50 Jahre OTH REGENSBURG

Fakultät Informatik und Mathematik

Informatik - 2023
Mathematik - 2024

(Almost) nobody knows what computer science is

A journey through time to the first computer science degree programme

at the Regensburg University of Applied Sciences

October 2023

Klaus Stampfer, graduate computer scientist (FH)

Student in the first degree programme in Computer Science at the FH Regensburg

50 years ago, the Computer Science degree programme was opened at Regensburg University of Applied Sciences. I was on the first degree programme. To mark the 50th anniversary, I would like to contribute my thoughts and go on a journey 50 years back to the early 1970s and try to describe what the term "computer science" meant back then.

In 1970, the book "Nobody knows what cybernetics is" [1] was published. Of course, this is not true in its absolute form, because if nobody really knew what cybernetics was, then the authors could not have written this book either. So there were a few insiders who did know what cybernetics was. Following the idea of this book, there could also have been a book with the title "Nobody knows what computer science is", because the term "computer science" was just as rarely heard in common parlance as "cybernetics".

The TU Dresden defines computer science today as follows: "The word computer science is made up of the words information and automation and describes the science of the systematic processing of information with the help of computer systems. The origins of computer science lie in mathematics, electrical engineering and communications engineering. Computer science designs and constructs mathematical machines that can independently process symbols, i.e. machines with which data can be transmitted, stored and automatically processed using chains of commands - the algorithms." [2]

But how was computer science defined 50 years ago when the new degree programme was introduced? Let's travel back in time to the beginning of the 1970s. Time travel is not physically possible. The mere fact that no travellers from the future visit us may be an indication that time travel will not be possible in the future either. But we can mentally transport ourselves to other eras. The best way to do this is to rely on our own memories. As with many storage media, a few bits may be lost in the brain and the overall picture may deviate slightly from reality as a result, but this should not affect the mental journey.

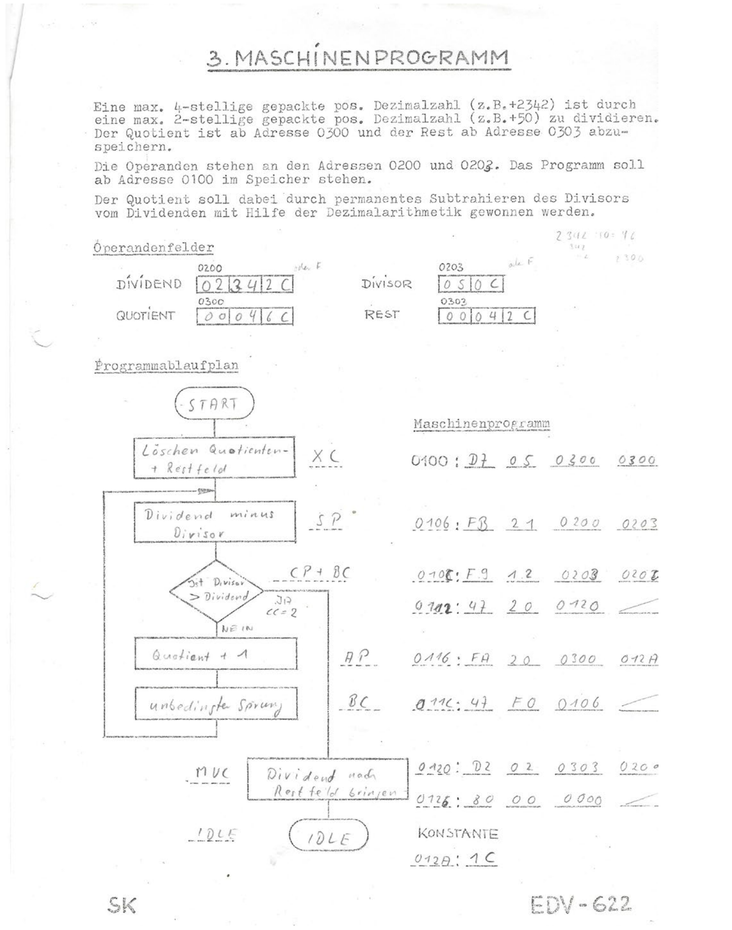
We are now in the early 1970s. For many decades, there have been machines whose mode of operation is controllable, i.e. programmable. A well-known example is the barrel organs at fairgrounds. Valves are operated with a perforated tape to produce the organ sounds. Different melodies can be played by changing the perforated tape. The perforated tape is regarded as the control of the barrel organs, nobody thinks of computer science.

