Master's thesis

Master's Programme in Computer Science

Exploring study paths and study success in undergraduate Computer Science studies

Dennis Ahlfors

March 4, 2022

Faculty of Science

University of Helsinki

Supervisor(s)

Dr. Matti Luukkainen, Assoc. Prof. Petri Ihantola, Dr. Erkki Kaila

Examiner(s)

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Dennis Ahlfors

Työn nimi — Arbetets titel — Title

Exploring study paths and study success in undergraduate Computer Science studies

Ohjaajat — Handledare — Supervisors

Dr. Matti Luukkainen, Assoc. Prof. Petri Ihantola, Dr. Erkki Kaila

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	March 4, 2022	68 pages, 16 appendix pages

Tiivistelmä — Referat — Abstract

While the role of IT and computer science in the society is on the rise, interest in computer science education is also on the rise. Research covering study success and study paths is important for understanding both student needs and developing the educational programmes further. Using a data set covering student records from 2010 to 2020, this thesis aims to find key insights and base research in the topic of computer science study success and study paths in the University of Helsinki. Using novel visualizations and descriptive statistics this thesis builds a picture of the evolution of study paths and student success during a 10-year timeframe, providing much needed contextual information to be used as inspiration for future focused research into the phenomena discovered.

The visualizations combined with statistical results show that certain student groups seem to have better study success and that there are differences in the study paths chosen by the student groups. It is also shown that the graduation rates from the Bachelor's Programme in Computer Science are generally low, with some student groups showing higher than average graduation rates. Time from admission to graduation is longer than suggested and the sample study paths provided by the university are not generally followed, leading to the conclusion that the programme structure would need some assessment to better incorporate students with diverse academic backgrounds and differing personal study plans.

ACM Computing Classification System (CCS)
Social and professional topics → Computing education → Computing education programs → Computer science education
Social and professional topics → Computing education → Model curricula
Social and professional topics → Computing education → Student assessment

Avainsanat — Nyckelord — Keywords

computer science education, computer science curriculum, study paths

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software study track

Contents

1 Introduction

Research focusing on computer science education is abudant, but research in study paths in relation to study success is less common. As the business world craves for more educated IT workforce, universities and other educational institutions scramble to improve IT education, including computer science. This thesis aimed to function as a base study that would provide a high-level point of view into computer science education in the University of Helsinki on a 10-year time period. The scope of this thesis was focused on observing the relationships between study success derived by statistical methods and study paths visualized from the same data set. The phenomena discovered in this thesis provide a starting point for focused studies that provide more precise insight into the factors that cause them - and how to further improve the strengths and mitigate the weaknesses inherent in the programme. Comparing historical data based on partly different approaches to more recent data provides valuable insights into the development and effects of changes and what issues should be taken into account in the future. A data set containing pseudonymized student admission and course completion records between the years 2010 and 2020 was used as the base data. Setting the scope into base research allowed for a broader look that covers multiple interesting topics without a need to omit important relationships between study paths and study success. A broader scope also allowed for partial validation of prior research conducted in the University of Helsinki through the combination of study success and study paths.

Dividing students to major groups by computer science curricula was the first step in facilitating proper comparison. Students were separated to students admitted after the major computer science programme overhaul in 2017 labeled as group A, students admitted and that completed studies before the overhaul labeled as group B and students that transitioned from the old programme to the new programme labeled as group C. This division allowed for a high-level comparison of two different curricula and the trends within them. The students were further divided into four subgroups for more precise insights. The groups were all students, graduated students, students with a large amount of Open University studies and students with a large amount of mathematics studies. The first two groups are self-explanatory, but the two other groups are specific to computer science students in the University of Helsinki. The admission methods have expanded to include MOOC and DEFA combined with the growing amount Open University courses available freely through the aforementioned admission paths and the FMSEI project, the number of students with both prior Open University studies and students that use Open University courses to complement their studies after admission has been increasing. As this student group becomes larger, it is more important to understand study success and study path choices within the group. Computer science is inherently a mathematical subject and the role of mathematics in computer science studies could be thought to be a major factor. Thus, the final subgroup of students contains students with heavy focus on mathematics during their studies.

Novel methods of data visualizations were required to properly convey the complex study paths that students choose. The combination of data visualization and statistical results garnered by methods of descriptive statistical analysis provided the necessary tools to answer the research questions. These results were augmented with a theoretical background, though as study paths in compluter science are not extensively researched the focus is on extracting new knowledge. Regarding topics more related to educational methods and study success, referencing research from foreign institutions with different educational standards and programme structures provides contextual data, but might not be directly applicable to the student groups used in this thesis.

The thesis is structured as follows. The second chapter contains the relevant background information explaining the theoretical framework. In this chapter a look into previous study and topics of special interest are followed by the motivation and research questions. A short introduction to the Bachelor's programme in Computer Science in the University of Helsinki and the Computer Science curricula relevant to the data set are also present in this chapter. The third chapter outlines the data set and the methodology used to gain the insights required to answer the research questions. The chapter also explains the student group divisions and their motivations, which become important in the following chapters. The fourth chapter contains the results of the applied methods. The chapter contains the statistical key distributions related to both the general student mass and the focused student groups and the study paths discovered by visualizations, discussing the relationships between different admission years, student groups and individual courses. The chapter also presents the results from the statistical analysis and the study path visualizations as answers to the research questions. Further discussion into the implications and relevance of the results and the further research they might warrant is also presented. The final chapter is dedicated to the conclusion and summarization of the whole thesis.

2 Background

In this chapter I present related works and research, explain my motivations in writing this thesis and the research questions that guided the research. I also explain computer science education and the Bachelor's Programme in computer science in the University of Helsinki. The chapter is finished with a look into the major computer science curricula iterations in the University of Helsinki during the time period of 2010 to 2021. It should be noted that the information presented in this chapter is intended to be contextual background information and for scoping reasons there is no deep analysis of curriculum contents and educational decisions that might have affected certain changes or differences in content.

2.1 Related work

Relevant previous research on study success and study paths in computer science in the University of Helsinki has in the last years focused mainly on different outlooks on programming, programming courses and student admission through the Open University paths of admission. The Open University admission paths are explained in more detail in section 2.5. Papers concerning study paths and success of most interest from the point of view of this thesis are related to student admission. Leinonen et al. [23] studied study success of students admitted through the MOOC admission path and concluded that the students perform better in their computer science studies and have a slightly better percentage for completion of studies. Pirttinen et al. [32] conducted a similar study, focusing on students admitted through the DEFA admission route. They concluded that students in their focus group completed more credits and showed better success in their studies when compared to students admitted through other routes. Both of the aforementioned studies note that student motivation can be perceived to be higher than in more traditionally admitted students. Other studies functioned as interesting contextual information, but are not directly related to the research questions presented in this thesis. Among others, *Pass Rates in Introductory Programming* by Simon et al. [37] and student retention studies by Barker et al. [2, 3] provided valuable context from the point of view of student retention, educational sciences and reasoning that explain in their own right the phenomena discovered.

Research originating from foreign universities is more abundant and provide valuable additional context in regards of understanding the phenomena presented in this thesis. Especially in general educational topics like student motivation as researched by Williams & Williams [45] and Mitchell et al. [28] show that challenging but simultaneously rewarding course contents that reinforce feelings of personal achievement combined with flexibility lead to better student motivation. Student motivation in general and in specific areas of study is a field with an adequate amount of research, but due to the lack of qualitative reference studies that would tie the general motivation related phenomena to the computer science programmes in the University of Helsinki there are limitations that affect their use. The specific limitations are later discussed in section 6.1.

Broadening the research to cover faculties other than computer science, slightly more relevant research can be found. Lindblom et al. [24], Hailikari et al. [15, 16] and Määttä [26] have quite recently conducted research on student motivation and study path choices in students studying humanities and law in the University of Helsinki. These studies provide an interesting qualitative point of view into student motivation and the results are most likely partly applicable to computer science students. These studies conclude that a student's own choices in how they approach both study and university life in general has a noticeable effect on the study paths and study success. Outside these studies there is only a limited amount of research that would be directly applicable to computer science studies in the University of Helsinki. Valuable additional context can mostly be drawn from publications in educational sciences or official statistics published by Statistics Finland and the Ministry of Education. The scoping of this thesis prevents a deep look into educational theory, rendering most additional context redundant before a more focused study into the topic of educational methods in computer science education is done.

2.2 Study success factors

Understanding why some students succeed in their studies and some do not is a complex and multidimensional topic that spans from curricula, educational methods and programme design to subjective factors like prior studies, academic affinity and personal life situations. It is almost impossible to build a comprehensive picture of student success that would satisfactorily explain the plethora of different combinations of unrelated factors that might or might not lead to either success or failure. Even framing the question to only cover CS studies in the University of Helsinki it is challenging to discuss student success based only on numbers and assumed success factors, as without accompanying longitudinal qualitative research the context is inherently superficial. Some studies, for example Lindblom et al. [24], have covered individual study success factors in first year students through interviews, but due to small sample sizes, different faculties and contextual differences the results cannot be generalized to cover a long-time view on CS studies. However, individual success factors that are thought to affect study success in a positive manner have been widely studied and using these factors it is possible to both facilitate and predict study success in groups of students. Fowler and Boylan [13] categorize these factors to academic and nonacademic success factors, which is a good enough distinction to be used in this thesis. Socioeconomic status and related external factors, such as family income and parent's educational attainment have an effect on study success. The official statistical organization of Finland, *Statistics Finland*, shows that a parent's educational level greatly affects the child's choices and thus the educational level and academic skills they have [18, 40]. With the inclination towards academic studies being to some extent hereditary it is hard to quantify how specific factors affect specific students. Completely covering study success factors is outside the scope of this thesis, so the focus is on factors that can be considered to be most relevant from a generalized point of view.

Non-academic success factors are the larger group of factors, consisting of among others subjective factors, personal skills, social support and motivation [13, 25]. Nonacademic factors could be framed as the complete framework of personal, interpersonal, social and motivational factors that together form the basis for student success. These factors can be divided into risk factors and success factors depending on whether they have negative or positive effects on success. Risk factors as categorized by Horton [19] can be further divided into two distinct categories, behavioral risks and environmental risks, where behavioral risks are factors that are dependent on the choices and behaviors exerted by the individual student and environmental risks the outside factors that affect success. Examples of behavioral risks are lack of self-discipline, procrastination, irresponsibility, lack of motivation, bad study practices, insecurity and fear of failure. Environmental risks contain background characteristics, such as minority status, academic unpreparedness, history of failure, language barriers, financial considerations, lack of advisory services and workforce issues among others. In addition, individual characteristics can be a risk factor. Individual characteristics include for example learning and physical disabilities, unrealistic goals, social skills, negative social networks and attitude issues. Individual characteristics are closely linked to behavioral risks, but are different enough to be considered to be a separate category of risk factors [19]. Success factors are in many cases the inverse of the risk factors, meaning that by mitigating a risk factor by addressing the underlying issue it is possible to turn the factor into a success factor. The above factors have been shown to affect student retention and success in a major way [13, 19, 25] and are applicable to University of Helsinki students, though individual environmental factors like the financial costs of degrees might not be relevant. In addition, some behaviors and skills might translate to study success in CS studies. Studies by Byrne and Lyons [5] and Boyle et al. [4] show that academic skills, preparedness and applicable cognitive skills translate to study success to some degree, though conclusive results cannot be drawn. Some of the possible nonacademic success factors that are relevant to this thesis are discussed later in section 4.3.

Academic success factors are related to things that are specifically academic in nature. Lotowski et al. [25] classify ACT assessment scores and high school GPA to be purely academic success factors, though both of these factors are a composite of nonacademic factors that have their roots in academic topics, such as academic goals, motivation, skills and self-confidence. While it is debatable whether these factors should also be classified as academic factors, the academic success factors in this context are numerically quantifiable values and the nonacademic success factors are more qualitative phenomena. That being said, academic success factors allow for certain predictions about student success using statistical methods. It is more likely that students with a stronger GPA and a history of academic success also succeed in tertiary level studies, though study success is a combination of both academic and nonacademic success factors [25]. Prior academic success alone does not directly translate to success in Bachelor studies and prior studies have also shown that for example strong background in math does not necessarily indicate direct success in CS studies [4].

The background material for this thesis consists of student records that allows for calculation of GPAs and numerical analysis. The extent of success factors that can be derived is limited to academic success factors in the context of GPA during Bachelors studies. This limitation should be taken into consideration when looking at the study paths and GPAs associated with them, as they represent the quantifiable numerical truth but do not take into consideration possible nonacademic factors that affect the results. Some factors, such as the effect of remote learning versus classroom instruction cannot be taken into account as the base data does not indicate which method of course participation each course has had. A study by Yukselturk and Bulut [47] shows that there are partly different success factors that lead to success in online education, adding another layer of complexity into study success analysis. The differences in remote and on-site learning in relation to retention and study success would be an interesting additional research topic, considering the ubiquitousness of online and MOOC learning in today's curriculum.

2.3 Understanding study paths

To properly contextualize the results presented later in chapter 5, a look into the concept of study paths and educational options is required. The starting point is the definition of the abstract concept of a *study path*. The term has multiple meanings and is commonly used to refer to e.g. collections of courses by a common theme, different course options and flows inside curricula and the complete study experience including both of the former combined with extracurricular activities and other personal experiences [24, 26]. It has been shown that study paths leading to more successful studies are often a combination of course choices that complement students' personal interests and motivation, personal factors like academic motivation and goals and an environment that both enables and preserves a positive outlook to studies [15, 16]. Professionalism and educational skills in teaching has also been noted as a success factor in study paths [26]. This observation is important, as the changing course structures and shift to more self-guided methods of study such as MOOC's and other self-paced online courses might affect study paths from the success side in addition to the more obvious pacing and structure of studies. Measuring study success between primarily distance learning students and on-site students would be an interesting research topic in the context of this thesis, but as the base data does not explicitly indicate the methods of study for each student and course completion - such comparisons cannot be reliably drawn.

In this thesis the focus is on course completions inside a curriculum as the lack of qualitative studies into personal factors as noted in the prior section 2.2 severely limits a more complete analysis of different study paths. While we do not know why students chose to complete courses in a certain order, the data could be used to model different study paths as orders of course completion. Modeling study paths as course completion orders and the statistical distributions with median completion points gives valuable insight into how course completion changes between admission years, curricula and the subgroups of students. The aim of the modeling was to provide insight into how course completion times differ between groups of students and if those results can be tied to specific subgroups of students. Understanding the potential shifting of course completion order between both curricula and student subgroups is valuable in curriculum design as such phenomena have not, as far as I know, been modeled before. The visualization of study path shifts presented in chapter 4.2 is a novel representation in the context of computer science studies in Finland.

2.4 Motivation and Research Questions

As stated in the introduction, research into this specific subtopic of computer science education remains comparatively scarce. Recent research has focused on more focused topics, aiming to extensively study smaller phenomena in great detail. Understanding how students approach the education, choose to complete courses and succeed in them is of paramount importance to future curriculum design and educational considerations within the Bachelor's Programme in Computer Science. To properly address the topic, a sufficiently large sample set from both a temporal and a quantitative perspective was required. Taking a historical view into study paths and study success is required to properly understand they have evolved during the years, leading to more recent results relevant to the current curriculum. Overall, the main motivation was to do basic research that would result in information that is both new and has real world applications. Discovering new topics of research that could later be expanded and providing a base for future study was another major motivator. The multi-disciplinary considerations in educational theory, advances in both curriculum and day-to-day education design are exciting possibilities that could lead to improvements in computer science education as a whole and in the University of Helsinki.

Fortunately, the University of Helsinki has kept digital digital student records that could be used for this kind of research. From the beginning it was clear that this area of study would be challenging and scoping the research into a thesis scale would be hard to do without omitting topics that would contextualize the results. In this light, I decided to focus on a more high-level research approach to form a knowledge base that could later function as a starting point for more specific studies into the phenomena discovered in my thesis. This means that general educational theories, their localized applications and benchmarks between educational styles and differences in yearly course iterations are scoped out, as are more practical reasonings in particular curriculum design, reasonings for study paths patterns and analysis of offending factors outside a general level. To explore different study paths more comprehensively, I had to look into yearly curriculum design and suggested study paths for new students, completion data between courses and key statistical distributions. Prior studies in related topics, such as admission and study success research in MOOC students by Leinonen et al. [23] and study success research in DEFA students by Pirttinen et al. [32] provided valuable context, though general admission and study success statistics are not the main focus of this thesis.

My three research questions are as follows:

RQ1. Do students follow the suggested study paths?

RQ2. How do students in the student groups complete courses?

RQ3. How does study success in the study paths differ between the student groups?

The three research questions are equally important to this thesis. Answering *RQ1* through the data provides important information about possible discrepancies between expectations set by the university and the actual study paths taken by the students. This information is very important from the point of view of curriculum design. Finding the right materials to answer *RQ1* required searching for old curricula guidebooks and suggested study paths from internal university sources. These materials were scanned and added as appendix, as they are no longer available from public sources. Answering *RQ2* is equally important, as answering it provides a more focused view into what kinds of study paths students choose. Following *RQ2*, *RQ3* is required to establish actual study success differences between students groups and finding out if there are differences that could be amounted to choosing different study paths. Combined, these three research questions provide valuable information for future curriculum planning and programme design.

2.5 Computer science education in the University of Helsinki

Computer science education in the University of Helsinki has been available since 1967 [41], evolving over the years with expanding education, quality programmes and research programmes. The education follows a common pattern of offering a Bachelor's Programme, Master's Programmes with different areas of specialization and a Doctoral Programme. As is usual in Finland students generally apply to both the Bachelor's and Master's programmes jointly, gaining study rights to both programmes simultaneously - though a completed Bachelor's degree is a prerequisite to a Master's degree. A Bachelor's degree consists of 180 ECTS (*European Credit Transfer And Accumulation System*) credits worth of compulsory and elective studies culminating in a scientifical capstone project and a Master's degree consists of 120 ECTS credits also culminating in a larger scientifical thesis project. The Finnish government has set a three-year target time for graduation for a Bachelor's degree and a two-year target time for a Master's degree. For a Bachelor's degree completion should happen within a year of the target time, while for a Master's degree a similar limit of four years is given. [8] As of 2021, students can be absent from their studies for one academic year of two terms - two academic years or four terms before 2015, maternity or paternity leave or military service without the absence counting towards study time limits [8]. These absences do not need to be consecutive, meaning that students it is possible for students to take one term off for personal reasons. The possibilities for absence are important in relation to personal curricula more closely examined in chapter 4.2.

In this thesis the focus is on the Bachelor's Programme. The current Bachelor's Programme consists of compulsory and elective studies in computer science, a 25 ECTS credit set of either mathematical or method studies, language studies, a set of elective studies that can be outside the field of computer science and a capstone project. The content of the programme receives minor updates yearly with larger overhauls happening infrequently mostly due to changing University requirements. In addition to the traditional computer science programme offered by the Department of Computer Science, the Faculty of Science offers a joint Bachelor's Programme in Science that incorporates mathematics, physics, chemistry and computer science. While the joint programme contains partly shared content and graduates can apply for Master's Programmes in traditional computer science, it should not be confused with the dedicated programme offered by the Department of Computer Science and this thesis is strictly related to the latter. Curriculum design and the educational content of computer science studies in the Bachelor's Programme is outlined in more detail in section 2.6 and appendices A and B.

In addition to courses offered exclusively to degree students, a variety of Open University, DEFA, FMSEI and MOOC courses have been available to degree students in all programmes across all faculties and non-degree students. DEFA stands for *Digital Education For All*, which is a national joint venture encompassing multiple universities offering first-year studies in computer science and other fields to anyone and anywhere. Courses offered by DEFA contained most of the first-year studies required by the Bachelor's Degree and credits attained are directly usable towards a degree should the student have or attain a study right to the programme. Many of the DEFA courses were MOOCs (*Massive Online Open Course*), constituting a slight paradigm shift from traditional lecture-based education to either time gated or self-paced online education. FMSEI [10] is a project by the University of Helsinki that aims to provide further education in skills relevant to modern work environment both professionals and other interested attendees. The courses offered through the FMSEI project form a different set of courses with some overlap with DEFA, further increasing the amount of available Open University courses. DEFA originally started as a pilot project during the academic year 2018 - 2019 with its latest iteration finishing at the end of 2021. No official information on the continuation of DEFA as a way of either taking courses or as a path of student intake is available as of writing this thesis. DEFA is a relevant factor in more recent studies and study paths, as we will later discover in chapters 4.1 and 4.2.

The main methods of student intake in the time frame of this thesis are high school matriculation examination results, entrance examinations, the Open University path, MOOC intake since 2012 [23] and most recently DEFA intake between 2018 and 2021. Since the changes in general University wide in admission policies in 2016, students with no prior study rights in other degree programmes are greatly favored. The number of reserved spots for first time students mainly affects the traditional paths of matriculation examination results and entrance examinations. Good matriculation examination results in certain subjects can grant the student an automatic admission. In the recent years automatic admission through the matriculation examination has become the largest intake method, as changes to the general tertiary education admission standards [9] greatly favor admission through this method. Entrance examinations used to form the largest part of student intake prior to the aforementioned changes and the same examination is shared between multiple universities, meaning that students can state their preferred places of study in the order of interest and take the same examination. Traditionally the Bachelor's Programme in the University of Helsinki has had the highest score requirements for admission. The Open University path requires students to complete 25 ECTS credits worth of basic studies in computer science with a GPA of 3.5/5 to apply for admission [32]. To be eligible for MOOC intake students must complete 90% of the assignments on two timed programming courses to be invited into a separate MOOC entrance examination [23]. From this examination 50 best students are given study rights to the degree programme annually [23]. The MOOC path has been discontinued for the application year 2021. The newest path of student intake was the DEFA intake. To have gained admission through the DEFA path, a student had to complete from 50 to 60 ECTS credits worth of studies in a year from the available DEFA courses. These studies had to include 25 ECTS credits of basic studies in computer science and 5 ECTS credits of compulsory mathematics. [32] While DEFA can be considered to be an extension of the Open University path, there is was no average GPA requirement. According to the intake statistics provided in prior research by Leinonen et al. [23] and Pirttinen et al. [32], between 150 to 200 students gain study rights through traditional methods, 50 by MOOC admission, around 20 to 30 students by the Open University path, though there has been a rise in admissions during the last few years and just under 20 students through the DEFA path. The admission statistics illustrate that there have been multiple ways to gain study rights and new students have started with very diverse backgrounds. The same starting group of students might have included students with up to 60 ECTS credits worth of DEFA, FMSEI and Open University studies and straight out of high school students with no prior studies.

2.6 The University of Helsinki computer science curricula

The University of Helsinki Computer science curricula have been affected to varying degrees by the joint ACM and IEEE computer science curricula, a brief explanation of which has been added as appendix B. The University of Helsinki curricula have in part evolved independently from the guidelines, but are mostly comparable in content to the ACM/IEEE join curricula. While the latest full release of the ACM and IEEE curriculum is from 2013 [11], a major structural update in the University of Helsinki Bachelor's programme and by extension the computer science curriculum happened in 2017. This warranted a contentual split to pre-2017 and post-2017 curricula, aptly named "the old curricula" and "the new curricula". These brief explanations also function as crucial background information in how the programme is designed, in contrast to the study path exploration done later in section 4.2.

The materials covering the old curricula used between 2008 and 2017 were found out to be scarce. The online resources that once provided information to the students have either been removed or have vanished during various migrations to new sites. While some sources remain, a complete collection of materials to be used as reference was impossible to build. Official curriculum guidebooks exist only as singular physical copies archived by the education coordinators of the Department of Computer Science and only in their Finnish language forms. Scans of these documents are provided in appendix A. Other additional materials, such as those that have possibly been handed out to new students and functioned as additional material to the official guidebooks, could not be traced and are most likely lost in time. The lack of additional materials raises a slight validity issue, as guidebooks for academic years between 2008 and 2012 contained detailed sample schedules for studies accompanied with a suggested timeline of completion. These schedule examples can be seen in figures A.2, A.3, A.5 and A.6. The following guidebooks for academic years between 2012 and 2016 omit these materials and focus on the course content and general student information, with the sample schedules having been distributed as separate documents. As these resources are no longer available, it is impossible to verify whether the suggested timelines and schedules have changed between 2012 and 2016. More obvious changes, such as changing course codes, splitting or combining of courses and inclusion of new courses have been individually accounted for in the figures later in sections 4.1 and 4.2.

A prerequisite system shown in figures A.3 and A.6 gives students a clear path of progression through the compulsory courses through the suggested three years of study time. The same prerequisite system was in effect until 2017 with minor course changes. The system follows a path starting from introduction to programming and branching out to different advanced topics, for example basics of software engineering leading to different individual projects and culminating in a larger software engineering project completed by a group of students. This progression system ensures that certain basic skills and knowledge is attained before moving further into more advanced topics. The charts show only computer science prerequisites though, some courses may have had suggested prerequisites in Mathematics or Statistics. For example, figure A.3 shows *Introduction to Discrete Mathematics* as a suggested prerequisite to *Data Structures* and *Introduction to Databases*. While this prerequisite is not shown in figure A.6, the same suggestion remained through the following curricula.

The general outline of the new, post 2017 curricula in the new Bachelor's programme is in essence similar to the prior curricula. The base structure of studies is identical and the compulsory courses required to complete the programme are for the most part the same. The most noteworthy changes in course structure are the ECTS credits rewarded for completion. Comparing for example the structures for admission years 2010 to 2012 shown in figure A.4 and 2014 to 2016 in figure A.10 to the newest course structure [33], it is apparent that the courses currently have a more uniform structure of 5 ECTS credits per course for a period of study. The older course structures show a wider range of credits awarded and longer courses. More recent changes include splitting the formerly 10 ECTS and two study period *Data Structures and Algorithms* course into two independent 5 ECTS courses. A wider array of both educational and completion methods is also utilized, giving some courses multiple paths of completion through among others lecture based study, MOOC's and self-paced individual study without formal examinations. Naturally the course offering and course content has evolved during the years, with outdated ones being replaced with more up to date courses relevant to modern computer science. This change mostly affects optional courses though and the compulsory core educational content is more or less the same.

The major differences between the old and new curricula are the introduction and incorporation of alternative admission routes, emphasis on flexibility through booth the Open University and changes in educational methods. The alternative admission routes and freely available courses discussed earlier in section 2.5 greatly diversifies the student material admitted to the programme. Students can have in some cases up to 60 ECTS credits completed prior to admission, forcing the students to pick more challenging courses and to plan very different personal study paths compared to the students that are admitted without prior studies. These students could be considered to be like transfer students, who are moving from a very different style of studying to another. Studies by Laanan [22, 34] and Kwik et al. [21] show that transfer students can have more difficulties in adjusting to their new university as degree students. As far as publicly available materials reveal, there is no special onboarding path for students admitted through the new admission paths. Kwik et al. [21] also note that transfer students show lower GPA's than their non-transfer peers, but that result is contradictory to the results shown by Leinonen et al. [23] and the results found in this thesis introduced later in chapter 4.1. Regardless, the more academically diverse student material offers new challenges to curriculum design.

On a general level the curricula adhere to the principles of CS2008 and CS2013, covering most if not all of the knowledge areas with multiple knowledge areas being present in some courses. The compulsory basic studies consisting of introductions to programming, databases, theoretical computer science and basics of software engineering cover 25 ECTS credits. Advanced studies, of which some are compulsory and some elective, cover 61 to 63 ECTS credits. With a single ECTS credit equating to approximately 27 hours of study the minimum amount of calculatory total study hours devoted to various computer science studies varies between 2322 and 2403 hours. This number includes both the instructional hours and individual work hours. While the number of instructional hours varies from course to course with the students also having an option to complete courses with separate examinations, it can be assumed that the minimum hours specified by CS2008 and CS2013 are achieved within the curricula. Compulsory courses in both basic and advanced studies contain topics from all of the core knowledge areas with elective courses widening the possibilities of specializations in certain areas. There are no major deviations from the spirit and principles of either CS2008 or CS2013 that would warrant further exploration. The additional flexibility through optional completion paths and more widely available courses moves the new curricula closer to the philosophy of broader knowledge sets with students choosing courses that interest them instead of a rigid and forced curriculum. The new programme has also introduced courses that reflect new and trending technologies, like the *Full Stack Open* course and *DevOps with Docker* courses to name a few. The splitting of larger courses into equal parts and unifying the credit hour requirements between courses is also as a move closer to the CS2013 principle of managing the size of essential knowledge.

3 Data, methodology and student groups

In this chapter I explain the data set used in this thesis and how I decided to approach the data. The first section gives a short introduction to the data set used. The second section discusses the methodology of statistical analysis and data visualization related to the thesis. The third, fourth, fifth and sixth sections outline the student groups chosen for further analysis. In the final section I present the more fine-grained subgroups within the student groups that allowed for more in-depth analysis of study paths and comparison of student success.

3.1 The data set

The data set used in this thesis consisted of student records between the years 2010 and 2020. The data was polled from the university systems in a raw format. Student identification numbers were pseudonymized before the data was handed over and could not be directly linked back to the real student, though individual students might have been recognizable should one have known the exact courses and grades attained by that specific student. Complete anonymization of the data was not possible due to individual row level data being required to explore study paths and study success. The main data content was delivered in two separate files, one with student information complete with the pseudonymized student identification, the programme they studied in, date of admission and whether the student has graduated or not. The second file contained course information with the pseudonymized student identification, course code, date of completion, grade and type of accomplishment. Supplementary files containing course information for the course codes were also supplied. In total, the data set contained 997 students who have started their studies with the new curriculum, 872 students who have started their studies with the old curriculum and have either graduated or paused their studies as so that they have not gained studying rights to the new curriculum and 536 students that have started studying with the old curriculum but have later transitioned to the new curriculum. The data set also contained 103 401 rows of course credit information, course information for 466 courses in the new curriculum and 3130 courses in the old curriculum.

3.2 Data analysis and visualization

The major methodological considerations in this thesis were related to data analysis and visualization. The first consideration was the implementation of research in educational sciences to the field of computer science education and more specifically their use as interpretative contextualization of phenomena discovered by other methods of research. The second consideration was the statistical analysis and interpretation of data to discover study success, differences between student groups and other distributions that provide additional insight into the research questions. The final consideration was the visualization of data to sufficiently convey differences in study paths between student groups while taking into account curricular differences and other factors that might affect the clarity of the visualization methods. The limitations in these methods are discussed later in section 6.1.

