Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

DISSERTATION

Student Performance and Collaboration in Introductory Courses to Theory of Computation

Dissertation with the aim of achieving the doctoral degree

Dr. rer. nat.

at the Faculty of Mathematics, Informatics and Natural Sciences

Department Informatics of Universität Hamburg

submitted by

Christiane Frede, M.Sc.

2021 in Hamburg

Assessors:

1. Univ.-Prof. Dr. Maria Knobelsdorf, Universität Wien
2. Prof. Dr. Peter Kling, Universität Hamburg

Day of the oral defense: February 10, 2022

This thesis is dedicated to my mother

Acknowledgement

First of all, I would like to thank my supervisor Prof. Dr. Maria Knobelsdorf from the Department of Computer Science at the Universität Wien for her guidance during my research. Her expertise and new impulses were irreplaceable. In the process, she gave me the opportunity to develop as a scientist and to grow beyond myself. I am very grateful that I had the opportunity to work in this field and to gain valuable new experiences.

Furthermore, I would like to thank Prof. Dr. Peter Kling, from the Department of Informatics at the Universität Hamburg. He accepted me into his working group and was open to my research. He always gave me the opportunity to give presentations and gave me helpful feedback and encouragement. I am very grateful that he trusted me to work independently and shared these years with me.

At the same time, I would also like to thank all those who have actively enabled and supported me in my studies: In particular, I want to thank Prof. Dr. Uwe Nestmann, Prof. Dr. Kirstin Peters, Tobias Prehn, Christine Rickmann, and Arno Wilhelm-Weidner at the Chair of Models and Theory of Distributed Systems of Technische Universität Berlin. They made it possible for me to conduct my studies, supported me over the years, and have always brightened up my time in Berlin. I want to thank all other teaching assistants and students for supporting and participating in my studies and sharing their time with me. I want to thank Prof. Dr. Hubert Knoblauch and Theresa Vollmer at the Chair of General Sociology at Technische Unversität Berlin for the valuable meetings in which they have interpreted my data with me and provided me a new perspective on my research. I want to thank the PhD students of Didaktik der Informatik and my colleagues at the working groups Computer Science Education and Theory of Efficient Algorithms at Universität Hamburg for their valuable feedback and encouragement over the years. I want to thank my friends Jenny Gabel, Felix Kiehn, Sandra Schulz, and Christoph Stanik for repeatedly discussing my research and providing feedback and encouragement. And I want to thank my fiancé Heiko Bornholdt for traveling this journey with me.

Last but not least, I would like to thank the support and great love of my friends and family who have accompanied me throughout the years. I would like to especially thank Ann-Kathrin Drews and Wiebke Langebeck for their unconditional friendship. Without all of you, this achievement would not have been possible. Thank you.

Abstract

In tertiary computer science education, computer science undergraduate programs usually include one or two compulsory courses in theory of computation. Although computer science curriculum recommendations indicate that theory of computation courses are a highly relevant part of computer science undergraduate programs, the courses suffer from high failure rates, and only a minority of students perform well. Several pedagogical approaches have been introduced in the last decade to address the problem and improve the situation in theory of computation courses. These approaches offer elaborated pedagogical solutions for engaging computer science students for theory of computation and lowering attrition and failure rates. Most of the existing approaches were developed with the assumption that students' difficulties with theory of computation are mainly caused by a lack of interest, motivation, or ability to understand the relevant concepts and theorems due to the abstract and formal nature of computation. Thereby, the assumptions are often based on occasional oral feedback or surveys with given answers which were conducted after the courses. This leads to the fact that none of the assumed student difficulties have been empirically validated in ways that would inform pedagogical considerations by detailed insights about the nature of students' actual difficulties.

In the present work, I have undertaken a detailed investigation of the difficulties of students with theory of computation. Thereby, I provide more sustained information than the general assumptions on which current pedagogy has been based. In more detail, I conducted two studies: (1) a quantitative study within an introductory course about Formal Languages and Automata to investigate the student performance in all assignments and topics covered, and (2) a qualitative study to explore students' difficulties in assignments selected based on the results of the quantitative study.

Using an exploratory data analysis approach and a one-way analysis of variance, I analyzed the final exam and homework performance of about 1500 students over

three consecutive years. The results show that all students perform low on almost all proof assignments, regardless of their final exam grades. While students performed worst in the final exam on an assignment that required a formal proof using the pumping lemma, performance on a similar homework assignment was not as low. Furthermore, I detected how one assignment of the first year of analysis had a significantly lower performance in the following years after a sub-task was added that required proof development. The results underline that students have most difficulties with formal proof assignments and add that this can affect students regardless of their final exam grade.

Based on the performance discrepancy between pumping lemma final exams and homework assignments, I conducted a qualitative study. Using a videography and a video interaction analysis, I observed three student groups working on two pumping lemma homework assignments. Thereby, I came to the following conclusions: Students have the same difficulties on the pumping lemma assignments in final exams and homework. However, when it comes to homework, students usually solve the problems while working together, so performance on homework solutions tends to be higher than on individual final exams. Nevertheless, through an analysis of student interactions, I found that there is a particular distribution of roles in the groups. Generally, one student acts as an explaining teacher, one as a questioning student, while all other students hardly participate in the group work, regardless of the group size. One possible explanation for this type of distribution lies in how students externalize and internalize their knowledge. They focus heavily on the tutor session and their tutor's explanations and use the sample solutions they receive online for various assignments as patterns for their own solutions. The overall study gives the impression that students in group work are trying to achieve a result that will earn them as many points as possible with the tutor, rather than really internalizing and learning the topics.

Through an extensive quantitative study and a detailed qualitative study, the present work offers new insights and explanations for the low performance and high failure rates in theory of computation courses. The findings offer starting points for changing the pedagogical design to improve the poor situation in theory of computation courses. In addition to teaching proof skills, special attention needs to be paid to collaborative teaching-learning situations.

Kurzfassung

Die Bachelorstudiengänge der Informatik umfassen in der Regel ein oder zwei verpflichtende Kurse aus dem Bereich der Theoretischen Informatik. Obwohl aktuelle Lehrpläne zeigen, dass diese Kurse eine hohe Relevanz für ein Informatikstudium haben, gelingt es nur einer geringen Anzahl der Studierenden, eine gute Leistung zu erbringen. In den letzten Jahren wurden mehrere didaktische Ansätze erprobt, um die Leistung der Studierenden zu verbessern und die Abbruchquoten zu senken. Die meisten dieser Ansätze wurden unter den Annahmen entwickelt, dass die Schwierigkeit der Studierenden mit den Themen der Theoretischen Informatik hauptsächlich auf mangelnde und Motivation oder die fehlenden Fähigkeit zurückzuführen sind, die relevanten Konzepte und Theoreme aufgrund der abstrakten und formalen Natur zu verstehen. Diese Annahmen beruhen dabei oft auf subjektiver Lehrerfahrung oder Umfragen mit vorgegebenen Antworten, die nach den Kursen durchgeführt werden. Bisher wurde keine der Annahmen empirisch in einer Weise validiert, die didaktischen Überlegungen durch detaillierte Einblicke in die Art der tatsächlichen Schwierigkeiten der Studierenden untermauern würde. In der vorliegenden Arbeit habe ich eine detaillierte Untersuchung der Schwierigkeiten von Studierenden mit der Theoretischen Informatik vorgenommen. Dafür habe ich zwei Studien durchgeführt: (1) eine quantitative Studie im Rahmen eines Einführungskurses über Formale Sprachen und Automaten, um die Leistung der Studierenden bei allen Aufgaben und behandelten Themen dieses Kurses zu untersuchen, und (2) eine qualitative Studie, um die Schwierigkeiten der Studierenden mit Aufgaben zu untersuchen, die basierend auf der quantitativen Studie ausgewählt wurden.

In der quantitativen Studie habe ich eine Explorative Datenanalyse und eine einfaktorielle Varianzanalyse verwendet, um die Abschlussprüfungen und Hausaufgaben in drei aufeinanderfolgenden Jahren von insgesamt etwa 1500 Studierenden zu analysieren. Die Ergebnisse zeigten, dass die Studierenden unabhängig von ihrer Note in der Abschlussprüfung bei fast allen Beweisaufgaben durchschnittlich schlechter abschneiden als bei den anderen Aufgaben. In der Ab-

schlussprüfung erwartete die Aufgabe mit der niedrigsten Leistung, die Entwicklung eins formalen Beweisens unter Verwendung des Pumping Lemmas. Die Leistung bei ähnlichen Hausaufgaben war hingegen deutlich höher. Ich konnte auch feststellen, dass eine Aufgabe aus dem ersten Jahr der Analyse in den beiden darauffolgenden Jahren deutlich schlechter ausfiel, nachdem eine Teilaufgabe hinzugefügt worden war, welche die Entwicklung eines Beweises erforderte. Die Ergebnisse unterstreichen, dass Studierende in Einführungskursen der Theoretischen Informatik die größten Schwierigkeiten mit formalen Beweisaufgaben haben. Außerdem zeigen die Ergebnisse auch, dass dies für alle Studierenden unabhängig von ihrer Note in der Abschlussprüfung zutreffen kann.

Ausgehend von der Leistungsdiskrepanz in der Pumping Lemma-Aufgabe zwischen Abschlussprüfung und Hausaufgabe, habe ich eine qualitative Studie durchgeführt. Mit Videografie und einer Videointeraktionsanalyse habe ich Studierendengruppen bei der Bearbeitung ihrer Hausaufgaben beobachtet. Die Ergebnisse zeigen, dass die Studierenden in der Abschlussprüfung und den Hausaufgaben bei ähnlichen Pumping Lemma-Aufgaben die gleichen Schwierigkeiten haben. Jedoch lösen die Studierenden ihre Probleme bei den Hausaufgaben meist gemeinsam, so dass die Hausaufgabenlösung besser ausfällt als die nachfolgenden Einzellösungen in den Abschlussprüfungen. Bei einer weitergehenden Analyse der Interaktion in den Gruppen habe ich außerdem festgestellt, dass es eine besondere Rollenverteilung gibt. In der Regel fungiert ein:e Studierende:r als erklärende Lehrperson und ein:e Studierende:r als fragende:r Schüler:in, während alle weiteren Studierenden sich kaum an der Gruppenarbeit beteiligen. Eine mögliche Erklärung für diese Art der Verteilung lässt sich in der Art und Weise finden, wie die Studierenden ihr Wissen externalisieren und internalisieren. Sie konzentrieren sich stark auf den Inhalt der Tutorien, die Erklärungen der Tutor:innen und verwenden Musterlösungen für ähnliche Aufgabe als Vorlage für ihre eigenenen Lösungen. Die gesamte Studie, erweckt den Eindruck, dass die Studierenden in der Gruppenarbeit eher versuchen, möglichst viele Punkte mit ihrer Lösung zu erzielen, anstatt die Themen zu verinnerlichen und zu lernen.

Anhand der umfangreichen quantitativen Studie und der detaillierten qualitativen Studie bietet die vorliegende Arbeit neue Einblicke und Erklärungen für die geringen Leistungen und hohen Durchfallquoten in Einführungskursen der Theoretischen Informatik. Die vorgestellten Erkenntnisse bieten Ansatzpunkte für eine Veränderung der didaktischen Gestaltung, um diese Situation zu verbessern. Dabei muss neben der Lehre der Beweisführungskompetenzen auch die kollaborative Lehr-Lern-Situation betrachtet werden.

Contents

1. Introduction

In tertiary Computer Science (CS) education, CS undergraduate programs usually include one or two compulsory courses in Theory of Computation (ToC). These courses and the corresponding introductory literature cover topics and concepts of automata theory, regular languages, grammar, logic, formal proofs, undecidability (e.g., [HMU01] [Sip12]) as well as algorithms, functions, sorting and order statistics, data structures, and complexity (e.g., [CLRS09]).

Although CS curriculum recommendations indicate that ToC courses and subjects are essential in CS majors (e.g., [JTFoCCS13] [Zuk16]), only a minority of students manage to complete the entire courses well. At German universities, it is not uncommon for pass rates to be less than 50%. Thus, many students either drop out early or fail final exams, which can even lead to students abandoning CS as a major. As I show in the following chapters, several pedagogical approaches have been introduced in the last decade to address the problem and improve the situation in ToC courses. Still, most of the approaches build on assumptions that have not yet been sufficiently empirically validated until now. Furthermore, the situation has not improved significantly so far, which can be seen from the fact that the failure rates remain high.

In 2015, I started to work on students' difficulties in ToC courses as part of my master's thesis. Based on existing approaches and ideas how to improve the situation in ToC topics, I wanted to provide a deeper insight into students' work processes rather than relying on undefined or non-empirically validated origins, such as delayed oral feedback, final written results, or individual teacher experiences (e.g., [CnGM04] [Ham04] [KAPG07]). Therefore, I conducted an observational study explicitly focused on analyzing students working in a group setting. Students were asked to develop a proof about NP-completeness, as the subjective teaching experience of several colleagues led me to suspect that students have difficulties with this topic. In the study conducted, I observed three groups of second-year students focusing on their usage of mathematical descriptions as they developed the required proof.