I completed my apprenticeship as a telecommunications technician at the Regensburg telecommunications office in 1970 and I work in the overhaul of telephone exchanges operated by the telecommunications offices of the German Federal Post Office. The W50 exchange system works electromechanically with relays and rotary and rotary-lift selectors and enables worldwide telephoning. Mechanics are subject to wear and tear. For this reason, the exchanges have to be overhauled once a year and burnt contacts in the relays and worn springs in the diallers have to be replaced. Your own and the dialled number determine which of the almost infinite possible connections is made. The telephone network is the world's largest machine and can be compared to the first computers, which were built with relays. Although it can be used to establish flexible connections and transfer information, the telephone network is not associated with IT.

I was interested in electronics during my apprenticeship. So I built amplifiers, simple receivers and, now that it's time-barred, small transmitters with a short range. All analogue technology. I bought the books in the Regensburg cathedral bookshop. Once I came across the book "Schaltungen der Digital-Elektronik" [3]. It describes a completely different, exotic area of electronics. My interest in AND, OR and NOT gates, astable and monostable multivibrators and flip-flops was awakened. So that's how you build computers. There are no pocket calculators (yet). So I use transistors, resistors, capacitors, switches and lights to build a 4-bit calculator that can perform the four basic arithmetic operations in the number range from 0 to 15. It is the size of a table top. The fact that nobody in my circle of friends, even among electronics enthusiasts, knows how computers work and that I have a little insight with my 4-bit calculator makes computers all the more fascinating to me.

Via the second educational pathway, vocational college and technical secondary school, I obtained my university entrance qualification. With my professional activity, what could be more obvious than to enrol on the Communications Engineering course at the University of Applied Sciences (FH) in Regensburg in 1973. But then I received a letter from the FH saying that there was a new degree programme, namely computer science, and I re-registered for it. What motivated me to do this? Computers have something to do with computer science. That much is clear to me. And computers mean the future, and this prospect is not unimportant for my career choice as a 20-year-old. That's the reason why I'm enrolling in computer science. I expect to learn how to build large computers. The terms "computer" and "computing" are used interchangeably.
I have heard from many friends that they are surprised that I am going to study computer science and that I am entering the media industry as a technology enthusiast. My acquaintances associate computer science with information dissemination, with the press, broadcasting, i.e. journalism. This is an example of how little the term "computer science" is known.
What do I associate with computer science? The construction of computers. Programming is part of it so that the hardware works. However, it is of secondary importance.
In the first two semesters, I learnt that computer science does not mean hardware development. I was somewhat disappointed by this. At the same time, I recognise the possibilities that software opens up and, together with other subjects, arouses my interest.
How does the FH understand the subject of computer science? Let's take a look at the lectures offered in the first two semesters: Practical programming with ALGOL 60 on the Zuse Z23, data processing systems, basics of computer science, mathematics, mathematical practical (instructions for calculating with the PC-1001 desktop computer), business administration, physics, chemistry, English.

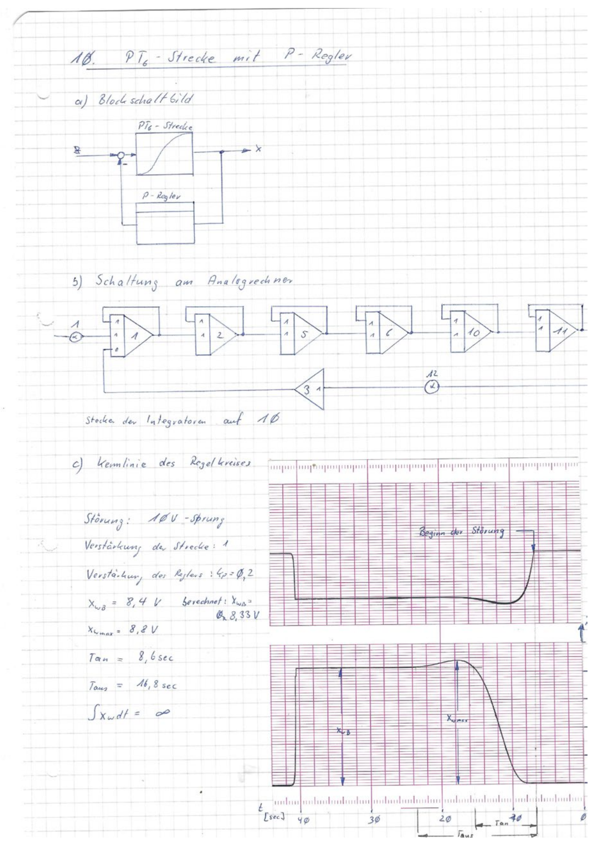
Fundamentals of computer science, programming in ALGOL60 and in machine code, as well as mathematical topics have priority. The other lectures give an impression of the breadth of topics that comprise computer science. The development of hardware does not appear to be part of computer science. It is used, development is apparently taught in other disciplines. Programming in Siemens 4004 machine code is only done on paper due to the lack of a computer.