Finding the relevant statistical distributions, means and significant comparisons between groups of students was one of the most important tasks to be tackled in this thesis. Statistical methods were directly required to answer both *RQ1* and *RQ3* - and to provide important context to the later discussions and conclusions about the insights found in this thesis. For *RQ1*, there was a need to eliminate sets of students from existing data sets through certain conditions while for *RQ3* a need for more standard statistical methods to find key figures of study success. The most significant choices were the statistical methods to be used with the data. With a plethora of possible statistical methods suitable for use, a large emphasis on understanding the data and results relevant to the research questions was required. Descriptive statistics played a key role in understanding the most important summarizations of both the data as a whole and the multiple subgroups of students that formed their own subsets of data. Using mostly methods central to descriptive statistics like averages, minimums, maximums, correlations and deviations [12], the key statistical values for comparisons in study success and important general context like the graduation rates were found. Simple bar charts were also used to visualize notable key distributions. Considering the more high-level scope of this thesis, application of more advanced statistical methods was not deemed necessary. That being noted, further research would certainly warrant the use of other statistical methods, such as time-scaled correlations between students of different admission years, variability of grades within certain courses or even probability distribution changes between the programmes.

The visualization of study paths formed another significant part of this thesis. To properly answer *RQ2* in an understandable manner, visualization of the study paths was deemed neccessary. The aim of using visualization to complement statistical results is to offer further insights and present data in a way that is both clearer and more understandable than simple numerical statistical data [42]. This high-level goal itself does not warrant visualization however, utilizing visualizations is also dependent on the use case and the value of the data being represented. Visualizing study paths through course orders is very challenging. Students tend to not complete courses outside the very basic courses in a logical order, as we later discover in chapter 4.2. With a sufficiently large data set with an even larger variety of possible course combinations, visualizing the study paths as Sankey and similar common visualization styles led to poor results. This was consistent with the insights provided by an earlier student group that tried visualizing similar data as a course project. Finding a balance of simplicity and conveyed information on a challenging data set was found through explorative visualization and a trial-and-error method. A satisfactory visualization style ended up requiring normalization of the data from dates as comparison attributes to integer values between admission dates and course completion dates. This allowed visualization through standard statistical plotting familiar to most readers. The use of boxplots was a logical choice to visualize differences between groups of students and their respective study paths and complement the statistical methods used, as standard deviations and distributions are important in descriptive statistics.

Incorporating multidisciplinary topics without the possibility of referring to relevant research conducted with similar research paradigms is challenging. Computer science education research in the University of Helsinki has been focused on other topics and no large scale studies have been conducted relating to study paths. Research in study success and specific admission routes has been done, with that research forming most the theoretical background on which many of the results in this thesis stand on. The theoretical background was expanded to foreign research after the formation of the base theoretical background from research conducted in Finnish universities. However, due to the doubts in comparability in foreign research, a major literature review would not have served its common purpose. This ultimately led to the usage of foreign research as supportive material that functions as possible explanations and clues on interesting approaches to future research within the University of Helsinki. The articles used were chosen both due to their generalist nature and relevance to all of the research questions.

3.3 Identifying student groups

To better understand the differences in study success and personal curricula for individual students, certain divisions within the data were made. The major curriculum changes in 2017 warrant a general division to two student groups. Focus group A thus consists of students that started studying under the new curriculum and student group B of students that started studying under the old curriculum, but have either graduated or dropped out without receiving study rights to the new programme. This distinction is important due to students having had the ability to either graduate from the old programme or move over to the new programme and continue their studies there. These curriculum transitioners have been separated to a third student group C.

The division to three student groups allows comparisons in how students have chosen their study paths in two distinct curriculum styles. Adding a third group that contains the curriculum transitioners to the comparison aimed to give some clues how the transition between programmes has affected study paths and success. Certain limitations do apply though, as later outlined in subsection 3.3.3. To complement general information provided by the larger curriculum-based student groups, a further division to specific subgroups was made. These subgroups are shared by all of the main student groups with the aim of giving more precise insight into the differences in study choices between curricula.

3.3.1 Group A: new curriculum, 2017 -

The first major student group consists of students that have only attained study rights for the Bachelor's programme modeled with the new curriculum for the academic year of 2017-2018. While the available courses and structure of studies looks quite similar on the surface compared to structures relevant to the curriculum followed by the students prior to 2017, there are major differences that warranted splitting this group from the other groups. The most obvious differences are changes in course content and structure, major changes in the ways a course credit can be attained, new and updated courses that better reflect current trends in the field of computer science and differences in student admission, as described earlier in section 2.5. The group can be considered to be a little more homogenous in terms of available courses and course content compared to the other student groups. Only a few major course content overhauls have occurred during the relatively short period of time since the introduction of the new curriculum.

Students in the new curriculum group contain students that have had an easier access to freely available Open University and DEFA studies and as we later find out in section 4.1.1, almost half of the students in this student group have attained credits in CS courses before attaining study rights to the programme. With the number of studies prior to attaining studying rights ranging from singular courses to DEFA students with 50-60 ECTS credits worth of studies, there is a larger variance in prior student knowledge in CS compared to the other student groups. The variance in prior knowledge combined with a modernized and more flexible curriculum is a possible cause for differing study paths compared to the other student groups. These study path choices are interesting in relation to study success and the subgroups introduced later in subsection 3.3.4. These points make this group interesting and combined with prior findings like better student retention and study success in both MOOC [23] and DEFA [32] admitted students, there is a motivation to examine the students in this group as a separate student group.

3.3.2 Group B: old curriculum, 2010 - 2017

The second major student group consists of students that had study rights to the Bachelor's programme based on the old curriculum and who have either graduated or dropped out from their studies without attaining study rights to the new programme. The distinction of either graduating or dropping out can be concluded by the term limitations described earlier in section 2.5, as the last students in the old curriculum were admitted in 2016 and reached term limitations during the academic year 2020 - 2021. Students who were admitted during the old curriculum and have since transitioned to the new curriculum have been separated to another student group discussed in detail in the following section.

As the student group spans a larger time, students in the student group were exposed to a larger variety of courses and changes in content. With the rapid development of both software development tools and the general IT landscape, it is clear that students who started in 2010 have faced a different approach to e.g. programming than students who started relatively recently in 2016. While the basics of computer science education, programming and mathematics have remained the same, the general approach to education with for example MOOC's, DEFA, evolving educational methods, additional course completion methods and co-operation with the IT-industry lead to a very different overall study environment between curricula. Contentual overhauls happen gradually, with students mixing updated courses with older courses until at some point the older courses are batched out in favor of the updated courses. The gradual change is less dramatic than major changes in programmes, even if old and new programmes tend to run in parallel until students in the old programme either graduate or switch programmes.

3.3.3 Group C: transitioners

The third and final student group contains the students that have attained study rights to both the old and new curricula. For the purpose of this thesis, these students are called transitioners. There are two distinct groups of students within, students who have switched programmes organically by either their own voluntary decision or by administrative decisions and students who have reached term limitations earlier during the old curriculum and have later gained study rights to the new programme by applying for extension with a detailed study plan [38], after which the student would gain study rights to the newer programme as the older programme has been phased out.

The students in this student group form an interesting mix of varying study paths and courses. Depending on the point of transition between curricula, the students could have incorporated both old and new studies to their degrees and possibly have had to retake courses depending on whether the credits are transferable to the new curriculum. The students thus cannot be considered to be part of either of the two other student groups and require their own group to differentiate from the more uniform groups. For certain points, such as graduation rates and retention the students in this group complement results in the old curriculum group, as curriculum transitions do not change their original admission date nor does the transition extend term limitations. In other topics like the prevalence of Open University studies, the students might complement the new curriculum group. In this light, separating transitioners from the uniform programme-based groups leads to the two other groups better representing the students in the context of this thesis and also forms an interesting reference group that could provide some basic insight into effects of major programme overhauls to study success and study paths.

It is important to note that the factors that lead to students being forced to transition programmes, as in them not being able to graduate within their own programme even if such option would be present, cannot be explored in the context of this thesis. Student absenteeism in addition to unique factors such as mandatory military service, parental leaves, entering the workforce and plain dropping out due to other reasons have not been studied in the context of computer science education. For example, entering the workforce or working full time during studies is noted to affect completion of studies on time in a negative way in a study done by the Finnish Ministry of Culture and Education [43]. Working while studying is also considered by many students to be a factor that delays studies [1] and taking into account the good job prospects in the IT field, it is more than likely that this is one of the factors that cause delays and programme transitions. However, more research into these factors and reasons for dropping out of studies is needed before they can be satisfactorily taken into account.

3.3.4 Further groupings

To gain better insight into the student groups explained in prior subsections, a further division into subgroups of interest was devised. The subgroups *graduated students*, *all students*, *students with a large amount of Open University studies* and *students with a large amount of mathematics studies*. These subgroups are student groups relevant to RQ2 and were prechosen to function as rough generalizations of common study paths. While further divisions and even equally important subgroups could no doubt be identified, in the scope of this thesis these four subgroups are sufficient to yield meaningful results.

The graduated student's subgroup contains students that have graduated and it is also a subset of the all students subgroup. The reason of including both graduated and all students is the fact that the graduation rates within the old curriculum group are much larger than in the new programme group. While the results in graduated students especially in the old curriculum group might not be very conclusive, adding non-graduated students allows for a more comprehensive look into the differences between the old and new curriculum groups. While looking at graduated students to establish patterns that led to a degree is important, due to somewhat poor graduation rates as shown later in table 4.1 and figure 4.1, it is also interesting to explore patterns in non-graduated students. Student retention in the transitioners group is of special interest in these two subgroups, as it will most likely show whether the transition has caused deviation in study paths compared to their peers in the old curriculum group.

The first special interest subgroup is *students with large amounts of Open University studies*. A large number of studies is considered to be over 25 ECTS credits worth of studies, with Open University studies done prior attaining study rights being accepted as a part of these studies. The first subgroup stems from prior research results by Leinonen et al. [23] and Pirttinen et al. [32] that show MOOC and DEFA students having better study results and retention compared to students admitted by other means. The core objective of the first subgroup is to explore whether the same effect can be observed in students that have done lots of Open University coursework without necessarily being admitted through MOOC or DEFA and if the effect is present in all of the three student groups.

The final subgroup, *students with a large amount of mathematical studies*, stems from the fact that computer science is a mathematical subject in its core and some of the currently most hyped fields of study, such as AI, Machine Learning and Data Science are all heavy in mathematical theory. Like in the subgroup dealing with Open University studies, a large amount is considered to be at least 40 ECTS credits worth of studies. The increased amount of required credits for this subgroup stems from the fact that all students are required to complete 25 ECTS worth of mathematical or method sciences studies. While no prior research in how mathematics correlates to computer science study success in the University of Helsinki have been done, other research concludes that affinity and strong background in mathematics has a measurable effect on student success. For example, Campbell McCabe [6] established a relationship between mathematics and computer science success back in 1984, with more recent studies complementing the results. For example, Hakkarainen [17] and Wilson [46] noted the positive effect of mathematics in relation to study success in general. Additionally, Mieskonen [27] and Nunez-Pena [29] individually confirm the negative effect of poor mathematics skills to general study success. Concerning course choices, both Sigurdson & Petersen [36] and Whalley et al. [44] established that while students feel that mathematics are important in relation to computer science studies, there were clear adverse reactions to choosing courses that are perceived to be hard or not directly applicable to computer science studies. Other results do however exist, e.g. Boyle et al. [4] established that students with a weaker mathematics background do not have distinct disadvantages in computer science studies. The results might thus be dependent on individual study paths, as students with weaker mathematics backgrounds might refrain from picking mathematics heavy courses. Replicating some of these results by modeling personal study paths would provide a good basis for future research discussed later in section 5.3.

4 Results

In this chapter I outline key statistical distributions and data visualizations from the dataset used as the backbone of this thesis. The chapter is divided into two major sections covering the individual methods followed by a look into possible reasons for the results in the final section.

4.1 Statistical distributions and study success

In this section I explore the statistical distributions. The first subsection contains a short look into key distributions between the student groups like graduation rate, distribution of students and the prevalence of Open University studies in personal study paths. The following subsections take a more in depth look into the three student groups and the related key statistical distributions.

4.1.1 Key distributions

To better understand the student groups introduced in section 3.3, a look into some key distributions is necessary. Firstly, looking at the graduation rates in table 4.1, only 9% of students in the new curriculum group have graduated, while 53% of students in the old curriculum group have graduated. Considering that the suggested time to complete a Bachelor's degree is three years, the graduated students in the new curriculum group mostly consists of students that started studying in 2017 and 2018. This is also apparent when referring to graduates compared to admissions in figure 4.1. 2020 was omitted from the figure due to no graduations as of yet. Running a simple Pearson's product-moment correlation calculation, it is apparent that the graduation rate correlates with the number of admissions with a correlation coefficient of *0.82* and a p-value of *0.02*. The implications about student retention rates between admitted and graduated students in the figure are harrowing, though factors such as military service, planned absences, slowing of studies due to entering the workforce must be taken into account. As explained in section 2.5, students can use a total of four academic years and one year of absence to complete a Bachelor's Degree, meaning that the graduation rates for students that were admitted from 2016 onwards may still rise after the data used in this thesis was polled from the student record system. This topic would certainly warrant further qualitative research into student retention factors. Within transitioners in the transitioners group the graduation rate is 15%, which is a low percentage. As the transitioners group consists mostly of students that were admitted in 2015 and 2016 and those students had the option to graduate from the old programme or move over to the new programme, this percentage is in part complementary to the graduation statistics for the old curriculum group. This is reflected in figure 4.1 showing slightly higher graduation rates for admission years 2015 and 2016.

Student distribution by student group		
Group	N, graduated	N, not graduated
A: New curriculum	88	909
B: Old curriculum	460	412
C: Transitioning students	81	455

Table 4.1: Student distribution by student group

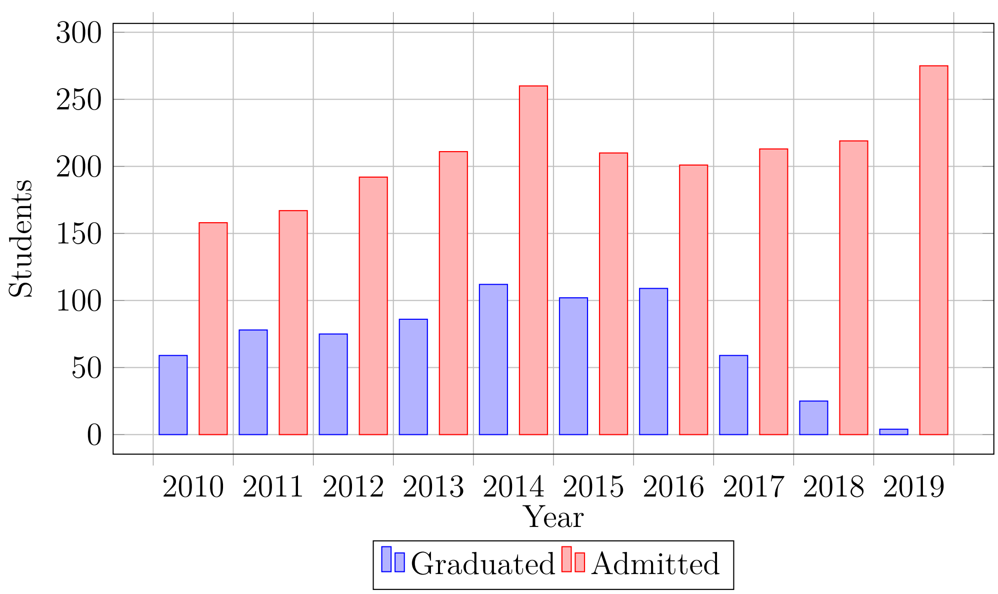


Figure 4.1: Graduated students per admission year, 2010 - 2019

Another interesting distribution is the number of years of studies between admission and graduation shown in table 4.2. The years in the table represent the academic year during which the student has graduated. The results for the transitioners group have been calculated between admission to the old programme and graduation from the new programme. In the new curriculum group, the first students are starting their fifth year of studies as of the writing of this thesis. From the table it is apparent that while a portion of the students graduate in the suggested time frame of 3 to 4 years, a sizable portion stretches their studies over that to five or even six or more years. This is a fascinating result with some important implications to student study success, study paths and student retention in the programmes in general. For some reason students do not graduate in the planned time and the causes are not deductible from statistical results alone. Some possible explanations and offending factors are discussed later in section 4.3.

Years of study between admission and graduation						
Group	1	2	3	4	5	6+
A	3	19	42	24		
B	2	17	84	115	97	145
C	0	1	12	22	23	23

Table 4.2: Years of study between admission and graduation

Looking at the distribution of students with Open University studies in table 4.3, the number of courses taken during or before studies is heavily skewed in favor of the new curriculum group. Of these students, 72% have completed one more course compared to only 16% of students in the old curriculum group. This is most likely due to Open University courses being more widely available and more closely integrated to the curriculum as explained in section 2.5. For students in the old curriculum group, the availability of Open University courses has generally been much lower. In the transitioners group, 31% of students have taken at least one course, which might be explained by the same factor as with the new curriculum group. This is also apparent as shown in figure 4.2, which shows that Open University studies have been quite marginal before admission years 2014 and 2015, after which the prevalence of these studies has been steadily rising. Prior to 2014 in the old curriculum group, it is quite likely that primarily students who were admitted through the Open University admission path have Open University studies, while in the new curriculum group students with no Open University studies are rarer.

Students with Open University studies		
Group	No studies	One or more courses
A: New curriculum	284	713
B: Old curriculum	731	141
C: Transitioning students	368	168

Table 4.3: Students with Open University studies

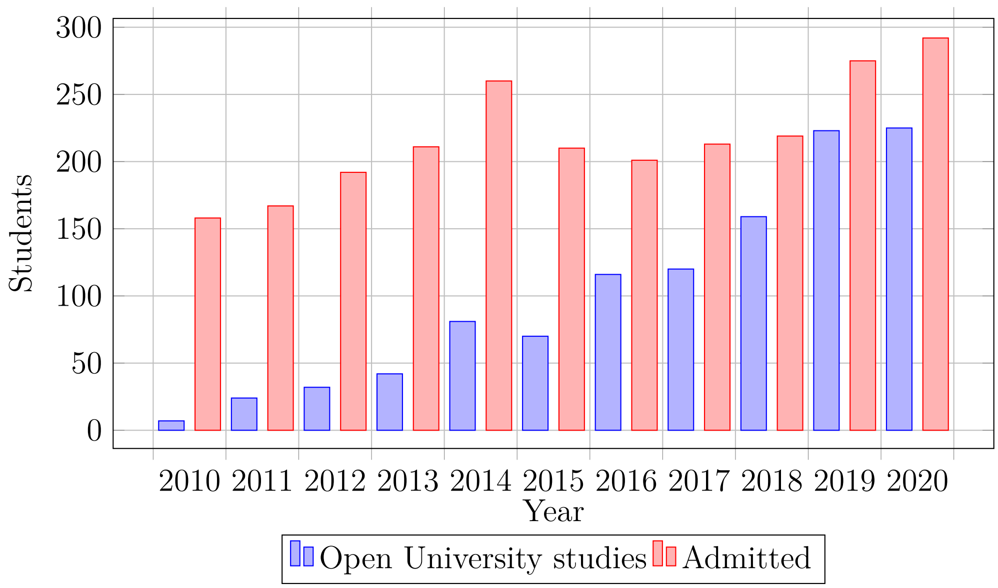


Figure 4.2: Number of students with Open University studies by admission year

Referring back to section 2.5 another key distribution is students with Open University studies in computer science prior to obtaining right to study. This distribution is shown in table 4.4. In the new curriculum group, 49% of students have completed at least one computer science course in the Open University, being consistent with the changed student intake methods and larger number of available courses. In the old curriculum group only 7% of students did the same, which is a low number considering that admission through Open University studies is not a new concept. In the transitioners group 17% of students have completed one or more of these courses, but this number also includes courses that might have been completed while the students had study rights connected to the old curriculum and that were completed before they transitioned over to the new curriculum. The percentage for the transitioners group can thus again be considered to be partly complementary to the results for the old curriculum group, but at the same time it indicates a similar trend as shown table 4.3 and figure 4.2 about the prevalence and importance of Open University being higher in the new curriculum.

Students with prior Open University studies in computer science		
Group	No studies	One or more courses
A: New curriculum	508	489
B: Old curriculum	807	65
C: Transitioning students	444	92

Table 4.4: Students with prior Open University studies in CS

4.1.2 Group A: new curriculum, 2017 -

Starting with the new curriculum group and looking at the distributions by subgroups in table 4.5, we notice a few interesting facts. The share of students with a large amount of Open University studies is 23%, which is a comparatively large percentage. It would seem that additional admission routes and greater availability of Open University courses leads to students incorporating said courses into their degrees. As noted in the prior section, the graduation rate remains low and thus the subgroup containing graduated students is also relatively small. Interestingly, compared to the other major student groups the number of students with heavy focus on mathematics is clearly lower. This might be explained in part with changes in curricula, but nevertheless show dwindling interest in picking mathematics courses. The graduation rates in the subgroups differ from the general graduation rate of the student group. Of students in the Open University group 10% of students have graduated, while in the mathematics group the rate is 35%. The implications of the graduation rates are discussed later in chapter 5.

Students in the new curriculum group by subgroup		
Subgroup	N, students	Percentage of total
Graduated	88	9%
Open University	232	23%
Mathematics	144	14%

Table 4.5: Students in new curriculum group by subgroup

Looking at the average GPA in all CS studies for the subgroups in table 4.6 the average GPA is quite high for all of the subgroups. This might be partially explained by some of the students especially from the more recent admission batches having not yet reached the more challenging courses. The distribution is likely skewed towards the early courses and thus cannot be directly compared to the more complete degrees in the other groups. The reason for a higher GPA for graduated students could be thought to be a result of more motivated and gifted students graduating, while their peers with absences or delays lagging behind. The GPA for graduated students will most likely drop with a larger sample size. It is still noteworthy however that students both in the graduated and mathematics groups have distinctly higher GPA than other subgroups. These results are in line with the results from the other groups shown in the next sections. The GPA for Open University group is also slightly higher compared with all students in the group, but the result is not as dramatic as in the old curriculum group. The results seem to be in line with the results in prior studies that indicate better study success in Open University and DEFA admitted students [23, 32]. This might be explained by the fact that both Open University studies and admissions through Open University studies are more prevalent in the group as a whole, leading to a situation where Open University studies is the norm.

GPA in CS studies, new curriculum group by subgroup	
Subgroup	GPA
All students	4.10
Graduated	4.29
Open University	4.14
Mathematics	4.28

Table 4.6: Average grades in CS studies, new curriculum group by subgroup

4.1.3 Group B: old curriculum, 2010 - 2017

Looking at the subgroups in the old curriculum group in table 4.7, it is noteworthy that while the percentages of students in the open university and mathematics subgroups look quite low, the numbers do not represent reality. The numbers should be viewed in combination with the numbers in the transitioners group presented in table 4.9, as it is more than likely that many of the students divided into the transitioners group are also students who for example completed Open University studies before transitioning between programmes. The numbers do however indicate that the share of students with a large amount of Open University studies is generally lower than in the new curriculum group, which most likely results from the differences between curricula. The results for the Open University subgroup are as expected. The mathematics subgroup is notably larger than in group A. This fact can be partly attributed to two known factors, the other being the possiblity of completing a larger set of mathematics studies as a full minor subject and some Master's Programmes either requiring or strongly advising completion of 60 ECTS credits worth of mathematics studies before admission. These factors might have led to more students investing in mathematics studies when compared to group A where such requirements and study paths are generally not available. Like in the new curriculum group, the graduation rates for the subgroup are different from the total graduation rate. Both in the mathematics group and in the Open University group all students have graduated. Like with the new curriculum group, the graduation rates are discussed later in chapter 5.

Students in the old curriculum group by subgroup		
Subgroup	N, students	Percentage of total
Graduated	460	53%
Open University	23	3%
Mathematics	270	31%

Table 4.7: Students in the old curriculum group by subgroup

The GPA's in CS studies presented in table 4.8 are lower than the ones for the new curriculum group, but as noted earlier, the GPAs in the new curriculum group are likely skewed towards early courses and have a smaller sample size for completed degrees. In the old curriculum group, the average GPA for all students and graduated students is similar, with the GPA in Open University and mathematics subgroups being slightly higher similarly to the new curriculum group. The result for the Open University subgroup is in line with prior results by Leinonen et al. [23] that used a sample of students included in the old curriculum group. The lower GPA in the mathematics subgroup compared to group A is an interesting find, though might be related to the fact that the GPA's in group A are most likely skewed.

GPA in CS studies, the old curriculum group by subgroup	
Subgroup	GPA
All students	3.74
Graduated	3.79
Open University	4.01
Mathematics	3.90

Table 4.8: Average grades in CS studies, the old curriculum group by subgroup

4.1.4 Group C: transitioners

The student distribution in the transitioners group illustrated in table 4.9 can be considered to be partly complementary to both the old and new curriculum groups. From this point of view, the results themselves are not that interesting as separate percentages. Taking into account a point made earlier in section 3.3.3 about the group possibly containing students who have applied for study time extensions due to absences, the percentages become more meaningful. While the data does not contain information about whether a student is an earlier absentee or not, it is surprising that the graduation rate in the group is as low as 15%. This would indicate that taking the effort to transition between curricula does not result in graduations nearly as much as would be expected. Similarly, to the other major student groups, the graduation rates for the distinct subgroups differ from the total graduation rate. Students in the mathematics group have a 35% graduation rate, while the Open University group has a slightly larger than average graduation rate of 20%.

Students in the transitioners group by subgroup		
Subgroup	N, students	Percentage of total
Graduated	81	15%
Open University	52	10%
Mathematics	102	20%

Table 4.9: Students in the transitioners group by subgroup

Looking at GPA's, the averages seem to be lower than in the other major student groups. The GPA for graduated students is similar to the old curriculum group as expected from students who transitioned from that programme, but the GPA's in all groups are lower. Whether this is a result of course content, difficulties in adapting to a new curriculum or from other factors is unknown. The result for the Open University subgroup is not in line with the results from the old and new curriculum groups, where similar subgroups have achieved better GPA's than other subgroups. One factor that could lead to a lower GPA is hinted at in table 4.2. The table shows that the majority of students in the group graduate four or more years, which is longer than the suggested time to graduation. It is possible that lack of practice due to absences or poorer performance caused by limited study time due to work or other constraints have an effect on the results. That being said, all of the factors that cause this discrepancy are not present in the data set and finding the cause of this phenomenon would warrant further research.

GPA in CS studies, the transitioners group by subgroup	
Subgroup	GPA
All students	3.60
Graduated	3.87
Open University	3.55
Mathematics	3.76

Table 4.10: Average grades in CS studies, the transitioners group by subgroup

4.2 Exploring study paths

The dispersion of data points into a comparatively long timeframe of ten years presented challenges. As each academic year has slightly different dates for examinations, course starts and period changes, an easy way of putting different curriculum, admission years and groups of students to equal terms had to be devised. It would have been a monumental task to correctly manually model each academic year to take into account possible examination dates, completion points and for example public holidays that might have affected specific years. The base data set only contained the dates of completion that might or might not be representative of a date within a specific period of study. Knowing that the processes for marking courses as done can differ between courses and individual teachers, it would have been almost impossible to correctly model each academic year.

Taking these challenges into account, the study paths were modeled by counting the difference in days between attainment of study rights and course completion. While a simple difference between two dates is not an accurate way to present complex study paths, it is good for visualizing the differences in course completion between both student groups and subgroups within them. A decision was made to limit the visualization to cover two years before attainment of study rights to account for possible Open University studies. An upper bound of six years was chosen to allow inclusion of students with absences and to account for possible extensions in study right. Course completions outside the eight-year frame were considered to be large outliers and were omitted from the model. To further increase the validity of the comparisons, only courses that are part of the compulsory core curricula for base and intermediate studies were chosen to be visualized.

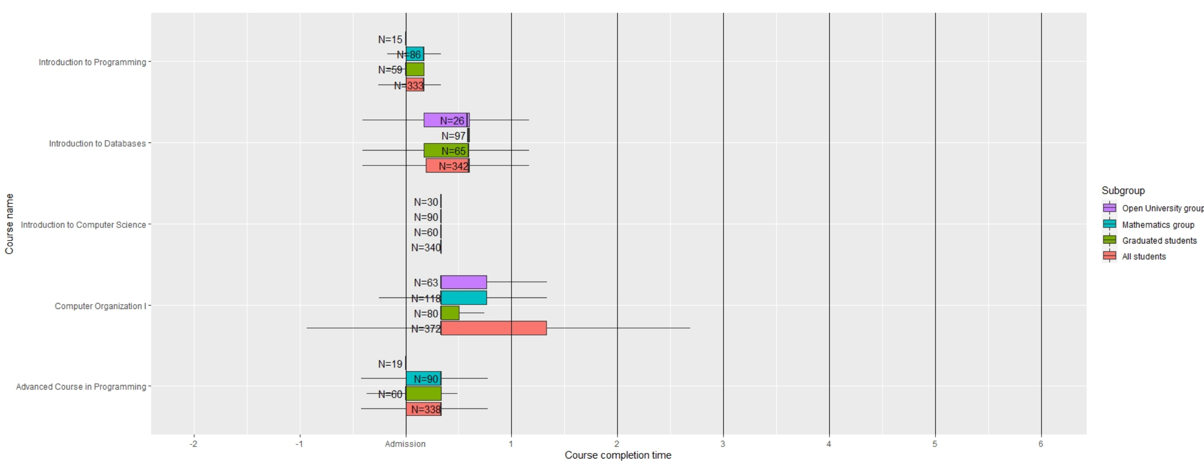
The data was first collected into database views containing a combination of course records and the students connected in various student groups and subgroups. The content of the views were processed with R to add new columns with time difference calculations between admission dates and course completion dates as integers. It should be noted that small sample sizes and either very small or very large dispersions can result in misleading plots. This includes singular data points that manifest as simple lines, large interquartile ranges that consist of only a few values with a large distance between them and skewed medians with non-normally distributed sets of values. These kinds of anomalous groupings are present in the data.