The results showed that the students had no structure in their work process and were unsure of what they had to prove and what was already given. Moreover, they understood the process of writing down only as a result of their work and hardly used the opportunity to record information for all to see. In addition, however, the students also indicated how interesting and exciting the assignment was, thus contradicting the common assumption in related work in the field that students lack interest in the subject matter or skills to understand the body of knowledge of the subject. The implication of this finding is that ToC-education research may need to be reoriented towards a pedagogy that focuses on teaching students how to practice ToC, alongside introducing them to the factual knowledge of the discipline. Based on the results also published in [KF16], I found that processes by which students work were a previously neglected source for valuable insights continuing student-oriented pedagogy fostering working proficiency in this field of education. As a basic step in addressing this, I found it essential to first examine in detail the performance of the students to understand if there are patterns here that warrant a more detailed examination. In the following, I explain which two studies I conducted during my dissertation project and why it was undertaken.

1.1. Motivation for the Quantitative Study

In discussing the results and implications of my master's thesis, I was often confronted by teachers with similar statements: for example, that since ToC topics are notoriously difficult to learn, it is not a surprising or unusual situation that some students can understand the topics and others cannot. Although it has never been rigorously demonstrated, this situation seems to contribute strongly to the impression that student performance in ToC courses is bimodal, reinforcing the "geek gene" belief in this area of CS [Lis11].

Patitsas et al. (2016) have questioned the prevailing belief that high performance among CS students is based on natural predispositions [PBCE19]. For that matter, the authors have provided strong evidence that CS students' final grades are not bimodal, an argument often used to support the "geek gene" hypothesis. They argued that the alleged bi-modality in student performance (like the "geek-gene" hypothesis) might rather be used as a social defense by CS faculty since "it is easier for computer science educators to blame innate qualities of their students for a lack of learning than it is for the educators to come to terms with the ineffectiveness of their teaching." [PBCE19, p. 120]. Students themselves can rely on the "geek-gene" hypothesis as an explanation of their performance in CS [SK07, p. 34], or they can distance themselves from their poor performance, accepting it as a matter of predisposition rather than questioning it [Dwe14].

For ToC teachers, it is probably no surprise that CS students might find challenges with the formal and abstract nature of ToC, and have difficulties understanding ToC concepts in general or lack motivation and interest in this field. However, it remains open whether all these aspects apply to all CS students (or, for example, just those failing final exams) and how strongly these reasons might impact students' overall performance. It is tempting to assume that students who did well in final exams have also gained an overall understanding of the entire course content, while students scoring low in final exams suffer from cognitive and motivational deficiencies. Such conclusions from one final exam as only data source are not necessarily an effective indicator, though, as it sums up an entire spectrum of domain-specific competences of ToC. Also, the extent to which high or low scores reflect students' overall high or low performance has not yet been sufficiently taken into account in research studies about ToC.

The first half of my doctoral thesis aims to provide a differentiated picture of student performance in ToC courses. Therefore, I conducted an extensive exploratory data analysis to contribute valuable insights into student performance in an introductory ToC course on one hand, and to challenge teacher beliefs of bimodality in this field on the other. For that reason, I chose to work with the same kind of data available to ToC teachers every year: homework and final exam assignments. I analyzed data from students attending an introductory ToC course at Technische Universität Berlin (Germany) in 2016 and validated the results with data from two more cohorts of 2017 and 2018 (over 1500 students in total). The results were partly published in [FK18] and [FK21].

1.2. Motivation for the Qualitative Study

A quantitative analysis can provide interesting insights into the assignments and topics that students perform low on final exams and homework assignments. However, during my analysis, it became apparent that performance in homework and final exams can differ even for similar assignments [FK18] [FK21]. The reason for the difference is presumably the pedagogical design of ToC courses, which I, therefore, examine more closely and include in my second study.

The pedagogical approach in ToC courses traditionally involves homework assignments that students are expected to complete collaboratively in self-organized small learning groups [KKB14, pp. 68 – 69]. Although this group work in the form of study groups working on homework is common in introductory ToC courses, students are required to take their final exams individually.

Overall, homework groups are not the only form of collaborative learning setting in CS courses. In general, such settings are strongly favored in different CS courses – for example, pair programming in programming courses (e.g., [BEW08] [MWBF06] [RPB17]), peer discussions in programming and computer architecture courses (e.g., [HA20] [PBLS13] [Zin14]), or computer-supported collaboration in software engineering courses (e.g., [KIP13]). Collaboration, therefore, is a significant element of CS pedagogy, and has motivated a considerable amount of research into improving student learning in CS. For example, Drury and Kay (2003) analyzed group work in undergraduate CS and reported that most students felt that they had "learned to learn independently" [DKL03, p. 83]. Moreover, Porter et al. (2010) and Zingaro (2014) reported that peer instruction led to reduced failure rates and increased final exam scores in different courses (e.g., CS1, computer architecture) [PBLS13] [Zin14]. Furthermore, McDowell et al. (2006) found that their analyzed student pairs in an introductory programming course produced higher quality programs, and the students were more confident in their work and had a higher enjoyment [MWBF06].

The results of my quantitative study could not – and did not aim to – provide insights into whether the students in ToC courses necessarily reach the same amount and quality of individual understanding as the quality of the homework group's results may imply. The reason for the assumption is, on the one hand, that a considerable number of students either drop out of ToC courses early or fail their individual final exams. On the other hand, only a minority of students manage to perform well throughout the entire course. Therefore, it seems likely that the students in homework groups contribute to a shared group knowledge beyond their own understanding and create a solution to a problem or a task that they would not have achieved on their own [Ros92].

Consequently, in contrast to the perceived benefits implied by research about group work in CS education, I question whether group work displays its full potential or rather provides reasons as to why many students cannot perform better or even fail in the final exam. To focus on this part of ToC courses, I conducted a second and qualitative study and analyzed how student groups work on assignments I selected based on the results of the quantitative study. First, I provide an analysis of students' pitfalls and challenges with the chosen assignments and topics. Second, I analyzed how students interact while working together on the assignments. Thereby, I did not only consider the use of mathematical descriptions as in my master's thesis, but I explicitly considered the collaborative aspect. Furthermore, I examined what influence student interaction has on the learning of the students in the group.

1.3. Thesis Content

The thesis consists of four parts:

I **Topic and Current State of Research**

a) Chapter 2 provides an overview of the historical roots of computer science and its development as an academic field. After illustrating the role of theory of computation in early and current computer science curricula, I present the traditional pedagogical structure of German Theory of Computation introductory courses and give a thematic example of lecture content and assignments.

b) Chapter 3 presents research concerning how to address low performance and high failure rates in Theory of Computation courses. First, I present approaches that have modified and extended existing courses and content. Second, I focus on software, systems, and tools developed to teach the concepts and topics interactively. Third, I present existing case studies which analyze actual students' difficulties with specific topics and assignments. Fourth, I give some insights into similar research in mathematics education.

II **Student Performance in Final Exams and Homework**

a) Chapter 4 explains the necessary background for the quantitative study as well as the research design. I begin by motivating the research questions before presenting the exploratory data analysis, the qualitative content analysis, and all the statistical analyses carried out. After presenting the data sources, I explain how the data analysis was done.

b) Chapter 5 presents the results and discussion of the quantitative study, which form the basis for the second part of my dissertation.

III **Collaboration and Learning within Student Groups**

a) Chapter 6 introduces the theoretical background for the second study. I provide definitions for the terms "collaboration" and "interaction" and explain how I used distributed cognition theory to define how students learn.

b) Chapter 7 presents the overall study design. First, the research questions are motivated before background information about the research methods of videography and video interaction analysis are presented. Next, the setup and data collection are described before information about the study participants is given.

c) Chapter 8 presents the results and discussion of the first data analysis about students' pitfalls and challenges with the related topics and assignments.

d) Chapter 9 presents the results and discussion of the second data analysis about students' interactions within their homework groups and students' externalization and internalizing of knowledge.

IV **Concluding Review**

a) Chapter 10 provides a summary of the conducted data analyses. Furthermore, I illustrate the scientific contribution of this doctoral thesis and present suggestions for future work.

Part I.

Topic and Current State of Research

2. Theory of Computation

The studies conducted in the present doctoral project were situated in courses of Theory of Computation (ToC). To understand the importance of ToC as part of Computer Science (CS) education, I begin the section with insights into the historical roots of computing followed by more details on the development of CS as an academic discipline. Next, I examine the role ToC occupied in CS curricula. I conclude the chapter by providing insight into how the historical roots also influence the way ToC is taught today. To do this, I illustrate the typical components of introductory ToC courses as well as an exemplary topic with example assignments.

2.1. Three Traditions of Computing

In the following section, I summarize the historical roots of CS as a combination of the paradigms of mathematics, engineering, and science. In the context of this doctoral project, is is particularly relevant how these paradigms have shaped the development of CS education. Therefore, I present insights into the development of CS as an academic discipline.

2.1.1. Historical Roots of Computer Science

Tedre and Sutinen (2008) state that "[e]ducators in the computing fields are often familiar with the characterization of computing as a combination of theoretical, scientific, and engineering traditions." [TS08, p. 153]. They see the origin of this tripartite reported by the Task Force on the Core of Computer Science stating that "the task force characterized the discipline of computing [...] rely[ing] on three different intellectual traditions (the task force called them paradigms): the mathematical (or analytical, theoretical, or formalist) tradition, the scientific (or empirical) tradition, and the engineering (or technological) tradition." [TS08, p. 153]. The authors further argue that this tripartite leads to the situation that when "the three traditions of computing are based on different principles, they have different aims, they employ different methods, and their products are very different." [TS08, p. 153].

In the following section, I provide the historical roots of the three traditions and their influence on the development of CS as an academic field. Figure 2.1 summarizes the main representatives, their life spans, and selected key developments on the way to the development of CS from 1800 to 2000.

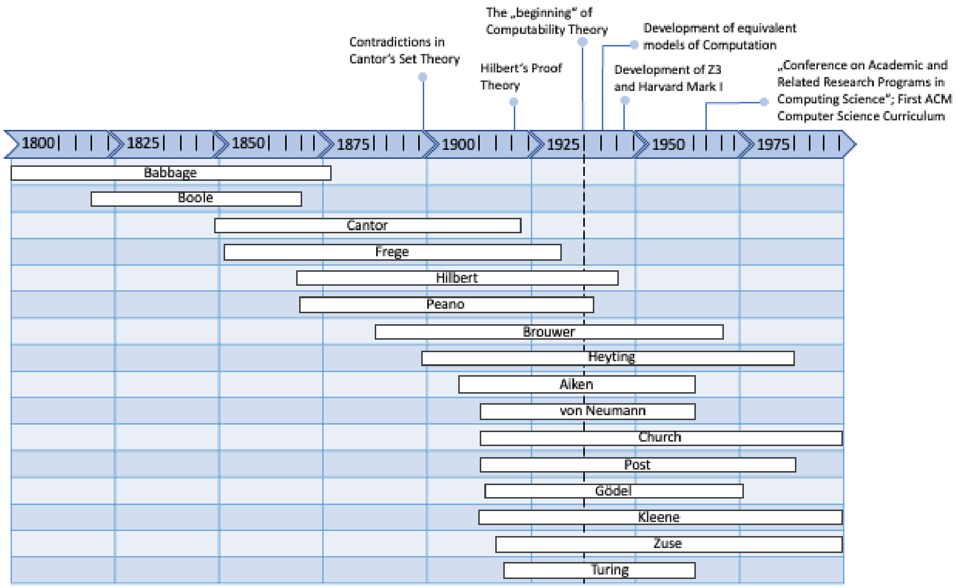

Figure 2.1.: Timeline from the years 1800 to 2000. The lower part illustrates the life spans of a number of representatives associated with the development of computer science. The upper part gives key developments of the historical roots of computer science.

Mathematics

Considering mathematics as a tradition of CS, Tedre and Sutinen (2008) describe how "it has been argued that a mathematical reductionist could say, somewhat facetiously, that the discipline of computing is nothing but a paradigm change in mathematics." [TS08, p. 154]. They outline how one of the characterizing features of the 1970s and 1980s in computing disciplines was a so-called "formal verification debate." They state that the debate between proponents and opponents was mainly about the question of whether a "formal verification can be used to prove that a computer system works correctly" [ibid., p. 155]. They summarize that even if "[c]omputers, the machines, are physical objects and although one could prove computer blueprints to be theoretically correct, the physical world does not work with mathematical certainty," and "[w]henever computers as physical machinery are in the picture pure mathematics turns out to be inadequate, and some other intellectual frameworks must be utilized" [ibid., p. 156].

In the following, I provide some of the mathematical foundations fostering the development of CS as a discipline. According to various authors, one of Cantor's[1] set theories was one of the starting points for discussing the non-contradiction of mathematics (e.g., [Hei91, p. 31] [Rob15, p. 19] [Bau96, p. 62]). Heintz (1991) states that although the discussion was already partly known, this contradiction eventually led to the development and linkage of three directions aimed at developing a basis for proving the non-contradiction of mathematics [Hei91, p. 19]:

1. **Logicism:** Logicism aimed to base mathematics on pure logic. Robič (2015) illustrates that even before Cantor's set theory, Boole[2] had attempted to express logical statements by algebraic expressions containing the operations "and," "or," and "not" and to use them for logical deduction. He clarifies how these considerations were later to lead to Propositional Logic. Building on Boole's considerations, Frege[3] and Peano[4] introduced quantified variables and another alphabet of symbols, laying the foundation for First-Order Logic. [Rob15, p. 23].

