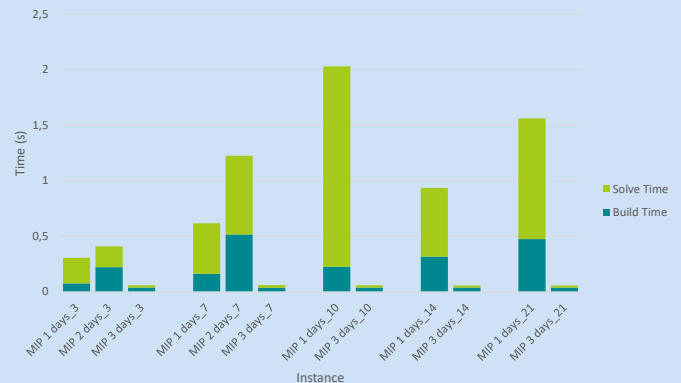


Figure 3: Experiment results of varying the number of days the planning covers

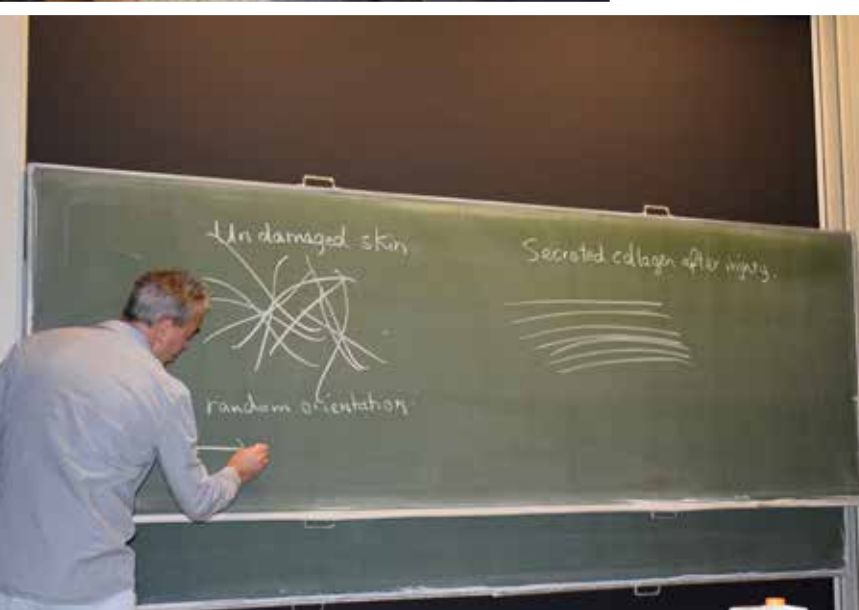
our client, one for TNO and the last for the committee of the TU Delft. Each of the presentations required different perspectives as each party had different interests; this led to three different kinds of presentations in order to be prepared for different kinds of questions being asked during the Q&A's. The presentations for our client and TNO were used as stepping stones for the last presentation, learning from the feedback gained and improving both the slides and Q&A accordingly. I can strongly recommend everybody to make use of multiple 'training' presentations, including at least one for your client, one for your supervisor and perhaps one for a potential user of your product. It is fun to do, lets you get used to the feeling and makes you more aware of what aspects of your project people are interested in. Everything you learn from them can then be used to boost your final presentation!

Take-aways

When the moment comes that you are the one being challenged by the BEP, remember the experience shared here with you and use it to your greatest benefit. I hope that it was made clear, by the emphasis on research, that research is a key component of the BEP. You are aiming for an academic degree, ignoring research is not an option anymore and it is also too much fun to want to skip. Here 5 things I hope you remember when starting your adventure:

1. Take your time to discover what preparations you have to make.
2. Prepare for the meetings and keep an option open to ask more questions.
3. Enjoy additional challenges throughout your BEP, it gives a crazy good feeling!
4. Use 'training' presentations to get used to the feeling and to learn what interests others most about your project.
5. Research is the key point of this challenge, use that to your advantage!

Interested in learning more about our project or want to know what a final report looks like? You can find our project in the TU Delft repository: <https://repository.tudelft.nl> (choose the education repository and just search on my name 'author:cas buijs').



[illegible]

The Hilbert-Pólya conjecture

Jan van Neerven

In Linear Algebra it is shown that each self-adjoint matrix (i.e., a matrix A whose coefficients satisfy $a_{ij} = \overline{a_{ji}}$) has real eigenvalues and is orthogonally diagonalizable (i.e., there is an orthogonal basis of eigenvectors). In the master's course Applied Functional Analysis we prove an infinite-dimensional generalization of this result, the so-called spectral theorem for bounded self-adjoint operators. There is also a version for unbounded self-adjoint operators such as the Laplace operator Δ . In this article, which is based on my presentation at the TU Delft Best Graduate of 2016 event, I will discuss a possible connection between the spectral theorem and the Riemann hypothesis, probably the best known and, according to many, the most important open problem in all of mathematics.

The Hilbert-Pólya conjecture is the conjecture that such a link indeed exists. It exemplifies the deep and mysterious relationship between mathematics and theoretical physics - except that in this case the relationship is conjectural: almost nothing has been proven rigorously!