Another challenge was that data for the more recent admission years is still lacking for some later courses, leading to smaller sample groups in them and most likely skewing the visualizations. Students admitted in 2020 were completely omitted from the visualizations, as their course completions are heavily skewed towards early courses. Due to the curriculum changes in 2017 both course codes and course names were changed, but the actual content of the courses in most cases did not change. Additional groupings were made to combine completions with both the old and new course codes. Some courses, like *Software Development Methods*, were moved from base studies to intermediate studies and completely redesigned in manner that renders the new iteration, *Software Development Methods* a completely different course. In these cases the completions were not combined. For the new curriculum and transitioners groups the courses are represented with the new course names, while for the old curriculum group old names are used. In the following sections only visualizations that provided value in either comparison with other groups or unique insights are presented, even though all combinations of subgroups and admission years were originally drawn.

4.2.1 Group A: new curriculum, 2017-

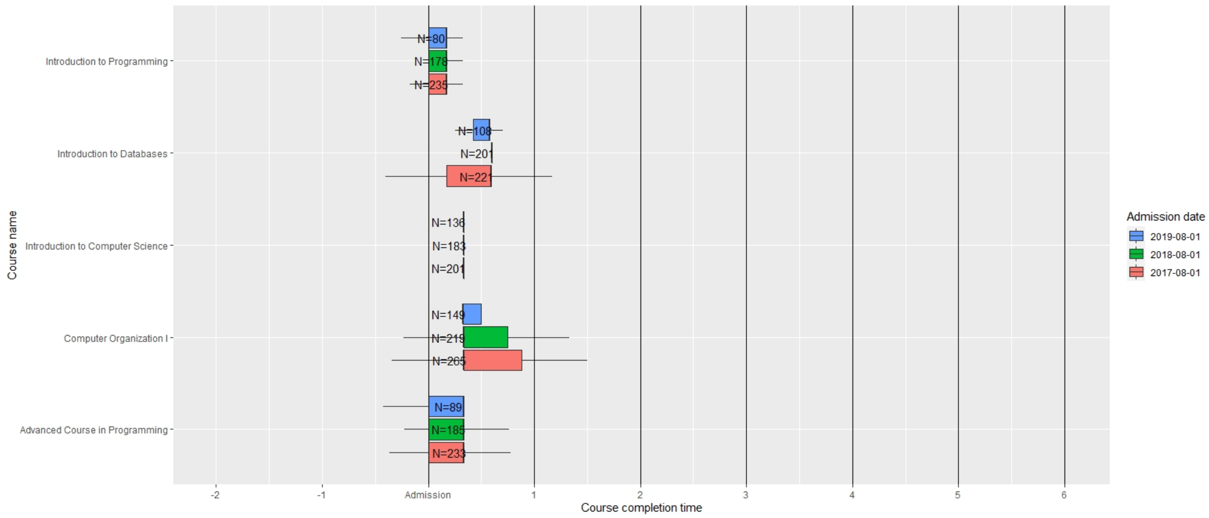
Starting with the new curriculum group and looking at non-Open University base courses in 4.3, we can see that base studies are completed in an uniform fashion. In the case of *Introduction to Computer Science*, 93 % of students complete the course in its intended time. The most surprising find is how the students in the mathematics group have a much larger distribution in *Computer Organization I*, as this observation is clearly out of line from the other groups. The number of students in the Open University -group that completed the non-Open University version of the same course is also much higher than in the other base studies, explained by the Open University version of *Computer Organization I* being a more recent addition to the course selection. The similarity of distributions in base courses is expected and from the point of view of study paths, it is natural that students tend to complete their first courses in similar time frames.

Figure 4.3: Group A: the new curriculum, completion of base courses



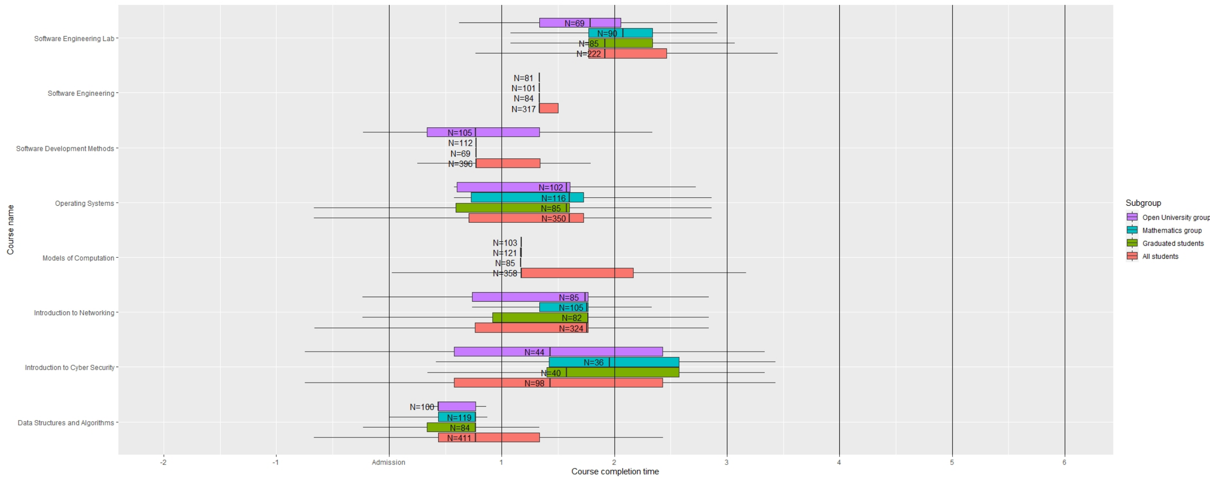
Looking at the same non-Open University base studies grouped by admission year in figure 4.4, we see that students tend to complete base studies in the same time through different admission years. The completions between admission years are even more uniform than between student subgroups. Changes in course orders like the elevation of *Computer Organization I* to be part of the five course base courses module in 2018 can be seen clearly, as students admitted in 2017 have a much larger range of completion compared to the later admission years. The same effect can be seen in *Introduction to Databases*.

Figure 4.4: Group A: the new curriculum, completion of base courses by admission year



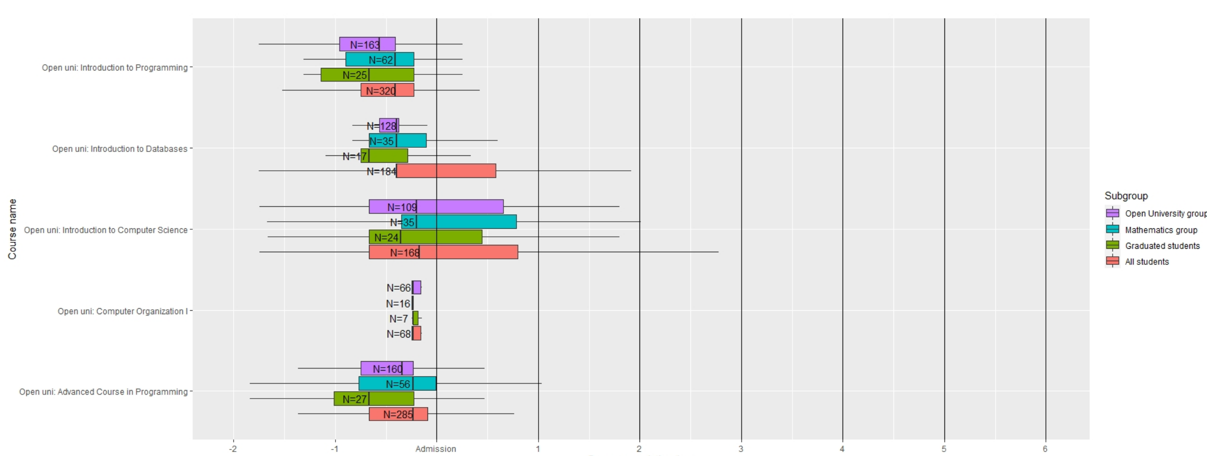
Moving on to intermediate studies in figure 4.5, it is surprising to see that all subgroups have quite similar distributions in all courses. Some courses show the all students -group having large distributions spanning multiple years while other groups have notably smaller distributions. This is an interesting observation, as when combined with graduation rates and study success figures presented earlier it would indicate that completing these courses in the suggested slot is a factor in study success and graduation. Otherwise, many courses show total ranges that span at least two academic years and interquartile ranges that span around one academic year, verifying the earlier observations about options in course choices and completion order leading to complex and differing study paths. In the case of the Open University group the little bit earlier completion can be explained by completion of base courses before admission as shown later in 4.6, leading to students moving to intermediate courses during the first year of studies. Other than that, it is interesting to see that students in the Open University group complete intermediate level courses in many cases in the same schedule as other students. This would indicate that completing base courses and even some intermediate courses before admission does not affect later course timing as much as could be expected.

Figure 4.5: Group A: the new curriculum, completion of intermediate courses



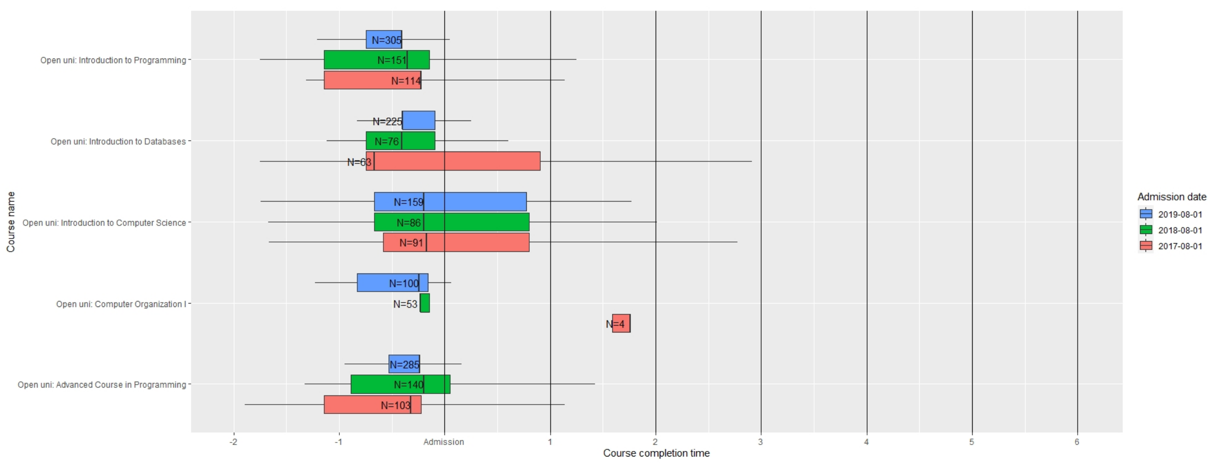
Open University base studies represented in figure 4.6 show that completing the studies before admission is common. Naturally students in the Open University -group have a comparatively large representation and tend to complete some of the studies slightly earlier than the other student groups. This is without doubt due to a large number of students admitted through Open University related paths present in the group. Interestingly, the completion of *Introduction to Computer Science* is quite common after admission, which is an unexpected result. This phenomenon cannot be explained through the data, but can be speculated to be related to a more flexible version being offered through the Open University that the students prefer over the standard iteration. Another noteworthy insight is that graduated students tended to complete these courses earlier than the other group, which might be attributed to both a small sample size and factors related to student motivation.

Figure 4.6: Group A: the new curriculum, completion of Open University base courses



More interesting observations arise from the distribution of course completions in Open University studies when grouped by admission year, shown in figure 4.7. Studies are completed mainly before admission and the completion points stay uniform between admission years. A very slight move towards earlier completion can be seen with smaller completion distributions. The key insight gained from the visualization is that the number of students completing Open University studies has been almost doubling every year. This is consistent with earlier observations about Open University related admission, it is clear that the availability of Open University courses and their incorporation into the admission system leads to more students having prior studies in Computer Science before admission.

Figure 4.7: Group A: the new curriculum, completion of Open University base courses by admission year



4.2.2 Group B: old curriculum, 2010 - 2017

The non-Open University studies in the old curriculum group grouped by subgroups presented in figure 4.8 show some interesting phenomena. From the figure it is apparent that students from the Open University -group hardly completed any non-Open University base studies. While the representations are small in the other major groups too as shown in figures 4.3 and 4.11, the complete lack of completions is surprising. This implies that students in the Open University -group in the old curriculum group have completed base studies as Open University versions, leading to the conclusion that these students might be almost exclusively from the Open University admission path. The groups show notably large ranges of completions, outlining the fact that students do not complete their studies in the suggested time and do not start studying immediately after admission.

Figure 4.8: Group B: the old curriculum, completion of base courses

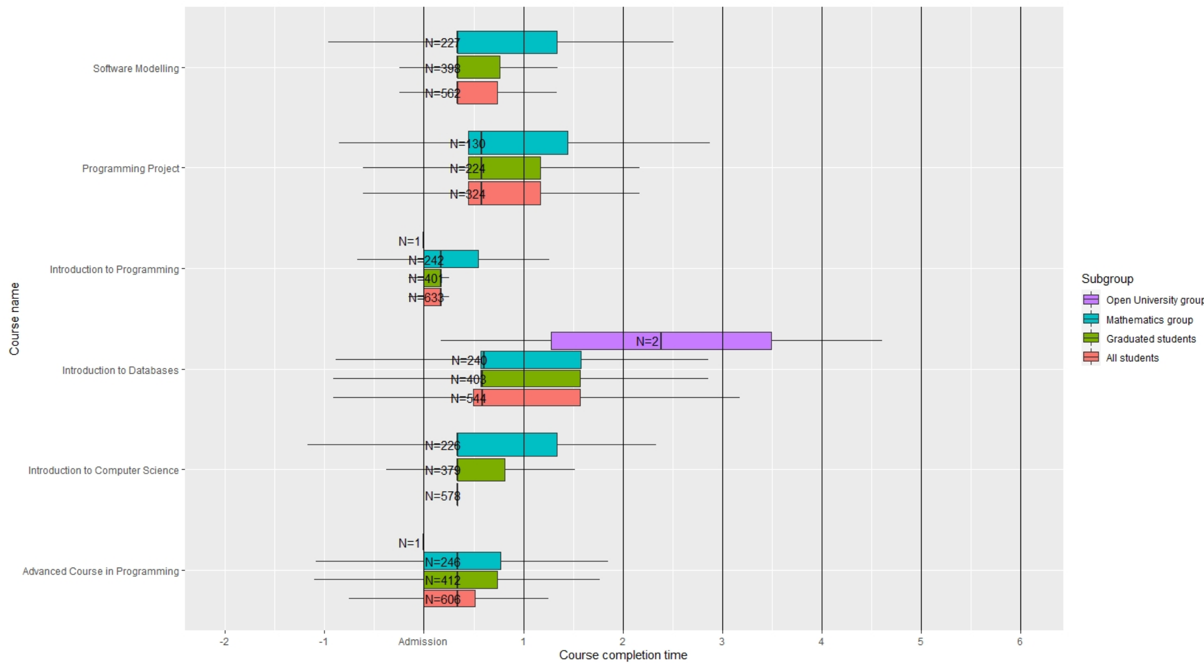
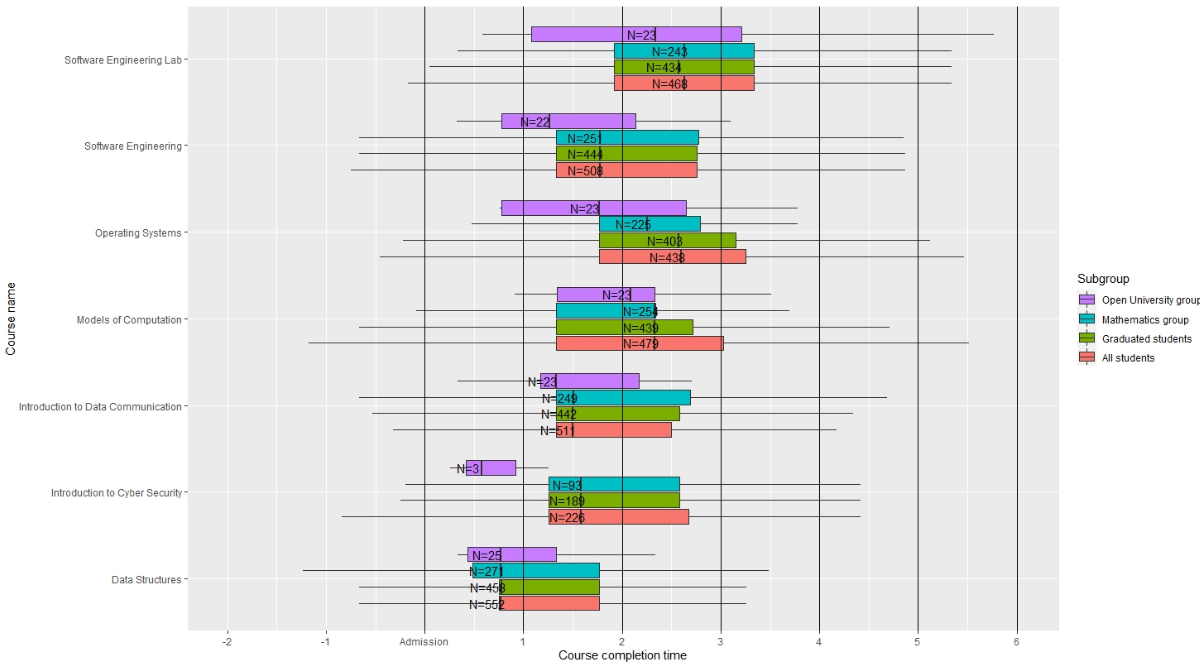


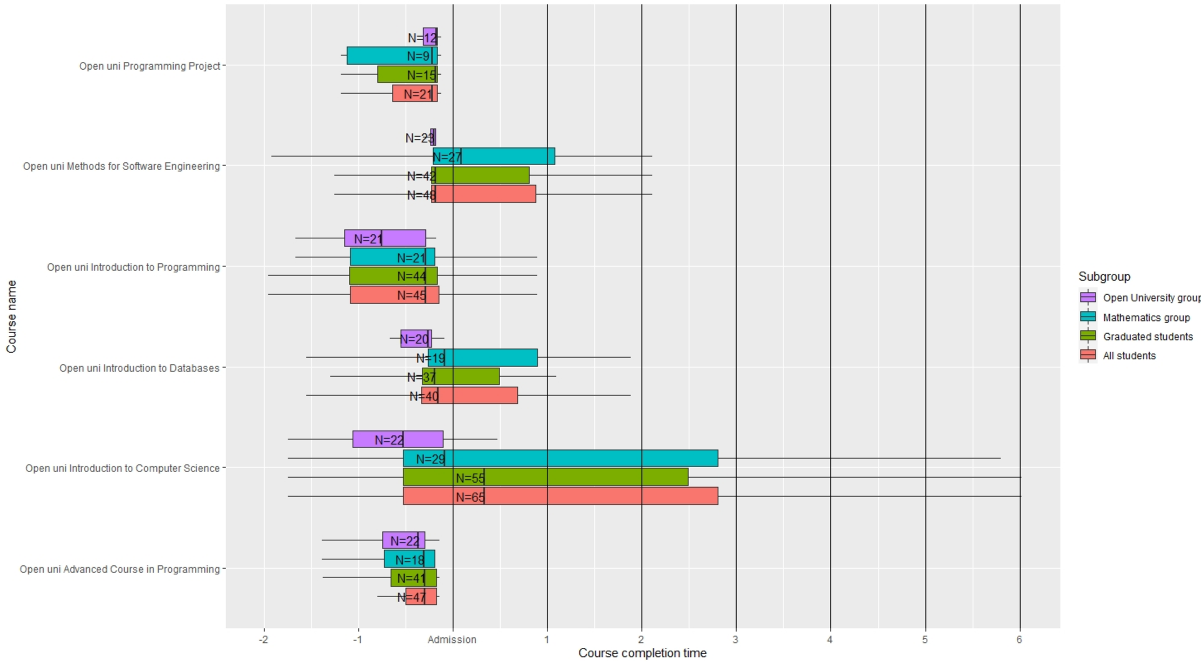
Figure 4.9 shows intermediate course completions grouped by subgroups of students. Students in the Open University -group completed the courses slightly earlier. The effect is more pronounced than in the new curriculum group shown in 4.5 and is closer to what could be expected considering that these students have completed their base courses in most cases before admission. It should be noted that for all groups the ranges cover many years - with interquartile ranges covering a full academic year. This shows that students have historically not followed the suggested study paths and most of them completed their studies in a time frame most suitable for themselves. Groupings by admission years yielded almost indentical results, though with some slight variations most likely caused by changes in when courses have been offered during the academic year. A slight movement towards earlier completion with newer admission years could be seen, but that is most likely caused by smaller sample sizes as many of the students admitted in 2015 and 2016 are part of of the transitioned students group.

Figure 4.9: Group B: the old curriculum, completion of intermediate courses



Moving on to Open University studies in figure 4.10 we observe similar results as in the new curriculum group. Open University base courses tend to be completed before admission, with the Open University admission path most likely playing a major role. Open University studies being free for degree students it is likely that students have used this in their advantage to complement missing courses or for scheduling reasons. It is also possible that some students prefer the educational style of the Open University versions of the courses more than the standard versions. Unsurprisingly students in the Open University group have completed the studies earlier than other groups.

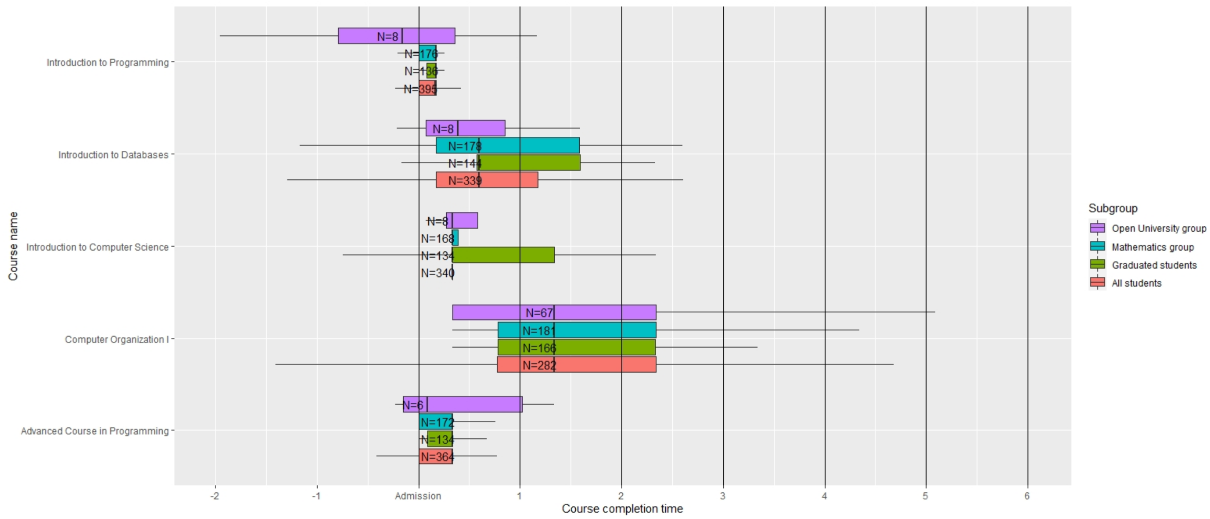
Figure 4.10: Group B: the old curriculum, completion of Open University base courses



4.2.3 Group C: transitioners

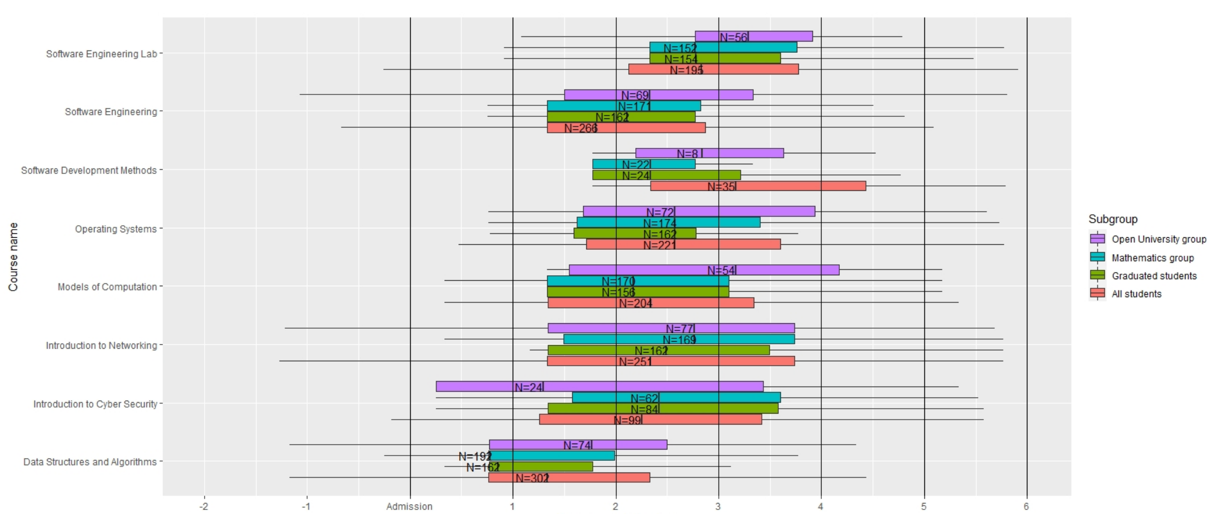
Starting again with non-Open University base courses as presented in figure 4.11 the students in the Open University group have completed the base courses slightly earlier than other groups. This result is in line with the other major groups A and B. Otherwise the completions between the subgroups are quite similar. The large amount completions in *Introduction to Databases* and *Computer Organization I* is an expected result due to them having been intermediate courses in the old curriculum that the transitioners might have completed before transition. The group includes students that were admitted earlier, were absent and upon returning to active studying have been transitioned to the new programme and have thus properly started studing after transition. Groupings by admission year follow a similar patterns of completion as the other major groups.

Figure 4.11: Group C: the transitioners group, completion of base courses



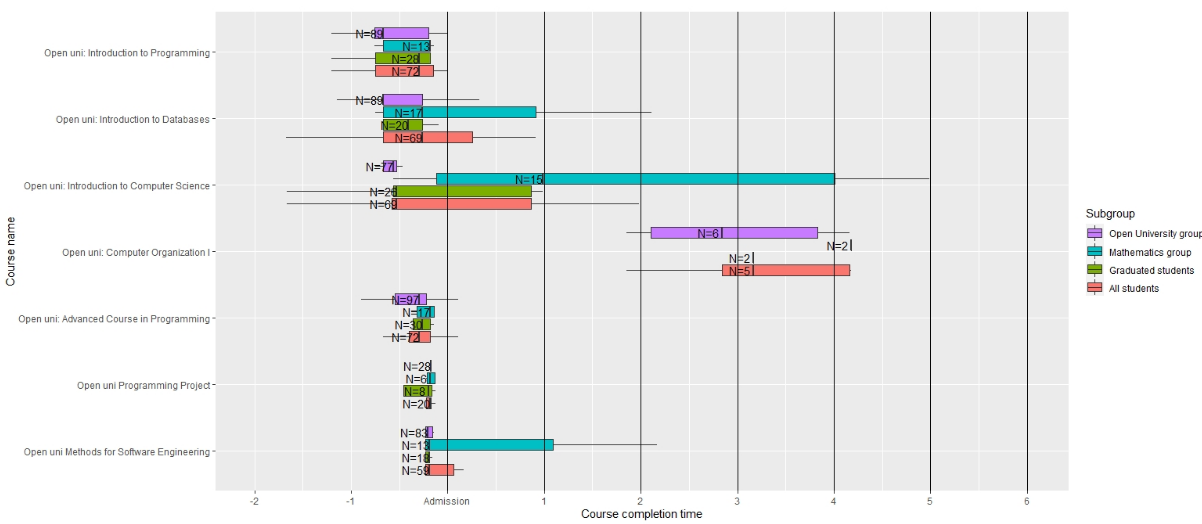
Intermediate courses presented in figure 4.12 provide some interesting insights. Like with the new curriculum group, it would seem that earlier completion of base studies does not translate into earlier completion of intermediate studies, specially for the Open University group. Indeed, the completions are roughly comparable to both the graduated students -group and the all students -group. The mathematics group shows slightly later completion than their peers in other groups in some cases. A noticeable portion of the studies have been completed during the third and even fourth years of study, meaning that if students completed the earlier intermediate courses during their third year of study, they most likely did not graduate in the three-year study time suggested by the University.

Figure 4.12: Group C: the transitioners group, completion of intermediate courses



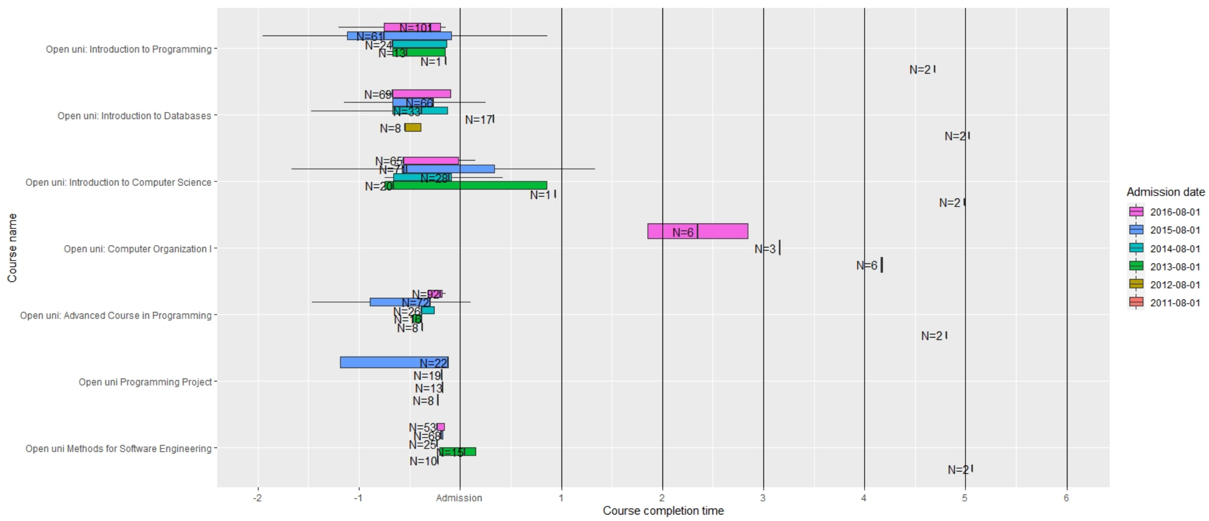
Moving on to Open University studies in figure 4.13, one very interesting thing is immediately apparent. While the completions generally conform to the model established in the other major groups, for some unknown reason *Introduction to Computer Science* sees a large distribution of completion times. Being a base studies course, one might expect completions to mirror other base studies courses. A reason for this cannot be drawn from the data used in this thesis. Another noteworthy phenomena is the low representation of the mathematics group in Open University studies compared to the all students group. This indicates that students in the mathematics subgroup of the transitioners group tended not to pick Open University versions of base studies, in direct contrast with results from the old and new curriculum groups.