[1] Georg Cantor, German mathematician, 1845 – 1918
[2] George Boole, English mathematician, philosopher, and logician, 1815 – 1864
[3] Gottlob Frege, German philosopher, logician, and mathematician, 1848 – 1925
[4] Giuseppe Peano, Italian mathematician, 1858 – 1932

2. **Intuitionism:** Brouwer[5] and Heyting[6] mainly advocated intuitionistic mathematics. Heintz (1991) describes how Brouwer and Heyting did not intend mathematics to be preserved merely by placing it on a new, non-contradictory foundation. Instead, they wanted to re-found mathematics based on a fundamental analysis of the nature of mathematical thinking [Hei91, p. 35]. Robič (2015) and Heintz (1991) explain how, in this process, large areas of mathematics were eventually discarded because they could not be reconstructed according to intuitionistic principles [Rob15, p. 20]; [Hei91, p. 36].

3. **Formalism:** Heintz (1991) and Baumann (1996) describe how formalistic mathematics no longer focused on interpretation and meaning (i.e., semantics), but only on structure (i.e., syntax), – that is, there was to be no reference to meaning outside the mathematical system [Hei91, p. 16]; [Bau96, p. 63]. They describe how formalism attempted to preserve all of classical mathematics. One of the best-known representatives was Hilbert[7] [Rob15, p. 26].

Heintz (1991) describes how Hilbert used the ideas and results of the other two directions to formulate his idea of proof theory within formalistic mathematics. With this theory, he wanted to prove the non-contradiction of mathematics by purely mathematical means. Finally, it was Gödel[8] in 1931 who realized that the non-contradiction could be proved only by showing that no contradiction arises, instead of proving the contradiction itself [Göd31] [Hei91, p. 47]. Robič (2015) describes how Gödel thereby challenged the previous assumption that there is always "true" or "false" and showed that there are statements that can neither be proved nor disproved by logical methods [Rob15, p. 77].

According to different authors, the further development of computing based on Gödel's work continued in the 1930s, primarily through Kleene[9] [Kle36], Church[10] [Chu36], Turing[11] [Tur37], and Post[12] [Pos36] and their ideas for models of

[5] Luitzen Egbertus Jan Brouwer, Dutch mathematician, 1881 – 1966
[6] Arend Heyting, Dutch mathematician, and logician, 1898 – 1980
[7] David Hilbert, German mathematician, 1862 – 1943
[8] Kurt Gödel, Austrian logician, mathematician and philosopher, 1906 – 1978
[9] Stephen Cole Kleene, American mathematician, 1909 – 1994
[10] Alonzo Church, American mathematician, and logician, 1903 – 1995
[11] Alan Turing, English mathematician, computer scientist, logician, philosopher, 1912 – 1954
[12] Emil Post, Polish mathematician, and logician, 1906 – 1978

computation [Bau96, p. 67]; [DM15, p. 1]. Denning and Martell (2015) argue that these models laid the mathematical and formal foundation for answering the question "What is computation?" [DM15, p. 1]. Robič (2015) states how the different models of computations were eventually found to be equivalent, as a computation in anyone could be realized in any other [Rob15, p. 76].

Engineering

According to Tedre and Sutinen's (2008) considerations about the three traditions of computing, "[t]he origins of modern computing lie equally strongly in engineering as they lie in mathematics. Many of the turning points in the history of computing come from technological breakthroughs, not only theoretical breakthroughs". [TS08, p. 161]. They describe how "the engineering character of computing fields relies on the view that the goal of computing is to design and construct useful things (Loui, 1995; Wegner, 1976)." [ibid., p. 162]. Tedre and Sutinen conclude how both traditions were equally necessary so that a CS as known today could develop by stating that "without engineers computing would still be a compartment of mathematics, or that without engineers the theories of computing would be just idle speculation." [ibid., p. 162].

In the following, I give a short overview of some key developments of the engineering tradition of CS. Denning and Martell (2015) emphasize how in the time the different theoretical models of computation were developed, "the terms 'computation' and 'computers' were already in common use in the sense of engineering and mechanization." [DM15, p. 1]. According to Baumann (1996), one of the roots of CS lies in "a historical process in the course of which man delegated more and more functions to technical devices and reserved the higher functions for himself. We refer to this process as automation. Initially, man delegated only physical activities to machines, but as automation progressed, he also delegated sensory and mental activities. Parallel to this development, the techniques of automatic message transmission and the control of machines or processes developed." [Bau96, p. 55].

Baumann (1996) addresses how even in ancient times, there were attempts to facilitate calculation, to mechanize repetitive arithmetic operations, and to store numbers, for example, through the Abacus. [Bau96, p. 69]. Therefore, the computation was understood as the mechanical steps followed to evaluate mathematical functions. He also summarizes how Babbage[13] designed the first program-controlled, automatic calculating machine around 1833. While Babbage could not build such machines during his lifetime, Zuse[14] and Aiken[15] built almost equivalent versions in 1941 and 1944, respectively [Bau96, p. 75].

According to Baumann (1996), it was von Neumann[16] who first stored not only the data internally in the machine but also the programs. Around 1945, he invented the von Neumann architecture and laid the foundation for the way most computers known today work. This general idea goes back to the universal Turing machine [Tur37] [Bau96, p. 78]. Bowen (2018) describes how both models function similarly, but the terminology and purpose are different: "A Turing machine is a theoretical model to aid in reasoning about computation, whereas the von-Neumann architecture is a more practical description of the configuration of a standard electronic digital computer." [Bow18, p. 207]. This connection again demonstrates how the mathematical and engineering traditions used different foci to describe similar developments.

Within the engineering tradition, several authors also emphasize the importance of computer-like machines during World War II. Bowen (2018) describes how Alan Turing's "unique mathematical abilities were recognized during his time at Cambridge and he was invited to join Bletchley Park, the secret centre of the United Kingdom's efforts, to break German codes." [Bow18, p. 203]. He states that since decryption by hand was not the way to succeed in the limited time available, Turing decided to use machines to tackle the problems. Also, the work of Zuse, Aiken, and von Neumann shaped the development of the war.

Science

Tedre and Sutinen (2008) stress that "[i]f one were to call computing a science one should understand the various meanings of the term science as well as the aims, methods, and limitations of science." Therefore, he concludes how "[o]ne should understand the complexity of argumentation, logic, confirmations, concepts, demonstrations, and consensus in the computing disciplines; as well as

[13] Charles Babbage, English mathematician, philosopher, engineer, 1791 – 1871

[14] Konrad Zuse, German engineer, computer scientist, inventor and businessman, 1910 – 1995

[15] Howard Hathaway Aiken, American physicist, and computer scientist, 1900 – 1973

[16] John von Neumann, Hungarian-American mathematician, physicist, computer scientist, and engineer, 1903 – 1957

problems with objectivity and the limits of scientific knowledge. [...] Finally, students of computing should be taught the proper use of the vocabulary of science". [TS08, p. 161]. They describe the difference between the paradigms of science and engineering that "although scientists and engineers both may spend most of their time building and refining their apparatus, the distinction between a scientist and an engineer is that the scientist builds in order to study while the engineer studies in order to build" [TS08, p. 162].

Denning and Martell (2015) describe the development of science as a paradigm for CS. Thereby, they illustrate the debate about whether CS could be accepted as science because its opponents stated that "true science deals with phenomena that occur in nature ("natural processes"), whereas computers are man-made artifacts." [DM15, p. 3]. The authors continue to reproduce that only when computers and CS were not only seen as "a tool for science but also a new method of thought and discovery in science" [ibid., p. 9] it was that scientists "began to acknowledge that natural information processes [...] can be studied with the same methods as the artificial information processes generated by computer". [ibid., p. 9] Hereafter, CS became accepted as "genuine science" around the 1980s [ibid., p. 10].

2.1.2. Computer Science as an Academic Field

Around the 1960s, CS began to be recognized as an academic subject. As Finerman et al. (1968) state, there were already about 100 different universities or four-year colleges with degree programs in CS in 1968 [Fin14, p. 170]. He summarizes how the pure existence of numerous academic programs did not indicate that all have common goals because "some programs are intended to train professional programmers and analysts who will be among the several hundreds of thousands required by industry in coming years. Others aim to train computer designers or systems architects. Still others attempt to educate a select few in the more theoretical aspects of computing science." [ibid., p. 197]. These different objectives of the programs already showed the tripartite division of CS.

The roots and influences of different disciplines and traditions were also a recurring point of discussion at the first Association for Computing Machinery (ACM) conference with international representation – the "Conference on Academic and Related Research Programs in Computing Science" – that was held at the State University of New York at Stony Brook in 1967 [Fin14, p. vii]. Finerman et al.

(1968) report from what they called a "schizophrenic situation." They describe how various arguments were advanced to support the rejection of CS as a standalone academic field, for example, "the computer is just a tool and a body of study based upon a tool is not a proper intellectual discipline; the importance of computers has been overrated and their acquisition and study is not warranted; computing science is not a coherent discipline but rather a collection of bits and pieces from other disciplines." [ibid., p. 194].

In order to become clear about the meaning of CS as an academic discipline, the participants tried to come up with a definition. For example, Perlis connected current considerations and the development of CS with the development of "computer programming and the digital computer." [Fin14, p. 88]. In doing so, he explained that although CS as an academic discipline is new, the "algorithms and their goals are very, very old" – and without the computer, "its goals and studies would remain as fragile and isolated as they have been in the past." [ibid., p. 70]. Thereby, Perlis already connected the mathematical and engineering roots of CS. Gill proceeds one step further and formulates: "I am one of those who believe that there does exist a new profession in the computer field. This does not mean I can define it precisely, and it is bound to have fuzzy edges. It will have much overlap with applied mathematics, with communication engineering, with accountancy, with electronics, with management science, and so on." [ibid., p. 117].

Finerman et al. (1968) describe how even the critics of CS undergraduate programs often build on the existing connections with other fields: "The contention is that an undergraduate student must take so wide a range of fundamental courses that no time is left for specialization." [Fin14, p. 202]. Thereby, different authors stated that since the specialization must be in a graduate program either way, the basics can also be covered by existing programs, and students in other disciplines can choose to specialize in CS at the graduate level. Oettinger also called on the broad field of CS: "In our rush to be accepted as scientists or engineers and to mold students in our image, I hope that we are not going to make the mistake of prescribing narrow curricula restricted entirely to technical subjects." [ibid., p. 29]. He feared that ignoring the fundamentals in other related fields could create an "army of technicians." [ibid. p. 202], and summarized that CS will "cover the spectrum from the purest of mathematics to the dirtiest of engineering." [ibid., p. 34].

2.2. Theory of Computation in Computer Science Education

Up to this point, I have compiled the traditions and historical roots that constitute CS and its development as an academic field. As this doctoral project relates to the teaching and learning of ToC, I have thereby clarified, among other things, the importance of the mathematical and theoretical roots of CS. After CS developed as an academic discipline, the question remains what role ToC occupied in the recommended CS curricula and how its meaning was communicated to teachers and students. Therefore, I summarize selected early and current curricula below, highlighting the content of ToC and the changes that have occurred over the years. I conclude the section with an example of how ToC is taught today.

2.2.1. Theory of Computation in Computer Science Curricula

I briefly point out that I use a generalization of the ToC topics for this section instead of listing each area separately. For a short overview about the topics typically taught in German introductory ToC courses based on internationally widespread introductory literature, I refer to Section 2.2.2. For now, I summarize these topics by the term " ToC topics". For the following section, note that there may have been major differences between "necessary" and "recommended" topics over time and among universities or other educational institutions. All the following curricula were only recommendations.

Curriculum from 1968

In one of the first CS curriculum presented in 1968 by the ACM [ACH$^+$68], CS was divided into three main content areas: (1) Information Structures and Processes[17], (2) Information Processing Systems[18], (3) Methodologies[19].

[17] "This subject division is concerned with representations and transformations of information structures and with theoretical models [...]." [ACH$^+$68, p. 4]

[18] "This subject division is concerned with systems having the ability to transform information. Such systems usually involve the interaction of hardware and software." [ACH$^+$68, p. 5]

[19] "Methodologies are derived from broad areas of applications of computing which have common structures, processes, and techniques." [ACH$^+$68, p. 5]

In addition to topics and content, the curriculum in 1968 included a detailed appendix on each topic, suggestions for appropriate course content, and recommended literature. The committee inconsistently mentioned what students should be able to do after completing the course. The formulation of the curricula also justified, in part, why each topic was particularly relevant to CS and computer scientists and why they could not be inserted into existing courses from other fields. The committee provided an overview of the interconnection of the various fields and thereby, demonstrate how the content interrelates [ACH$^+$68, p. 7].