Let us first explain the Riemann hypothesis. Our starting point is Euler's formula

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

There are many ways to prove this formula, three of which could be given in the bachelor's curriculum: an elementary calculus proof, a proof using Fourier analysis, and a proof using the theory of complex functions. The latter proof easily gives more: by the same method one obtains the identities

$$\sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}, \quad \sum_{n=1}^{\infty} \frac{1}{n^6} = \frac{\pi^6}{945}, \quad \sum_{n=1}^{\infty} \frac{1}{n^8} = \frac{\pi^8}{9450}, \quad \dots$$

and more generally

$$\sum_{n=1}^{\infty} \frac{1}{n^{2k}} = B_{2k} \frac{(-1)^{k+1} (2\pi)^{2k}}{2(2k)!},$$

where B_0, B_1, B_2, \dots are the so-called *Bernoulli numbers*. These are defined as the coefficients in the Taylor series of the function $z/(e^z - 1)$:

$$\frac{z}{e^z - 1} = \sum_{j=0}^{\infty} B_j \frac{z^j}{j!}.$$

The values $B_2 = \frac{1}{6}$, $B_4 = -\frac{1}{30}$, $B_6 = \frac{1}{42}$, $B_8 = -\frac{1}{30}$ give the aforementioned Euler series. What about other powers? The exact value of the

sum

$$\sum_{n=1}^{\infty} \frac{1}{n^3} = 1.2020569031 \dots$$

is not known. It was considered a major breakthrough when Apéry in 1978 proved that the sum is an irrational number. (A shorter proof was found a few years later by the Dutch mathematician Frits Beukers).

This brings us to the definition of the *Riemann ζ -function*: for $z \in \mathbb{C}$ with real part $\Re z > 1$ the sum

$$\zeta(z) := \sum_{n=1}^{\infty} \frac{1}{n^z}$$

converges absolutely and defines an analytic function. It can be demonstrated that this function has a (unique) analytic extension to $\mathbb{C} \setminus \{1\}$. For us it is not so important to know exactly how this works; it suffices to know that the extension is given on the basis of certain identities that can be deduced for $\zeta(z)$. Using this, one can determine certain values of $\zeta(z)$ explicitly: for instance, $\zeta(0) = -\frac{1}{2}$ and $\zeta(-1) = -\frac{1}{12}$. The deeper meaning of the "formula"

$$1 + 2 + 3 + \dots = -\frac{1}{12}$$

in terms of normalization is explained very nicely in [2].

The Riemann ζ -function has zeros at $z = -2, -4, -6, \dots$ (the so-called trivial zeros), while all other zeros (the so-called non-trivial zeros, it is known that there are infinitely many) lie in the vertical strip $\{0 < \Re z < 1\}$. The *Riemann hypothesis* is the 150-year-old conjecture that all non-trivial zeros lie on the *critical line* $\Re z = \frac{1}{2}$ (see Figure 1).

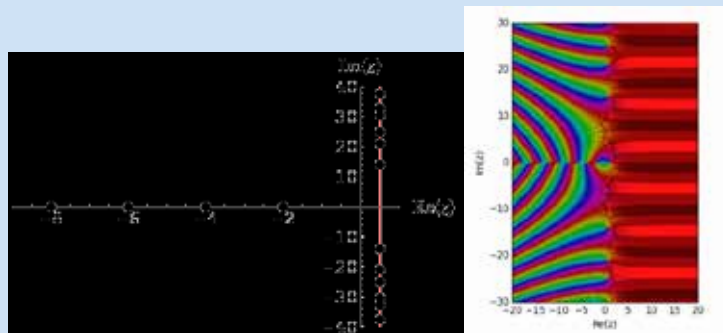


Figure 1: Schematic representation (left) and a so-called domain colouring plot (right) of the Riemann ζ -function. Sources: mathworld.wolfram.com/CriticalLine.html, en.wikipedia.org/wiki/Riemann_zeta_function.

There are numerous partial results in the direction of the Riemann hypothesis: it is known that at least 1/3 of all non-trivial zeros lie on the critical line (this is a famous theorem of Levinson from 1974). Numerically, it has been rigorously verified that the first 10^{13} non-trivial zeros lie on the critical line. The Riemann hypothesis belonged to the so-called 23 problems of Hilbert in



1900 and is one of the seven one million dollar Millennium problems published in 2000 by the Clay Institute.

Why is the Riemann hypothesis so important? The identity

$$\zeta(z) = \prod_p \left(1 - \frac{1}{p^z}\right)^{-1},$$

where the product extends over all prime numbers p , is a simple consequence of the fact that each positive integer admits a unique prime factorization. And indeed there are deep relationships between the Riemann ζ -function and the distribution of prime numbers. For example, the *prime number theorem*, which asserts that the number of prime numbers not exceeding n is asymptotically given by

$$\pi(n) \sim \frac{n}{\ln n} \text{ as } n \rightarrow \infty,$$

can be derived from the fact that the Riemann ζ -function has no zeros on the line $\Re z = 1$. There are countless results that are known to be true under - or even equivalent to - the Riemann hypothesis. These are recorded in the book [4], a must-read for anyone interested in this material.

Let us return to self-adjoint operators. Because they play the role of 'observables' in quantum theory. It has long been speculated that the Riemann hypothesis has a physical interpretation. Recalling that the spectral values of a self-adjoint operator are real, the zeros on the critical line could then be the spectrum of an operator of the form $\frac{1}{2} + iH$, with H , a hypothetical 'Hamiltonian', a self-adjoint operator that describes the energy of some (unknown) quantum mechanical system. If this conjecture, known as the *Hilbert-Pólya conjecture* (see [1] for historical details) is correct, the Riemann hypothesis immediately follows. In the following, I will try to explain why this conjecture is more than a speculation.