Figure 4.13: Group C: the transitioners group, completion of Open University base courses



The grouping by Open University studies admission year shown in figure 4.14 does not offer any major new insights compared to the previous figures and groups. The figures follow a similar pattern with grouping by subgroups in figure 4.13, though the ranges are notably larger. This can partly be explained by smallish sample sizes for older admission years. This does not represent the whole truth however, as the distributions for admission year 2015 are much larger than admission year 2016 even with comparably sized sample groups. Again, the data itself offers no specific clues to why this has happened.

Figure 4.14: Group C: the transitioners group, completion of Open University base courses by admission year

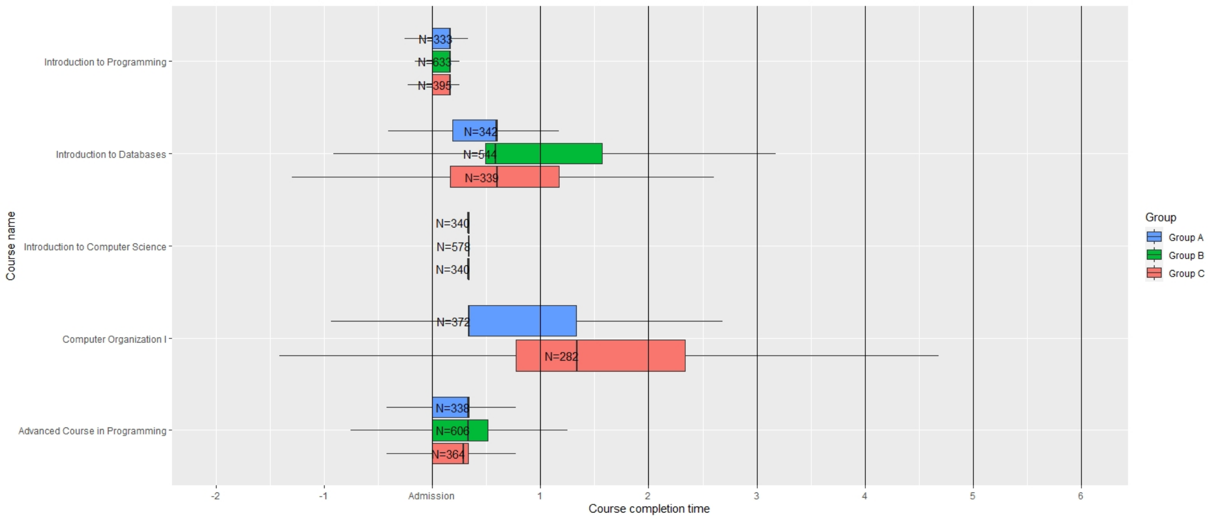


4.2.4 Group comparison

Another question is the differences in course completion between the major student groups. The following figures follow the model established in the previous sections with one the completions being grouped by student groups and courses having been combined by name to facilitate more understandable visualizations. The same limitations in intermediate courses as in the previous sections also apply with the studies chosen for comparison being those that are similar in both of the curricula. Course names from the new curriculum have been used for all groups in the following visualizations.

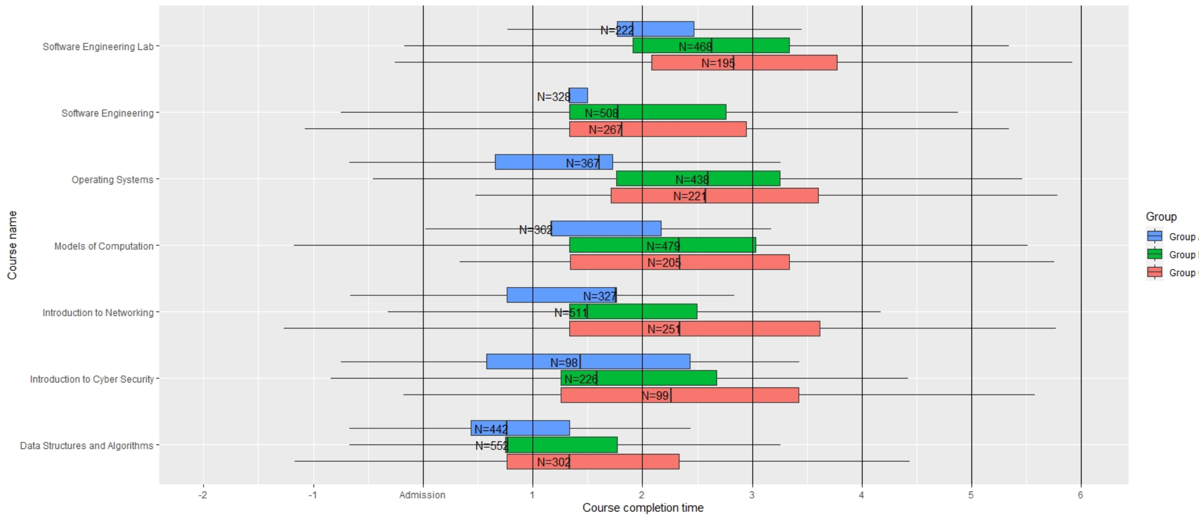
Looking at the non-Open University base courses shown in figure 4.15 it is clear that completion of base courses has followed a similar pattern regardless of the group or curriculum. The slight differences can be explained by differences in when the courses have been offered in the programme. Interestingly, the interquartile ranges indicate that students in the new curriculum group generally completed courses before or at the median point of the range. Only small tails reach completion points after the median where most of the students have completed the courses. Other groups show much larger upper quartiles which would indicate that students had to retake the course or examinations. It would be interesting to know the reason for this - are the new courses easier, have educational methods developed dramatically or are the students more gifted? A part of the larger interquartile ranges in the old curriculum and transitioners groups is explained by a larger sample sets and it is more than likely that when time goes on, the ranges for the new curriculum group become larger. Students in the transitioners group show tails leading up to third or fourth years of study, which is a long time since admission to complete base studies.

Figure 4.15: Comparison of student groups in non-Open University base courses



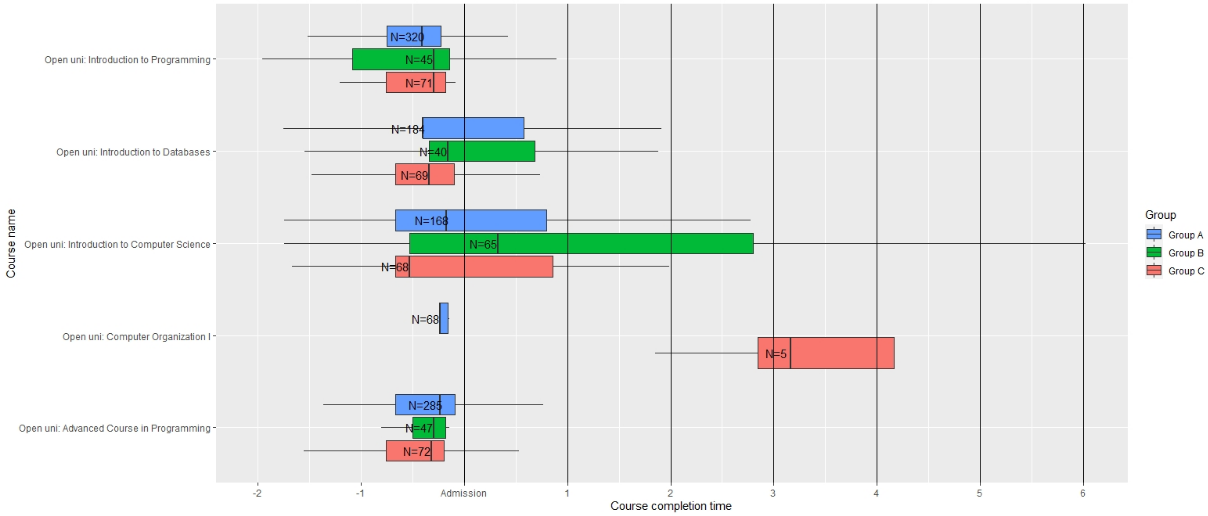
The corresponding intermediate courses shown in figure 4.16 show larger differences between the programmes. Students in the new curriculum group completed their intermediate courses earlier and with a similar effect of small upper quartiles and shorter tails. The suggested completion schedules of the courses are comparable between the programmes, meaning that for some undiscovered reason the new programme leads to earlier completion with less need for retaking examinations. The sample sets for the groups are large enough to warrant such an observation and while the tails might stretch with future admission batches, it is unlikely that the medians move dramatically without major changes to suggested course completion times. Earlier visualizations for the new curriculum group established that there is no dramatic difference in completion of intermediate studies between students in the Open University -group and their peers, meaning that earlier completion of intermediate studies is an effect caused by the curriculum changes. The results for the transitioners group verify the results in section 4.2.3 showing that the students complete their intermediate courses later and with a larger variance in completion time than their peers from the old curriculum group. Indeed, the differences in completion between the old curriculum and transitioners groups are very noticeable, with transitioning between programmes clearly indicating a later completion of intermediate studies and thus by an extent later graduation.

Figure 4.16: Comparison of student groups in non-Open University intermediate courses



The results for Open University base courses in figure 4.17 do not show any new results outside those already established in the previous sections. Most of the students complete their Open University studies before admission with some courses having ranges that stretch beyond admission. As established before, the role of Open University studies has become more important in the new curriculum and thus the new curriculum group. The numbers of completions for these courses are significantly higher and the completion points slightly earlier than for the other groups. This highlights a large shift in student study paths leading to more students completing studies prior to admission. Other than that, the visualizations show that the completion of Open University base studies before admission has remained in a similar trend, though choosing Open University studies was much rarer before the curriculum overhaul in 2017.

Figure 4.17: Comparison of student groups in Open University base courses



4.3 Regarding general distributions, study success and study paths

The results in section 4.2 give the general outlines in how certain groups of students have both performed and the distribution of students between certain course completion-based subgroups. These results are helpful in detecting some greater trends, like a noticeable increase students with large amounts of Open University studies shown in table 4.5 compared to earlier student groups shown in tables 4.7 and 4.9. Similarly, a decline in students with large amounts of mathematics studies is observable, for which the data offers no explanations. The Open University results can be directly traced to both changes in curricula and in admissions, leading us to a conclusion that a more flexible curriculum that incorporates or accepts Open University courses lead to students choosing those courses in favor of traditional courses. Purely statistical analysis does not provide the reasons why this phenomenon has manifested in such a distinguishable way after the 2017 curriculum changes, but prior research offers some clues. Studies by Zheng et al. [48], Leinonen et al. [23] and Pirttinen et al. [32] show that performance in MOOC and DEFA courses is better compared to traditional students and that factors such as student motivation are higher. Another possible affecting factor is only slightly covered in the aforementioned studies, but could be of importance in understanding the phenomenon. This factor is course structure and content, which can be thought to be of better quality when compared to traditional courses. The Open University courses, especially continuously evolving MOOCs, tend to be newer and their content is constantly curated, updated and maintained. The content thus stays fresh and is structured in a way that allows independent study with a variety of resources, such as links to appropriate helping sources and active online chat channels that provide both official and peer guidance. It is also more than likely that experimental educational methods tried-and-tested in MOOC's are later incorporated to the standard versions of the course. The students can also provide feedback easily and feedback is readily asked by the interactive online environment. When compared to lecture-based studies it is possible that these educational methods provide a more interactive and engaging experience, as noted by for example the student testimonials covered in Zheng et al [48]. Overall, the GPAs in computer science studies are above average of 3 on a scale from 1 to 5, meaning that students who complete their courses generally do so with decent grades. In fact, averages that are closer to or over 4 can be considered excellent. This would indicate that student motivation is high enough and the education is of good enough quality that such high averages are possible.

The graduation rates shown in figure 4.1 and table 4.2 show that the graduation rates are not very high and students don't seem to complete their studies in the three-to-four-year target time given by the university [8]. Tying the graduation rates and visualized study paths to the background presented in sections 2.2 and 2.3, it is clear that there are unknown variables that affect the study paths of students in such a way that graduation is not achieved - or atleast studies take much longer than suggested. The big question is of course what these variables could be. No sources seem to indicate major issues in either educational content, course structure, general curricula or university policies. Factors such as military service, parental leave, working while studying and other absences covered in sections 2.5 and 3.3.3 most likely play a significant role, but no studies in how these factors affect CS students in particular have been done. It would be unsurprising if receiving a job offer and either dropping out or postponing graduation was not a major factor, as there has been a major shortage of IT professionals in the business world and there might not be a career incentive to graduating as a formal education is not often required for junior IT roles. Other studies, including for example Barker et al. [2, 3] and Giannakos et al. [14], have explored CS student retention, with the results being generally applicable to the groups of students in this thesis though student retention is outside the scope of this thesis. Nevertheless, student retention is a major factor that affects the general perception of study success and study paths.

5 Discussion

In this chapter, I discuss the results from chapter 4 and answer the research questions. The answers are followed by my personal opinions on what the results imply and a look into further research topics that the results would warrant.

5.1 Insight from the results

Understanding the implications of the theoretical background presented in sections 2.2 and 2.3 in relation to the results from the previous chapter is not as straightforward as one might expect. There are multitudes of possible factors that can affect either a single students study path or even whole admission years. The global COVID19 pandemic happening during the time of the writing of this thesis is one such event that will no doubt affect the study paths and study success of students currently studying and will hopefully be a topic of research in the future. Even on a smaller scale, how courses have been available and possible experimental educational methods applied to them might have caused effects that have changed the study paths of some students in relation to their peers. Such effects are unknown variables, as the data alone does not indicate the presence of anomalous phenomena. As of such, singular deviant results shown earlier in the results can only be speculated upon.

Regarding study success, as noted before more recent admission years show better study success than prior admission years. Wether this is because of evolving educational methods, the growing variety of completion methods leading to easier ways of achieving good grades or other reasons is unknown. From an educational methods and study success factors perspective, the general structure of the courses and the supporting facilities such as feedback and discussion options can be seen as major factors to student motivation and by extension success and retention as presented earlier in section 2.2. Exercises that are intellectually challenging, offer a variety of open-ended and closed-ended questions and contain clear goals, such as the programming tasks with multiple levels that are independently graded are of exactly of the style that has been shown to increase student motivation [28, 35]. The five key ingredients for student motivation as presented by Williams & Williams [45] are the student, the teacher, the content, the method and the environment.

The flexible educational style, access to discussions and feedback, engaging framework and a sense of achievement with instant feedback among other similar benefits present in this style of courses are elements of the five key ingredients for student motivation. Combined with unlimited tries within the weekly deadlines, it might be deductible that the course structure and educational style in modernized course iterations seems to be more fit to modern computer science education in programming courses than the traditional lecture-based style, at least for a substantial number of students. Adding the findings by Leinonen et al. [23] and Pirttinen et al. [32] that show better GPA and the results in table 4.6 showing high GPA even for a larger sample of students, the phenomenon looks more understandable.

Study paths however leave much more up for speculation. As we cannot know the subjective reasons for student decisions, we cannot know why for instance the sample study paths provided by the university are not followed at all. Possible explanations ranging from absenteeism to changes in educational methods have been presented in the earlier chapters, but they offer only partial answers. The visualizations in section 4.2 show that students take a long time graduate and that course completions are very spread out. This indicates that the study paths chosen by the students are much more based on subjective reasons than what the university considers to be a good study path. It would be interesting to know how much a students academic motivation and the general academic environment of the programme affect the study paths taken and wether the new programme with more flexible study path choices leads to more students completing their studies on time. Indeed, without insight into student reasoning we are left speculating why the visualizations show this phenomenon, but do not know the actual reasons behind it.

5.2 Answering the research questions

This section provides detailed answers to the research questions presented in section 2.4. Research questions are answered through the results provided by the methods.

5.2.1 RQ1. Do students follow the suggested study paths?

The Bachelor's programme used to provide detailed suggestions to students about what a desirable study path could look like. For example, the ones shown in the official guidebooks for students admitted between 2008 and 2011 shown in figures A.3 and A.6 included all of the compulsory courses required for graduation and suggested placement with an accuracy of a specific study period. The later guidebooks do not show the sample course structures, but as noted earlier in section 2.6 similar suggestions were still present even if not printed in the guidebooks. A study path suggestion tailored for the new curriculum [33] is provided online, though its course placement is quite rigid when compared to the suggested completion points defined in course descriptions. I thus consider the course descriptions to provide a more realistic view of a suggested study path than the online sample schedule. Students that followed the sample study path suggestions are almost non-existent. One factor might be the prevalence of Open University studies, because it is impossible for students with any prior studies to follow the suggested path - having even the two base programming courses done complicates course selection starting from the first study period.

The same phenomenon was present when modeling the suggested study paths for the old curriculum, only a handful of students followed the suggested paths. For example, a total of 305 students were admitted in 2010 and 2011 and were instructed to consider the suggested study path shown in A.6. Only one student out of the 305 followed the suggested study path for CS courses the first year of their studies. This is a shocking find, as one would expect that more students would have followed the path deemed appropriate by the faculty. After some manual adjustments to simulate possible sample schedules after changed course orders for admission years between 2012 and 2016, the results did not change for the better. Indeed, in the old curriculum students that followed the complete suggested path for the first year of their studies are so rare that they can be considered to be outliers. The conclusion is that students did not and do not follow the suggested study paths almost at all. Thus, the answer to *RQ1* is no, students do not follow the suggested study paths. This could have some implications to curriculum design as the suggested study paths affect how courses are offered. Because students do not follow the suggestions, does it matter as much when the courses are offered? Personally, I think that a certain figurative backbone of courses in a logical order from admission to graduation should be outlined, but the timing of courses offered during the academic year could be more flexible to better serve students.

5.2.2 RQ2. How do students in the student groups complete courses?

The answer to *RQ2* requires a more open-ended look on two levels. Firstly, looking at the comparison figures presented in section 4.2.4 some general level distinctions between the major student groups can be seen. As noted in the aforementioned section, students in the new programme seem to complete intermediate courses earlier, while base studies and open university studies are completed in quite similar fashion if slightly earlier as students in the old programme and the students that transitioned between the programmes. The interquartile ranges for students in the old programme are also much larger, spanning multiple years of study in most cases. This is most likely caused by a larger sample set that includes students with long study times. It is more than likely that the plots will be closer to the plots for the old programme as the sample set grows. Should the transitioned students be considered as new programme students, the visualizations would be very different. As is apparent from the visualizations, students in the transitioners group have completed studies during the third or fourth years of study, which would shift the distributions of the new programme group. Thus, the most important clue from the major student group visualizations is that active students tend to complete courses earlier, while a noticeable tail of students progressing slower than expected move the medians noticeably and stretch the distributions. This is important as it gives a more realistic picture of how computer science studies are completed in general. As noted in the answer for *RQ1*, students do not follow sample study paths at all. Should students follow the suggested study paths, the completions should follow a more uniform pattern and have much narrower distributions. This causes is discrepancy between university expectations and actual completions.

The distributions for the more fine-grained subgroups of students in sections 4.2.1, 4.2.2 and 4.2.3 provide additional insights. On a general level through all of subgroups in all major student groups, the overarching pattern is that students with large amounts of Open University studies complete studies slightly earlier than other groups, with the largest difference being in base studies that are most often completed before admission. Interestingly, intermediate studies are often completed in tandem with other groups, meaning that in many cases completing base studies before admission does not lead to starting intermediate studies during the first year of studies. The reason for this might be that guidance in courses is lacking or that courses simply are not available. The first autumn is generally reserved for base studies, which the students might have already completed, leading to a lack of available courses. The sample study paths do not account for prior studies, leading to situations where students are unavailable to choose intermediate courses available in the autumn periods as their prerequisites are only offered during the spring periods. This is a known problem to students with prior studies before admission that would require addressing as the prevalence of prior studies in admitted students is growing with every admission year. Moving back to the student subgroups, graduated students have similar distributions with slightly narrower distributions than students in the all students -group. This is not surprising considering that graduation at four, five or even six or more years is not exactly uncommon as shown earlier in figure 4.2. Students with large amounts of mathematics studies generally complete their computer science studies a little bit later than the other groups of students, but this is explainable with a need to balance courses that might be offered on similar schedules. It would not be unrealistic to assume that the flexibly available courses computer science courses and completion methods affect how students in the new programme choose to complete courses.

As a conclusion it could be said that course completions are much more spread out than expected and differ greatly from the sample study paths planned by the faculty. There are also noticeable differences in how different student groups complete courses. Students with a large amount of Open University studies is a group that would require additional attention from an education planning point of view. That being said, the visualization through normalization of data to boxplots provides a generalized view of phenomena and only a partial truth. Without accompanying qualitative study that would explain reasoning for long graduation times, course choices and prioritization or student motivation, it is impossible to draw complete conclusions.

5.2.3 RQ3. How does study success in the study paths differ between the student groups?

The earlier results concerning GPAs for student groups and subgroups were presented in chapter 4.1. Notably, the GPAs are higher for students in the new curriculum group. This is an interesting find that can partly be explained by differences in sample sets, as students that transitioned between the curricula have completed same courses but have lower GPAs. One explanation is that the data is likely biased towards including a larger number of early courses that can be considered to be easier, while the other major student groups include also more advanced and harder courses. This however is not the whole truth. Personal communication with Dr. Matti Luukkainen, one of the supervisors of this thesis, led to one interesting tidbit of information. In his words course completion methods have changed drastically during the last years, with more courses being able to be completed by exercises and project work only without taking an examination, or alternatively the weight of the exam in grading is lower. Older course iterations tended to be more focused on examinations and the course grade could be mostly dependent on examination performance. This has led to a situation where a good grade is attainable with rigorous completion of coursework, yielding on average better grades than before for all students. Taking this into account, the GPAs for the student groups are actually not directly comparable without separating courses by completion method. The GPAs between students in the old programme and students that transitioned between programmes are more comparable. Looking at GPAs alone thus cannot tell us whether studying in the new programme is better for real study success compared to the old programme, even if the inflated number so indicate. This highlights the fact that comparing GPAs is dependent on context and might yield misleading results should they be compared without criticism. In my personal opinion it is quite unlikely that the student material is suddenly noticeably more talented, sporting GPAs almost half a grader higher with comparable grading practices. It is more likely that a slightly biased sample combined with evolving course completion methods leads to inflation in grades compared to prior students.

More meaningful answers can be found when looking at the general trends in the student subgroups. Even though the issue of incomparable GPAs between the groups persists, a look on the general trends yields answers to the research question. As expected, the GPA is lowest for the all students -group - most likely due to it containing all students including dropouts and low performers. Graduated students fared slightly better when compared to all students, but the difference is small enough to indicate that graduating does not translate to good grades. The groups with large amount of mathematics studies have consistently higher GPA than other students. While the precise reasons are unknown, it could be assumed that students investing in mathematics gain an advantage over their peers in computer science courses. Computer science theory is by nature mathematical, so it is not unreasonable to assume that mathematics and computer science complement each other well. It could also be that these students are inherently more apt and motivated for computer science studies and thus perform better than their peers. Indeed, studies show that high mathematical aptitude and prior studies translate to general study success in computer science and studies in general [6, 17, 46]. Overall, this result would indicate that encouraging students to choose mathematics courses to complement their degrees - and by a stretch encouraging students in Bachelor's programmes in mathematics and statistics to choose computer science as their academic minors. In this light, it is sad to see that the amount of students choosing to invest heavily in mathematics studies is on the decline. However, as mathematical aptitude most likely plays a role in both investing in mathematics studies and study success, encouraging mathematics studies might not translate to study success for all students. Mathematical anxiety and low aptitude affects study success, as shown by for example Nunez et al. [29] and Mieskonen [27]. Students with large amounts of Open University students also succeed better than their peers, consistent with the findings by Leinonen et al. [23] and Pirttinen et al. [32]. This validates earlier studies and highlights the fact that even without focused guidance and issues with course planning, study success for students with Open University backgrounds or studies is better compared to their peers not counting the mathematics group. The outlier here is the transitioners group, where the GPA for the Open University group is lower than in the two other major student groups. There is no explanation to this phenomenon that can be derived from the data and it is not consistent with prior study.

One could say that the most important study success metric is actually graduating, a good GPA does not amount to much should the student not graduate. It is clear judging by graduation rates presented in chapter 4.1 that students choosing to pursue study paths with a large amount of mathematics or Open University studies have better graduation rates. In the old programme, all of the students in the Mathematics and Open University groups graduated, compared to the average graduation rate of 53%. Even in the new programme, 35% of the students in the mathematics group have graduated. This most likely means that students choosing to invest in mathematics are more likely graduate in the suggested time, as the general graduation rate in the whole student group is only 9%. Additionally, the results would indicate that students choosing to pursue more diverse study paths are more likely to graduate. Combined with better GPA's, it is clear that study success is better on these distinct study paths compared to the general mass of students.

5.3 Implications and further research

Making definitive conclusions from high-level perspective such as the one take in this thesis is both hard and not very wise. The results do have very interesting and significant implications, but I believe that there is not enough focused research on many of the observed issues to allow for bold claims. Certain implications however are clear enough, students have not, do not and in many cases are not able to follow the sample schedules provided. This casts a shadow of doubt on their usefulness. Sample schedules guide curricula design, but on the same time provide a model that is not used, leading to a situation where they are initially useful for a very small minority of students. On a conceptual level, it could be beneficial to provide different suggestions for different student groups, e.g. providing one for students with a certain amount of prior Open University studies. Keeping in mind the results in chapter 4.1, many students do not graduate in the three suggested years of study, forming a need to take this into account. This could mean investing into methods that would favor graduating in the allotted time and making sure that students have access to courses, referring to the fact that students with prior studies might have to wait until courses are available and might thus have difficulties in finding suitable courses in the meantime. One solution to this issue could be running some courses in an on-demand fashion, allowing students to enroll and complete on their own volition or starting a course in any period should the minimum required attendees enroll. This might be hard to implement, but some optional courses do already allow for this kind of flexible enrollment and completion. Sample study paths designed for two-year completion might also be useful for students with large amount of Open University studies. It might also be beneficial to provide sample schedules for study plans taking longer than the suggested time. It could be motivating for students who know that they cannot graduate in three years due to personal reasons to know that it is not the end of the world and there are alternative study paths that lead to graduation in 3.5, 4 or even 5 years. Offering customized study paths that aim for specific Master's programmes, such as Data Science and Bioinformatics could also be offered, further motivating students that have goals in specific fields of study. In the end, graduation is the goal for both the faculty and the student and looking at the poor graduation rates from earlier, any measure that leads to increase would be welcome. Officially endorsing diverse paths to graduation could present the programme in a good light and show that old educational structures can be redesigned. The three-year completion is however a built-in design in the educational system as a whole and is as of such not the fault of the Faculty of Computer Science in the University of Helsinki.

Further minor and major research topics that stem from this thesis are plentiful. Deeper understanding regarding the study paths and their effect on the educational methods and applied educational theory would be one interesting high-level topic that would deepen the understanding on how students study computer science and how to design education to better suit student wants and needs. More specific research into how for example industry focused optional courses affect graduation rates and study success would also be interesting. Such research could shed light into the role of immediately applicable work skills in contrast to study motivation and retention. Regarding study success, further research into the factors affecting success specifically in the University of Helsinki could yield interesting results. Taking a more fine-grained approach with how individual teaching methods, course completion policies and even individual courses might highlight success factors most relevant to different course models. A lack of research into computer science specific educational methods from an educational sciences viewpoint would also warrant further research. Such research could positively affect general curricula design and programme structure to further improve the current system.

Noted multiple times in the prior chapters, the lack of qualitative research leaves a noticeable gap in the understanding of study paths and study success factors in the University of Helsinki. I would strongly suggest investing research resources to this issue, as without focused qualitative results a complete picture of the issue raised in this thesis and to certain extent other studies is impossible to form. Without the knowledge of *why* some phenomena appear the insight presented by merely finding them is not enough. I believe that that longitudinal study spanning multiple admission years with attempts to understand historical data through focused interviews with former students would result in a treasure trove of information that when combined with statistical data would allow major research-based development of the Bachelor's Programme in Computer Science.

6 Conclusion

Key statistical distributions show that both the graduation rates for the programme related student groups and admission years are quite low. The numbers also show that time until graduation is in most cases noticeably longer than the suggested three years. The results for the different student groups show that there are study success differences between the student groups. Students in the new curriculum group had higher average GPAs through all the subgroups than their peers in the other groups. This can be explained with the sample set being skewed towards students earlier in their studies and changes in course completion methods, with more recent course iterations offering completion without examinations, which could result in better grades overall. Through all the groups certain trends were present. Students that graduated had slightly better GPAs than students in the all students -group. Students with large amount of Open University studies had better GPAs when compared to both all students and graduated students, consistent with results from prior research. These students also had a larger graduation rate than the general graduation rate for all students. Students with large amount of mathematics studies consistently have a noticeably better GPAs than students in other groups and the graduation rates for the students is significantly better than the rate for all students in their reference groups. The underlying causes can only be speculated upon, but it would seem that both mathematical affinity and a heavy investment in mathematics studies could result to better study success in computer science.