The authors emphasized that mathematical, and engineering foundations are recommended to be developed within the respective existing fields. Especially relevant for the context of this doctoral thesis is how the committee behind the first curriculum already appreciated the mathematical roots by stating that "computer science must be well based in mathematics since computer science draws so heavily upon mathematical ideas and methods." [ACH$^+$68, p. 11].

At that time, theoretical topics were recommended in courses like "Data Structures," "Introduction to Discrete Structures," "Theory of Computability," and "Formal Languages and Syntactic Analysis" [ACH$^+$68, p. 6 – 10]. In the course description of "Introduction to Discrete Structures," it is motivated how pure mathematics courses cannot cover the algorithms seen in the CS curriculum despite their discrete structure by stating that "[t]his course provides the student with an introduction to the basic numerical algorithms used in scientific computer work – thereby complementing his studies in beginning analysis – and affords him an opportunity to apply the programming techniques he has learned in Course B1 [Introduction to Computing]. Because of these aims, many of the standard elementary numerical analysis courses now offered in mathematics departments cannot be considered as substitutes for this course." [ACH$^+$68, p. 8].

Curriculum from 1978

The curriculum of 1968 would be adjusted again only in 1978; the authors emphasized that ongoing efforts and significant developments had changed the previous version. One of their particular focuses was to identify the CS core material that should be common to all CS undergraduate programs with the aim that students could achieve the objectives of the undergraduate major. The discussions led to eight core areas, but apart from two basic areas, the other areas were again interwoven and connected throughout the curriculum [ACH$^+$68, p. 121].

Furthermore, the committee was even more clear about the connection between CS and mathematics: "An understanding of and the capability to use a number of mathematical concepts and techniques are vitally important for a computer scientist. Analytical and algebraic techniques, logic, finite mathematics, aspects of linear algebra, combinatorics, graph theory, optimization methods, probability, and statistics are, in various ways, intimately associated with the development of computer science concepts and techniques." [ACH+68, p. 132].

In 1978, the ToC topics were taught through courses such as "Data Structures and Algorithms Analysis" or "Organization of Programming Languages" [ACH+68, p. 123 – 131]. While the overall structure of the curriculum was heavily reorganized, the courses' objectives were consistently added within the core areas. With this expansion, the authors presented the benefits and usefulness of the topics and courses in a more structured way, for example "[t]he objectives of [Organization of Programming Languages] are: to develop an understanding of the organization of programming languages, especially the run-time behavior of programs; to introduce the formal study of programming language specification and analysis; to continue the development of problem solution and programming skills introduced in the elementary level material." [ibid., p. 126]

Current Curricula

Today, there are two predominant CS curriculum recommendations that are similar in parts but also slightly different in how courses are structured.

ACM and IEEE Computer Society (2013): The first curriculum is from the Joint Task Force on Computing Curricula, ACM and Institute of Electrical and Electronics Engineers (IEEE) Computer Society. The curriculum is organized into so-called *Knowledge Areas* that correspond to topical fields of CS [JTFoCCS13, p. 14]. The Knowledge Areas are explicitly and thoroughly interwoven, such that they are not intended to describe specific courses. The curriculum also reports examples of actual CS courses and illustrates how topics of Knowledge Areas may be combined and covered in several different ways.

Furthermore, instead of inconsistently listing objectives of specific courses, the authors of the curriculum provide a set of topics and "learning outcomes" that students are expected to achieve. The authors assigned the learning outcomes to one of three "levels of mastery" they developed [JTFoCCS13, p. 34]:

- *Familiarity:* "The student understands what a concept is or what it means. This level of mastery concerns a basic awareness of a concept as opposed to expecting real facility with its application. It provides an answer to the question 'What do you know about this?' "
- *Usage:* "The student is able to use or apply a concept in a concrete way. Using a concept may include, for example, appropriately using a specific concept in a program, using a particular proof technique, or performing a particular analysis. It provides an answer to the question 'What do you know how to do?' "
- *Assessment:* "The student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem. This level of mastery implies more than using a concept; it involves the ability to select an appropriate approach from understood alternatives. It provides an answer to the question 'Why would you do that?' "

Also in 2013, the importance of ToC topics within a CS curriculum was emphasized. In the recommendations, the Knowledge Areas "Algorithms and Complexity (AL) and "Discrete Structures" (DS) cover most ToC topics and received the second- and third-highest number of recommended hours[20] in a so-called Core Tier-1[21] after Software Development Fundamentals (SDF) [ibid., p. 37]. Specifically, 19 Core Tier-1 hours were recommended for AL[22]. For DS, 37 Core-Tier 1 hours are recommended[23].

In the following, I present how learning outcomes for Basic Automata Computability and Complexity in the Knowledge Area Algorithms and Complexity (AL) are formulated [ibid., p. 59]:

[20] "An 'hour' corresponds to the time required to present the material in a traditional lecture-oriented format". Therefore, it does not include any practice and tutorial sessions or individual preparation and follow-up time [JTFoCCS13, p. 32]

[21] Core Tier-1 topics are understood as "a required part of every Computer Science curriculum" since these "topics are those with widespread consensus for inclusion in every program" [JTFoCCS13, p. 30]

[22] two for basic analysis; five for algorithmic strategies; nine for fundamental data structures and algorithms; three for basic automata, computability, and complexity [ibid., pp. 56 – 61]

[23] four for sets, relations, and functions; nine for basic logic; 10 for proof techniques; five for basics of counting; three for graphs and trees; six for discrete probability [ibid., pp. 77 – 81]

- Topics
  - Finite-state machines
  - Regular expressions
  - The halting problem
- Learning outcomes:
  1. "Discuss the concept of finite state machines. [Familiarity]"
  2. "Design a deterministic finite state machine to accept a specified language. [Usage]"
  3. "Generate a regular expression to represent a specified language. [Usage]"
  4. "Explain why the halting problem has no algorithmic solution. [Familiarity]"

Gesellschaft für Informatik (2016): The curriculum of the Gesellschaft für Informatik e.V. (GI) is a second example [Zuk16]. This curriculum builds, amongst others, on the decisions of [JTFoCCS13]; furthermore, the authors use a slightly different distribution of the curriculum content. Following an "outcome orientation," they first describe the requirements for computer scientists from the perspective of working life and then derive core competencies related to different content areas. These competencies[24] are differentiated concerning the level of requirements and the context of application and are recommended to be taught in every CS degree program [Zuk16, p. 5]. Similar to the level of mastery in [JTFoCCS13], the authors use "cognitive competence dimensions" by adapting the Anderson Krathwohl Taxonomy [AK01]. Therefore, for every content field, descriptions of competencies are inserted within the following dimensions: Understand, Apply (Transfer), Analyse (Evaluate), and Create [JTFoCCS13, p. 10].

As ToC topics, the GI recommends "Algorithms and Data Structures" [Zuk16, p. 13 – 14], "Discrete Structures" [ibid., pp. 21 – 22], and "Formal Languages and Automata" [ibid., p. 23]. For example, the competence dimensions for Formal Languages and Automata are [ibid., p. 23]:

[24] Following Weinert [Wei01], competences are understood here as learnable cognitive abilities and skills that enable an individual to solve problems in a context of action, including the necessary motivational, volitional and social dispositions and skills [ibid., p. 9].

- *Understand:* Explain basic concepts of describing formal languages in declarative form or by means of grammars. Explain the classification of languages in the Chomsky hierarchy and the associated automata models. Understand transformations between the individual forms of description. Explain Turing machines, computability and non-determinism. Explain the computability of functions and the decidability of languages through Turing machines and comprehend them for individual examples.
- *Apply (Transfer):* Define grammars, regular expressions and automata for formal languages and transform them into equivalent models. Prove equivalences between different forms of description. Use non-determinism to obtain more effective automata. Assess computability and decidability for simple examples. (Use parser generators or lexers.)
- *Analyse (Evaluate):* Classify formal languages into the correct levels of the Chomsky hierarchy. Evaluate and, if necessary, optimize models.
- *Create:* -

In summary, the curricula presented illustrate the high status that ToC topics have had in CS education in the past and today. While the importance of mathematical roots was illustrated in the early curricula, today's curricula recommend a focus on demonstrating the learning outcomes and competencies that CS students can and should develop through participation in ToC courses. In this section, it becomes even more apparent how important it is to examine the teaching of ToC more closely in order to prevent students from failing essential ToC topics and potentially dropping out of their CS studies as a result.

2.2.2. Theory of Computation Introductory Courses

In the following section, the focus of the descriptions is on teaching ToC in Germany, since the present doctoral project is being conducted at a German university. The traditional course setup of German ToC introductory courses will be presented, followed by the topics usually presented and an example of lecture content and assignments.

Course Setup

The number of students in introductory courses in ToC at German universities varies depending on admission restrictions and the capacity of the university. Usually, the number lies between 150 and 350 students. Depending on the study programs offered, the largest number of participants are CS students, the next largest number are likely to be those in computer science-related courses (e.g., business informatics, software engineering); finally, there are participants who have CS as a minor. Whether introductory courses are taught in the first, second, or third year is not uniform across universities. Usually, students are also expected to take an introductory mathematics course beforehand or in parallel.

Despite their differences, the pedagogical approach in German ToC courses traditionally consists of a combination of these components [KKB14, pp. 66 – 69]:

- An optional lecture per week (often 90 minutes) given by a lecturer who presents the course topics, central concepts, algorithms, and their proofs, illustrating them with examples and using slides, live annotation, and a blackboard in a lecture hall. All material is available online.
- A weekly tutorial session (often 90-minute and optional) with various design options. For example, a teacher/tutor (usually senior students or research associates) solves practice assignments visible for everyone. Another possibility is that students are expected to present solutions to homework assignments they solved beforehand or during the session. Sometimes, sample solutions for the presented assignments are given to the students after the session.
- A number of homework assignments that students are required to work in self-organized groups without direct teacher/tutor support and then submit for evaluation. Depending on the university, the assignments must be solved during the tutorial or in their free time.
- A final exam at the end of the course consisting of assignments similar to the tutorial session and homework. Typically, 50% of the points are necessary to pass the course.

Course Content and Topics

It is not uncommon to find about two introductory courses on ToC in Germany – at least one on Formal Languages and Automata (FLAT) and one on Algorithms, Data structures, and Complexity (ADC). Depending on the university, the title and division of topics may differ somewhat, but in principle, these courses cover the following topics based on widespread international literature (e.g., [HMU01] [Sip12] [CLRS09]).

Formal Languages and Automata

- **Mathematical Basics:** In general, the aim is to introduce and discuss the basic mathematical objects, tools, and notations that will be used later in the course. Especially, the connection between *sets*, *functions*, *relations* as well as an introduction to *logic* is formally presented and exemplified by *graphs* [Sip12, pp. 3 – 28]. Depending on the course, these topics can also be covered in mathematics courses.
- **Automata and Languages:** Within this course, different computational models are introduced. The aim is to use them "to set up a a manageable mathematical theory of them directly" [Sip12, p. 31]. First, it usually begins with *Finite Automata* which are introduced with an example or riddle from the real world, before the formal version is introduced – that is, a number of formal definitions that describe the very idea of an automaton in his most abstract form [ibid., p. 31 – 36]. In the course of this, *Deterministic Automata* and *Non-deterministic Automata* are also illustrated followed by their formal versions [ibid., p. 48 – 58]. Within this part, *regular and non-regular languages* and necessary proofs are presented as a second model to describe languages [ibid., p. 77 – 82]. Second, *context-free and non-context free languages* as more powerful computational models are introduced including also *Pushdown Automata* (again with textual explanation followed by a formal definition) [ibid., pp. 102 – 146]. Third, models for general purpose computers are presented, including *Turing machines* and its variants. Furthermore, the notion of algorithm by means of the *Church-Turing thesis* is defined [ibid., pp. 156 – 187]. Fourth, the *decidability and undecidability* of algorithms is introduced to establish and prove that there are algorithms

that cannot be solved algorithmically [ibid., pp. 193 – 210]. Lastly, *reducibility* is introduced as "primary method for proving that problems are computationally unsolvable" [ibid., p. 215].

Automata, Data Structures, and Complexity

- **Basics and Sorting:** At the beginning, the definition and use of an *algorithm* is given. *A simple sorting algorithm* is often used as a first example of an algorithm [CLRS09, pp. 3 – 30]. In addition to incremental algorithms, *divide and conquer algorithms* [ibid., pp. 67 – 99] and *probabilistic and randomized algorithms* are presented [ibid., pp. 115 – 130]. In the course of analyzing algorithms, *Big-O notation* is introduced and used as a method to formally analyze the running time of algorithms [ibid., p 45 – 55]. Furthermore, *recursion equations* are introduced as a mathematical way of calculating and transforming functions [ibid., p. 85].
- **Data Structures:** Various *data structures* are presented (e.g., stacks, queues, lists, trees, hash tables) as applications for constructing efficient algorithms [CLRS09, pp. 147 – 332]. Thereby, the running time is also formally analyzed. In addition to the mathematical knowledge required up to now, probability theory can also be used.
- **Graph Algorithms:** This is about algorithms that work on *graphs*. In the course of this, topics like depth-first search, breadth-first search, spanning trees, shortest paths can introduced [CLRS09, pp. 595 – 748].
- **Additional Topics:** Various other topics can be covered depending on the university, for example, multi-threading [CLRS09, pp. 785], efficient algorithms working on matrices [ibid., pp. 827], linear programming [ibid., pp. 857], the complexity classes P and NP [ibid., pp. 1059]. In all of these cases, also existing (proving) methods are applied to new contexts and expanded.