The story starts with an article from Montgomery from 1973 [9]. Under the assumption that the Riemann hypothesis is correct, a result is obtained describing certain correlations between successive zeros on the critical line. Based on this result it is conjectured that for each pair of positive real numbers $\alpha < \beta$ the number of pairs of consecutive zeros z and z' satisfying

$$\frac{2\pi\alpha}{\ln T} < |z - z'| < \frac{2\pi\beta}{\ln T}$$

asymptotically scales as

$$\int_{\alpha}^{\beta} 1 - \left(\frac{\sin \pi t}{\pi t}\right)^2 dt \text{ as } T \rightarrow \infty. \quad (1)$$

This formula appears to fit perfectly with reality. Using modern numerical methods, Odlyzko [10] determined the first 10,000 zeros following the 10^{12} -th zero on the critical line. Their two-point correlation plot shows indeed a striking fit with the line predicted on the basis of (1) (see Figure 2 for a more recent improvement of this calculation).

Now it gets interesting. The integral in (1) turned out to be an old friend in a very different segment of mathematical literature: it describes the correlations of consecutive eigenvalues of symmetric random matrices! Such matrices were introduced in the 1950s by the physicist Wigner to model the energy levels of heavy atomic nuclei. The precise energy levels of such nuclei are too complicated to be calculated explicitly, and Wigner devised a heuristic that suggested that their distribution is well described by the distribution of the eigenvalues of a symmetric random matrix with coefficients that are independent and Gaussian distributed. Indeed, there is a striking resemblance in the distribution of the zeros of the ζ -function, the eigenvalues

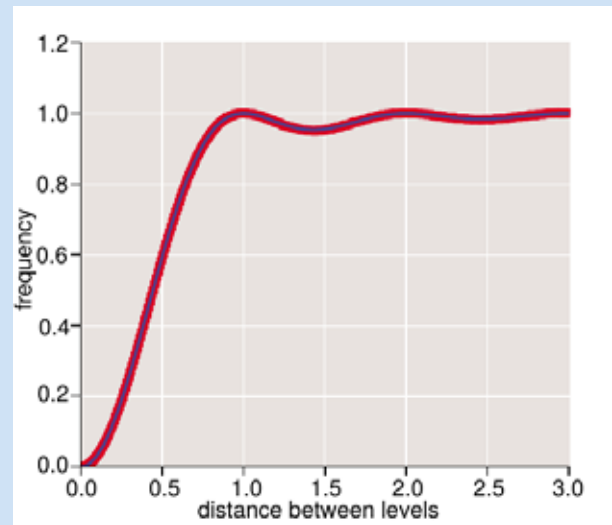


Figure 2: Pair correlations of the first 10^9 zeros following the 10^{23} -th zero (red) and its prediction of the basis of (1) (blue). Source: [6] (data by A.M. Odlyzko).

of symmetric random matrices, and the energy levels of heavy atomic nuclei (see Figure 3 (a) - (c)). The uniform distribution and the distribution of prime numbers, on the other hand, yield very different pictures (see Figure 3 (d) - (e)). The relation between the zeros of the Riemann ζ -function and the



Figure 3: From top to bottom: (a) positions of the first 100 zeros following the 10^{22} -th zero; (b) the middle 100 eigenvalues of a 300×300 random symmetric matrix; (c) the energy values erbium-166; (d) 100 independent draws from the uniform distribution; (e) 100 consecutive large prime numbers. Source: [6].

eigenvalues of symmetric random matrices is largely speculative and based on experimental data, although there are also some precise results such as those of Keating and Snaith [7]. The use of these matrices in the modelling of heavy elements is also speculative. Nonetheless, the combination of these heuristics leads to the concrete suspicion that the zeros of the Riemann ζ -function could represent the energy levels of some hypothetical 'element'. There is even a name for it, *Riemannium*, and its properties can be studied in [6, 8].

In physics literature there is much speculation about the nature of the




hypothetical Hamiltonian of Hilbert and Pólya, which we shall denote by H in what follows. The most famous proposal is from Berry and Keating [3]. They give a heuristic argument that the operator $H = xp$, where x and p are the quantum mechanical position and impulse operators respectively, is a good candidate. Using a clever but hard-to-justify truncation argument and further ‘educated guesses’, they argue that the number of energy states of xp with energy not exceeding E asymptotically equals

$$N_H(E) = \frac{E}{2\pi} \left(\ln\left(\frac{E}{2\pi}\right) - 1 \right) + \frac{7}{8} + \dots$$

These are exactly the first two terms of the formula for the number of zeros in the critical strip up to height T , given by the classical von Mangoldt formula:

$$N_\zeta(T) = \frac{t}{2\pi} \left(\ln\left(\frac{T}{2\pi}\right) - 1 \right) + \frac{7}{8} + O(\ln(T)).$$

Incidentally, this formula was the starting point of the BEP of Joris van Tatenhove [11] who performed a number of numerical simulations à la Odlyzko. More recent work has shown that the Hamiltonian proposed by Berry and Keating cannot be the sought-after Hilbert-Pólya Hamiltonian. The idea of Berry and Keating was inspired by another proposal by Connes [5] a few years earlier, based on deep ideas from algebra geometry. In Connes’s model, the zeros of the Riemann ζ -function on the critical line correspond exactly to the absorption spectrum (missing spectral lines) of a certain Hamiltonian. In order to prove the Riemann hypothesis, however, an as yet unproven ‘trace formula’ is needed - the proof which could be as difficult as the Riemann hypothesis itself. For further study, a good starting point is the article [12] which also contains an extensive list of references. Even better, you could make a stop at the Analysis section! 