Study path visualizations show that the completion of courses is quite fragmented. The used visualization method shows that in most intermediate courses, it is possible to point the median point of completion, but the interquartile ranges usually cover at least a whole academic year either way of the median. The visualizations when combined with suggested study paths provided by the university showed that students do not follow the suggested study paths and tend to complete their studies flexibly in a way that best suit their personal needs. This is most likely related to the long study times before graduation and also prior Open University studies that both invalidate the sample schedules built on a premise of zero prior course completions and a three-year study time until graduation. The visualizations also show that students in the new programme complete courses slightly earlier than the other groups and complete Open University courses more after admission. The earliness of completion will most likely be changed as the sample group grows and late joiners start to complete their courses. On the other hand, the visualizations highlight the large role of Open University courses as a complement to the general course offering, leading to a larger amount of flexibility in personal study path planning which the students seem to utilize.

As a conclusion, it is clear that further research into the factors that cause these phenomena would be required. Statistical results augmented by visualizations provides a partial truth and the underlying reasons of why the phenomena happen remains unknown until longitudinal qualitative research is conducted. Drawing overarching conclusions by numbers alone is hard, but certain speculations are possible. Through the results presented in this thesis it is apparent that more focus on study path planning and student retention measures should be taken. As students are admitted with more diverse academic backgrounds, it would be beneficial to re-evaluate how course structures are designed and to provide alternative study paths for students. More flexible course offerings and on-demand enrollments could make study path planning easier for students and thus motivate them further to complete their studies. Officially endorsing diverse study paths over the three-year model with both faster and slower paths to graduation could result in better student retention and higher graduation rates. Evolving the Bachelor's Programme in Computer Science to be more inclusive for students from all backgrounds and taking a more student centric approach in providing the means for flexible personal study path design and a larger variety of course completion options could be very beneficial for all the parties involved. It is possible that boldly upgrading the programme to answer current issues could lead to advances in educational theory, educational methods and curriculum design that would result in interesting novel approaches to computer science education.

6.1 Limitations

Research in computer science education from foreign universities is abundant, but differences in the educational practices and methods make them of limited use. Tertiary education systems in Europe follow the Bologna process rendering them mutually comparable in terms of degree requirements [39], but the process does not dictate the contents of a programme. Tertiary education systems in the Americas and Asia again have their own distinct systems, casting doubts on direct comparisons between the degree programmes without taking into account the specific differences. The ACM/IEEE curricula detailed in appendix B functions as global guideline in computer science education design, but the similarity of educational content might not be enough should the degree requirements be very different. This leads to issues in comparison, as results always need to be contextualized to their specific educational systems. This does not mean that the results would be completely useless, but their usefulness is limited to generalized observations that could be relevant in the context of the Finnish Education System.

Additionally, the mentioned lack of large-scale qualitative studies into student success, paths and motivation specifically in the University of Helsinki leaves many phenomena up to speculation. A data-based view into study paths through statistical means and visualizations provides only a partial truth with many unanswered questions. A lack of more focused research on which to draw conclusions from is a major limitation for this thesis and additional research would be welcome.

It is challenging to create visualizations in a topic that requires novel visualization ideas combined with purpose made data normalization. The undestandability of visualizations is a subjective matter and perfect methods that would convey the points perfectly to all readers is impossible to find. [30, 31] Thus, some readers might find the chosen visualization style hard to understand and would gain better understanding through different forms of visualization.

Bibliography

[1] S. Aho, S.-M. Hynninen, H. Karhunen, and M. Vanttaja. "Opiskeluaikainen työssäkäynti ja sen vaikutukset". In: *Työ ja yrittäjyys 26/2012* (2012).

[2] L. Barker, C. L. Hovey, and L. D. Thompson. "Results of a large-scale, multi-institutional study of undergraduate retention in computing". In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. 2014, pp. 1–8. DOI: 10.1109/FIE.2014.7044267.

[3] L. J. Barker, C. McDowell, and K. Kalahar. "Exploring Factors That Influence Computer Science Introductory Course Students to Persist in the Major". In: *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*. SIGCSE '09. Chattanooga, TN, USA: Association for Computing Machinery, 2009, pp. 153–157. ISBN: 9781605581835. DOI: 10.1145/1508865.1508923. URL: https://doi-org.libproxy.helsinki.fi/10.1145/1508865.1508923.

[4] R. Boyle, J. Carter, and M. Clark. "What Makes Them Succeed? Entry, progression and graduation in Computer Science". In: *Journal of Further and Higher Education* 26 (Aug. 2000). DOI: 10.1080/03098770120108266.

[5] P. Byrne and G. Lyons. "The Effect of Student Attributes on Success in Programming". In: *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*. ITiCSE '01. Canterbury, United Kingdom: Association for Computing Machinery, 2001, pp. 49–52. ISBN: 1581133308. DOI: 10.1145/377435.377467. URL: https://doi-org.libproxy.helsinki.fi/10.1145/377435.377467.

[6] P. F. Campbell and G. P. McCabe. "Predicting the success of freshmen in a computer science major". In: *Communications of the ACM* 27.11 (1984), pp. 1108–1113.

[7] L. Cassel, A. Clements, G. Davies, M. Guzdial, R. McCauley, A. McGettrick, B. Sloan, L. Snyder, P. Tymann, and B. W. Weide. *Computer Science Curriculum 2008: An Interim Revision of CS 2001*. Tech. rep. New York, NY, USA, 2008. DOI: 10.1145/2593246.

[8] *Duration of the degree*. https://studies.helsinki.fi/instructions/article/duration-degree. Accessed: 2021-07-12.

[9] *Education ministry, universities weigh admission rate increases*. https://yle.fi/news/3-10875026. Accessed: 2021-12-13.

[10] *FMSEI, Flexible Study Modules in Software Engineering and ICT*. https://blogs.helsinki.fi/edu-kumpula/2019/06/17/avoimet-verkkokurssit-koodarikoulutus-tietoturva-tai-tekoaly/. Accessed: 2021-12-15.

[11] C. T. Force. *Computing Curricula 2020: Paradigms for Global Computing Education*. New York, NY, USA: Association for Computing Machinery, 2020. ISBN: 9781450390590. DOI: 10.1145/3467967.

[12] D. Forsyth. *Probability and statistics for computer science*. Springer, 2018.

[13] P. R. Fowler and H. R. Boylan. "Increasing Student Success and Retention: A Multidimensional Approach." In: *Journal of Developmental Education* 34.2 (2010), p. 2.

[14] M. N. Giannakos, I. O. Pappas, L. Jaccheri, and D. G. Sampson. "Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness". In: *Education and Information Technologies* 22.5 (2017), pp. 2365–2382.

[15] T. Hailikari, R. Sund, A. Haarala-Muhonen, and S. Lindblom-Ylänne. "Using individual study profiles of first-year students in two different disciplines to predict graduation time". In: *Studies in Higher Education* 45.12 (2020), pp. 2604–2618.

[16] T. Hailikari, T. Tuononen, and A. Parpala. "Students' experiences of the factors affecting their study progress: Differences in study profiles". In: *Journal of Further and higher Education* 42.1 (2018), pp. 1–12.

[17] A. Hakkarainen. "Matematiikan ja lukemisen vaikeuksien yhteys toisen asteen koulutuspolkuun ja jatko-opintoihin tai työelämään sijoittumiseen". PhD thesis. Itä-Suomen yliopisto, 2016.

[18] *Hereditaryness of education*. https://www.stat.fi/artikkelit/2009/art_2009-03-16_002.html?s=0. Accessed: 2021-10-03.

[19] J. Horton. "Identifying at-risk factors that affect college student success". In: *International Journal of Process Education* 7.1 (2015), pp. 83–101.

[20] A. f. C. M. (Joint Task Force on Computing Curricula and I. C. Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: Association for Computing Machinery, 2013. ISBN: 9781450323093. DOI: 10.1145/2534860.

[21] H. Kwik, B. Xie, and A. J. Ko. “Experiences of Computer Science Transfer Students”. In: *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ICER ’18. Espoo, Finland: Association for Computing Machinery, 2018, pp. 115–123. ISBN: 9781450356282. DOI: 10.1145/3230977.3231004. URL: https://doi-org.libproxy.helsinki.fi/10.1145/3230977.3231004.

[22] F. S. Laanan, S. S. Starobin, and L. E. Eggleston. “Adjustment of community college students at a four-year university: Role and relevance of transfer student capital for student retention”. In: *Journal of College Student Retention: Research, Theory & Practice* 12.2 (2010), pp. 175–209.

[23] J. Leinonen, P. Ihantola, A. Leinonen, H. Nygren, J. Kurhila, M. Luukkainen, and A. Hellas. “Admitting Students through an Open Online Course in Programming: A Multi-year Analysis of Study Success”. English. In: *ICER ’19 - Proceedings of the 2019 ACM Conference on International Computing Education Research*. The fifteenth annual ACM International Computing Education Research (ICER), ICER 2019 ; Conference date: 11-08-2019 Through 14-08-2019. United States: ACM, July 2019, pp. 279–287. DOI: 10.1145/3291279.3339417. URL: https://icer.acm.org/.

[24] S. Lindblom-Ylänne, A. Haarala-Muhonen, L. Postareff, and T. Hailikari. “Exploration of individual study paths of successful first-year students: an interview study”. In: *European Journal of Psychology of Education* 32.4 (2017), pp. 687–701.

[25] V. A. Lotkowski, S. B. Robbins, and R. J. Noeth. “The Role of Academic and Non-Academic Factors in Improving College Retention. ACT Policy Report.” In: *American College Testing ACT Inc* (2004).

[26] K. Määttä and S. Uusiautti. “How to enhance the smoothness of university students’ study paths”. In: *International Journal of Research Studies in Education* 1.1 (2012), pp. 47–60.

[27] E. Mieskonen et al. “Opintomenestyksen ennustaminen ja opiskelijavalinnan vaikutuksen arviointi Aalto-yliopiston kauppakorkeakoulussa”. In: (2017).

[28] M. Mitchell, J. Sheard, and S. Markham. “Student Motivation and Positive Impressions of Computing Subjects”. In: *Proceedings of the Australasian Conference on Computing Education*. ACSE ’00. Melbourne, Australia: Association for Computing Machinery, 2000, pp. 189–194. ISBN: 1581132719. DOI: 10.1145/359369.359398. URL: https://doi-org.libproxy.helsinki.fi/10.1145/359369.359398.

[29] M. I. Núñez-Peña, M. Suárez-Pellicioni, and R. Bono. “Effects of math anxiety on student success in higher education”. In: *International Journal of Educational Research* 58 (2013), pp. 36–43.

[30] A. Perer and B. Shneiderman. “Integrating statistics and visualization for exploratory power: From long-term case studies to design guidelines”. In: *IEEE Computer Graphics and Applications* 29.3 (2009), pp. 39–51.

[31] A. Perer and B. Shneiderman. “Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 2008, pp. 265–274.

[32] N. Pirttinen, J. Leinonen, and K. Lemström. “Digital Education For All: Better Students Through Open Doors?” English. In: *Proceedings of the 26th Annual Conference on Innovation and Technology in Computer Science Education*. Annual Conference on Innovation and Technology in Computer Science Education , ITiCSE ; Conference date: 28-06-2021 Through 01-07-2021. Apr. 2021. URL: https://iticse.acm.org/.

[33] *Sample schedule for 2021-2022*. https://www.helsinki.fi/fi/koulutusohjelmat/tietojenkasittelytieteen-kandiohjelma/opiskelu/tietojenkasittelytieteen-opintojen-rakenne-ja-aikataulu. Accessed: 2021-09-16.

[34] F. Santos Laanan. “Studying transfer students: Part II: Dimensions of transfer students’ adjustment”. In: *Community College Journal of Research and Practice* 31.1 (2007), pp. 37–59.

[35] S. Sharmin, D. Zingaro, and C. Brett. “Weekly Open-Ended Exercises and Student Motivation in CS1”. In: *Koli Calling ’20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research*. Koli Calling ’20. Koli, Finland: Association for Computing Machinery, 2020. ISBN: 9781450389211. DOI: 10.1145/3428029.3428035. URL: https://doi-org.libproxy.helsinki.fi/10.1145/3428029.3428035.

[36] N. Sigurdson and A. Petersen. “A Survey-Based Exploration of Computer Science Student Perspectives on Mathematics”. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE ’19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, pp. 1032–1038. ISBN: 9781450358903. DOI: 10.1145/3287324.3287416. URL: https://doi-org.libproxy.helsinki.fi/10.1145/3287324.3287416.

[37] Simon, A. Luxton-Reilly, V. V. Ajanovski, E. Fouh, C. Gonsalvez, J. Leinonen, J. Parkinson, M. Poole, and N. Thota. "Pass Rates in Introductory Programming and in Other STEM Disciplines". In: *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. ITiCSE-WGR '19. Aberdeen, Scotland Uk: Association for Computing Machinery, 2019, pp. 53–71. ISBN: 9781450375672. DOI: 10.1145/3344429.3372502. URL: https://doi-org.libproxy.helsinki.fi/10.1145/3344429.3372502.

[38] *Study right extensions*. https://studies.helsinki.fi/instructions/article/if-your-right-study-expires. Accessed: 2021-08-08.

[39] U. Teichler. "Changing structures of the higher education systems: The increasing complexity of underlying forces". In: *Higher Education Policy* 19.4 (2006), pp. 447–461.

[40] *The effect of parent's education on child's educational choices*. https://www.tilastokeskus.fi/tietotrendit/artikkelit/2016/vanhempien-koulutus-vaikuttaa-lasten-valintoihin/. Accessed: 2021-10-03.

[41] *The history of University of Helsinki computer science department*. https://www.cs.helsinki.fi/historia/. Accessed: 2021-07-10.

[42] J. J. Van Wijk. "The value of visualization". In: *VIS 05. IEEE Visualization, 2005.* IEEE. 2005, pp. 79–86.

[43] L. Vanhanen-Nuutinen, J. Saari, H. Kotila, and K. Mäki. "Opintojen aikainen työssäkäynti - ongelma vai mahdollisuus ammattikorkeakouluopinnoissa?" In: *Opetus- ja kulttuuriministeriön julkaisuja 2018:10* (2018).

[44] J. Whalley, A. Petersen, and P. Denny. "Mathematics, Computer Science and Career Inclinations — A Multi-Institutional Exploration". In: *Koli Calling '20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research*. Koli Calling '20. Koli, Finland: Association for Computing Machinery, 2020. ISBN: 9781450389211. DOI: 10.1145/3428029.3428046. URL: https://doi-org.libproxy.helsinki.fi/10.1145/3428029.3428046.

[45] K. C. Williams and C. C. Williams. "Five key ingredients for improving student motivation". In: *Research in Higher Education Journal* 12 (2011), p. 1.

[46] B. C. Wilson and S. Shrock. "Contributing to success in an introductory computer science course: a study of twelve factors". In: *Acm sigcse bulletin* 33.1 (2001), pp. 184–188.

[47] E. Yukselturk and S. Bulut. "Predictors for student success in an online course". In: *Journal of Educational Technology & Society* 10.2 (2007), pp. 71–83.

[48] S. Zheng, M. B. Rosson, P. C. Shih, and J. M. Carroll. "Understanding Student Motivation, Behaviors and Perceptions in MOOCs". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW '15. Vancouver, BC, Canada: Association for Computing Machinery, 2015, pp. 1882–1895. ISBN: 9781450329224. DOI: 10.1145/2675133.2675217. URL: https://doi-org.libproxy.helsinki.fi/10.1145/2675133.2675217.

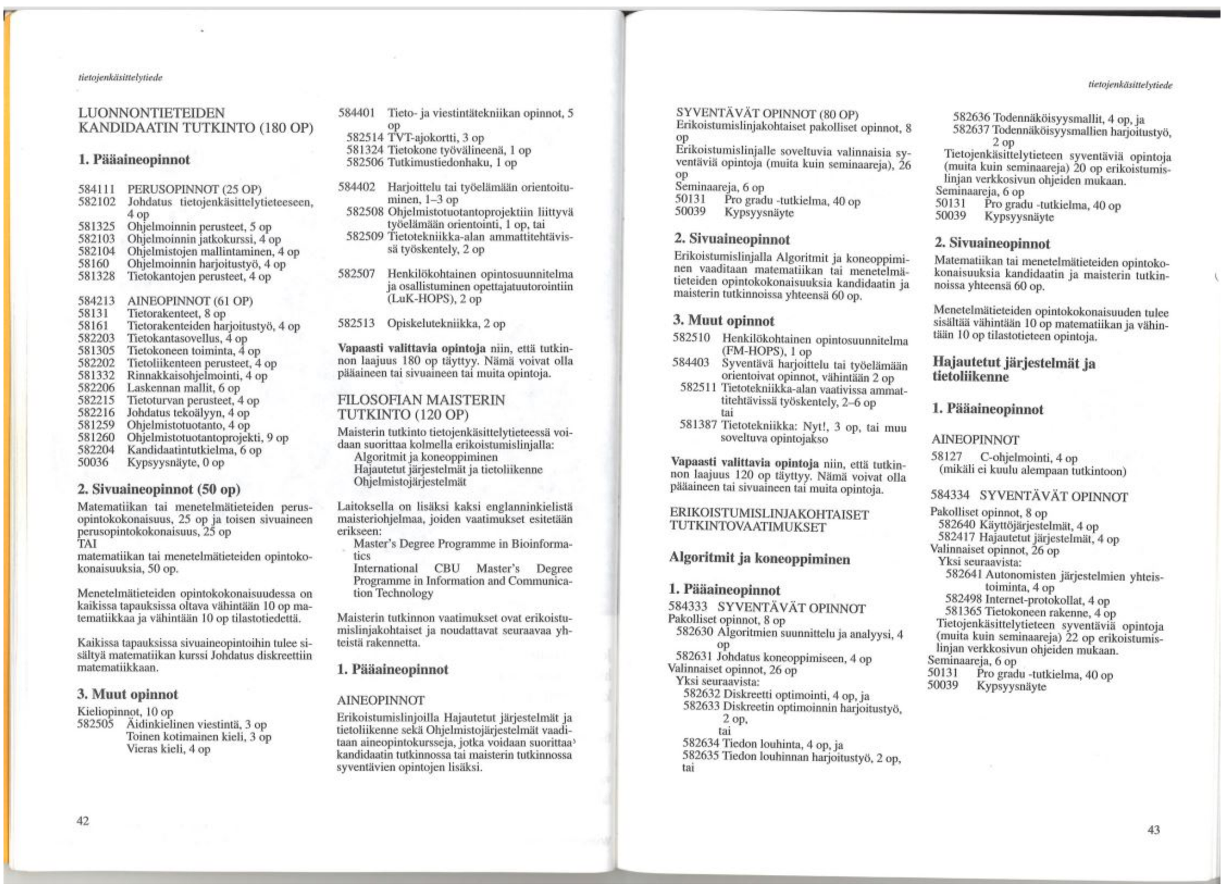
List of Tables

List of Figures

Appendix A University Of Helsinki CS Curriculum 2008 - 2017

The following figures are scans of curriculum guidebooks through the years between 2008 and 2016. The material is currently only available in Finnish and in singular surviving booklets archived by the faculty. The booklets were provided to be used and scanned in this thesis by *Reijo Sivén*, the education coordinator responsible for CS studies. Additional materials that may have been distributed to students are most likely lost in time.

Figure A.1: General structure Bachelor level CS studies, 2008-2010



tietojenkäsittelytiede

LUONNONTIETEIDEN KANDIDAATIN TUTKINTO (180 OP)

1. Pääaineopinnot

584111 PERUSOPINNOT (25 OP)
582102 Johdatus tietojenkäsittelytieteeseen, 4 op
581325 Ohjelmoinnin perusteet, 5 op
582103 Ohjelmoinnin jatkokurssi, 4 op
582104 Ohjelmistojen mallintaminen, 4 op
58160 Ohjelmoinnin harjoitustyö, 4 op
581328 Tietokantojen perusteet, 4 op

584213 AINEOPINNOT (61 OP)
58131 Tietorakenteet, 8 op
58161 Tietorakenteiden harjoitustyö, 4 op
582203 Tietokantasovellus, 4 op
581305 Tietokoneen toiminta, 4 op
582202 Tietoliikenteen perusteet, 4 op
581332 Rinnakkaisohjelmointi, 4 op
582206 Laskennan mallit, 6 op
582215 Tietoturvan perusteet, 4 op
582216 Johdatus tekoälyyn, 4 op
581259 Ohjelmistotuotanto, 4 op
581260 Ohjelmistotuotantoprojekti, 9 op
582204 Kandidaatintutkielma, 6 op
50036 Kypsyysnäyte, 0 op

2. Sivuaineopinnot (50 op)

Matematiikan tai menetelmätieteiden perusopintokokonaisuus, 25 op ja toisen sivuaineen perusopintokokonaisuus, 25 op
TAI
matematiikan tai menetelmätieteiden opintokokonaisuuksia, 50 op.

Menetelmätieteiden opintokokonaisuudessa on kaikissa tapauksissa oltava vähintään 10 op matematiikkaa ja vähintään 10 op tilastotiedettä.

Kaikissa tapauksissa sivuaineopintoihin tulee sisältyä matematiikan kurssi Johdatus diskreettiin matematiikkaan.

3. Muut opinnot

Kieliopinnot, 10 op
582505 Äidinkielinen viestintä, 3 op
Toinen kotimainen kieli, 3 op
Vieras kieli, 4 op

584401 Tieto- ja viestintätekniikan opinnot, 5 op
582514 TVT-ajokortti, 3 op
581324 Tietokone työvälineenä, 1 op
582506 Tutkimustiedonhaku, 1 op

584402 Harjoittelu tai työelämään orientoituminen, 1–3 op
582508 Ohjelmistotuotantoprojektiin liittyvä työelämään orientointi, 1 op, tai
582509 Tietotekniikka-alan ammattitehtävissä työskentely, 2 op

582507 Henkilökohtainen opintosuunnitelma ja osallistuminen opettajatuutorointiin (LuK-HOPS), 2 op

582513 Opiskelutekniikka, 2 op

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 180 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

FILOSOFIAN MAISTERIN TUTKINTO (120 OP)

Maisterin tutkinto tietojenkäsittelytieteessä voidaan suorittaa kolmella erikoistumislinjalla:
Algoritmit ja koneoppiminen
Hajautetut järjestelmät ja tietoliikenne
Ohjelmistojärjestelmät

Laitoksella on lisäksi kaksi englanninkielistä maisteriohjelmaa, joiden vaatimukset esitetään erikseen:
Master's Degree Programme in Bioinformatics
International CBU Master's Degree Programme in Information and Communication Technology

Maisterin tutkinnon vaatimukset ovat erikoistumislinjakohtaiset ja noudattavat seuraavaa yhteistä rakennetta.

1. Pääaineopinnot

AINEOPINNOT

Erikoistumislinjoilla Hajautetut järjestelmät ja tietoliikenne sekä Ohjelmistojärjestelmät vaaditaan aineopintokursseja, jotka voidaan suorittaa kandidaatin tutkinnossa tai maisterin tutkinnossa syventävien opintojen lisäksi.

42

SYVENTÄVÄT OPINNOT (80 OP)
Erikoistumislinjakohtaiset pakolliset opinnot, 8 op
Erikoistumislinjalle soveltuvia valinnaisia syventäviä opintoja (muita kuin seminaareja), 26 op
Seminaareja, 6 op
50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte

2. Sivuaineopinnot

Erikoistumislinjalla Algoritmit ja koneoppiminen vaaditaan matematiikan tai menetelmätieteiden opintokokonaisuuksia kandidaatin ja maisterin tutkinnoissa yhteensä 60 op.

3. Muut opinnot

582510 Henkilökohtainen opintosuunnitelma (FM-HOPS), 1 op
584403 Syventävä harjoittelu tai työelämään orientoivat opinnot, vähintään 2 op
582511 Tietotekniikka-alan vaativissa ammattitehtävissä työskentely, 2–6 op
tai
581387 Tietotekniikka: Nyt!, 3 op, tai muu soveltuva opintojakso

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 120 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

ERIKOISTUMISLINJAKOHTAISET TUTKINTOVAATIMUKSET

Algoritmit ja koneoppiminen

1. Pääaineopinnot

584333 SYVENTÄVÄT OPINNOT
Pakolliset opinnot, 8 op
582630 Algoritmien suunnittelu ja analyysi, 4 op
582631 Johdatus koneoppimiseen, 4 op
Valinnaiset opinnot, 26 op
Yksi seuraavista:
582632 Diskreetti optimointi, 4 op, ja
582633 Diskreetin optimoinnin harjoitustyö, 2 op,
tai
582634 Tiedon louhinta, 4 op, ja
582635 Tiedon louhinnan harjoitustyö, 2 op,
tai

tietojenkäsittelytiede

582636 Todennäköisyysmallit, 4 op, ja
582637 Todennäköisyysmallien harjoitustyö, 2 op
Tietojenkäsittelytieteen syventäviä opintoja (muita kuin seminaareja) 20 op erikoistumislinjan verkkosivun ohjeiden mukaan.
Seminaareja, 6 op
50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte

2. Sivuaineopinnot

Matematiikan tai menetelmätieteiden opintokokonaisuuksia kandidaatin ja maisterin tutkinnoissa yhteensä 60 op.

Menetelmätieteiden opintokokonaisuuden tulee sisältää vähintään 10 op matematiikan ja vähintään 10 op tilastotieteen opintoja.

Hajautetut järjestelmät ja tietoliikenne

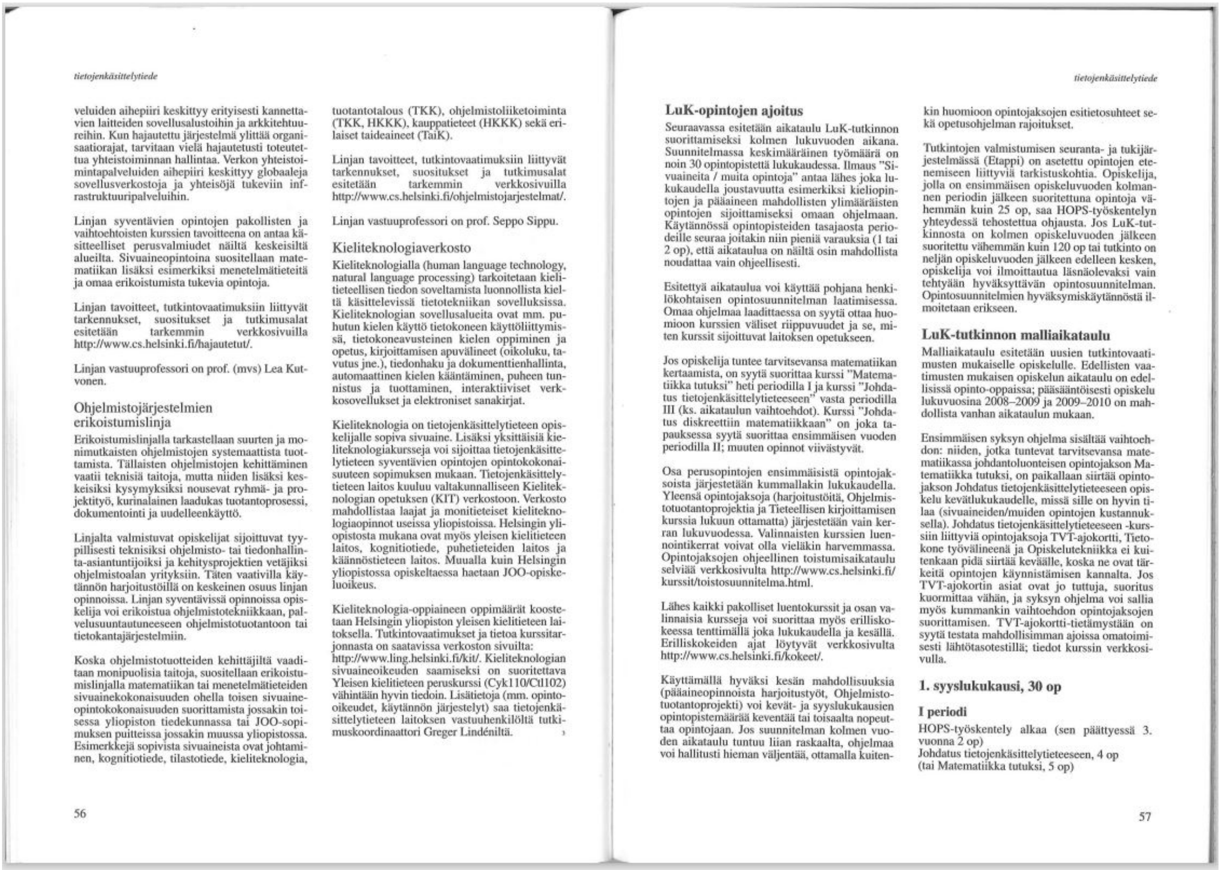
1. Pääaineopinnot

AINEOPINNOT
58127 C-ohjelmointi, 4 op
(mikäli ei kuulu alempaan tutkintoon)

584334 SYVENTÄVÄT OPINNOT
Pakolliset opinnot, 8 op
582640 Käyttöjärjestelmät, 4 op
582417 Hajautetut järjestelmät, 4 op
Valinnaiset opinnot, 26 op
Yksi seuraavista:
582641 Autonomisten järjestelmien yhteistoiminta, 4 op
582498 Internet-protokollat, 4 op
581365 Tietokoneen rakenne, 4 op
Tietojenkäsittelytieteen syventäviä opintoja (muita kuin seminaareja) 22 op erikoistumislinjan verkkosivun ohjeiden mukaan.
Seminaareja, 6 op
50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte

43

Figure A.2: Proposed timing of Bachelor level CS studies, 2008-2010



veluiden aihepiiri keskittyy erityisesti kannettavien laitteiden sovellusalustoihin ja arkkitehtuureihin. Kun hajautettu järjestelmä ylittää organisaatiorajat, tarvitaan vielä hajautetusti toteutettua yhteistoiminnan hallintaa. Verkon yhteistoimintapalveluiden aihepiiri keskittyy globaaleja sovellusverkostoja ja yhteisöjä tukeviin infrastruktuuripalveluihin.

Linjan syventävien opintojen pakollisten ja vaihtoehtoisten kurssien tavoitteena on antaa käsitteelliset perusvalmiudet näiltä keskeisiltä alueilta. Sivuaineopintoina suositellaan matematiikan lisäksi esimerkiksi menetelmätieteitä ja omaa erikoistumista tukevia opintoja.