Overall, every mentioned part is intensively framed by various theorems, lemmas, and corollaries who describe essential attributes of the models, algorithms, and their relations. These are then formally proven and, in turn, used for further formal proofs and analyses. Illustrated examples from the real world are used at most at the beginning, before turning to the abstract and formal version. This focus on formalism is necessary because Hilbert's formalist proof theory often enabled the development of the topics described in this section (see Section 2.1.1). In the course of ADC, all algorithms are also presented by using pseudo-code – that is, an abstract or simple form of programming code so that it can be transferred in any other programming language. Nevertheless, descriptive names or terms are also rarely used here to correspond with the formalized versions of the algorithms.

Example of Lecture Content and Assignments

In the following section, I present an example topic from a ToC course in more detail. I offer details about the content and present how this topic is treated in the individual components of the respective course. Finally, I present explicit assignments, including their solution approaches, that were part of an existing regular course.

The Pumping Lemma for Regular Languages The Pumping Lemma (PL) for regular languages is a common element of automata theory and is used in student assignments to prove that a given formal language is not regular – that is, it cannot be accepted by a finite automaton [Sip12, pp. 77 – 79]. For that matter, students are introduced to a specific scheme to be used for creating a proof by contradiction. Hence, among ToC assignments requiring formal proofing, PL assignments are schematic and do not require students' particular creative ideas or new approaches to develop a proof structure; rather, they follow the given proof pattern.

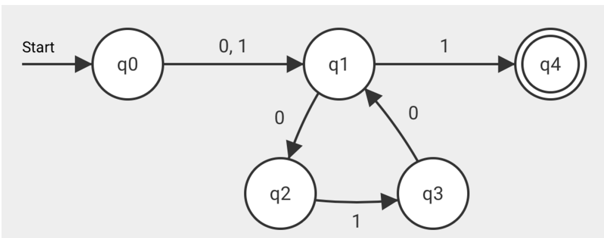

Figure 2.2.: A finite automaton.

Since an understanding of finite automata is useful for understanding the PL, a simple finite automaton M is visualized in Figure 2.2. According to Sipser (2012) and Hopcroft (1979), M is formally described by $M = (Q, \Sigma, \delta, q0, F)$, where [HMU01] [Sip12]:

- Q is a finite set of states (i.e., $Q = q0, q1, q2, q3, q4$). These are normally represented by the circles in Figure 2.2.
- Σ is a finite set of input symbols (i.e., $\Sigma = \{0, 1\}$). These are the numbers on the edges in Figure 2.2.
- $\delta : Q$ x $\Sigma \longrightarrow Q$ is a transition function returning a subset for a state of Q and an input symbol of Σ with the possible next states e.g., with a 1 you can only go from $q1$ to $q4$.
- $q0 \in Q$ is an initial state and has an incoming edge without previous state
- $F \subseteq Q$ is the set of final states (i.e., $F = \{q4\}$) and is represented as a double bordered circle.

Single accepted strings are called *words*, the set of all accepted words is the *language* of M. The class of languages accepted by a finite automaton are called *regular* languages.

In general, the PL captures the idea that all words from a regular language are also regular – that is, in every word there exists a part that can be repeated many times without violating the attributes of the language. A proof by contradiction is sufficient to show that a language is not regular by finding one word of the language and one part in it that will violate the attributes of the language after being repeated. The following formal scheme is usually used in ToC courses to create this contradiction and, therefore, prove that a given language L is not regular:

1. Assume that L is a regular language and choose a $n \in N$ arbitrary but fixed.
2. Choose a word $w \in L$ with the minimum word length n.
3. Choose an arbitrary decomposition of xyz of w, where the following applies:
    a) $|y| \geq 1$: y is not empty.
    b) $|xy| \leq n$: The words x and y have a maximum length of n.

c) $\forall k \geq 0 : xy^k z \in L$: For all natural numbers k that are greater than or equal to 0, the word $xy^k z$ is part of L.

4. Choose one specific k, calculate $xy^k z$ and check if the decomposition $xy^k z \in L$. If $xy^k z \notin L$, then the assumption that L is a regular language led to a contradiction and is, therefore, incorrect.

Considering the visualized finite automaton in Figure 2.2, x is the substring from the initial state to a reused state (i.e., 0 or 1). The substring within the cycle is y (i.e., 010 any number of times). The substring z considers the last reused state until the finite state (i.e., 1).

Lectures and Tutorial Sessions The PL is usually taught in a course that also covers topics concerning formal languages and automata, and is presented after Finite Automata, Deterministic Automata, Non-deterministic Automata, regular expressions, and languages (see Section 2.2.2). Depending on the introductory literature, the basic ideas behind automata are usually introduced via an easier-to-understand context or riddle[25]. In a next step, the context is removed from the automata, revealing a *formalized* variant (see Figure 2.2). I call this variant the formalized variant because it parallels Hilbert's idea of formalization that instead of contextual examples or interpretations of the automata, only mathematical symbols shall remain and be used (see Section 2.1.1). Since all symbols are defined and classified according to mathematical regularities, they can later be used to lead a formal proof.

The necessary mathematical notation to use the PL to develop a formal proof is introduced in the current or previous CS lectures, while some of which are also familiar from mathematics classes. Usually, the basic scheme for the PL is then explained in the lecture, which the lecturer works through with one or more examples and only limited student activity (see Section 2.2.2). Knobelsdorf et al. (2014) already highlight problems with this kind of traditional lecture: "The pedagogical approach behind these course components assumes that

[25]For example: "A man with a wolf, goat, and cabbage is on the left bank of a river. There is a boat large enough to carry the man and only one of the other three. The man and his entourage wish to cross the right bank, and the man can ferry each across, one at a time. However, if the man leaves the wolf and goat unattended on either shore, the wolf will surely eat the goat. Similarly, if the goat and cabbage are left unattended, the goat will eat the cabbage. Is it possible to cross the river without the goat or cabbage being eaten"? [HMU01, p. 14]

students understand the presented concepts, theorems, and proofs during the lectures." [KKB14, p. 69]. During the course, students are usually provided with the presented lecture slides and the formal PL scheme.

Often, sample assignments are only developed step-by-step with the PL scheme in the tutorial sessions. Depending on the course setup, students are required to solve practice assignments in groups within the tutorial session and with the possibility of teacher/tutor support. In other cases, students are required to solve these assignments without teacher/tutor support in self-organized study groups independent of their time in the university. Then, the students either have to present their solutions in one of the following tutorial sessions or follow the solution approach presented by their teachers/tutors. Furthermore, it may be desired that students submit their solutions to their teachers/tutors for assessment. Knobelsdorf et al. (2014) also recognized the inactivity of the students in tutorial sessions by sharing that "our tutors reported that most students remained very passive during the student sessions and did not participate in discussions even when their own solutions contained mistakes." [KKB14, p. 69].

Example Assignments The following first two assignments were part of real existing student homework, while the third assignment was used in a final exam at the end of a FLAT course:

Assignment 1. Prove that the language $L_1 = \{a^m a^l cb^{m+l} \; | m, l \in N\}$ is not regular.

Solution Approach. (1) The students must conclude that the occurrence of a and b within a word from $L_1$is always equal despite the different exponents notation. (2) Therefore, a word must be chosen depending on the length n e.g., by choosing $n = m + l$ resulting in $w = a^n cb^n$ (3) They can chose their distribution of xyz by assigning e.g. $x = a^i ; y = a^i$, with the new exponents $i, j \leq n$. Then, z is represented by subtracting the occurrences a in x and y, e.g. $z = a^{n-i-j} cb^n$.(4) Any k can be chosen (but $k \neq 1$ is not appropriate as it is the same word) and used to calculate the new word which violates the assumption that $xy^k z \in L_1$ and proves that L_1 is not regular.

Assignment 2. Prove that the language $L_2 = \{bxc^m | x \in \{a, b\} * \wedge m \in N \wedge |bx|_a - |bx|_b > m\}$ is not regular.

Solution Approach. (1) The students have to realize that x must consist of any a and b. Thereby, it applies that the number of as is greater than the number of bs, i.e., $a > b$. (2) A word must be chosen depending on n, e.g. $w = b^{n+1}a^{n+2}c^n$ (3) Next, a distribution must be chosen of xyz by assigning e.g., $x = b^i, y = b^j$, with the new exponents $i, j \leq n$.Then, z is represented by subtracting the occurrences of b in x and y, e.g. $z = b^{n+1-i-j}a^{n+2}c^n$. (4) Any k can be chosen (but $k \neq 1$ is not appropriate as it is the same word) and used to calculate the new word which violates the assumption that $xy^kz \in L_2$ and proves that L_2 is not regular.

Assignment 3. Prove that the language $L_3 = \{xy|x \in \{a,b\}^* \wedge |x|_a mod 2 = 0 = |x|_b \wedge y \in \{c,d\}^* \wedge |y|_c > |y|_d\}$ is not regular.

Solution Approach. The students must figure out that they can also choose $x = \lambda$ to simplify the language. (2) A word must be chosen depending on n, e.g., $w = c^{(n+1)}d^n$. (3) Next, a distribution must be chosen of xyz by assigning e.g., $x = c^i; y = c^j$, $i, j \leq n$. Then, z is represented by subtracting the occurrences of z in $x$and y e.g., $z = c^{(n+1-i-j)}d^n$. (4) Any k can be chosen (but $k \neq 1$ is not appropriate as it is the same word) and used to calculate the new word which violates the assumption that $xy^kz \in L_3$ and proves that L_3 is not regular.

The language of the second assignment expected, in addition to the basic notation of languages, that they apply knowledge from set theory and the mathematical notation of the length of languages. Additionally, the final exam assignment expected students to understand and use the modulu operation and, depending on how they choose to distribute xyz, a case distinction.

As can be seen, the presented assignments and solutions are formalized according to the proof theory and hence mathematical tradition of CS (see Section 2.1.1). Given the formalistic roots of many of the mentioned ToC topics, it is not surprising that formalism is the dominating way in which the body of knowledge in ToC is presented and communicated in corresponding courses today. The obvious question is if CS students are expected to assume the role of the mathematical tradition during their ToC courses, whether they are expected to assume a different role in courses shaped by the other traditions. After all, as I reproduced before, it is apparent that the different traditions of CS have different aims and methods (see Section 2.1.1). Considering that students have to experience such an understanding of several traditions in their introductory courses, it seems like a significant challenge. Especially when they found themselves more in the engineering or science tradition of CS. Therefore, the demand in ToC, to formalize the body of knowledge in question could be a factor for the high dropout rates in

ToC courses. In the following chapter, I summarize the reasons that have been assumed so far for the high failure rates and how attempts are being made to address them.

2.3. Summary

In this chapter, I have summarized and discussed related work that emphasizes on the importance of ToC as part of CS formation. Together with engineering and sciences, the mathematical ideas that fostered the development of ToC form a tripartite of three traditions. They have had a significant impact on how CS has developed as a discipline and academic field. Although the foundations and ideas for facilitating computation date back to antiquity, it was primarily ideas and developments between 1800 and 2000 that contributed to the development of CS, e.g., Hilbert's formalistic mathematics, the theoretical Turing machine, or the precursors of today's modern computer. On the one hand, the three traditions have mutually developed and benefited from each other. On the other hand, this tripartite division also led to ambiguities about the objective of a CS university degree and the emphases that should be placed in the academic discipline, since the three traditions pursue different goals, use different methods, and produce different products. For this reason, not only were there several different CS-related courses in the 1960s but the first attempts to develop an CS curriculum covering the relevant topics and areas had intensely interrelated courses.

While CS curricula covered the various disciplines and traditions that formed the basis for CS, ToC also has an established place in the recommended CS curricula to this day. While the importance of mathematical roots was illustrated in the early curricula, an emphasis in today's curricula is on presenting the learning outcomes and competencies that CS students can and should develop by taking ToC courses. In this chapter, it becomes even more apparent how important it is to examine the teaching of ToC more closely in order to prevent students from failing essential ToC topics and potentially dropping out of their CS studies as a result. As a content example, I have explained how these components are structured for PL for regular languages. Such an illustration underlines how interpretation and meaning are also reduced away from the content in ToC until only mathematical symbols remain, which in turn goes back to the basic idea of Hilbert's formalism.

3. Current State of Research

In the following section, I describe existing research on how to address low performance and high failure rates in ToC courses. First, I present approaches that have modified and extended existing courses and content. Second, I focus on software, systems, and tools developed to teach the concepts and topics interactively. Third, I present existing case studies which analyze actual students' difficulties with specific topics and assignments. Fourth, I give a brief overview of related work in mathematics education. Finally, I summarize how the presented research led to the decision to undertake this dissertation project.

3.1. Modify and Extend Courses and Content

In this section, I detail approaches that have modified and extended existing courses and content to make topics more relevant and applicable, thereby engaging students. Table 3.1 provides an overview of existing approaches and also contains the course or topic that was extended or changed with the individual approach. I have also listed the assumption on which the approach was developed and where this assumption originated (*undefined* if the origin is not apparent from the article).