References

- [1] www.dtc.umn.edu/~odlyzko/polya/
- [2] J. Baez, “On the number 24”, [youtube.com/watch?v=vzjbRhYjELo](https://www.youtube.com/watch?v=vzjbRhYjELo).
- [3] M. Berry and J. Keating, “ $H = xp$ and the Riemann Zeros”, in *Supersymmetry and Trace Formulae: Chaos and Disorder*, vol. 370 of NATO ASI Series B Physics, pp. 355–368, Plenum Press, 1999.
- [4] P. Borwein, S. Choi, B. Rooney, and A. Weirathmueller, “The Riemann Hypothesis: A Resource For The Afficionado And Virtuoso Alike”, Springer-Verlag, 2007.
- [5] A. Connes, “Formule de trace en géométrie non-commutative et hypothèse de Riemann”, C. R. Acad. Sci., Paris 323, 1231–1236, 1996.
- [6] B. Hayes, “The Spectrum of Riemannium”, *American Scientist*, Volume 91, Number 4, 2003, 296–300.
- [7] J. Keating and N.C. Snaith. “Random matrix theory and $\zeta(\frac{1}{2} + it)$ ”, *Comm. Math. Phys.* 214, no. 1 (2000): 57–89.
- [8] P. Leboeuf, A. G. Monastera, and O. Bohigas, “The Riemannium”, arXiv preprint nlin/0101014, 2001.
- [9] H.L. Montgomery, “The pair correlation of zeros of the zeta function”, in *Analytic Number Theory (Proc. Sympos. Pure Math., Vol. XXIV, St. Louis Univ., St. Louis, MO., 1972)*, pp. 181–193, Providence, R.I.: Amer. Math. Soc., 1973.
- [10] A.M. Odlyzko, “On the distribution of spacings between zeros of the zeta function”, *Math. Comp.*, vol. 48, pp. 273–308, 1987.
- [11] J. van Tatenhove, “De verschillende aspecten van Riemanns ζ -functie”, BSc eindverslag, TU Delft, 2014.
- [12] M. Wolf, “Will a physicist prove the Riemann Hypothesis?”, arxiv:1410.1214, 2014.



Mathematical Career Orientation Day

with

1-on-1 company talks with mathematicians

SAVE THE DATE:

4 APRIL, WEDNESDAY

Interested?
wisv.ch/MCOD



A dotted line graph on a coordinate plane. The line starts at a high y-value on the left, remains horizontal for a short distance, then curves downward in a smooth, concave-up fashion, and finally becomes horizontal again at a lower y-value on the right.



Internship at Google

Tim van der Lippe

In the summer of 2017, I did an internship at the Polymer team in the San Francisco office of Google. This internship included training by top engineers about the inner-workings of Google as a company, how to use the tooling for the daily developer workflow as well as a real-world project. The real-world project concerned the load time of large web applications, focused on one of the biggest users of Polymer: YouTube.

Working at Google

The first 2 weeks consisted of training by top Google engineers in Mountain View, which explained how Google operates and introduces new hires into the Google workflow. As the Polymer team is primarily open source, my work was all on GitHub, which I was very familiar with as a result of my work during my courses. Nonetheless, whenever terms were used in discussions with Google engineers, I was able to follow the conversation and understand the concepts they were talking about. Besides that, the knowledge I obtained, during courses such as Software Quality and Testing¹ and Software Engineering Methods², was crucial to be able to have efficient and constructive discussions with my host and second reviewer.



On a Google bike in Mountain View

During the internship I experienced a different working environment in comparison to the environment at my university, Delft University of Technology. There is a stark difference between prioritization and assignment of tasks between industry and university life. During the 12 weeks, I was tasked to answer

a (seemingly) simple question. There was no pre-specified set of sub-tasks required to be able to answer this question, this was my own responsibility. Of course I had (almost daily) meetings with my host and second reviewer, as well as other team members, who I could discuss approaches and problems with. This ad-hoc approach allowed me to work on problems and discuss blocking issues right as I encountered them (or after an hour being stuck on the same issue).

The emphasis on testing is very prevalent at Google. Besides work on my project tool, all of my contributions to (open source) projects required the existence of tests that either tested the new functionality or were regression tests to make sure bugs were fixed. Full Test-Driven-Development (TDD) was not something I had to practice, although I mostly used TDD for debugging bug reports and verifying that the issue was resolved. Google uses a Test Automation Platform (TAP)³ to test changes made by a developer. Since there is an average of 1 commit per second in the single repository⁴, running all tests on every single change list is infeasible and computationally impossible. Instead, tests are only run in specific intervals focusing on running frequently-breaking tests. During my internship, I experienced the debugging experience second-handed, when a team member was triaging a TAP failure. The test that broke was in a project not owned by the Polymer team. However, as the change list of the team member broke the test, it was their responsibility to figure out the root issue. If the root issue cannot be found, instead of waiting and continuing the search, the change is rolled back to unblock all other engineers from working. Rolling back changes is crucial, as Google operates in this single repository. In my specific example, the engineer and I discussed and reasoned about the test failure. Even though we had no experience with the test, or the complete system, the team member and I were still able to deduce why the test failed.