Linjan tavoitteet, tutkintovaatimuksiin liittyvät tarkennukset, suositukset ja tutkimusalat esitetään tarkemmin verkkosivuilla http://www.cs.helsinki.fi/hajautetut/.

Linjan vastuuprofessori on prof. (mvs) Lea Kutvonen.

Ohjelmistojärjestelmien erikoistumislinja

Erikoistumislinjalla tarkastellaan suurten ja monimutkaisten ohjelmistojen systemaattista tuottamista. Tällaisten ohjelmistojen kehittäminen vaatii teknisiä taitoja, mutta niiden lisäksi keskeisiksi kysymyksiksi nousevat ryhmä- ja projektityö, kurinalainen laadukas tuotantoprosessi, dokumentointi ja uudelleenkäyttö.

Linjalta valmistuvat opiskelijat sijoittuvat tyypillisesti teknisiksi ohjelmisto- tai tiedonhallinta-asiantuntijoiksi ja kehitysprojektien vetäjiksi ohjelmistoalan yrityksiin. Täten vaativilla käytännön harjoitustöillä on keskeinen osuus linjan opinnoissa. Linjan syventävissä opinnoissa opiskelija voi erikoistua ohjelmistotekniikkaan, palvelusuuntautuneeseen ohjelmistotuotantoon tai tietokantajärjestelmiin.

Koska ohjelmistotuotteiden kehittäjiltä vaaditaan monipuolisia taitoja, suositellaan erikoistumislinjalla matematiikan tai menetelmätieteiden sivuainekokonaisuuden ohella toisen sivuaineopintokokonaisuuden suorittamista jossakin toisessa yliopiston tiedekunnassa tai JOO-sopimuksen puitteissa jossakin muussa yliopistossa. Esimerkkejä sopivista sivuaineista ovat johtaminen, kognitiotiede, tilastotiede, kieliteknologia, tuotantotalous (TKK), ohjelmistoliiketoiminta (TKK, HKKK), kauppatieteet (HKKK) sekä erilaiset taideaineet (TaiK).

Linjan tavoitteet, tutkintovaatimuksiin liittyvät tarkennukset, suositukset ja tutkimusalat esitetään tarkemmin verkkosivuilla http://www.cs.helsinki.fi/ohjelmistojarjestelmat/.

Linjan vastuuprofessori on prof. Seppo Sippu.

Kieliteknologiaverkosto

Kieliteknologialla (human language technology, natural language processing) tarkoitetaan kielitieteellisen tiedon soveltamista luonnollista kieltä käsittelevissä tietotekniikan sovelluksissa. Kieliteknologian sovellusalueita ovat mm. puhutun kielen käyttö tietokoneen käyttöliittymissä, tietokoneavusteinen kielen oppiminen ja opetus, kirjoittamisen apuvälineet (oikoluku, tavutus jne.), tiedonhaku ja dokumenttienhallinta, automaattinen kielen kääntäminen, puheen tunnistus ja tuottaminen, interaktiiviset verkkosovellukset ja elektroniset sanakirjat.

Kieliteknologia on tietojenkäsittelytieteen opiskelijalle sopiva sivuaine. Lisäksi yksittäisiä kieliteknologiakursseja voi sijoittaa tietojenkäsittelytieteen syventävien opintojen opintokokonaisuuteen sopimuksen mukaan. Tietojenkäsittelytieteen laitos kuuluu valtakunnalliseen Kieliteknologian opetuksen (KIT) verkostoon. Verkosto mahdollistaa laajat ja monitieteiset kieliteknologiaopinnot useissa yliopistoissa. Helsingin yliopistosta mukana ovat myös yleisen kielitieteen laitos, kognitiotiede, puhetieteiden laitos ja käännöstieteen laitos. Muualla kuin Helsingin yliopistossa opiskeltaessa haetaan JOO-opiskeluoikeus.

Kieliteknologia-oppiaineen oppimäärät koostetaan Helsingin yliopiston yleisen kielitieteen laitoksella. Tutkintovaatimukset ja tietoa kurssitarjonnasta on saatavissa verkoston sivuilta: http://www.ling.helsinki.fi/kit/. Kieliteknologian sivuaineoikeuden saamiseksi on suoritettava Yleisen kielitieteen peruskurssi (Cyk110/Ctl102) vähintään hyvin tiedoin. Lisätietoja (mm. opinto-oikeudet, käytännön järjestelyt) saa tietojenkäsittelytieteen laitoksen vastuuhenkilöltä tutkimuskoordinaattori Greger Lindéniltä.

LuK-opintojen ajoitus

Seuraavassa esitetään aikataulu LuK-tutkinnon suorittamiseksi kolmen lukuvuoden aikana. Suunnitelmassa keskimääräinen työmäärä on noin 30 opintopistettä lukukaudessa. Ilmaus "Sivuaineita / muita opintoja" antaa lähes joka lukukaudella joustavuutta esimerkiksi kieliopintojen ja pääaineen mahdollisten ylimääräisten opintojen sijoittamiseksi omaan ohjelmaan. Käytännössä opintopisteiden tasajaosta periodeille seuraa joitakin niin pieniä varauksia (1 tai 2 op), että aikataulua on näiltä osin mahdollista noudattaa vain ohjeellisesti.

Esitettyä aikataulua voi käyttää pohjana henkilökohtaisen opintosuunnitelman laatimisessa. Omaa ohjelmaa laadittaessa on syytä ottaa huomioon kurssien väliset riippuvuudet ja se, miten kurssit sijoittuvat laitoksen opetukseen.

Jos opiskelija tuntee tarvitsevansa matematiikan kertaamista, on syytä suorittaa kurssi "Matematiikka tutuksi" heti periodilla I ja kurssi "Johdatus tietojenkäsittelytieteeseen" vasta periodilla III (ks. aikataulun vaihtoehdot). Kurssi "Johdatus diskreettiin matematiikkaan" on joka tapauksessa syytä suorittaa ensimmäisen vuoden periodilla II; muuten opinnot viivästyvät.

Osa perusopintojen ensimmäisistä opintojaksoista järjestetään kummallakin lukukaudella. Yleensä opintojaksoja (harjoitustöitä, Ohjelmistotuotantoprojektia ja Tieteellisen kirjoittamisen kurssia lukuun ottamatta) järjestetään vain kerran lukuvuodessa. Valinnaisten kurssien luennointikerrat voivat olla vieläkin harvemmassa. Opintojaksojen ohjeellinen toistumisaikataulu selviää verkkosivulta http://www.cs.helsinki.fi/kurssit/toistosuunnitelma.html.

Lähes kaikki pakolliset luentokurssit ja osan valinnaisia kursseja voi suorittaa myös erilliskokeessa tenttimällä joka lukukaudella ja kesällä. Erilliskokeiden ajat löytyvät verkkosivulta http://www.cs.helsinki.fi/kokeet/.

Käyttämällä hyväksi kesän mahdollisuuksia (pääaineopinnoista harjoitustyöt, Ohjelmistotuotantoprojekti) voi kevät- ja syyslukukausien opintopistemäärää keventää tai toisaalta nopeuttaa opintojaan. Jos suunnitelman kolmen vuoden aikataulu tuntuu liian raskaalta, ohjelmaa voi hallitusti hieman väljentää, ottamalla kuitenkin huomioon opintojaksojen esitietosuhteet sekä opetusohjelman rajoitukset.

Tutkintojen valmistumisen seuranta- ja tukijärjestelmässä (Etappi) on asetettu opintojen etenemiseen liittyviä tarkistuskohtia. Opiskelija, jolla on ensimmäisen opiskeluvuoden kolmannen periodin jälkeen suoritettuna opintoja vähemmän kuin 25 op, saa HOPS-työskentelyn yhteydessä tehostettua ohjausta. Jos LuK-tutkinnosta on kolmen opiskeluvuoden jälkeen suoritettu vähemmän kuin 120 op tai tutkinto on neljän opiskeluvuoden jälkeen edelleen kesken, opiskelija voi ilmoittautua läsnäolevaksi vain tehtyään hyväksyttävän opintosuunnitelman. Opintosuunnitelmien hyväksymiskäytännöstä ilmoitetaan erikseen.

LuK-tutkinnon malliaikataulu

Malliaikataulu esitetään uusien tutkintovaatimusten mukaiselle opiskelulle. Edellisten vaatimusten mukaisen opiskelun aikataulu on edellisissä opinto-oppaissa; pääsääntöisesti opiskelu lukuvuosina 2008–2009 ja 2009–2010 on mahdollista vanhan aikataulun mukaan.

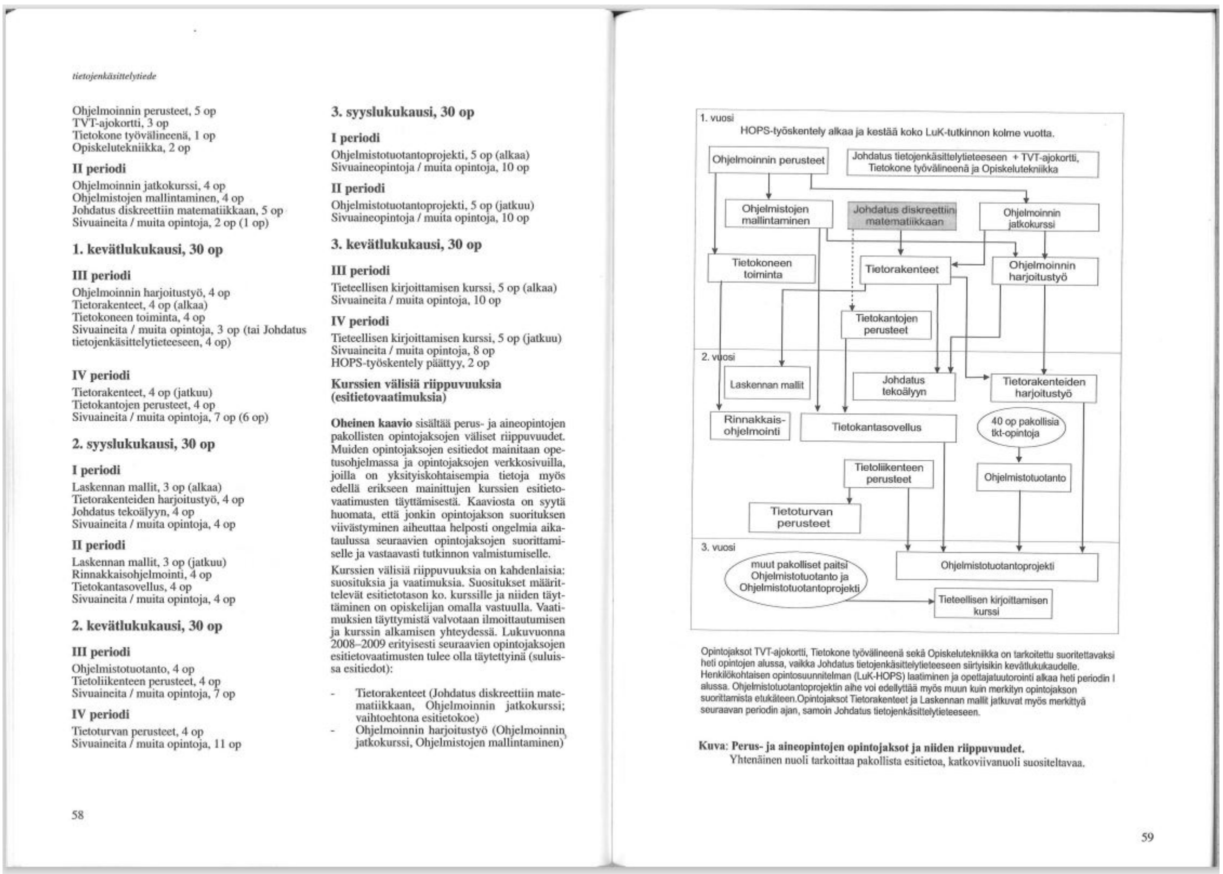
Ensimmäisen syksyn ohjelma sisältää vaihtoehdon: niiden, jotka tuntevat tarvitsevansa matematiikassa johdantoluonteisen opintojakson Matematiikka tutuksi, on paikallaan siirtää opintojakson Johdatus tietojenkäsittelytieteeseen opiskelu kevätlukukaudelle, missä sille on hyvin tilaa (sivuaineiden/muiden opintojen kustannuksella). Johdatus tietojenkäsittelytieteeseen -kurssiin liittyviä opintojaksoja TVT-ajokortti, Tietokone työvälineenä ja Opiskelutekniikka ei kuitenkaan pidä siirtää keväälle, koska ne ovat tärkeitä opintojen käynnistämisen kannalta. Jos TVT-ajokortin asiat ovat jo tuttuja, suoritus kuormittaa vähän, ja syksyn ohjelma voi sallia myös kummankin vaihtoehdon opintojaksojen suorittamisen. TVT-ajokortti-tietämystään on syytä testata mahdollisimman ajoissa omatoimisesti lähtötasotestillä; tiedot kurssin verkkosivulla.

1. syyslukukausi, 30 op

I periodi

HOPS-työskentely alkaa (sen päättyessä 3. vuonna 2 op)
Johdatus tietojenkäsittelytieteeseen, 4 op
(tai Matematiikka tutuksi, 5 op)

Figure A.3: Sample structure of Bachelor level CS studies, 2008-2010



tietojenkäsittelytiede

Ohjelmoinnin perusteet, 5 op
TVT-ajokortti, 3 op
Tietokone työvälineenä, 1 op
Opiskelutekniikka, 2 op

II periodi

Ohjelmoinnin jatkokurssi, 4 op
Ohjelmistojen mallintaminen, 4 op
Johdatus diskreettiin matematiikkaan, 5 op
Sivuaineita / muita opintoja, 2 op (1 op)

1. kevätlukukausi, 30 op

III periodi

Ohjelmoinnin harjoitustyö, 4 op
Tietorakenteet, 4 op (alkaa)
Tietokoneen toiminta, 4 op
Sivuaineita / muita opintoja, 3 op (tai Johdatus tietojenkäsittelytieteeseen, 4 op)

IV periodi

Tietorakenteet, 4 op (jatkuu)
Tietokantojen perusteet, 4 op
Sivuaineita / muita opintoja, 7 op (6 op)

2. syyslukukausi, 30 op

I periodi

Laskennan mallit, 3 op (alkaa)
Tietorakenteiden harjoitustyö, 4 op
Johdatus tekoälyyn, 4 op
Sivuaineita / muita opintoja, 4 op

II periodi

Laskennan mallit, 3 op (jatkuu)
Rinnakkaisohjelmointi, 4 op
Tietokantasovellus, 4 op
Sivuaineita / muita opintoja, 4 op

2. kevätlukukausi, 30 op

III periodi

Ohjelmistotuotanto, 4 op
Tietoliikenteen perusteet, 4 op
Sivuaineita / muita opintoja, 7 op

IV periodi

Tietoturvan perusteet, 4 op
Sivuaineita / muita opintoja, 11 op

58

3. syyslukukausi, 30 op

I periodi

Ohjelmistotuotantoprojekti, 5 op (alkaa)
Sivuaineopintoja / muita opintoja, 10 op

II periodi

Ohjelmistotuotantoprojekti, 5 op (jatkuu)
Sivuaineopintoja / muita opintoja, 10 op

3. kevätlukukausi, 30 op

III periodi

Tieteellisen kirjoittamisen kurssi, 5 op (alkaa)
Sivuaineita / muita opintoja, 10 op

IV periodi

Tieteellisen kirjoittamisen kurssi, 5 op (jatkuu)
Sivuaineita / muita opintoja, 8 op
HOPS-työskentely päättyy, 2 op

Kurssien välisiä riippuvuuksia (esitietovaatimuksia)

Oheinen kaavio sisältää perus- ja aineopintojen pakollisten opintojaksojen väliset riippuvuudet. Muiden opintojaksojen esitiedot mainitaan opetusohjelmassa ja opintojaksojen verkkosivuilla, joilla on yksityiskohtaisempia tietoja myös edellä erikseen mainittujen kurssien esitietovaatimusten täyttämisestä. Kaaviosta on syytä huomata, että jonkin opintojakson suorituksen viivästyminen aiheuttaa helposti ongelmia aikataulussa seuraavien opintojaksojen suorittamiselle ja vastaavasti tutkinnon valmistumiselle.

Kurssien välisiä riippuvuuksia on kahdenlaisia: suosituksia ja vaatimuksia. Suositukset määrittelevät esitietotason ko. kurssille ja niiden täyttäminen on opiskelijan omalla vastuulla. Vaatimuksien täyttymistä valvotaan ilmoittautumisen ja kurssin alkamisen yhteydessä. Lukuvuonna 2008–2009 erityisesti seuraavien opintojaksojen esitietovaatimusten tulee olla täytettyinä (suluissa esitiedot):

- Tietorakenteet (Johdatus diskreettiin matematiikkaan, Ohjelmoinnin jatkokurssi; vaihtoehtona esitietokoe)
- Ohjelmoinnin harjoitustyö (Ohjelmoinnin jatkokurssi, Ohjelmistojen mallintaminen)

Opintojaksot TVT-ajokortti, Tietokone työvälineenä sekä Opiskelutekniikka on tarkoitettu suoritettavaksi heti opintojen alussa, vaikka Johdatus tietojenkäsittelytieteeseen siirtyisikin kevätlukukaudelle. Henkilökohtaisen opintosuunnitelman (LuK-HOPS) laatiminen ja opettajatuutorointi alkaa heti periodin I alussa. Ohjelmistotuotantoprojektin aihe voi edellyttää myös muun kuin merkityn opintojakson suorittamista etukäteen.Opintojaksot Tietorakenteet ja Laskennan mallit jatkuvat myös merkittyä seuraavan periodin ajan, samoin Johdatus tietojenkäsittelytieteeseen.

Kuva: **Perus- ja aineopintojen opintojaksot ja niiden riippuvuudet.**
Yhtenäinen nuoli tarkoittaa pakollista esitietoa, katkoviivanuoli suositeltavaa.

59

Figure A.4: General structure Bachelor level CS studies, 2010-2012

tietojenkäsittelytiede

Lukuvuonna 2010-2011 opetusperiodit ovat:

I	6.9.–24.10.2010
II	1.11.–19.12.2010
III	17.1.–6.3.2011
IV	14.3.–22.5.2011 *)

Lukuvuonna 2011-2012 opetusperiodit ovat:

I	5.9.–23.10.2011
II	31.10.–18.12.2011
III	16.1.–4.3.2012
IV	12.3.–20.5.2012 *)

*) Periodi IV sisältää pääsiäisviikon sekä intensiivikursseille tarkoitetun kahden viikon jakson periodin lopussa; periodin normaaliopetuksen kesto on sama kuin muillakin periodeilla.

Ilmoittautuminen kevään 2011 ja 2012 ohjelmistotuotantoprojekteihin, tieteellisen kirjoittamisen kurssille ja seminaareihin: 10.–25.11.2010 ja 9.–24.11.2011

Ilmoittautuminen syksyn 2011 ja 2012 ohjelmistotuotantoprojekteihin, tieteellisen kirjoittamisen kurssille ja seminaareihin 11.–26.5.2011 ja 9.–24.5.2012

Kesän 2011 ja 2012 ohjelmistotuotantoprojektiin ilmoittautuminen 1.–20.4.2011 ja 3.–29.4.2012.

TUTKINTOVAATIMUKSET

Nopeasti kehittyvänä alana tietojenkäsittelytieteen tutkintovaatimuksia päivitetään säännöllisesti. Tämän opinto-oppaan sisältämät uudet tutkintovaatimukset tulevat voimaan 1.8.2010 ja ovat lukuvuosien 2005–2008 ja 2008–2010 vaatimuksista poikkeavat.

Näiden tutkintovaatimusten mukaan opiskelevat lukuvuonna 2010–2011 opintonsa aloittavat opiskelijat. Ennen 1.8.2010 opintonsa aloittaneet voivat suorittaa tutkintonsa aloitusvuotensa tai sen jälkeisten tutkintovaatimusten mukaan. On huomattava, että vuosien 2005–2008 tutkintovaatimukset ovat voimassa vain 31.7.2011 asti sekä kandidaatin että maisterin tutkintojen osalta.

Vuosien 2005–2008 ja 2008–2010 tutkintovaatimukset löytyvät edellisistä opinto-oppaista ja laitoksen Opiskelu-verkkosivun kautta.

LUONNONTIETEIDEN KANDIDAATIN TUTKINTO (180 OP)

1. Pääaineopinnot

584111 PERUSOPINNOT (25 OP)
582102 Johdatus tietojenkäsittelytieteeseen, 4 op
581325 Ohjelmoinnin perusteet, 5 op
582103 Ohjelmoinnin jatkokurssi, 4 op
582104 Ohjelmistojen mallintaminen, 4 op
58160 Ohjelmoinnin harjoitustyö, 4 op
581328 Tietokantojen perusteet, 4 op

584213 AINEOPINNOT (63 OP)
58131 Tietorakenteet, 8 op
58161 Tietorakenteiden harjoitustyö, 4 op
582203 Tietokantasovellus, 4 op
581305 Tietokoneen toiminta, 4 op
582202 Tietoliikenteen perusteet, 4 op
581332 Rinnakkaisohjelmointi, 6 op
582206 Laskennan mallit, 6 op
582215 Tietoturvan perusteet, 4 op
582216 Johdatus tekoälyyn, 4 op
581259 Ohjelmistotuotanto, 4 op
581260 Ohjelmistotuotantoprojekti, 9 op
582204 Kandidaatintutkielma, 6 op
50036 Kypsyysnäyte, 0 op

2. Sivuaineopinnot (50 op)

Matematiikan tai menetelmätieteiden perusopintokokonaisuus, 25 op ja
toisen sivuaineen perusopintokokonaisuus, 25 op
tai
matematiikan tai menetelmätieteiden opintokokonaisuuksia, 50 op.

Menetelmätieteiden opintokokonaisuudessa on kaikissa tapauksissa oltava vähintään 10 op matematiikkaa ja vähintään 10 op tilastotiedettä.

Kaikissa tapauksissa sivuaineopintoihin tulee sisältyä matematiikan kurssi Johdatus diskreettiin matematiikkaan.

3. Muut opinnot

Kieliopinnot, 10 op

42

582505 Äidinkielinen viestintä, 3 op
Toinen kotimainen kieli, 3 op
Vieras kieli, 4 op

584401 Tieto- ja viestintätekniikan opinnot, 5 op
582514 TVT-ajokortti, 3 op
581324 Tietokone työvälineenä, 1 op
582506 Tutkimustiedonhaku, 1 op

584402 Harjoittelu tai työelämään orientoituminen, 1–3 op
582508 Ohjelmistotuotantoprojektiin liittyvä työelämään orientointi, 1 op, tai
582509 Tietotekniikka-alan ammattitehtävissä työskentely, 2 op

582507 Henkilökohtainen opintosuunnitelma ja osallistuminen opettajatuutorointiin (LuK-HOPS), 2 op

582513 Opiskelutekniikka, 2 op

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 180 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

FILOSOFIAN MAISTERIN TUTKINTO (120 OP)

Maisterin tutkinto tietojenkäsittelytieteessä voidaan suorittaa kolmella erikoistumislinjalla:

Algoritmit ja koneoppiminen (Algorithms and Machine Learning)
Hajautetut järjestelmät ja tietoliikenne (Networking and Services)
Ohjelmistojärjestelmät

Laitoksella on lisäksi kaksi englanninkielistä maisteriohjelmaa, joiden vaatimukset esitetään erikseen:

Master's Degree Programme in Bioinformatics
International CBU Master's Degree Programme in Information and Communication Technology

Maisterin tutkinnon vaatimukset ovat erikoistumislinjakohtaiset ja noudattavat seuraavaa yhteistä rakennetta.

tietojenkäsittelytiede

1. Pääaineopinnot

AINEOPINNOT

Erikoistumislinjoilla Hajautetut järjestelmät ja tietoliikenne sekä Ohjelmistojärjestelmät vaaditaan aineopintokursseja, jotka voidaan suorittaa kandidaatin tutkinnossa tai maisterin tutkinnossa syventävien opintojen lisäksi.

SYVENTÄVÄT OPINNOT (80 OP)

Erikoistumislinjakohtaiset pakolliset opinnot, 8 op
Erikoistumislinjalle soveltuvia valinnaisia syventäviä opintoja (muita kuin seminaareja), 26 op
Seminaareja, 6 op
50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte

2. Sivuaineopinnot

Erikoistumislinjalla Algoritmit ja koneoppiminen vaaditaan matematiikan tai menetelmätieteiden opintokokonaisuuksia kandidaatin ja maisterin tutkinnoissa yhteensä 60 op.

3. Muut opinnot

582510 Henkilökohtainen opintosuunnitelma (FM-HOPS), 1 op

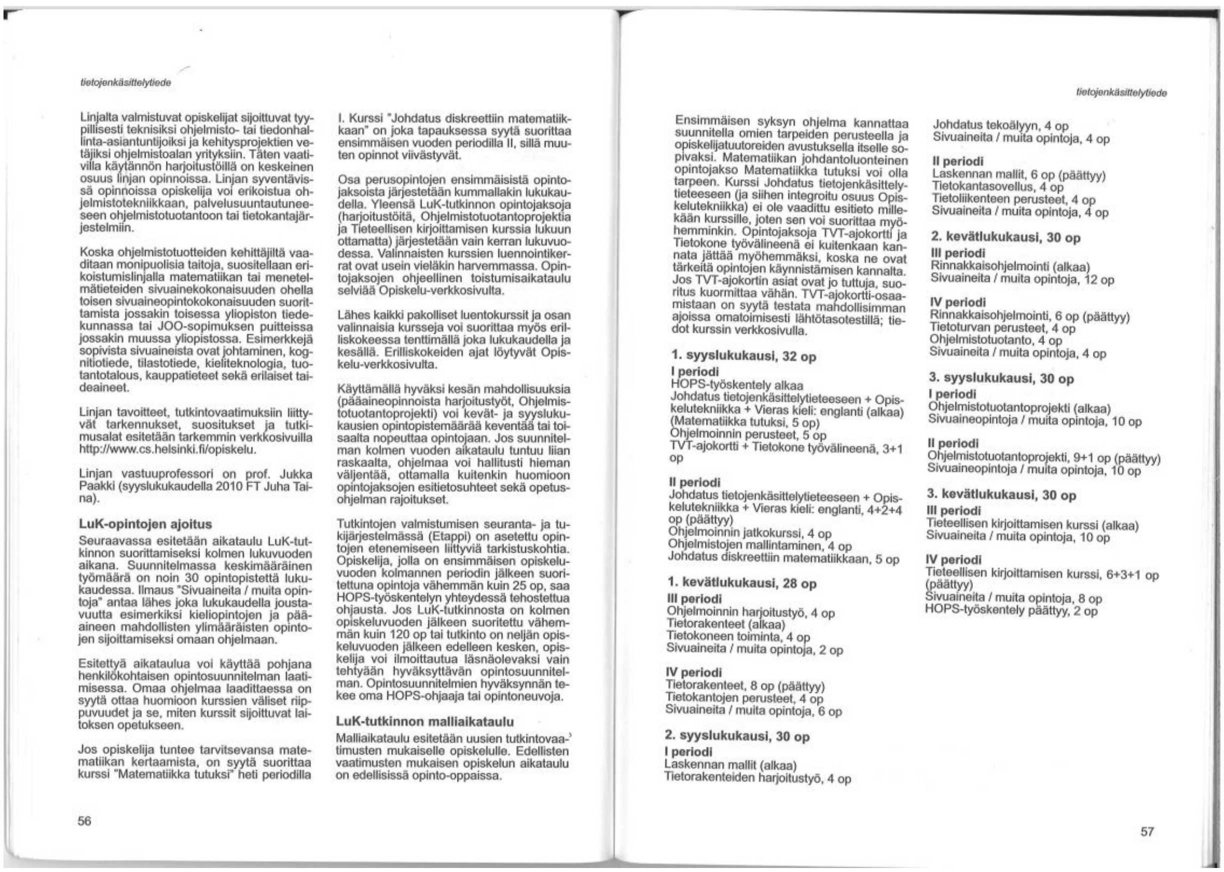
582517 Orientation to MSc Studies, 2 op, mikäli opiskelija suorittaa laitoksella vain FM-tutkinnon.

Englannin kielen opinnot, 0–4 op, laitoksella hyväksyttävän henkilökohtaisen opintosuunnitelman (FM-HOPS) mukaisesti.

584403 Syventävä harjoittelu tai työelämään orientoivat opinnot, vähintään 2 op
582511 Tietotekniikka-alan vaativissa ammattitehtävissä työskentely, 2–6 op
tai
582516 Software Factory Work Experience, 7 op
tai
582515 Tietotekniikka-ala ammattina, 2 op

43

Figure A.5: Proposed timing of Bachelor level CS studies, 2010-2012



tietojenkäsittelytiede

Linjalta valmistuvat opiskelijat sijoittuvat tyypillisesti teknisiksi ohjelmisto- tai tiedonhallinta-asiantuntijoiksi ja kehitysprojektien vetäjiksi ohjelmistoalan yrityksiin. Täten vaativilla käytännön harjoitustöillä on keskeinen osuus linjan opinnoissa. Linjan syventävissä opinnoissa opiskelija voi erikoistua ohjelmistotekniikkaan, palvelusuuntautuneeseen ohjelmistotuotantoon tai tietokantajärjestelmiin.

Koska ohjelmistotuotteiden kehittäjiltä vaaditaan monipuolisia taitoja, suositellaan erikoistumislinjalla matematiikan tai menetelmätieteiden sivuainekokonaisuuden ohella toisen sivuaineopintokokonaisuuden suorittamista jossakin toisessa yliopiston tiedekunnassa tai JOO-sopimuksen puitteissa jossakin muussa yliopistossa. Esimerkkejä sopivista sivuaineista ovat johtaminen, kognitiotiede, tilastotiede, kieliteknologia, tuotantotalous, kauppatieteet sekä erilaiset taideaineet.

Linjan tavoitteet, tutkintovaatimuksiin liittyvät tarkennukset, suositukset ja tutkimusalat esitetään tarkemmin verkkosivuilla http://www.cs.helsinki.fi/opiskelu.

Linjan vastuuprofessori on prof. Jukka Paakki (syyslukukaudella 2010 FT Juha Taina).