Chesñevar et al. (2004) point out that the level of abstraction in courses on FLAT makes the topics difficult to teach and learn [CnGM04]. They report that from their teaching experiences with second-year students, they felt that many students were not as motivated to take interest in these topics. They also note that this is not only because the topics are too mathematical, but also because they lack relevance to other CS topics. Surveys of students revealed that many of them apply the theoretical concepts mechanically instead of developing meaningful knowledge. The authors describe the results as a lack of "significant learning" in the sense of "adequate mental models that will be available for use in different contexts" [CnGM04, p. 7]. To overcome this problem, the authors introduced some pedagogical strategies along with their traditional curricula to enrich the content of the course. Their overall goal was to emphasize the fact that the course introduces mathematical foundations necessary for CS rather than presenting standalone abstract mathematical concepts. One idea they discuss is to introduce the course content in the historical context of CS to show its relevance in the overall context – for example, with biographical notes and videos on the course website. Their experience has shown that discussing the development of ToC between 1930 and 1950 helps students see the importance of different theoretical concepts. In addition, they suggest using simulator software to provide a motivating and interactive link between theory and practice and to promote active student learning. They explicitly point out that students benefit from trying to solve the same task with different simulators. Furthermore, the authors suggest linking the topics to current programming languages so as to counteract the prejudice that the topics are only mathematical in nature. Finally, they introduce several immersion and extension activities, such as articles on applications of concepts in real-world problems. Overall, they try to combine different teaching strategies to make the topics more interesting and attractive to students. They described the results followed by their changes as "highly satisfactory."

Similarly, Habiballa and Kmet (2004) contextualize ToC topics with practical examples from professional life [HK04]. They summarize the general problems of teaching CS by focusing on whether CS is more a mathematical or technical discipline, on how it is unclear at what educational level CS should be taught, and on the unclear interaction among the different fields of CS. For an experiment, the authors used a so-called "application concept" to connect the theoretical topics with interactive programming. This concept starts with the general concept, extends it with "practical technique" and adds an "application" to increase motivation and help students understand the essence of the procedures used. The authors studied an experimental group and a control group to compare the "application concept" with standard teaching concepts at their university. Their finding is that the application concept showed statistically better results in some settings, while the standard techniques were better in other aspects. They see their small sample size as a problem, but they nevertheless conclude that the application concept was promising overall.

Hamilton et al. (2003) describe how they reorganized their tutorials against the background of "problem-based learning" to make learning more "student-centered" [HHP03]. They argue how students struggle with the theoretical topics more so than with basic programming because they can not see the relevance to their studies. The result are groups of six students working collaboratively on assignments under the supervision of a teacher during a two-hour tutorial session. This is designed to counteract the problem that it is often "tempting for tutorial to lapse into mini-lecture mentality, with students asking few questions and hoping to be able to copy down solutions rather than work through the given problems" [HHP03, p. 2]. This group work is intended to encourage students to work on the issues themselves rather than waiting for someone else to talk about them. In addition, they provide an assessment to motivate students further to participate in the tutorial. No comprehensive evaluation has been conducted, but from several pieces of feedback over the years, it appears that students generally support this approach over "standard" tutoring.

Similarly, Hämäläinen (2004) changed her course by using another "problem-based method" to deal with topics that students report disliking because of their "mathematical and theoretical nature" [Ham04, p. S1H]. She states how often what is missing are practical applications and a focus on the meaning of the problems in the real-world. Her goal is to get students to become more active and not just be passive recipients of new information. Behind her problem-based learning is the idea of using problems or puzzles as a starting point for learning. In her experiment, she taught new concepts using a typical problem-based learning cycle consisting of seven steps: defining unclear concepts, defining problems, brainstorming, constructing hypotheses, defining learning goals, self-studying, and sharing the results. In addition to a traditional tutorial, she redesigned the lecture: Half of the lecture is worked through following the seven steps in groups for specific topics, and the other half is set aside for lectures or problem-solving games. During the group work, the lecturer is available for tutoring. In addition to these changes, the students keep a learning diary in which they observed their own learning and attempted to create an overall scheme of what they had learned. The problem reports from their group work and the learning diary are incorporated into the course grade at the end. Overall, student feedback on the method after the course was mixed. Most students were satisfied with the problem-based method but wished they had more time for the course.

Sigman (2007) also assumes that students are unsure about the relevance of the required mathematical material to CS [Sig07]. Furthermore, he states that the students find the material especially difficult because it differs from the material in other CS courses. He constructed a course using the learning technique known as the "Moore method", which directly targets students' engagement with this material. In general, this method represents a family of approaches that "share a common commitment," namely, "letting students discover their own capabilities to create and learn" [Sig07, p. 451]. He further describe how the Moore method consists of constructing a series of problems in such a way that students working through the problems *discover* the material central to a particular course. An actual course meeting consisted of students presenting solutions to problems, which were then discussed with fellow students. The tutor's role was to guide, critique, and evaluate, but not to lecture. According to Sigman, in its purest form, the Moore method contains no lecture per se, although many variations of the approach do contain a small lecture element. He found the method fitting for FLAT since "[t]he most striking example of this is the acknowledged ability of the method to teach students how to make proofs. This skill includes the ability to master abstraction, to think logically, and to communicate clearly. [...] The problem-based nature of the method fosters student engagement with the material throughout the semester since students are aware that they will be routinely called upon to present solutions" [ibid., p. 451]. Apart from the student presentations, the course design stayed traditional with lectures and textbooks. The presentations also were part of the course grade. Due to the small course size (four students) and the fact that the author offered the course only once, any evidence of its effectiveness is therefore anecdotal. The two students with the least mathematical experience showed the most significant improvements in their ability to construct proofs over the semester. However, in the 25 years that the author has taught mathematics and CS courses, the course achieved higher levels of student engagement and satisfaction than any he had previously taught.

Korte et al. (2007) used a constructivist approach with game-building [KAPG07]. They assume that students find the ToC material difficult compared to their other courses and lack motivation because they do not believe that the material is relevant to the rest of their studies. The authors argue that the problem lies with the topics being usually presented in an abstract manner with few real examples. They also state that since ToC covers so much material, it is not easy to find an appropriate new teaching method for every topic. They find that "modelling skills" (e.g., for automata, grammars) are a good starting point to address these problems and make the topic "more accessible and more relevant" [KAPG07, p. 53], thereby improving students' understanding of modeling. In their study, the authors used their learning through game-building for finite automata and regular expressions. Accordingly, students were asked to construct a finite automaton given specific requirements, choosing a personally meaningful context. However, students were not allowed to use the usual 5-tuple – but instead to use the game-building approach. They also conducted a second study concerning the Turing machine (TM), in which students were given a basic game framework with a 5-tuple TM to use in developing their own game. Eighty-seven percent of all students successfully completed their games, and informal observations showed that the weaker students were more enthusiastic than the stronger students who preferred the more traditional assignments.

A more incisive approach is suggested by Brookes (2004) [Bro04]. He states that theoretical topics require new ways to be taught in an interesting and relevant way and that the amount of ToC topics covered could be reduced. According to him, this idea is not new and already resulted in more "practical" study programs, such as software engineering. Brookes's approach addresses the problem of making ToC topic more relevant to students. In doing so, he integrated ToC into a course covering a range of fundamentals from different fields and tried to connect them with more "popular" technologies. This idea is part of a curriculum redesign that has repeatedly integrated theoretical topics into other courses, thereby replacing standalone courses. For example, the concept of trees and regular expressions is connected with teaching Extensive Markup Language (XML) technologies. The survey of 50 students showed promising results, but the theoretical material still seemed challenging. Overall, this approach spreads the curriculum of ToC topics across multiple courses and years and could lead students to no longer see ToC as a singular field in CS.

Table 3.1.: Approaches that modified and extended existing courses and content

Article	Course/Topic	Assumption	Source
[CnGM04]	FLAT	students' lack of motivation and interest; topics are too mathematical	teaching experiences
[HK04]	Regular and Context Free Languages	students' lack of motivation and knowledge	undefined
[HHP03]	FLAT	students cannot comprehend the relevance for future career	undefined
[Ham04]	Theoretical Foundations of Computer Science	students found the topics difficult and boring	student feedback
[Sig07]	several ToC topics (e.g. automata, grammars, complexity)	students cannot comprehend the relevance for future career	undefined
[KAPG07]	Programming	students' lack of motivation due to unclear relevance	undefined
[Bro04]	Distributed Computing	students cannot comprehend the relevance for future career	undefined

Table 3.2.: Software, tools, and systems to teach ToC topics

	Article	Course/Topic	Assumption	Source
Software	[WD05]	FLAT	mathematical nature is a hurdle to students	undefined
	[HW06]	FLAT	undefined	undefined
	[Chu07]	FLAT	undefined	undefined
	[CGG+05]	ToC	undefined	undefined
	[GKL+02]	Finite State Automaton	unclear connection to software development and unclear relevance	undefined
	[RBFR06]	FLAT	undefined	undefined
	[Ver05]	ToC	students' lack of motivation and interest	undefined
Physical Tools	[Zin08]	not specified	lack of motivation	undefined
	[BJJ01]	ToC	students' lack of engagement	undefined
Automated assessment systems	[GOMSJVGP08]	FLAT	the topics are too theoretical, dry, and difficult	undefined
	[CEK13]	Algorithms, Data Structure and Complexity	students' lack of motivation and unclear usefulness of subjects	teaching experience
	[DDP00]	FLAT	monotony while capturing the concepts and abstract topics	undefined

3.2. Usage and Adaption of Software, Physical tools, and Systems

Another set of approaches uses software, physical tools, and systems to engage students and bring them closer to the topics in different ways. I have divided the approaches into subsections for clearer discussion and comprehension, but Table 3.2 again gives an overview of all mentioned approaches.

3.2.1. Software

Wermelinger and Dias (2005) propose a Prolog[1] toolkit for FLAT courses. They aim for students to be able to easily map an implementation onto the mathematical definitions given in the lectures [WD05]. The Prolog toolkit for FLAT is a library of predicates to define and manipulate various kinds of languages and automata. Furthermore, the toolkit should also provide students with a library to implement other concepts and algorithms. In contrast to existing simulators, their Prolog toolkit would help students understand how the automata work and aid them in developing and debugging automata for accepting a given language. Moreover, students should study and extend the source code – for example, with individual notation. Students would also be able to execute the automata so that the toolkit can help solve assignments. This would also provide a bridge between abstract mathematical and formal concepts and their practical realization. In doing so, the Prolog toolkit complements existing graphical software and allows students to understand the FLAT concepts in ways other than just visual observations. Nonetheless, to benefit from the toolkit, the teaching of these ToC topics has to be linked with teaching Prolog or it has to be ensured that students have learned Prolog beforehand. Unfortunately, I was not able to test the toolkit.

Another learning environment for teaching FLAT is presented by Hielscher and Wagenknecht [HW06]. The learning environment AtoCC[2] can be used in teaching abstract automata, formal languages, and some of its applications in compiler construction. AtoCC aims to address a broad range of different learning activities (i.e. exercises and small projects), forcing the students to engage actively with

[1] http://www.swi-prolog.org, last visit 24 September 2021
[2] https://atocc.de, last visit 24 September 2021

the subjects being taught. In this way, the environment thereby represents four interconnected tools to expose four teaching aims:

- *presentation* through AutoEdit: can be used to visualize automata.
- *exercise* through AutoEdit Workbook: can be used to do exercises or share self-designed exercises with the community.
- *understand* through T-Diag: Tombstone diagramms used for compiler applications and development.
- *apply* through VCC: the VCC can be used to develop own compilers in different programming languages and to use them in T-Diag.

Unfortunately, I could not download the learning environment, but the authors provided tutorials and screenshots of their website that are found in Figure 3.1. I only show Autoedit, but the structure of T-Diag and VCC is similar to provide a uniform appearance. Although the authors provide and explain all the tools on one website, and the tools can cover different areas through all the interconnection, it poses a challenge for students to learn different tools before using them. In addition, the link with compiler construction has a particular use case.

Because of the mathematical and abstract nature of theoretical topics, there are even more approaches to work with visualization and simulation software to make the concepts more accessible: Chudá (2007) states the possibility of using visualization in the education of ToC [Chu07]. She describes her goals in visualization as "enhancing understanding of concepts and processes, making invisible visible and as effective presentation of significant features" [Chu07, p. IV.15-1]. She further describes how to use a storyboard to develop meaningful visualizations that she later provided through a Moodle e-learning course to her students. Moodle[3] is an open source learning management system that can be used to create online courses with grading possibilities and exercises of different kinds as well as community features such as discussions and "likes".