Work-life balance

On a less technical note, my host and I also worked on my work-life balance. During my study, I was used to working 7 days a week. On weekdays, I would start at 10 AM and work till 6 PM, eat dinner and sometimes work more between 8 PM and 10PM/1AM. My work hours largely depended on the size of my assignments and motivation, because sometimes I was excited working on a particular assignment. One occurrence, I recall, I was working on the course Concepts of Programming Languages⁵ till 1 AM in the morning, as I suddenly had inspiration and was able to solve the assignment (even though the deadline was still days away).

Being used to these working hours, at the start of my internship I had trouble having a healthy work-life balance. In the first couple of weeks, I would work hours similar to what I was used to at university. However, most of the Google

¹ http://studiegids.tudelft.nl/a101_displayCourse.do?course_id=34556&-SIS_SwitchLang=en

² http://studiegids.tudelft.nl/a101_displayCourse.do?course_id=43888&-SIS_SwitchLang=en

³ <https://research.google.com/pubs/pub45861.html>

⁴ <https://research.google.com/pubs/pub45424.html>

⁵ http://studiegids.tudelft.nl/a101_displayCourse.do?course_id=43899&-SIS_SwitchLang=en



engineers left at 6 PM, while a couple of team members and I would stick around. We had a smallish group of engineers eating dinner every day in the office. After dinner, we would go back to continue working in the office.

Roughly in the middle of my internship, my host started a discussion about my work-life balance. Implicitly, it was impacting my work performance and I was working less efficiently than I could, but I was not aware of it being that impactful. This discussion was tough and trying to change my working hours was hard, but in the last weeks of my internship I did change my day. Instead of returning to the office after dinner, I would join colleagues in playing games (pool, pinball, table tennis). Even though I reduced the total amount of working hours per day, I was more productive and able to resolve problems faster. Learning to manage my work-life balance was very valuable and I think that without working in industry and having peers discussing these problems, it is hard to make sure you are working in a healthy manner.

All in all, working at Google is not extremely different compared to university life. However, priorities and aspects such as a work-life balance are very much different in industry compared to being a student. The knowledge I obtained from my university courses was sufficient to be able to have thoughtful conversations with my colleagues. However, practical experience outside course assignments and exams is required to be able to effectively write industry code. Luckily, I could obtain experience with my open source contributions which I did in parallel to my study. (The open source contributions to Polymer eventually led the Polymer team to reach out to me for this internship)

The project

Aside from obtaining experience from working at Google, the most important part of the internship was the real-world project. Load time of applications is of great concern for YouTube as well as various other big users of Polymer⁶. Polymer is a JavaScript library that helps you create custom reusable HTML elements, and use them to build performant, maintainable apps.

As such, YouTube requested the Polymer team for improvements to their load performance. Earlier example of improved load performance was the introduction of lazyRegister⁷. lazyRegister specifically aimed to reduce the amount of registration and setup work Polymer did when registering elements. As YouTube is a big application, it builds and uses a lot of custom elements⁸ on every page. However, not all elements are crucial to be fully available on first load. Examples include elements that are in hidden visual components (drop down menus, dialogs, etc...) or are used in only a subset of the various pages. Late registration was introduced to only do the necessary registration work once an element was instantiated and thus actually used.

In a similar fashion, the Polymer team identified computational work on load time that could be prevented. This work concerned the metaprogramming of Polymer: in-memory representations of the definitions and thus behavior of custom elements. On load, Polymer parses the definitions of custom elements, performs the transformation to the in-memory representation and uses this representation in all further interaction with the element.

⁶ <https://www.polymer-project.org/>

⁷ <https://github.com/Polymer/polymer/releases/tag/v1.4.0>


⁸ https://developer.mozilla.org/en-US/docs/Web/Web_Components/Custom_Elements

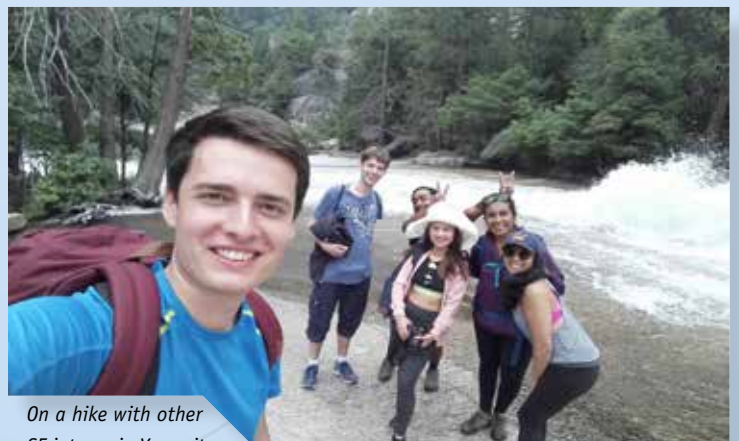
The so-called binding metadata remains largely the same in between page loads. As such, every time a user opens the page, the binding metadata has to be calculated.

The project thus focused on the trade-off between reduction in computation time versus increase of network time, because we are including the in-memory representation in the website payload.

The internship consisted of two parts: doing initial performance tests to assess the feasibility of building a tool and the second part actually building the tool. As such, I had to first investigate and proof the worthiness, before programming the tool.

All in all, the initial performance tests were successful and I built the tool. On the various test-cases, the tool improved website loading by 1-3%, which is marginal.

If you would like to read the full report and detailed results, please check out <https://github.com/TimvdLippe/internship-report-google> 



*On a hike with other
SF interns in Yosemite
National Park*



Science trends

Rebecca Glans

On this page you will find some brief information on recent scientific breakthroughs or interesting news. Whether they're big or small, if we think they might interest you, we will mention them here! Do you miss a certain trend or want to inform your fellow readers of an interesting innovation? Feel free to contact us.