LuK-opintojen ajoitus

Seuraavassa esitetään aikataulu LuK-tutkinnon suorittamiseksi kolmen lukuvuoden aikana. Suunnitelmassa keskimääräinen työmäärä on noin 30 opintopistettä lukukaudessa. Ilmaus "Sivuaineita / muita opintoja" antaa lähes joka lukukaudella joustavuutta esimerkiksi kieliopintojen ja pääaineen mahdollisten ylimääräisten opintojen sijoittamiseksi omaan ohjelmaan.

Esitettyä aikataulua voi käyttää pohjana henkilökohtaisen opintosuunnitelman laatimisessa. Omaa ohjelmaa laadittaessa on syytä ottaa huomioon kurssien väliset riippuvuudet ja se, miten kurssit sijoittuvat laitoksen opetukseen.

Jos opiskelija tuntee tarvitsevansa matematiikan kertaamista, on syytä suorittaa kurssi "Matematiikka tutuksi" heti periodilla I. Kurssi "Johdatus diskreettiin matematiikkaan" on joka tapauksessa syytä suorittaa ensimmäisen vuoden periodilla II, sillä muuten opinnot viivästyvät.

Osa perusopintojen ensimmäisistä opintojaksoista järjestetään kummallakin lukukaudella. Yleensä LuK-tutkinnon opintojaksoja (harjoitustöitä, Ohjelmistotuotantoprojektia ja Tieteellisen kirjoittamisen kurssia lukuun ottamatta) järjestetään vain kerran lukuvuodessa. Valinnaisten kurssien luennointikerrat ovat usein vieläkin harvemmassa. Opintojaksojen ohjeellinen toistumisaikataulu selviää Opiskelu-verkkosivulta.

Lähes kaikki pakolliset luentokurssit ja osan valinnaisia kursseja voi suorittaa myös erilliskokeessa tenttimällä joka lukukaudella ja kesällä. Erilliskokeiden ajat löytyvät Opiskelu-verkkosivulta.

Käyttämällä hyväksi kesän mahdollisuuksia (pääaineopinnoista harjoitustyöt, Ohjelmistotuotantoprojekti) voi kevät- ja syyslukukausien opintopistemäärää keventää tai toisaalta nopeuttaa opintojaan. Jos suunnitelman kolmen vuoden aikataulu tuntuu liian raskaalta, ohjelmaa voi hallitusti hieman väljentää, ottamalla kuitenkin huomioon opintojaksojen esitietosuhteet sekä opetusohjelman rajoitukset.

Tutkintojen valmistumisen seuranta- ja tukijärjestelmässä (Etappi) on asetettu opintojen etenemiseen liittyviä tarkistuskohtia. Opiskelija, jolla on ensimmäisen opiskeluvuoden kolmannen periodin jälkeen suoritettuna opintoja vähemmän kuin 25 op, saa HOPS-työskentelyn yhteydessä tehostettua ohjausta. Jos LuK-tutkinnosta on kolmen opiskeluvuoden jälkeen suoritettu vähemmän kuin 120 op tai tutkinto on neljän opiskeluvuoden jälkeen edelleen kesken, opiskelija voi ilmoittautua läsnäolevaksi vain tehtyään hyväksyttävän opintosuunnitelman. Opintosuunnitelmien hyväksynnän tekee oma HOPS-ohjaaja tai opintoneuvoja.

LuK-tutkinnon malliaikataulu

Malliaikataulu esitetään uusien tutkintovaatimusten mukaiselle opiskelulle. Edellisten vaatimusten mukaisen opiskelun aikataulu on edellisissä opinto-oppaissa.

56

tietojenkäsittelytiede

Ensimmäisen syksyn ohjelma kannattaa suunnitella omien tarpeiden perusteella ja opiskelijatuutoreiden avustuksella itselle sopivaksi. Matematiikan johdantoluonteinen opintojakso Matematiikka tutuksi voi olla tarpeen. Kurssi Johdatus tietojenkäsittelytieteeseen (ja siihen integroitu osuus Opiskelutekniikka) ei ole vaadittu esitieto millekään kurssille, joten sen voi suorittaa myöhemminkin. Opintojaksoja TVT-ajokortti ja Tietokone työvälineenä ei kuitenkaan kannata jättää myöhemmäksi, koska ne ovat tärkeitä opintojen käynnistämisen kannalta. Jos TVT-ajokortin asiat ovat jo tuttuja, suoritus kuormittaa vähän. TVT-ajokortti-osaamistaan on syytä testata mahdollisimman ajoissa omatoimisesti lähtötasotestillä; tiedot kurssin verkkosivulla.

1. syyslukukausi, 32 op

I periodi
HOPS-työskentely alkaa
Johdatus tietojenkäsittelytieteeseen + Opiskelutekniikka + Vieras kieli: englanti (alkaa)
(Matematiikka tutuksi, 5 op)
Ohjelmoinnin perusteet, 5 op
TVT-ajokortti + Tietokone työvälineenä, 3+1 op

II periodi
Johdatus tietojenkäsittelytieteeseen + Opiskelutekniikka + Vieras kieli: englanti, 4+2+4 op (päättyy)
Ohjelmoinnin jatkokurssi, 4 op
Ohjelmistojen mallintaminen, 4 op
Johdatus diskreettiin matematiikkaan, 5 op

1. kevätlukukausi, 28 op

III periodi
Ohjelmoinnin harjoitustyö, 4 op
Tietorakenteet (alkaa)
Tietokoneen toiminta, 4 op
Sivuaineita / muita opintoja, 2 op

IV periodi
Tietorakenteet, 8 op (päättyy)
Tietokantojen perusteet, 4 op
Sivuaineita / muita opintoja, 6 op

2. syyslukukausi, 30 op

I periodi
Laskennan mallit (alkaa)
Tietorakenteiden harjoitustyö, 4 op
Johdatus tekoälyyn, 4 op
Sivuaineita / muita opintoja, 4 op

II periodi
Laskennan mallit, 6 op (päättyy)
Tietokantasovellus, 4 op
Tietoliikenteen perusteet, 4 op
Sivuaineita / muita opintoja, 4 op

2. kevätlukukausi, 30 op

III periodi
Rinnakkaisohjelmointi (alkaa)
Sivuaineita / muita opintoja, 12 op

IV periodi
Rinnakkaisohjelmointi, 6 op (päättyy)
Tietoturvan perusteet, 4 op
Ohjelmistotuotanto, 4 op
Sivuaineita / muita opintoja, 4 op

3. syyslukukausi, 30 op

I periodi
Ohjelmistotuotantoprojekti (alkaa)
Sivuaineopintoja / muita opintoja, 10 op

II periodi
Ohjelmistotuotantoprojekti, 9+1 op (päättyy)
Sivuaineopintoja / muita opintoja, 10 op

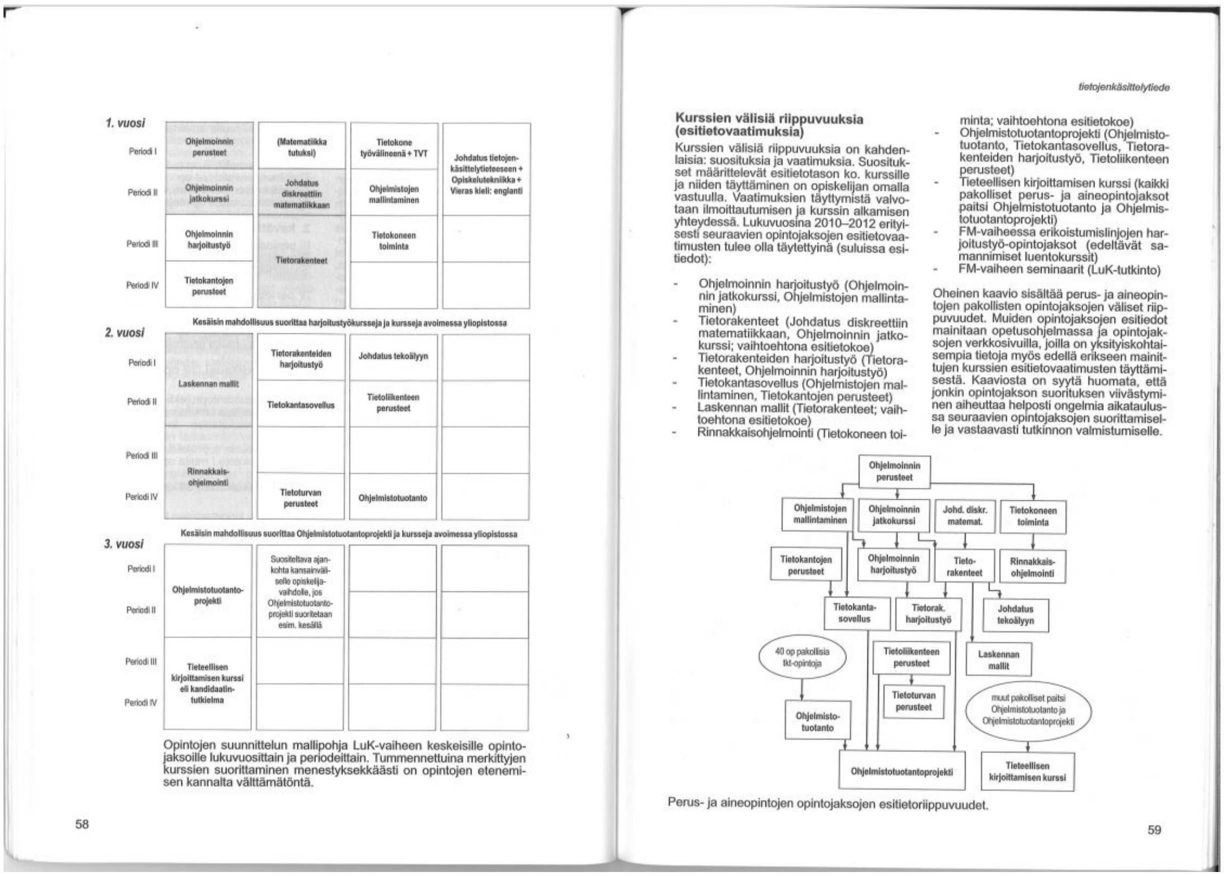
3. kevätlukukausi, 30 op

III periodi
Tieteellisen kirjoittamisen kurssi (alkaa)
Sivuaineita / muita opintoja, 10 op

IV periodi
Tieteellisen kirjoittamisen kurssi, 6+3+1 op (päättyy)
Sivuaineita / muita opintoja, 8 op
HOPS-työskentely päättyy, 2 op

57

Figure A.6: Sample structure of Bachelor level CS studies, 2010-2012



1. vuosi

Periodi I	Ohjelmoinnin perusteet	(Matematiikka tutuksi)	Tietokone työvälineenä + TVT	Johdatus tietojenkäsittelytieteeseen + Opiskelutekniikka + Vieras kieli: englanti
Periodi II	Ohjelmoinnin jatkokurssi	Johdatus diskreettiin matematiikkaan	Ohjelmistojen mallintaminen	
Periodi III	Ohjelmoinnin harjoitustyö	Tietorakenteet	Tietokoneen toiminta	
Periodi IV	Tietokantojen perusteet			

Kesäisin mahdollisuus suorittaa harjoitustyökursseja ja kursseja avoimessa yliopistossa

2. vuosi

Periodi I	Laskennan mallit	Tietorakenteiden harjoitustyö	Johdatus tekoälyyn	
Periodi II		Tietokantasovellus	Tietoliikenteen perusteet	
Periodi III	Rinnakkaisohjelmointi			
Periodi IV		Tietoturvan perusteet	Ohjelmistotuotanto	

Kesäisin mahdollisuus suorittaa Ohjelmistotuotantoprojekti ja kursseja avoimessa yliopistossa

3. vuosi

Periodi I	Ohjelmistotuotantoprojekti	Suositeltava ajankohta kansainväliselle opiskelijavaihdolle, jos Ohjelmistotuotantoprojekti suoritetaan esim. kesällä		
Periodi II				
Periodi III	Tieteellisen kirjoittamisen kurssi eli kandidaatintutkielma			
Periodi IV				

Opintojen suunnittelun mallipohja LuK-vaiheen keskeisille opintojaksoille lukuvuosittain ja periodeittain. Tummennettuina merkittyjen kurssien suorittaminen menestyksekkäästi on opintojen etenemisen kannalta välttämätöntä.

58

tietojenkäsittelytiede

Kurssien välisiä riippuvuuksia (esitietovaatimuksia)

Kurssien välisiä riippuvuuksia on kahdenlaisia: suosituksia ja vaatimuksia. Suositukset määrittelevät esitietotason ko. kurssille ja niiden täyttäminen on opiskelijan omalla vastuulla. Vaatimuksien täyttymistä valvotaan ilmoittautumisen ja kurssin alkamisen yhteydessä. Lukuvuosina 2010–2012 erityisesti seuraavien opintojaksojen esitietovaatimusten tulee olla täytettyinä (suluissa esitiedot):

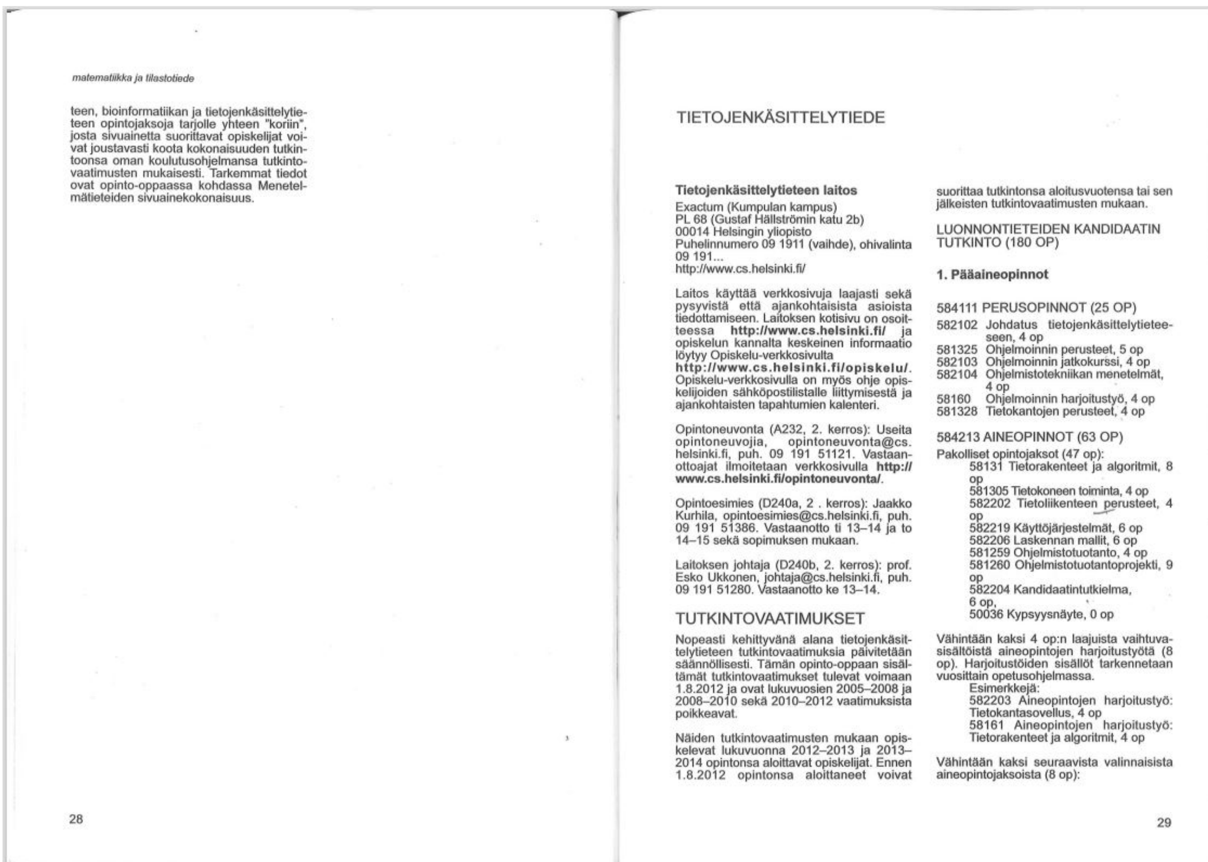
- Ohjelmoinnin harjoitustyö (Ohjelmoinnin jatkokurssi, Ohjelmistojen mallintaminen)
- Tietorakenteet (Johdatus diskreettiin matematiikkaan, Ohjelmoinnin jatkokurssi; vaihtoehtona esitietokoe)
- Tietorakenteiden harjoitustyö (Tietorakenteet, Ohjelmoinnin harjoitustyö)
- Tietokantasovellus (Ohjelmistojen mallintaminen, Tietokantojen perusteet)
- Laskennan mallit (Tietorakenteet; vaihtoehtona esitietokoe)
- Rinnakkaisohjelmointi (Tietokoneen toiminta; vaihtoehtona esitietokoe)
- Ohjelmistotuotantoprojekti (Ohjelmistotuotanto, Tietokantasovellus, Tietorakenteiden harjoitustyö, Tietoliikenteen perusteet)
- Tieteellisen kirjoittamisen kurssi (kaikki pakolliset perus- ja aineopintojaksot paitsi Ohjelmistotuotanto ja Ohjelmistotuotantoprojekti)
- FM-vaiheessa erikoistumislinjojen harjoitustyö-opintojaksot (edeltävät samannimiset luentokurssit)
- FM-vaiheen seminaarit (LuK-tutkinto)

Oheinen kaavio sisältää perus- ja aineopintojen pakollisten opintojaksojen väliset riippuvuudet. Muiden opintojaksojen esitiedot mainitaan opetusohjelmassa ja opintojaksojen verkkosivuilla, joilla on yksityiskohtaisempia tietoja myös edellä erikseen mainittujen kurssien esitietovaatimusten täyttämisestä. Kaaviosta on syytä huomata, että jonkin opintojakson suorituksen viivästyminen aiheuttaa helposti ongelmia aikataulussa seuraavien opintojaksojen suorittamiselle ja vastaavasti tutkinnon valmistumiselle.

Perus- ja aineopintojen opintojaksojen esitietoriippuvuudet.

59

Figure A.7: General structure Bachelor level CS studies, 2012-2014



matematiikka ja tilastotiede

teen, bioinformatiikan ja tietojenkäsittelytieteen opintojaksoja tarjolle yhteen "koriin", josta sivuainetta suorittavat opiskelijat voivat joustavasti koota kokonaisuuden tutkintoonsa oman koulutusohjelmansa tutkintovaatimusten mukaisesti. Tarkemmat tiedot ovat opinto-oppaassa kohdassa Menetelmätieteiden sivuainekokonaisuus.

28

TIETOJENKÄSITTELYTIEDE

Tietojenkäsittelytieteen laitos
Exactum (Kumpulan kampus)
PL 68 (Gustaf Hällströmin katu 2b)
00014 Helsingin yliopisto
Puhelinnumero 09 1911 (vaihde), ohivalinta 09 191...
http://www.cs.helsinki.fi/

Laitos käyttää verkkosivuja laajasti sekä pysyvistä että ajankohtaisista asioista tiedottamiseen. Laitoksen kotisivu on osoitteessa **http://www.cs.helsinki.fi/** ja opiskelun kannalta keskeinen informaatio löytyy Opiskelu-verkkosivulta **http://www.cs.helsinki.fi/opiskelu/**. Opiskelu-verkkosivulla on myös ohje opiskelijoiden sähköpostilistalle liittymisestä ja ajankohtaisten tapahtumien kalenteri.

Opintoneuvonta (A232, 2. kerros): Useita opintoneuvojia, opintoneuvonta@cs.helsinki.fi, puh. 09 191 51121. Vastaanottoajat ilmoitetaan verkkosivulla **http://www.cs.helsinki.fi/opintoneuvonta/**.

Opintoesimies (D240a, 2 . kerros): Jaakko Kurhila, opintoesimies@cs.helsinki.fi, puh. 09 191 51386. Vastaanotto ti 13–14 ja to 14–15 sekä sopimuksen mukaan.

Laitoksen johtaja (D240b, 2. kerros): prof. Esko Ukkonen, johtaja@cs.helsinki.fi, puh. 09 191 51280. Vastaanotto ke 13–14.

TUTKINTOVAATIMUKSET

Nopeasti kehittyvänä alana tietojenkäsittelytieteen tutkintovaatimuksia päivitetään säännöllisesti. Tämän opinto-oppaan sisältämät tutkintovaatimukset tulevat voimaan 1.8.2012 ja ovat lukuvuosien 2005–2008 ja 2008–2010 sekä 2010–2012 vaatimuksista poikkeavat.

Näiden tutkintovaatimusten mukaan opiskelevat lukuvuonna 2012–2013 ja 2013–2014 opintonsa aloittavat opiskelijat. Ennen 1.8.2012 opintonsa aloittaneet voivat suorittaa tutkintonsa aloitusvuotensa tai sen jälkeisten tutkintovaatimusten mukaan.

LUONNONTIETEIDEN KANDIDAATIN TUTKINTO (180 OP)

1. Pääaineopinnot

584111 PERUSOPINNOT (25 OP)
582102 Johdatus tietojenkäsittelytieteeseen, 4 op
581325 Ohjelmoinnin perusteet, 5 op
582103 Ohjelmoinnin jatkokurssi, 4 op
582104 Ohjelmistotekniikan menetelmät, 4 op
58160 Ohjelmoinnin harjoitustyö, 4 op
581328 Tietokantojen perusteet, 4 op

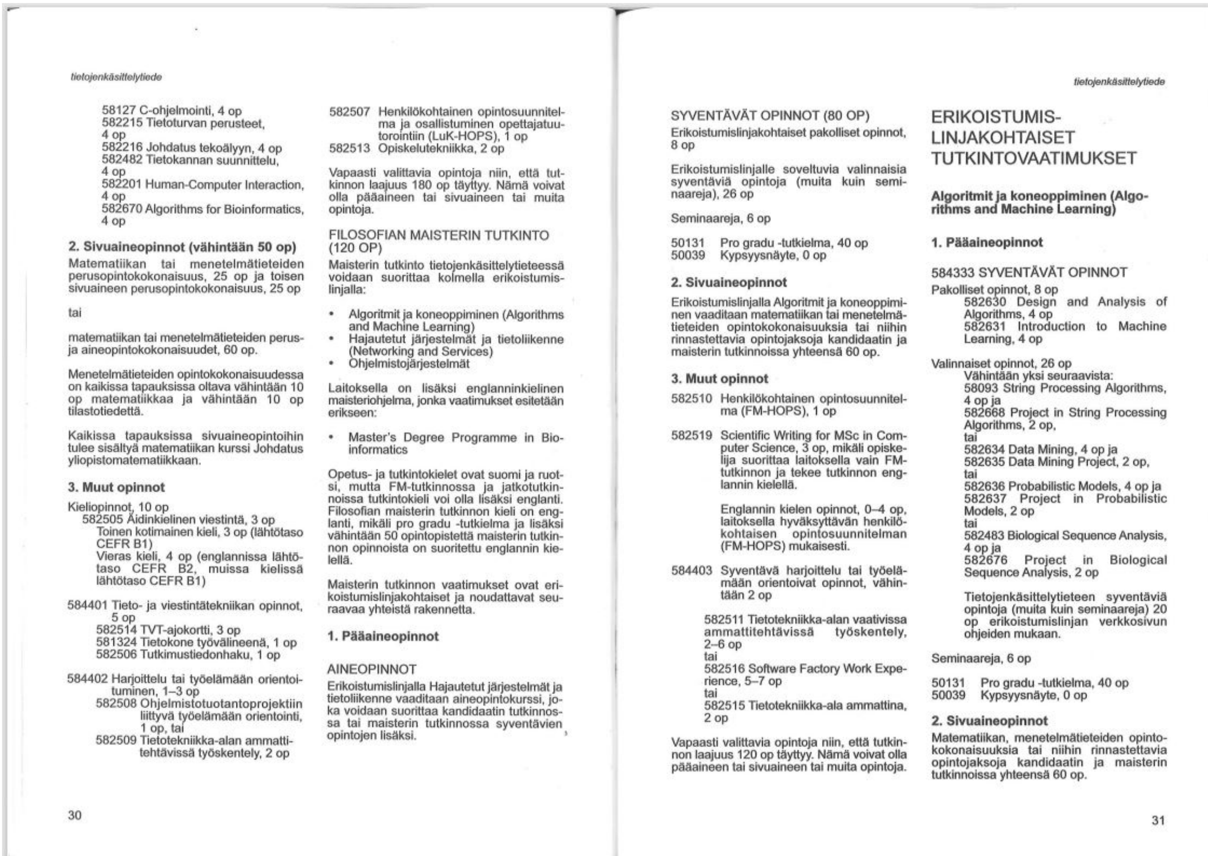
584213 AINEOPINNOT (63 OP)
Pakolliset opintojaksot (47 op):
58131 Tietorakenteet ja algoritmit, 8 op
581305 Tietokoneen toiminta, 4 op
582202 Tietoliikenteen perusteet, 4 op
582219 Käyttöjärjestelmät, 6 op
582206 Laskennan mallit, 6 op
581259 Ohjelmistotuotanto, 4 op
581260 Ohjelmistotuotantoprojekti, 9 op
582204 Kandidaatintutkielma, 6 op,
50036 Kypsyysnäyte, 0 op

Vähintään kaksi 4 op:n laajuista vaihtuvasisältöistä aineopintojen harjoitustyötä (8 op). Harjoitustöiden sisällöt tarkennetaan vuosittain opetusohjelmassa.
Esimerkkejä:
582203 Aineopintojen harjoitustyö: Tietokantasovellus, 4 op
58161 Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit, 4 op

Vähintään kaksi seuraavista valinnaisista aineopintojaksoista (8 op):

29

Figure A.8: The structure of Bachelor level CS studies, 2010-2012



tietojenkäsittelytiede

58127 C-ohjelmointi, 4 op
582215 Tietoturvan perusteet, 4 op
582216 Johdatus tekoälyyn, 4 op
582482 Tietokannan suunnittelu, 4 op
582201 Human-Computer Interaction, 4 op
582670 Algorithms for Bioinformatics, 4 op

2. Sivuaineopinnot (vähintään 50 op)

Matematiikan tai menetelmätieteiden perusopintokokonaisuus, 25 op ja toisen sivuaineen perusopintokokonaisuus, 25 op

tai

matematiikan tai menetelmätieteiden perus- ja aineopintokokonaisuudet, 60 op.

Menetelmätieteiden opintokokonaisuudessa on kaikissa tapauksissa oltava vähintään 10 op matematiikkaa ja vähintään 10 op tilastotiedettä.

Kaikissa tapauksissa sivuaineopintoihin tulee sisältyä matematiikan kurssi Johdatus yliopistomatematiikkaan.

3. Muut opinnot

Kieliopinnot, 10 op
582505 Äidinkielinen viestintä, 3 op
Toinen kotimainen kieli, 3 op (lähtötaso CEFR B1)
Vieras kieli, 4 op (englannissa lähtötaso CEFR B2, muissa kielissä lähtötaso CEFR B1)

584401 Tieto- ja viestintätekniikan opinnot, 5 op
582514 TVT-ajokortti, 3 op
581324 Tietokone työvälineenä, 1 op
582506 Tutkimustiedonhaku, 1 op

584402 Harjoittelu tai työelämään orientoituminen, 1–3 op
582508 Ohjelmistotuotantoprojektiin liittyvä työelämään orientointi, 1 op, tai
582509 Tietotekniikka-alan ammattitehtävissä työskentely, 2 op

582507 Henkilökohtainen opintosuunnitelma ja osallistuminen opettajatuutorointiin (LuK-HOPS), 1 op
582513 Opiskelutekniikka, 2 op

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 180 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

FILOSOFIAN MAISTERIN TUTKINTO (120 OP)

Maisterin tutkinto tietojenkäsittelytieteessä voidaan suorittaa kolmella erikoistumislinjalla:

- Algoritmit ja koneoppiminen (Algorithms and Machine Learning)
- Hajautetut järjestelmät ja tietoliikenne (Networking and Services)
- Ohjelmistojärjestelmät

Laitoksella on lisäksi englanninkielinen maisteriohjelma, jonka vaatimukset esitetään erikseen:

- Master's Degree Programme in Bioinformatics

Opetus- ja tutkintokielet ovat suomi ja ruotsi, mutta FM-tutkinnossa ja jatkotutkinnoissa tutkintokieli voi olla lisäksi englanti. Filosofian maisterin tutkinnon kieli on englanti, mikäli pro gradu -tutkielma ja lisäksi vähintään 50 opintopistettä maisterin tutkinnon opinnoista on suoritettu englannin kielellä.

Maisterin tutkinnon vaatimukset ovat erikoistumislinjakohtaiset ja noudattavat seuraavaa yhteistä rakennetta.

1. Pääaineopinnot

AINEOPINNOT

Erikoistumislinjalla Hajautetut järjestelmät ja tietoliikenne vaaditaan aineopintokurssi, joka voidaan suorittaa kandidaatin tutkinnossa tai maisterin tutkinnossa syventävien opintojen lisäksi.

30

SYVENTÄVÄT OPINNOT (80 OP)

Erikoistumislinjakohtaiset pakolliset opinnot, 8 op

Erikoistumislinjalle soveltuvia valinnaisia syventäviä opintoja (muita kuin seminaareja), 26 op

Seminaareja, 6 op

50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte, 0 op

2. Sivuaineopinnot

Erikoistumislinjalla Algoritmit ja koneoppiminen vaaditaan matematiikan tai menetelmätieteiden opintokokonaisuuksia tai niihin rinnastettavia opintojaksoja kandidaatin ja maisterin tutkinnoissa yhteensä 60 op.

3. Muut opinnot

582510 Henkilökohtainen opintosuunnitelma (FM-HOPS), 1 op

582519 Scientific Writing for MSc in Computer Science, 3 op, mikäli opiskelija suorittaa laitoksella vain FM-tutkinnon ja tekee tutkinnon englannin kielellä.

Englannin kielen opinnot, 0–4 op, laitoksella hyväksyttävän henkilökohtaisen opintosuunnitelman (FM-HOPS) mukaisesti.