Using a hypertextbook[4] Cogliati et al. (2005) – see also [GKL$^+$02] – present another visualization tool [CGG$^+$05]. In their article, they present several tools

[3] https://moodle.com, last visit 24 September 2021

[4] "The hypertextbook is a novel teaching and learning resource built around web technologies that incorporates text, sound, pictures, illustrations, slide shows, video clips, and – most importantly – active learning models of the key concepts of the theory of computing into an integrated resource." [CGG$^+$05, p. 1]

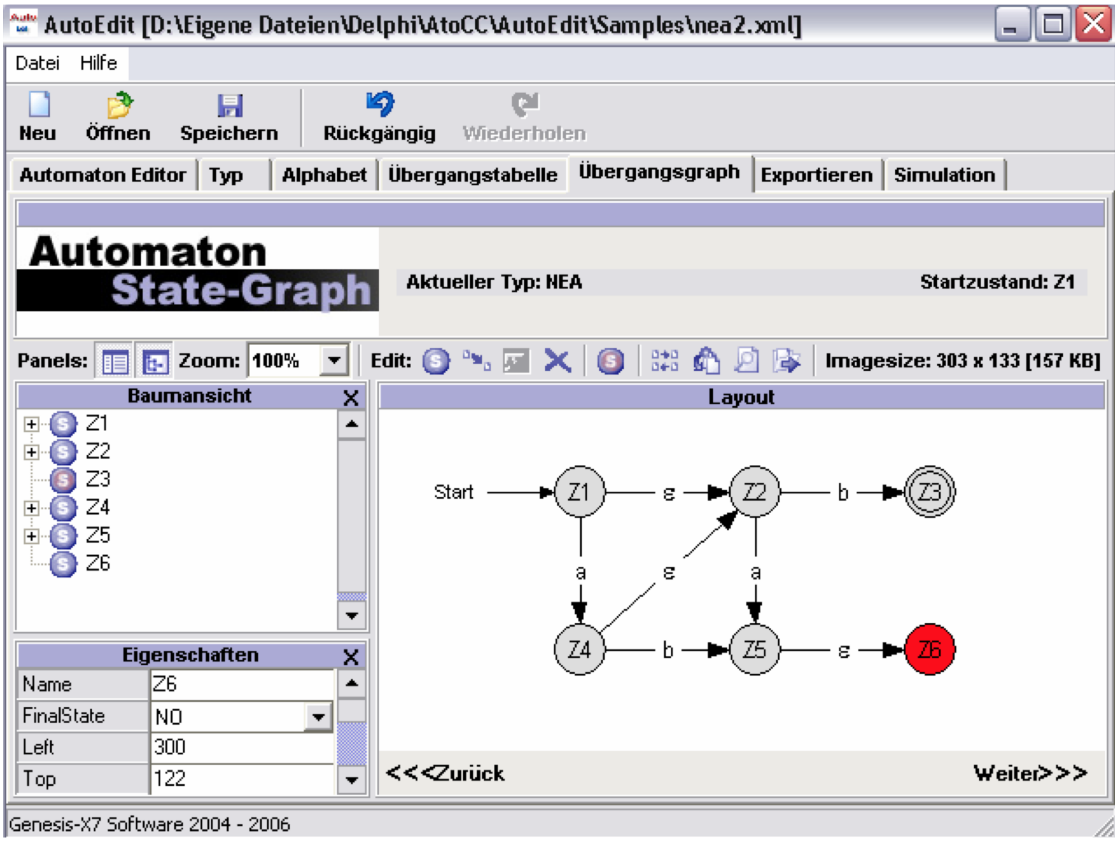

Figure 3.1.: Graphical User Interface of AutoEdit visualizing an automaton [Mic].

in more detail but state that "It should be clear that each of these active learning model and tool applets could be quite useful on their own. However, as noted at the beginning of this article, for many reasons stand-alone applets do not seem to be widely used in the classroom. So it is important that a comprehensive teaching and learning resource be developed and disseminated that seamlessly integrates standard text presentations of the material with the active learning models" [CGG+05, p. 13]. They report that a hypertextbook should be accessible in standard Web browsers, should work on any computer, be distributable by media, incorporate different levels of presentation for different levels of learners and be easily modifiable. Figure 3.2 presents a screenshot of the hypertextbook, but it is no longer available online. They tested the different tools and parts of the hypertextbook for the first chapter in a traditional ToC course: finite state automata, regular expressions, and regular grammars [GKL+02]. The authors state, "While we won't be so bold as to say that students will actually love to learn the theory of computing as a result of having access to these modules, we can confidently say from our own experience that they will find learning the theory to be more fun." [ibid., p. 371]. By employing a hypertextbook, the authors aim to work against several difficulties while using visualization software; for example, the teachers have to learn the software, install it, integrate it into an existing course, and teach it to the students.

Another well-known tool for visualizing formal languages topics is JFLAP[5], as described by Rodger et al. [RBFR06]. They explicitly state that the proof type of exercises should not be removed from the course, but "rather to supplement them with hands-on explorations of related topics." [RBFR06, p. 379]. In addition, it is possible to build a Turing machine or enter grammars. Figure 3.3 shows the possibilities as well as the graphical user interface for finite automata and solving a Pumping Lemma (PL) assignment.

Verma (2005) enhanced JFLAP and integrated it into his course [Ver05]. He decided to add a "debug" button to the interface, among other enhancements. Furthermore, he integrated additional simulation software into his course; and although he state that the students reacted positively to the software integration, the drop-out rate remained steady.

3.2.2. Physical tools

The literature also showcases ideas to use physical tools to support students' learning and understanding of theoretical topics. For example, Zingaro (2008) built his research on the "mental resistance" students seem to have when studying formal methods [Zin08]. As he notes, "The term Mental Resistance [7] has been used to characterize the attitude students bring to the study of formal methods. There have been many attempts to motivate students to want to study the material and to overcome this resistance". [Zin08, p. 56] He designed a book for first-year undergraduates that uses Java[6] as the programming language and does not require formal proofs. Instead, the focus is on programming problems that are much more easily solved when using invariants, and culls examples from various problem domains. Therefore, the approach only requires "sufficiently formal

[5] http://www.jflap.org, last visit 24 September 2021
[6] https://dev.java, last visit 27 September 2021

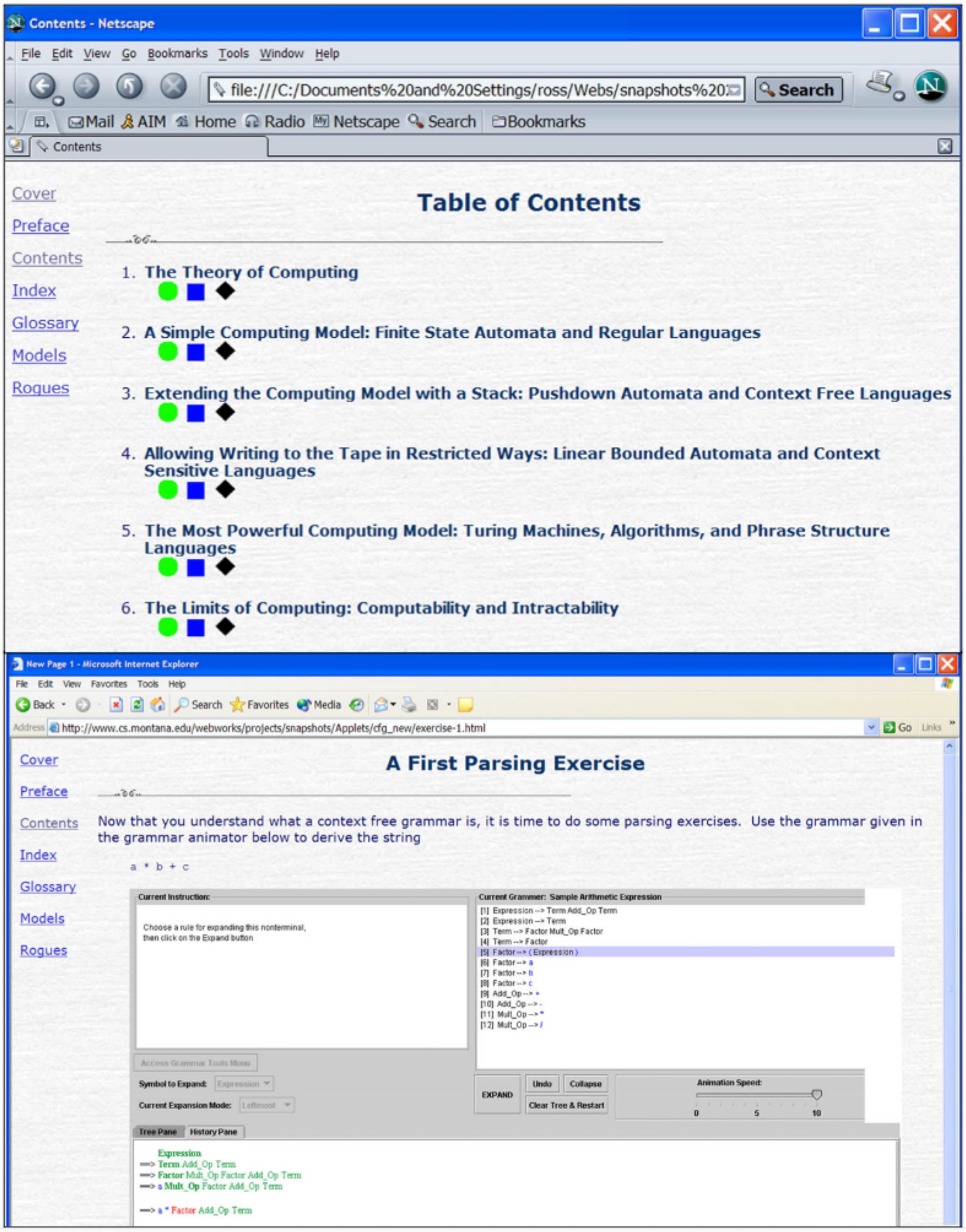

Figure 3.2.: Table of contents and example page of the hypertextbook. [CGG+05]

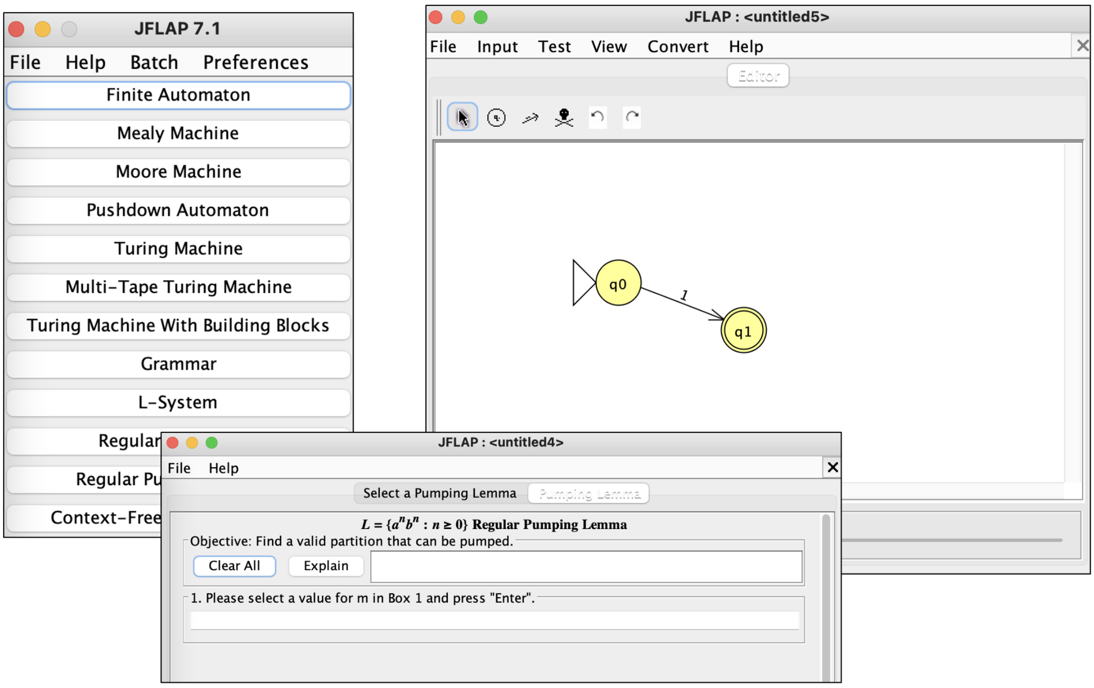

Figure 3.3.: The screenshots show the possibilities of JFLAP as well as an constructed automaton and the first step of solving a Pumping Lemma assignment step-by-step with JFLAP.

postconditions" [ibid., p. 56]. The aim is to introduce students to the ideas and concepts of formal proof in program development as early as possible. Nonetheless, he states that it may be possible that students will understand their approach as another "Java programming" book and not benefit from the improvements.

Berque et al. (2001) offer another idea by describing a course taught in an electronic classroom equipped with pen-based computers, a touch-sensitive electronic whiteboard, and locally written groupware [BJJ01]. The focus is on concepts and topics that are difficult to describe orally or can hardly be communicated using a keyboard. This course concept aims at improving the ability of teachers and students to share written information. With this approach, students can use their own pen on their own display, which is then transferred to the lecturer's display for all to see. By sketching, they can ask questions more easily and specifically. Furthermore, when the teacher writes something on his whiteboard, it is automatically transferred to the students' tablets, so that they can make personal annotations with their pens. Through this approach the students can be kept as actively engaged as possible with the course material during class time. This approach was also used in ToC courses, such as when explaining Deterministic Automaton (DFA). (Unfortunately, the costs to acquire additional technology like this should not be underestimated.) The authors conducted surveys for several courses but not especially for the ToC topics. For other courses, the results were as follows. When tablets were used for this approach, students could often become just as distracted as with personal devices, since tablets can also be used for web browsing and instant messaging. This outcome also depends on the teaching abilities of the lecturers and how well they keep the students motivated and involved. Nevertheless, students found this distraction to be similar to traditional courses and would still recommend the course to others.