(Action) Games are good for you!

The University of Geneva led an international team of psychologists to research how action video games impact our cognition. They assembled relevant data from the years 2000 to 2015, but the impact of video games has been studied since the 80's. In this study they focused on the impact of action games (war, shooter) which produced two meta-analyses.

The first meta-analysis showed that the cognition of gamers was indeed better than non-gamers. Even though this was concluded with data of almost 9000 participants, between the ages of 6 and 40, it didn't show whether action games were the cause.


With the second analysis they hoped to find if gamers develop their cognitive abilities by playing action games or if they play action games because of their high cognitive skills. They let almost 3000 men and women play either an action game or a control game (SIMS, Tetris) up to 50 hours in total in a time span of 12 weeks. Afterwards they saw that action games indeed had increased cognition. You can read the full publication in the journal Psychological Bulletin.[1]

At least Einstein was right

In August last year, the collision and merger of two neutron stars caused some difficulties for theories concerning dark energy. The merger caused gravitational waves (rippling in space-time) and a burst of light, both observed by detectors and telescopes around the globe. The detection- and thus arrival time of the light on earth is an important test for dark energy theories, as it was only 2 seconds after the detection of the gravitational waves. Many theories "survived", especially the more simple ones. For instance the "cosmological constant" theory introduced by Albert Einstein remains viable. But, theories which state that the arrival of the light signature of the star merger would occur after up to millions of years should either be modified or scrapped.

Nonetheless, scientists still don't have a clear understanding of dark energy and this star merger brings new insights. Dark energy can be described as antigravity, as it pushes matter away instead of attracting it. Dark energy causes the expansion of the universe to accelerate and makes up 68% of the universe's total mass and energy. More about the study can be found in [2].[5]

"ABC, easy as 1-2-3"

The ABC conjecture is a mathematical problem concerning the addition of numbers: $a + b = c$. It thus challenges a fundamental property of numbers. The problem was proposed in the 1980's and even though it was never proven, mathematicians believed it to be true. In 2012 Shinichi Mochizuki posted a 500-page long proof for the ABC conjecture developing a new type of mathematics along with it. The latter is called Inter-universal Teichmüller Theory and is very difficult to understand for experienced mathematicians. This is also the reason the proof has not been published by any journals yet. After many frustrating years, Mochizuki has maybe found a Japanese journal willing to publish the proof this year. Those who understand the proof or believe it will have an important impact on number theory, are looking forward to the publication, while others remain sceptical. One of the reasons is the supposed journal being Japanese, with Mochizuki as editor-in-chief.[3,4] 

References

- [1] Action games expand the brain's cognitive abilities, study suggests (2017, December 12). Retrieved January 10, 2018, from <https://www.sciencedaily.com/releases/2017/12/171212102158.htm>
- [2] Jose María Ezquiaga, Miguel Zumalacárregui (2017, December 18). Dark Energy After GW170817: Dead Ends and the Road Ahead. Retrieved January 10, 2018, from <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.119.251304>
- [3] Mathematician set to publish ABC proof almost no one understands (2017, December 18). Retrieved January 10, 2018, from <https://www.newscientist.com/article/2156623-mathematician-set-to-publish-abc-proof-almost-no-one-understands/>
- [4] Mathematician's anger over his unread 500-page proof (2015, January 7). Retrieved January 10, 2018, from <https://www.newscientist.com/article/dn26753-mathematicians-anger-over-his-unread-500-page-proof/>
- [5] Star mergers: A new test of gravity, dark energy theories (2017, December 18). Retrieved January 10, 2018, from <https://www.sciencedaily.com/releases/2017/12/171218131317.htm>



Historical figure: Ada Lovelace

Daniël van Gelder

People often say that science and engineering are fields dominated by men. Such was also the case in the early 1800s, when in London, a young aristocratic girl called Augusta Ada Byron, Countess of Lovelace – more commonly known as ‘Ada Lovelace’ – was born.

Life and career

Ada Lovelace is widely considered to be the one of the founders of the computer science field and the first computer ‘programmer’. Although it was unusual for a woman to study the sciences in 19th century England, Ada was pushed from a young age to learn mathematics by her mother and showed great talent. When she was seventeen, Ada met Charles Babbage, a mathematician who is known as the father of the computer. Babbage proposed a mechanical computing engine which could do calculations on numbers. When Ada translated an article about the engine she added her own notes and ideas about the engine, which ended up being three times longer than the original article. She described how codes could be developed for the machine in order to let it handle symbols alongside numbers and also theorized a method for repeated calculations, nowadays known as a loop. These ideas give her the reputation that she is known for today, namely being the first computer programmer.

Although her ideas and theories were innovative and forward-thinking at that time, she only got little recognition for them. It was only after her death in 1852, at the surprisingly young age of 36, that she received her well-deserved scientific rewards.

Work

We know that throughout her life, Ada was fascinated by the relation between nature and mathematics. For instance, she had a desire to construct a mathematical model for how the brain constructs thoughts and how nerves create sensation: ‘a calculus of the nervous system’, as she called it. In addition, she also worked on showing a relationship between music and mathematics. She also researched magnetism and carried out extensive electronic experiments. None of these ventures ever progressed into any publications however, which was partly due to her unexpected early death.