Extract from the machine programming script

After the second semester, I am called up for civilian service. In the second degree programme, I can continue my studies after my civilian service with the first internship semester. I do this at the municipal utilities in Regensburg. Several machines are in use there for processing punched cards, such as sorting, coding, etc. These machines are operated at a height of approx. 30 cm. These machines are programmed with boards measuring approx. 30 cm by 30 cm, which are fully equipped with sockets. The machines are programmed by inserting wire strips with banana plugs into the sockets. Various programmes can be executed by exchanging the plates. There is also a computer, a cabinet-sized Siemens 4004/16 with 16 KB of main memory, a punch card reader and puncher, a high-speed printer and a disc drive in the computer room. Among other things, the municipal utilities use it to process the annual bills for electricity, water, etc. for Regensburg households. Programming is done in Assembler.
So what does computer science look like in practice? Large machines and very limited programme memory. Every byte in the programme code has to be double-checked to see if it is necessary. The hardware is the dominant component. Computer manufacturers only sell the computer and the input and output devices for a lot of money; the software (operating system, parts of the application software) is included free of charge or is included in the price of the hardware.
Lectures in the fourth and fifth semesters: technical cybernetics, network technology, physical measurements, physical practical course, system and assembler programming, statistics, numerical mathematics, computer organisation, programming practice with PL1, Boolean algebra, medical informatics.
In IT organisation we hear that sufficient space must be planned in the computer centre when setting up the printers and disk drives in order to be able to open the doors of the enclosures and that sufficient waste bins must also be planned.

We learn another higher programming language, PL1. We write the programmes on punched cards, which is a big step forward compared to the punched strips with the ALGOL60 programmes for the ZUSE Z23. Unfortunately, we don't have a computer to translate and test our PL1 programs. So a fellow student takes the stacks of punched cards to the computer centre of a Regensburg company. A few days later, the listings (with the error messages) and the punched cards can be picked up. Once the errors have been corrected, the process starts all over again. Several cycles are required until the software runs without errors, and it takes a correspondingly long time until the programme runs without errors.
I chose medical informatics as my AW subject (AW = general science), which will be a great help to me later on.
I spend the second semester of my internship at Siemens in Erlangen. I test COBOL programs and work as an operator in the computer centre on a Siemens 4004/151 costing several million DM, which consists of several cabinets several metres wide and almost two metres high, contains 2 MB of main memory and to which four high-speed printers, two plotters, 36 disk drives and several magnetic tape drives are connected. The 36 disc drives, each measuring an estimated 80 cm by 80 cm by 100 cm (W x D x H), add up to a storage capacity of 780 MByte, an unimaginable size. The computer centre is a huge hall. Up to 90 users are connected to the computer via telephone lines using 50, 300 and 1200 baud modems, from Erlangen to Karlsruhe. Siemens' Kraftwerk-Union (KWU) uses it to plan its nuclear power plants, among other things.
A main memory of 2 MByte is considered limitless. Without the main memory limit, new application possibilities open up that can be completed much faster with software than manually. This includes the planning of large projects, including the automatic ordering of tens of thousands of parts. Without the memory limitation and the ever-increasing clock frequencies of computer hardware, software and therefore IT will become increasingly important. Computer manufacturers are announcing that they will charge for software.
Lectures in the seventh and eighth semesters: Formal languages and translators, BCPL programming language, technical cybernetics and control engineering - with practical training with analogue computers, business IT applications, computer-aided simulation, computer technology and electronics, exemplary treatment of an operating system, operations research, use of microprocessors and process computers, multiprocessor systems, data protection.
In the cybernetics practical, we set up pneumatic controllers with nozzles and cans. With the knowledge we have gained so far about what computer science is, we marvel at what can be done with compressed air, but consider pneumatics to be



Script of a test with an analogue computer

less compatible with the algorithms used in computer science. The situation is somewhat different with analogue computers. The PID controllers (proportional-integral-differential controllers) are built with operational amplifiers and programmed with the exchange of resistors and capacitors. The fact that they are electrically operated is something they have in common with digital computers. From a mathematical point of view, analogue computers are used to solve differential equations in real time. Analogue computers are far superior to digital computers in terms of speed and accuracy, not to mention their price and space requirements.
After understanding during the course of my studies that computer science is the science and technology of solving tasks using software with digital computers, I learnt in the last few semesters what I had missed in the first few semesters - how to build computers. To my delight, microprocessors were added to the analogue computers. I won't be building analogue computers in my future career, but I will be building them for hobby use. Digital computers will soon be faster, more powerful than analogue computers and also cheaper. The development of microprocessor systems with hardware and software would later be my bread and butter for 19 years.
My thesis is a new formal programming language especially for statistical calculations, which is executed by an interpreter. I am writing the interpreter in PL1. The computing centre at the University of Regensburg has granted me an exception to use 190 KB of main memory for the translation. However, the translation may only run at night because otherwise the other users would hardly have any main memory left.