584403 Syventävä harjoittelu tai työelämään orientoivat opinnot, vähintään 2 op

582511 Tietotekniikka-alan vaativissa ammattitehtävissä työskentely, 2–6 op
tai
582516 Software Factory Work Experience, 5–7 op
tai
582515 Tietotekniikka-ala ammattina, 2 op

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 120 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

tietojenkäsittelytiede

ERIKOISTUMISLINJAKOHTAISET TUTKINTOVAATIMUKSET

Algoritmit ja koneoppiminen (Algorithms and Machine Learning)

1. Pääaineopinnot

584333 SYVENTÄVÄT OPINNOT

Pakolliset opinnot, 8 op
582630 Design and Analysis of Algorithms, 4 op
582631 Introduction to Machine Learning, 4 op

Valinnaiset opinnot, 26 op
Vähintään yksi seuraavista:
58093 String Processing Algorithms, 4 op ja
582668 Project in String Processing Algorithms, 2 op,
tai
582634 Data Mining, 4 op ja
582635 Data Mining Project, 2 op,
tai
582636 Probabilistic Models, 4 op ja
582637 Project in Probabilistic Models, 2 op
tai
582483 Biological Sequence Analysis, 4 op ja
582676 Project in Biological Sequence Analysis, 2 op

Tietojenkäsittelytieteen syventäviä opintoja (muita kuin seminaareja) 20 op erikoistumislinjan verkkosivun ohjeiden mukaan.

Seminaareja, 6 op

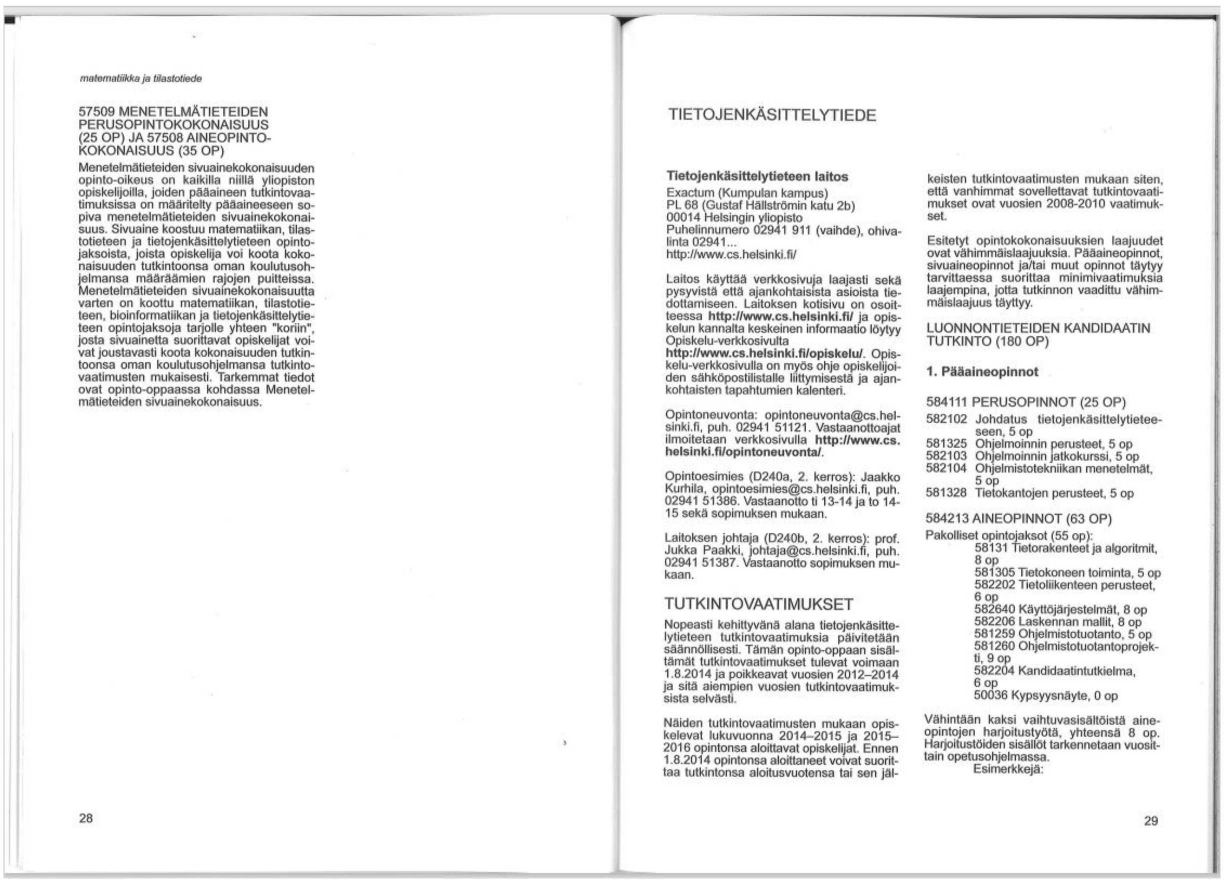
50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte, 0 op

2. Sivuaineopinnot

Matematiikan, menetelmätieteiden opintokokonaisuuksia tai niihin rinnastettavia opintojaksoja kandidaatin ja maisterin tutkinnoissa yhteensä 60 op.

31

Figure A.9: General structure Bachelor level CS studies, 2014-2016



matematiikka ja tilastotiede

57509 MENETELMÄTIETEIDEN PERUSOPINTOKOKONAISUUS (25 OP) JA 57508 AINEOPINTO-KOKONAISUUS (35 OP)

Menetelmätieteiden sivuainekokonaisuuden opinto-oikeus on kaikilla niillä yliopiston opiskelijoilla, joiden pääaineen tutkintovaatimuksissa on määritelty pääaineeseen sopiva menetelmätieteiden sivuainekokonaisuus. Sivuaine koostuu matematiikan, tilastotieteen ja tietojenkäsittelytieteen opintojaksoista, joista opiskelija voi koota kokonaisuuden tutkintoonsa oman koulutusohjelmansa määräämien rajojen puitteissa. Menetelmätieteiden sivuainekokonaisuutta varten on koottu matematiikan, tilastotieteen, bioinformatiikan ja tietojenkäsittelytieteen opintojaksoja tarjolle yhteen "koriin", josta sivuainetta suorittavat opiskelijat voivat joustavasti koota kokonaisuuden tutkintoonsa oman koulutusohjelmansa tutkintovaatimusten mukaisesti. Tarkemmat tiedot ovat opinto-oppaassa kohdassa Menetelmätieteiden sivuainekokonaisuus.

28

TIETOJENKÄSITTELYTIEDE

Tietojenkäsittelytieteen laitos
Exactum (Kumpulan kampus)
PL 68 (Gustaf Hällströmin katu 2b)
00014 Helsingin yliopisto
Puhelinnumero 02941 911 (vaihde), ohivalinta 02941...
http://www.cs.helsinki.fi/

Laitos käyttää verkkosivuja laajasti sekä pysyvistä että ajankohtaisista asioista tiedottamiseen. Laitoksen kotisivu on osoitteessa **http://www.cs.helsinki.fi/** ja opiskelun kannalta keskeinen informaatio löytyy Opiskelu-verkkosivulta **http://www.cs.helsinki.fi/opiskelu/**. Opiskelu-verkkosivulla on myös ohje opiskelijoiden sähköpostilistalle liittymisestä ja ajankohtaisten tapahtumien kalenteri.

Opintoneuvonta: opintoneuvonta@cs.helsinki.fi, puh. 02941 51121. Vastaanottoajat ilmoitetaan verkkosivulla **http://www.cs.helsinki.fi/opintoneuvonta/**.

Opintoesimies (D240a, 2. kerros): Jaakko Kurhila, opintoesimies@cs.helsinki.fi, puh. 02941 51386. Vastaanotto ti 13-14 ja to 14-15 sekä sopimuksen mukaan.

Laitoksen johtaja (D240b, 2. kerros): prof. Jukka Paakki, johtaja@cs.helsinki.fi, puh. 02941 51387. Vastaanotto sopimuksen mukaan.

TUTKINTOVAATIMUKSET

Nopeasti kehittyvänä alana tietojenkäsittelytieteen tutkintovaatimuksia päivitetään säännöllisesti. Tämän opinto-oppaan sisältämät tutkintovaatimukset tulevat voimaan 1.8.2014 ja poikkeavat vuosien 2012–2014 ja sitä aiempien vuosien tutkintovaatimuksista selvästi.

Näiden tutkintovaatimusten mukaan opiskelevat lukuvuonna 2014–2015 ja 2015–2016 opintonsa aloittavat opiskelijat. Ennen 1.8.2014 opintonsa aloittaneet voivat suorittaa tutkintonsa aloitusvuotensa tai sen jälkeisten tutkintovaatimusten mukaan siten, että vanhimmat sovellettavat tutkintovaatimukset ovat vuosien 2008-2010 vaatimukset.

Esitetyt opintokokonaisuuksien laajuudet ovat vähimmäislaajuuksia. Pääaineopinnot, sivuaineopinnot ja/tai muut opinnot täytyy tarvittaessa suorittaa minimivaatimuksia laajempina, jotta tutkinnon vaadittu vähimmäislaajuus täyttyy.

LUONNONTIETEIDEN KANDIDAATIN TUTKINTO (180 OP)

1. Pääaineopinnot

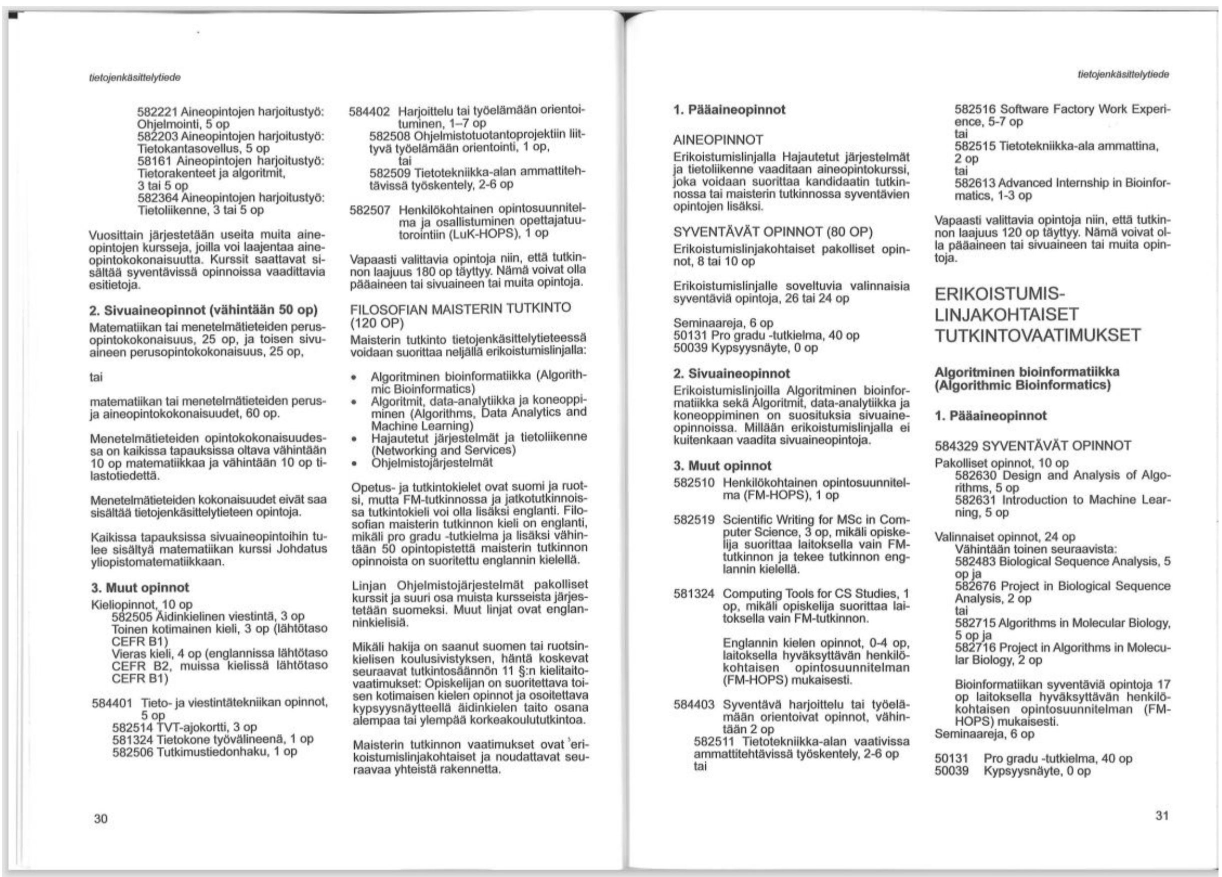
584111 PERUSOPINNOT (25 OP)
582102 Johdatus tietojenkäsittelytieteeseen, 5 op
581325 Ohjelmoinnin perusteet, 5 op
582103 Ohjelmoinnin jatkokurssi, 5 op
582104 Ohjelmistotekniikan menetelmät, 5 op
581328 Tietokantojen perusteet, 5 op

584213 AINEOPINNOT (63 OP)
Pakolliset opintojaksot (55 op):
58131 Tietorakenteet ja algoritmit, 8 op
581305 Tietokoneen toiminta, 5 op
582202 Tietoliikenteen perusteet, 6 op
582640 Käyttöjärjestelmät, 8 op
582206 Laskennan mallit, 8 op
581259 Ohjelmistotuotanto, 5 op
581260 Ohjelmistotuotantoprojekti, 9 op
582204 Kandidaatintutkielma, 6 op
50036 Kypsyysnäyte, 0 op

Vähintään kaksi vaihtuvasisältöistä aineopintojen harjoitustyötä, yhteensä 8 op. Harjoitustöiden sisällöt tarkennetaan vuosittain opetusohjelmassa.
Esimerkkejä:

29

Figure A.10: The structure of Bachelor level CS studies, 2014-2016



tietojenkäsittelytiede

582221 Aineopintojen harjoitustyö: Ohjelmointi, 5 op
582203 Aineopintojen harjoitustyö: Tietokantasovellus, 5 op
58161 Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit, 3 tai 5 op
582364 Aineopintojen harjoitustyö: Tietoliikenne, 3 tai 5 op

Vuosittain järjestetään useita muita aineopintojen kursseja, joilla voi laajentaa aineopintokokonaisuutta. Kurssit saattavat sisältää syventävissä opinnoissa vaadittavia esitietoja.

2. Sivuaineopinnot (vähintään 50 op)

Matematiikan tai menetelmätieteiden perusopintokokonaisuus, 25 op, ja toisen sivuaineen perusopintokokonaisuus, 25 op,

tai

matematiikan tai menetelmätieteiden perus- ja aineopintokokonaisuudet, 60 op.

Menetelmätieteiden opintokokonaisuudessa on kaikissa tapauksissa oltava vähintään 10 op matematiikkaa ja vähintään 10 op tilastotiedettä.

Menetelmätieteiden kokonaisuudet eivät saa sisältää tietojenkäsittelytieteen opintoja.

Kaikissa tapauksissa sivuaineopintoihin tulee sisältyä matematiikan kurssi Johdatus yliopistomatematiikkaan.

3. Muut opinnot

Kieliopinnot, 10 op
582505 Äidinkielinen viestintä, 3 op
Toinen kotimainen kieli, 3 op (lähtötaso CEFR B1)
Vieras kieli, 4 op (englannissa lähtötaso CEFR B2, muissa kielissä lähtötaso CEFR B1)

584401 Tieto- ja viestintätekniikan opinnot, 5 op
582514 TVT-ajokortti, 3 op
581324 Tietokone työvälineenä, 1 op
582506 Tutkimustiedonhaku, 1 op

584402 Harjoittelu tai työelämään orientoituminen, 1–7 op
582508 Ohjelmistotuotantoprojektiin liittyvä työelämään orientointi, 1 op,
tai
582509 Tietotekniikka-alan ammattitehtävissä työskentely, 2-6 op

582507 Henkilökohtainen opintosuunnitelma ja osallistuminen opettajatuutorointiin (LuK-HOPS), 1 op

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 180 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

FILOSOFIAN MAISTERIN TUTKINTO (120 OP)

Maisterin tutkinto tietojenkäsittelytieteessä voidaan suorittaa neljällä erikoistumislinjalla:

- Algoritminen bioinformatiikka (Algorithmic Bioinformatics)
- Algoritmit, data-analytiikka ja koneoppiminen (Algorithms, Data Analytics and Machine Learning)
- Hajautetut järjestelmät ja tietoliikenne (Networking and Services)
- Ohjelmistojärjestelmät

Opetus- ja tutkintokielet ovat suomi ja ruotsi, mutta FM-tutkinnossa ja jatkotutkinnoissa tutkintokieli voi olla lisäksi englanti. Filosofian maisterin tutkinnon kieli on englanti, mikäli pro gradu -tutkielma ja lisäksi vähintään 50 opintopistettä maisterin tutkinnon opinnoista on suoritettu englannin kielellä.

Linjan Ohjelmistojärjestelmät pakolliset kurssit ja suuri osa muista kursseista järjestetään suomeksi. Muut linjat ovat englanninkielisiä.

Mikäli hakija on saanut suomen tai ruotsinkielisen koulusivistyksen, häntä koskevat seuraavat tutkintosäännön 11 §:n kielitaitovaatimukset: Opiskelijan on suoritettava toisen kotimaisen kielen opinnot ja osoitettava kypsyysnäytteellä äidinkielen taito osana alempaa tai ylempää korkeakoulututkintoa.

Maisterin tutkinnon vaatimukset ovat erikoistumislinjakohtaiset ja noudattavat seuraavaa yhteistä rakennetta.

30

1. Pääaineopinnot

AINEOPINNOT

Erikoistumislinjalla Hajautetut järjestelmät ja tietoliikenne vaaditaan aineopintokurssi, joka voidaan suorittaa kandidaatin tutkinnossa tai maisterin tutkinnossa syventävien opintojen lisäksi.

SYVENTÄVÄT OPINNOT (80 OP)

Erikoistumislinjakohtaiset pakolliset opinnot, 8 tai 10 op

Erikoistumislinjalle soveltuvia valinnaisia syventäviä opintoja, 26 tai 24 op

Seminaareja, 6 op
50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte, 0 op

2. Sivuaineopinnot

Erikoistumislinjoilla Algoritminen bioinformatiikka sekä Algoritmit, data-analytiikka ja koneoppiminen on suosituksia sivuaineopinnoissa. Millään erikoistumislinjalla ei kuitenkaan vaadita sivuaineopintoja.

3. Muut opinnot

582510 Henkilökohtainen opintosuunnitelma (FM-HOPS), 1 op

582519 Scientific Writing for MSc in Computer Science, 3 op, mikäli opiskelija suorittaa laitoksella vain FM-tutkinnon ja tekee tutkinnon englannin kielellä.

581324 Computing Tools for CS Studies, 1 op, mikäli opiskelija suorittaa laitoksella vain FM-tutkinnon.

Englannin kielen opinnot, 0-4 op, laitoksella hyväksyttävän henkilökohtaisen opintosuunnitelman (FM-HOPS) mukaisesti.

584403 Syventävä harjoittelu tai työelämään orientoivat opinnot, vähintään 2 op
582511 Tietotekniikka-alan vaativissa ammattitehtävissä työskentely, 2-6 op
tai
582516 Software Factory Work Experience, 5-7 op
tai
582515 Tietotekniikka-ala ammattina, 2 op
tai
582613 Advanced Internship in Bioinformatics, 1-3 op

Vapaasti valittavia opintoja niin, että tutkinnon laajuus 120 op täyttyy. Nämä voivat olla pääaineen tai sivuaineen tai muita opintoja.

ERIKOISTUMIS-LINJAKOHTAISET TUTKINTOVAATIMUKSET

Algoritminen bioinformatiikka (Algorithmic Bioinformatics)

1. Pääaineopinnot

584329 SYVENTÄVÄT OPINNOT

Pakolliset opinnot, 10 op
582630 Design and Analysis of Algorithms, 5 op
582631 Introduction to Machine Learning, 5 op

Valinnaiset opinnot, 24 op
Vähintään toinen seuraavista:
582483 Biological Sequence Analysis, 5 op ja
582676 Project in Biological Sequence Analysis, 2 op
tai
582715 Algorithms in Molecular Biology, 5 op ja
582716 Project in Algorithms in Molecular Biology, 2 op

Bioinformatiikan syventäviä opintoja 17 op laitoksella hyväksyttävän henkilökohtaisen opintosuunnitelman (FM-HOPS) mukaisesti.
Seminaareja, 6 op

50131 Pro gradu -tutkielma, 40 op
50039 Kypsyysnäyte, 0 op

31

Appendix B The ACM and IEEE Computer Science curricula

The ACM and IEEE Computer Science Curricula are joint ventures by the aforementioned organizations aiming to provide generally applicable guidelines for computer science education in tertiary education[7, 20]. These curricula function in a guiding role, helping universities in building and structuring undergraduate programmes in a way that covers all of the topics considered essential. The curricula contain course structure ideas, topic implementation guidance and pedagogical discussions related to educational viewpoints in specific topics. With regular updates, content reviews and specialist panels, the curriculum versions keep evolving to address current topics and to cover emerging fields. The Computer Science Curricula are subset of a wider Computing Curricula, which cover fields such as Computer Engineering, Cybersecurity, Information Systems, Information Technology and Data Science[11]. While certain educational and procedural topics are shared between different fields, the field specific curricula are self contained units that evolve based on both the prior versions and latest developments in the fields. Thus, the latest complete Computer Science Curriculum release version, Computer Science Curriculum 2013 (*CS2013*) is based on Computer Science Curriculum 2008 (*CS2008*), which in turn is based on Computer Science Curriculum 2001 (*CS2001*)[7, 20]. In this thesis the focus is on CS2008 and CS2013, which are the two newest complete curricula. More specifically, the focus is on the educational content as the scope of this thesis does not support a larger and more multidisciplinary research into contentual and pedagogical differences between the localized versions and the guiding principles.

The CS2008 consists of 14 principles that guide the work, developments in the field of computing, changes to the curriculum since the last version and thoughts on specific topics that were of concern in 2008. The guiding principles evolved from similar principles outlined in CS2001,with the principles specified as follows[7].

On Computing

1. Computing is a broad field that extends well beyond the boundaries of computer science.

2. Like CS2001, CS2008 should seek to identify the fundamental skills and knowledge that all computing students must possess.

3. CS2008 must continue to strive to be internal in scope.

4. CS2008 must include updated professional practice as an integral component of the undergraduate curriculum.

5.The Interim Review needs to present the Computer Science discipline in as positive a light as possible, as seen by current applicants

On Computer Science

6. Computer science continues to to draw its foundations from a wide variety of disciplines.

7. Computer science education, moreover, must seek to prepare students for lifelong learning that will enable them to move beyond today's technology to meet the challenges of the future.

8. The required body of knowledge, i.e. the core, should be updated from CS2001 to reflect changes in the discipline.

9. The development of CS2008 must remain broadly based.

10. The Interim Review Task Force must take into account relevant feedback from industry and seek to address their current needs.

Course design and implementation

11. CS2008 should follow the style of CS2001 and go beyond knowledge units to offer significant guidance in terms of individual course design in particular areas.

12. CS2008 must include discussions of strategies and tactics for implementation along with high-level recommendations.

13. It has taken the view that the crisis should be addressed mainly during pedagogical considerations.

The Review Process

14. The rapid evolution of computer science requires an ongoing review of the corresponding curriculum.

In addition to guiding principles, the CS2008 outlines knowledge areas security topics, concurrent computing and net-centric computing as emerging areas of special concern[7]. These areas were not properly addressed in CS2001 and are one of the major contentual evolutions between CS2001 and CS2008. CS2008 outlines 14 major knowledge areas that are relevant to undergraduate level studies. The knowledge areas are outlined in table

B.1. While the knowledge areas are identical to the ones presented in CS2001, the content and focus of these areas shifted as per requirements posed by the areas of special concern and developments in the field[7]. From a curriculum design and adaptation standpoint, each of these knowledge areas does not need to have a specific single area of knowledge type course tied to it. CS2008 contains a set of core knowledge from various knowledge areas that covers around 280 hours of instruction time. A minimum core coverage time in instructional hours to be devoted to certain topics inside the knowledge areas is also given. The core topics are split into compulsory and elective topics. A single course can cover multiple knowledge areas and multiple topics, as long as the learning objectives of each topic are achieved. It is explicitly noted that the CS2008 only forms a guideline of content and universities should adapt the curriculum to fit their course structures[7].

CS2008 knowledge areas	
Discrete Structures (DS)	Human-Computer Interaction (HC)
Programming Fundamentals (PF)	Graphics and Visual Computing (GV)
Algorithms and Complexity (AL)	Intelligent Systems (IS)
Architecture and Organization (AR)	Information Management (IM)
Operating Systems (OS)	Social and Professional Issues (SP)
Net-Centric Computing (NC)	Software Engineering(SE)
Programming Languages (PL)	Computational Science (CN)

Table B.1: CS2008 knowledge areas

Additionally, CS2008 gives the characteristics that a successful graduate should possess. These characteristics are *system level perspective, appreciation of the interplay between theory and practice, familiarity with common themes and principles, significant project experience, attention to rigorous thinking and adaptability.* In addition to these characteristics, the graduate should also possess *cognitive capabilities and skills relating to computer science*, such as modeling, requirements management, understanding of computational thinking, critical evaluation, testing and professional responsibility. The graduate should also possess *practical capabilities and skills relating to computer science*, such as design and implementation, evaluation, information management, human-computer interaction and risk assessment. In addition, as *additional transferable skills*, the graduate should be versed in communication skills, teamwork, self management and professional development.[7]

Like CS2008, the newest iteration of computer science curricula - the CS2013, consists of principles that guided the work, developments in the field of computing, changes to the curriculum and thoughts on specific topics of special concern[20]. CS2013 contains major overhauls in content and a more flexible onlook towards course and curriculum structure. The role of mathematics in computer science studies is briefly covered. The general standpoint is that mathematics is an integral part of many computer science topics and that mathematical studies are recommended in addition to computer science studies. Institutions should be mindful of the prerequisites for mathematical studies, as computer science studies still need to be accessible for students without extensive backgrounds in mathematics[20]. CS2013 also briefly covers MOOCs as a course and learning model, which is especially relevant in comparisons with the University Of Helsinki CS curriculum from 2017 onwards outlined in section 2.5.

The high level themes that guided the development of CS2013 were *a big tent view of CS* - meaning that computer science studies should be modeled to be multi-disciplinary, *managing the size of the curriculum* - meaning that while the field of computer science is expanding the curriculum should stay in a manageable content size, *actual course exemplars* - adding exemplary courses structures from various higher education institutions to help other institutions in course and curriculum building and lastly *institutional needs* - meaning that the content of CS2013 should be explicitly flexible to suit various localized needs.[20]. The main principles in CS2013 are quite similar to the principles in CS2008. While the more general computing principles have been dropped in favor of principles more closely related to computer science as a field, the general theme of providing students with the right skills and knowledge to succeed in the field after graduation and facilitating lifelong learning with a multi-disciplinary point of view are present[7, 20]. The principles of CS2013 are as follows[20].

> *1. Computer science curricula should be designed to provide students with the flexibility to work across many disciplines.*
>
> *2. Computer science curricula should be designed to prepare graduates for a variety of professions, attracting the full range of talent to the field.*
>
> *3. CS2013 should provide guidance for the expected level of mastery of topics by graduates.*
>
> *4. CS2013 must provide realistic, adoptable recommendations that provide guidance and flexibility, allowing curricular designs that are innovative and track recent developments in the field.*

5. The CS2013 guidelines must be relevant to a variety of institutions.

6. The size of the essential knowledge must be managed.

7. Computer science curricula should be designed to prepare graduates to succeed in a rapidly changing field.

8. CS2013 should identify the fundamental skills and knowledge that all computer science graduates should possess while providing the greatest flexibility in selecting topics.

9. CS2013 should provide the greatest flexibility in organizing topics into courses and curricula.

10. The development and review of CS2013 must be broadly based.

CS2013 expands the knowledge areas from CS2001/CS2008 shown in table B.1. Four new knowledge areas were added and the existing ones received in some cases major overhauls to make them more relevant to modern practices[20]. Knowledge units were also moved between knowledge areas to form more coherent bodies of content. The knowledge areas in CS2013 are presented in table B.2. Like in CS2008, the knowledge areas contain core skills and topics that each student should learn and how many instructional hours should be dedicated to learning these skills[20]. These topics are called knowledge units. However, CS2013 splits knowledge units into tier 1 and tier 2 level skills, with tier 1 containing the required topics in the units and tier 2 the recommended but elective parts of the units. The change was introduced to allow for more flexibility in the localized curricula and to move away from a "every student needs to know everything" mentality[20]. The amount of instructional hours increased slightly with the change, going up to 308 hours in combined tier 1 and tier 2 studies, compared to 280 hours in CS2008. Another change introduced is how introductory courses are structured. CS2013 allows for a more flexible introductory course structure to accommodate students with various backgrounds, prior computing experience and to better suit non-major students. This change allows departments to be more flexible in how they design curricula in relation to local needs[20].

Like CS2008, CS2013 provides a list of desired qualities for a graduate. These qualities are *technical understanding of computer science, familiarity with common themes, appreciation of the interplay between theory and practice, system-level perspective, problem solving skills, project experience, commitment to life-long learning, commitment to professional responsibility, communication and organizational skills, awareness of the broad applicability of computing* and *appreciation of domain-specific knowledge*[20]. These characteristics

CS2013 knowledge areas	
Algorithms and Complexity (AL)	Architecture and Organization (AR)
Computational Science (CN)	Discrete Structures (DS)
Graphics and Visualization (GV)	Human-Computer Interaction (HCI)
Information Assurance and Security (IAS)	Information Management (IM)
Intelligent Systems (IS)	Networking and Communications (NC)
Operating Systems (OS)	Platform-based Development (PBD)
Parallel and Distributed Computing (PD)	Programming Languages (PL)
Software Development Fundamentals (SDF)	Software Engineering (SE)
Systems Fundamentals (SF)	Social Issues and Professional Practice (SP)

Table B.2: CS2013 knowledge areas

are expanded from the ones presented in CS2001 and CS2008. The qualities in CS2013 are more abstract and focus more on broader tasks that the graduate should be able to accomplish rather than the more specific singular skills in the previous curriculum versions. This change reflects the change from a rigid to flexible curricula and course design that CS2013 aimed to achieve.