As mentioned earlier, the biggest breakthrough of Ada was her work on Charles Babbage’s ‘Analytical Engine’, a proposed mechanical general-purpose computer. Babbage had finished a device called the ‘Difference Engine’ earlier, which could calculate results from polynomial functions and differences between

values of that function. The proposed ‘Analytical Engine’ was never actually built. However, Lovelace was fascinated by the idea of a computer and her extensive article about the machine featured many elaborate details about the machine and how complicated its structure had to be. Her notes were divided up into seven sections, each denoted by a letter. In section G, the final section of her notes, she proposed a program to calculate Bernoulli numbers. This was the first algorithm specifically designed for a machine ever, and this work is what earned her all her post-mortuary fame. Another interesting part of her notes were her dismissal of the idea of artificial intelligence. She wrote:

“The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform. It can follow analysis; but it has no power of anticipating any analytical relations or truths.” (Lovelace, Ada. 1843. Section G)

This idea would cause much debate and discussion. Alan Turing, for example, was known to be opposed to this idea. Within her notes, Ada also noted how a general-purpose computer like the ‘Analytical Engine’ had the potential to go beyond mere numerical computations. Such a machine which could solve problems of a much higher complexity could even produce writing, or pieces of music in her idea. Numbers could represent other entities than ‘quantities’ but also symbols and letters, much like the way we represent symbols with bits. An inspiration for this idea was how industrial weaving machines could create beautiful patterns on cloth based on an instructive punch card. Many scientists were fond of this notion and supported Lovelace; most of this, however, came after her death.

Legacy

We can fairly say that Ada Lovelace’s ideas about computing were far ahead of her time. It was unfortunate that during her life, little of her work was recognized by the scientific community. Perhaps this was due to the roles of women in 19th century England or due to the novelty of her ideas. In commemoration to Lovelace, the United States Department of Defense developed a programming language ‘Ada’ in her name during the 1980’s. An annual event called ‘Ada Lovelace Day’ is celebrated each year in mid-October to raise the profile of women in science and engineering fields in hope that many great figures like her will rise to share many more new wonderful ideas with the world. 

References

- [1] Wikipedia. (2017). Ada Lovelace. Retrieved from: https://en.wikipedia.org/wiki/Ada_Lovelace
- [2] Biography.com Editors. (2017). Ada Lovelace Biography.com. Retrieved from <https://www.biography.com/people/ada-lovelace-20825323>
- [3] Sydney Padua. (2017). Who was Ada?. Retrieved from: <https://findingada.com/about/who-was-ada/>





Mathematical puzzle

Even/odd Sudoku

An even/odd Sudoku is solved just like a Sudoku by filling in the numbers 1 to 9 into the blank cells. The grey cells may only be filled with even numbers and the white cells may only be filled with odd numbers.

				8		4		2
		1	6					
						5		
			7		3		1	
2	4							
5								
4	9			5				
			1				7	

Stairstep Sudoku

A stairstep Sudoku is also solved like a normal Sudoku. The only difference is the staircase-shaped boundary of the regions. Good luck!

		2		6				
	8				4			
1				5		3		
5		1	3	8				
						1		9
				1	2			7
						5		
					6			1
9		3	4			6		2



Computer science puzzle

Can you figure out what this program will print without running it on your computer? There's one thing you'll need to know: the character encoding on this machine is ASCII. The answer to this puzzle will be published in the next MaCHazine.

```
@SuppressWarnings("ALL")
public class MaChazinePuzzle {
    public static void main(String[] args) {
        System.out.println(
            "M" + (char)((0b0110 ^ 0b1001) - 0b111)
            + 'a' + (char) ('C' + 32 + (int) ((0.7 - 0.3) / 0.1 * 3))
            + (char) (0x19 == 25 ? (('h' + 'a' + 'z' + 'i' + 32)/4) : 420 % 42)
            + (char) ('n' + 3 * Math.PI)
            + (!System.out.equals(42) ? 'e' : '\r') + args.getClass().getSimpleName().charAt(2)
        );
    }
}
```

Last issue's puzzle solutions

Computer Science Puzzle

[1] "The main method creates a Pet instance representing Fido and invokes its live method. The live method, in turn, creates and starts a Thread that repeatedly executes the eat, play and sleep methods from the enclosing Pet instance. Forever. Each of those methods prints a single line, so one would expect the program to print these three lines repeatedly:

```
Fido: Mmmmm, beef
Fido: Woof Woof
Fido: Zzzzzzz...
```

If you tried the program, you found that it won't even compile. The compiler error is less than helpful:

```
Pet.java:28: cannot find symbol
symbol: method sleep()
sleep();
^
```

Why can't the compiler find the symbol? It's the right there in black and white. The problem stems from the details of the overload resolution process. The compiler searches for the method in the innermost enclosing scope containing a method with the correct name [JLS 15.12.1]. For the sleep invocation in our program, that scope is the anonymous class containing the invocation, which inherits the methods Thread.sleep(long) and Thread.sleep(long, int). These are the only methods named sleep in that scope, and neither is applicable to this invocation because both require parameters. As neither candidate for the invocation is applicable, the compiler prints an error message.

The sleep methods inherited into the anonymous class from Thread shadow [JLS 6.3.1] the desired sleep method. You should avoid shadowing. The shadowing in this puzzle is indirect and unintentional, which makes it even more insidious than usual.

The obvious way to fix the program is to change the name of the sleep method in Pet to snooze, doze, or nap. Another way to fix the problem is to name the class explicitly in the method invocation, using the qualified 'this' construct [JLS 15.8.4]. The resulting invocation is Pet.this.sleep().