It became 1978. I complete my studies as a graduate computer scientist (FH). Three years later, I received my diploma in computer science (FH).

After graduation, we can assess the extent to which the educational recommendations for a computer science degree programme [4] from the Gesellschaft für angewandte Mathematik und Mechanik (GAMM) and the NTG were implemented in our computer science degree programme. To start with, there is a large overlap between the recommendations and our degree programme. The topics of automata theory, formal languages, Turing machines, coding theory, information theory, programming languages, system programming, data management systems, input/output devices, hybrid computers, data transmission, operational data acquisition and processing, network technology, higher mathematics, numerical mathematics and error analysis were also included in our degree programme.
Little or no mention was made of switchgear design, digital memory, human-machine communication, availability and maintenance, automatic document processing and system planning.
To this end, subjects were taught that go beyond the GAMM/NTG recommendation. These were business administration, chemistry, English, technical cybernetics, statistics, computer-aided simulation, the use of microprocessors and data protection.

Here I end the mental journey through time from my memory (and the scripts from my student days).

In the course of my studies and then in 35 years as a development engineer for hardware and software, I have learnt and used 12 higher programming languages, from ALGOL60 to FORTRAN, COBOL, PL1, BCPL, ELAN, PLM, PASCAL, C, C++, JAVA to C#, 12 assemblers from Siemens 4004, IBM360, NCR TCU, Z80, 6502, Intel 4004, 8080, 8085, 8086, 80186, 80188 to 80286. Programming techniques were also developed further. During our studies, we discussed whether the idea of dispensing with the GOTO command was even feasible. Today, no developer knows the command any more. Object-orientated programming and frameworking were introduced, to name just a few techniques. For 19 years I developed the electronics and firmware for document encoding devices for processing cheques and bank transfers, for data communication devices for connection to the circuit-switched data network X.21 and the packet-switched data network X.21 of Deutsche Telekom, as well as ISDN terminal adapters in all variants. I worked for a further 16 years as a software developer for ATMs and as a project manager.

Today's smartphones, which fit in a trouser pocket, cost just a few hundred euros, have several times more memory and are faster than the largest computers 50 years ago, which cost many millions of DM. This has created the conditions that have led to the immense importance of IT today. This importance will continue to grow in the future. As long as the supply of electricity and the raw materials required for electronics is guaranteed, nothing stands in the way.
There are professions where you never stop learning new things. Computer science is one of them. That's why I find computer science so exciting, thrilling, interesting and never boring. Once you've been bitten by the bug, you can't stop. I am now retired and have equipped ten nesting boxes for great tits, blue tits and starlings and a feeding box with microprocessor-controlled light barriers and cameras.[5] By recording the information on when and how often a bird flies into a nesting box, how long it spends inside and outside, and correlating this with the data from my weather station, I gain new insights into the behaviour of birds that even many ornithologists are not yet aware of. Of course, the data in my flight models and in the household are recorded and analysed using microprocessor systems that I developed and built myself.
I am grateful to all the professors - it was only men who introduced me to this fascinating subject of computer science. I would particularly like to mention Prof Dr Pöppl for the AW lecture on Medical Informatics. I had a heart attack in 2020. With the knowledge from his lecture, I was able to recognise the symptoms quickly, go to the doctor immediately and two hours after the first symptoms, two stents were inserted in the heart and blood flow was restored. Thanks to the rapid response, the heart remained virtually undamaged. So studying computer science is not just good for earning a living.

Sources / References:

[1] Nobody knows what cybernetics is. A comprehensible introduction to a modern science Hardcover - 1 January 1970, by Rolf Lohberg (Author), Theo Lutz (Author), Fidel Nebehosteny (Illustrator)
[2] https://tu-dresden.de/ing/informatik/ressourcen/dateien/studium/dateien/sonstige_dokumente/zugangsvoraussetzungen/zv_Was_ist_Informatik_060509.pdf?lang=de
[3] Schaltungen der Digital-Elektronik, (TOPP-Schaltungen Teil 15), author Wolfgang Link, 1968
[4] The establishment of computer science in the Federal Republic of Germany, page 29
[5] https://www.klaus-stampfer.de