[...]

In summary, beware of the unintentional shadowing, and learn to recognize compiler errors that indicate its presence. For compiler writers, do your best to generate error messages that are meaningful to the programmer. In this case, for example, the compiler could alert the programmer to the existence of a shadowed method declaration that is applicable to the invocation."

Optimize your wife!

This problem, which is called the marriage problem, is also called the secretary problem, sultan's dowry problem, fussy suitor problem, the google game or the best choice problem. The question is about the optimal strategy (stopping rule) to maximize the probability of selecting the best applicant. If the decision can be deferred to the end, this can be solved by the simple maximum selection algorithm of tracking the running maximum and who achieved it, and selecting the overall maximum at the end. However, in this

case it is not that simple. The girlfriends will not wait while you try another one. You have to make a decision before going on to the next.

The problem has different ways to come to the solution. The shortest rigorous proof known so far is provided by the odds algorithm (Bruss, 2000). An easy analysis implies that the optimal win probability is always at least 1/

A free ticket to the movies!

To start with, we assume you don't know anyone else's birthday and that birthdays are uniformly distributed throughout the year (of 365 days). This should have been stated in the problem itself, if this stopped you from solving the problem, just stop here and try it again.

The probability $p(n)$ of getting a free ticket when you are the n th person line is:

(probability that none of the first $n - 1$ people share birth dates) * (probability that you share birthday with one of the first $n - 1$ people)

So, $p(n) = [1 * (\frac{364}{365}) * \frac{363}{365} * \dots * (\frac{365-(n-2)}{365})] * [\frac{(n-1)}{365}]$,
Here, $0 < n \leq 365$.

Now the least n such that $p(n) > p(n+1)$, or $\frac{p(n)}{p(n+1)} > 1$.

Now, $\frac{p(n)}{p(n+1)} = \frac{365}{(366-n)} * \frac{(n-1)}{n}$

$\Rightarrow 365n - 365 > 366n - n^2$

$\Rightarrow n^2 - n - 365 > 0$

$\Rightarrow (n - \frac{1+\sqrt{1461}}{2}) * (n - \frac{1-\sqrt{1461}}{2})$

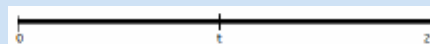
$\Rightarrow n = \frac{1+\sqrt{1461}}{2} = 19.6115148536$

$\Rightarrow n = 20$ (ceiling of computed value)

Hence standing on the 20th position maximizes your chance on a free ticket.

Getting home early

To solve this problem, you can use a time line. In this image, t stands for the time of a normal one-way trip to pick you up. The time your friend leaves home equals zero. So the time he takes to return after being at the station is $2t$. This makes a regular day looks like this



Starting to walk 60 minutes before t , and will lead to the following:



Your friend will leave his house at 0, like he regularly does. Your return time equals to $2t - 20$, the time you were picked-up is $t - 10$.

The result is found by:

$Walkingtime = pick - uptime - starttime$

$Walkingtime = (t - 10) - (t - 60)$

Which results in the fact that you've walked 50 minutes. [🔗](#)

References

[1] Bloch, J. Gafter, N. (2005). "Java Puzzlers - Traps, pitfalls and corner cases"



Calendar



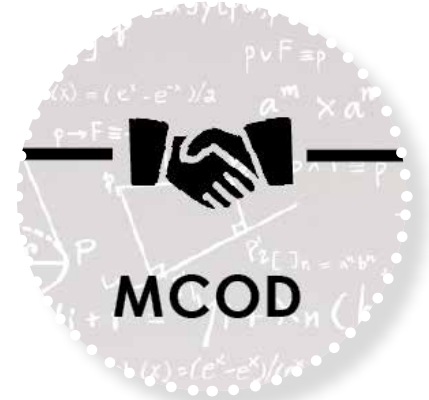
DDB

'De Delftse Bedrijvendagen' (DDB) will have both their solicitation workshops as well as the presentation days this month. 150 companies will present themselves here at the largest technology oriented career event in the Benelux. For more information, go to ddb.tudelft.nl



Area FiftyLAN

Lots of exciting events will happen during March! Area FiftyLAN is our annual LAN-party at which everybody can join is to play plenty of games or compete in competitions. Furthermore, we will also visit many companies during the Business Tour and celebrate the birthday of the association in the Dies week.



MCOD

Mathematical Career Orientation Day is the perfect day for Mathematics students to see what companies have to offer to them. Mathematicians companies will come to EEMCS to talk to students about what they can do there when it comes to both internships and jobs.

Februari

- 13 DDB solicitation workshops
- 13 Department Symposium
- 14 DDB solicitation workshops
- 14 Double Degree Dinner
- 15 DDB solicitation workshops
- 16 Study visit committee
Information Lunch
- 20-21 DDB Presentation Days
- 21 SjaarCie Party
- 27 T.U.E.S.Day Education Lecture
- 27 Graduation Panel
- 28 MeisCie Activity

March

- 2-4 Area FiftyLAN
- 5 General Assembly
- 6 T.U.E.S.Day Lecture
- 7 iCom Information Lunch
- 8 Career College 3.1
- 19-25 Business Tour
- 26-29 Dies week

April

- 3 T.U.E.S.Day lecture
- 4 MCOD
- 5 Career College
- 5 MatCH Activity
- 24 T.U.E.S.Day lecture
- 24 AkCie Activity
- 28-29 Hackathon