3.2.3. Automated assessment systems

Automated assessment systems are also frequently used to give students immediate feedback for their approaches and ideas and to structure their solutions. For example, García-Osorio et al. (2008) present their version of a tool (which they called "Thoth") to support the teaching of formal languages and automata theory [GOMSJVGP08]. They argue that students have problems with ToC because the topics are too theoretical and difficult. Thoth supports regular expressions, finite automata, context-free grammars, push-down automata, and Turing machines. With the help of the graphic user interface, students avoid doing the exercises by hand and are free to experiment with their step-by-step development of algorithms more easily and quickly. Unfortunately, I could not test the system due to its unavailability on the given website.

Crescenzi et al. (2013) – see also [EKN$^+$11] – focused their work on the concept of NP-completeness [CEK13]. The authors report how students have trouble seeing the usefulness of the subjects and especially of computational complexity and related proofs. For example, they see the problems with proving reduction as existing on three levels: "to come up with the idea for a reduction, to prove that the reduction is correct, and to describe what the implications of the existence of a reduction are" [CEK13, p. 16]. Accordingly, every level has its own challenges: "getting the direction of the reduction right in a proof is considered hard. Now, getting an idea for a reduction is very similar to getting an idea for any algorithm that we want to design. Proving it to be correct is connected to mathematical skills and knowledge of proof techniques" [ibid., p. 16]. The authors modified their courses with various activities to improve the learning of computational complexity. In particular, they used "Kattis"[7] – also an automatic programming assessment system – which could give students direct feedback on their solution. Furthermore, they used "A1ViE"[8] – an algorithm visualization system. They tested their approach in an ADC course. At the beginning, they asked former students who were now working in industry to provide real-life examples and discuss the relevance of computational complexity to the students attending the current course. After evaluating a questionnaire, they found that demonstrating the concept of reduction in smaller steps would be beneficial for students, and they suggest teaching the three levels separately and showing how they relate to each other.

Devedzic et al. (2000) also describe an intelligent tutoring system with the basic idea of systematically introducing students to the topics of FLAT [DDP00] – FLUTE (Formal Languages and Automata Environment)[9]. At the time of their research, no other similar intelligent tutoring systems had been reported for this CS field. The authors state that students' motivation is low due to missing practical applications. Therefore, they aim to provide a number of examples. Their goal is to accomplish this in accordance with both the logical structure and the

[7] https://kth.kattis.com, last visit 26 September 2021
[8] Unfortunately, the system is not available anymore
[9] Unfortunately, the system is not available anymore

personal background knowledge and learning abilities of each student. This approach is not a substitute for human teachers, but it does help students to learn individually.

Interactive Theorem Provers (ITP) are a special group of automated assessment systems that focus only on proofs and theorems. Existing research on ITP shows how the overarching aim is to provide proactive, automated, and (in the best case) instant feedback on formal proof tasks and to support students in using formal notation. Mostly, ITP are used in advanced courses or by advanced users and experts. The approaches for introductory courses are very similar overall, with different ITP adapted for various proof types to scaffold the outline of a proof. Furthermore, they use IDEs comparable to typical programming IDE – that is, by using different windows. The user can manually step through the proof and receive instant feedback on the specific steps. Because of the similarity, I do not provide further details here for every available ITP or include the research in Table 3.2. Instead, I only provide an overview of examples that have been used and developed for CS and in mathematics introductory courses, as seen in the following:

- Billingsley and Robinson (2007) explore the use of the automatic proof assistant Isabelle/HOL[10] as a model to support first-year exercises in which students write proofs in number theory. They also made adaptions to the interface to assist novices in learning to use automated proof assistants [BR07].
- Summer and Nuckols (2004) present the EPGY Theorem Proving Environment [SN04].
- Autexier et al. (2012) adapted the proof assistant system Ω*mega* for teaching textbook-style mathematical proofs [ADS12].
- Knobelsdorf et al. (2017) used the ITP Coq[11]. The authors based their adaption on an "infomation hiding" principle to make Coq usable with predefined procedures even for novices [KFBK17].

Overall, these approaches were evaluated on a small scale and demonstrated that ITP can be used to support how students learn to develop formal proofs through immediate and constant feedback. Nonetheless, the approaches have not been

[10] https://isabelle.in.tum.de, last visit 26 September 2021
[11] https://coq.inria.fr, last visit 26 September 2021

widely used, among other reasons, because of the lack of expertise to adapt such a system.

3.3. Students' Difficulties with Specific Topics and Assignments

Sections 3.1 and 3.2 show how several suggested pedagogical approaches build on the assumption that students have problems understanding the mathematical nature of the subjects and, therefore, are presumed to be inadequately motivated to study ToC concepts. Apart from seeking to improve the situation in ToC courses by building on unexplored assumptions, a few case studies have investigated students' actual difficulties in ToC; but these studies have been limited to specific topics or concepts of related courses. In the following, I provide an overview of this existing research.

Armoni et al. (2006, 2009) conducted several studies to analyze how students work through reduction proofs. In [AGEH06] and [AGE06], the authors present a study that addresses the reductive thinking of undergraduate CS students in different contexts – that is, whether they tend to use reductive solutions (around 60 students) or whether they transfer their reductive thinking from algorithms to the area of formal language theory (around 30 students). Based on the previous studies, Armoni (2009) used quantitative methods to analyze a larger population [Arm09]. She analyzed students' home assignments and final exams concerning which questions could be solved with reduction. Overall, the results of these studies suggest that students experienced difficulty when thinking at a high level of abstraction and understanding reduction as a method for problem solving. Furthermore, they were particularly challenged when having to apply formal reasoning methods appropriately.

Gal-Ezer and Trakhtenbrot (2006, 2013, 2016) recently reached the same conclusion after researching students' misconceptions about reduction proofs and pumping lemma assignments. They built their research and assumptions on experiences and observations in their courses. In [GET16] and [Tra13], the authors presented reduction-related misconceptions that have been observed during five academic years of teaching the course "Theory of Computation and Complexity" with 650 students involved. In [GET06], the authors used a similar approach for misconceptions related to the pumping lemma. Both studies aimed to provide a way to help students understand and avoid the detected misconceptions as early as possible. Therefore, the authors developed a series of instructive examples and used them proactively in tutorial sessions for both topics.

Smith and McCartney (2013) analyzed the skills students lacked at the end of a FLAT course for successfully solving formal proof assignments of regular languages [SM14]. Accordingly, the authors used the final exams of 42 students as a data source and analyzed the student errors. Students seemed to have trouble with applying reasoning involving quantifiers, understanding symbolic formulations, and forming abstractions.

Pillay (2009) used document analysis to consider the solutions to three tests and weekly tutorials in a course on FLAT for 13 students [Pil10]. She then reported the differences experienced by students in learning regular languages, transducers, context-free languages, and Turing machines. It was also noted that a "general difficulty experienced by students was the conceptualization of proofs to theorems such as the Pumping Lemma" [Pil10, p. 49]. Nonetheless, the main difficulty experienced by students was problem solving. She reported that often visualization software assists students and these tools do not directly aid them in further developing their problem-solving skills to the level necessary for FLAT courses.

Based on an interview with two undergraduate students working on a complexity assignment requiring big-O notation, Parker and Lewis (2014) also found that students primarily struggled with abstract mathematical function usage rather than the concept of complexity itself [PL14]. Comparable results have been revealed by an observational study conducted during my master's thesis (see Chapter 1). Thereby, I analyzed how student groups consisting of second-year CS students worked on an assignment about NP-completeness [KF16].

Most studies that have examined student difficulties either in homework or final exams have used written solutions to assignments as analysis objects. The difficulties of working on homework as compared to completing final exam solutions have not been explicitly compared. Enström (2014) conducted a broader analysis of student difficulties with ToC concepts in a course about ADC [Ens14]. She investigated mainly third-year students and used an automated assessment system, surveys, oral feedback, and student grades to measure student performance. Her paper opens explicitly with the observation that many students "complained that computational complexity, complexity classes, and reductions were much more difficult than algorithms and data structures" [Ens14, p. 52], and she concluded with the finding that students displayed problems with proof assignments and that complexity proofs seemed to be particularly difficult.

3.4. Mathematics Students' Difficulties with Proof Assignments

The results from section 3.3 seem to show that mathematical and formal proofs challenge CS students the most. Due to the similarities between ToC and mathematics, it seemed appropriate to examine (apart from ITP) what research exists that deals with how students of mathematics experience proofs. Overall, there is evidence that mathematics undergraduates also face difficulties with formal proofing and abstraction just as CS undergraduates do.

In research that is often cited, Moore (1994) conducted several studies, including non-participant observation of a class of 16 students, tutorial sessions, and interviews with the lecturer and five of the 16 students. He presented seven major sources of difficulties for students in mathematics education when constructing proofs [Moo94]. Thereby, these sources cover the general structure of the proof as well as a lack of knowledge, for example, the students did not know how to begin the proofs; the students did not know how to obtain the overall structure of the proof; the students were unable to understand and use mathematical notation; the students did not know the definitions; the students had no intuitive understanding of the concepts). Other studies have used a broader data base for further results.

Anapa and Şamkar (2010) surveyed students' perceptions about proving among 444 students (271 students were attending a department of Mathematics and Computer Science and 173 students were attending a Department of Elementary School Mathematics Teaching) [AŞ10]. They conclude from their study that "even the majority of students finding themselves successful in mathematics do not trust their proving abilities". However, students were able to understand proofs when they examined it [AŞ10, p. 2706]. These results suggest that students cannot fully benefit from the proving methods. In addition, students who were positive about proving were more likely to visit the math and CS department and perhaps aspire to become professional mathematicians in the future.

Almeida (2000) also conducted a survey a few years earlier [Alm00]. They asked 473 undergraduate mathematics students subjective questions and statements (e.g., "I am confident in my ability to prove results for myself") as well as content-related questions (e.g., "A proof in mathematics both verifies and explains"). Furthermore, 25 students volunteered to be interviewed after the questionnaire and discuss the reasons for their responses. These students were categorized into four types considering their acceptance of the need to work with formal proofs and formal or informal proving usage. Overall, the author found that students "favour" visual methods of proving instead of formal proofing with mathematical inscriptions [Alm00, p. 879].

Stylianou et al. (2015) present the results of a study of 535 undergraduate students [SBR15]. The purpose of the study was to describe students' views about the importance of proof and to find out how these views are related to their attitudes and beliefs about proof as well as their experiences with learning proof in the classroom. The study consisted of several questionnaires (e.g., demographic questions, beliefs about proof, and prior classroom experiences) and a multiple-choice test (on students' views about the role and functions of proof and competence in evaluating what can be considered an "example" of a mathematical proof). For fewer students, the authors also examined a written test (requiring students to construct proofs) and an interview. Among other findings, the authors noted that "high-performing students tended to hold a more positive and active stance with respect to their beliefs about proof than their low-performing counterparts."

Weber (2001) showed that undergraduate students often know and can apply the facts needed to prove a statement but are still unable to prove it [Web01]. He used two groups of participants (four undergraduates in CS and mathematics and four graduate students completing their dissertations in an algebraic topic). He gave them a list of sentences and observed their proof attempts in sessions where their process was to be "thought out loud." The students were then questioned on various details about prior knowledge or solution methods. Based on these studies, Weber constructed different types of strategic knowledge that graduate students possessed and that undergraduates seemed to lack: (1) knowledge about the proof techniques of the field, (2) knowledge about which theorems are essential and when they are helpful, (3) knowledge about when to use "syntactic" strategies and when not to. The analysis in this study suggests that one of the leading causes of student failure may be a lack of strategic knowledge.

Mejia-Ramos et al. (2012) describe assessment questions on how to assess student understanding [MRFW$^+$12]. The first three assessment types referred to students' understanding of only one or a small number of statements within the proof. The remaining four assessment types related to students' understanding of the proof as a whole. The authors found that despite the importance of understanding proofs in advanced mathematics courses and the widespread complaints that students do not understand the proofs they read, there is little empirical research on this topic.

Based on these and similar results, pedagogical approaches were proposed to address the difficulties of mathematics students. One option has been to lower the formal level for beginning students in developing proofs using generic or pseudo-proofs. For example, Biehler and Kempen (2013) describe a bridge course on logic, proof methods, inductions, and functions to connect high school mathematics and university mathematics [BK13]. Therefore, they use generic proofs to enable students to find the general argument and understand the main idea of a proof. For their study, they examined the tasks and solutions of 64 students. It turns out that few students understood the idea of a generic example and had problems understanding the explanatory power of generic proofs. In addition, when moving to formal proof, students had difficulty with the formal language of mathematics, symbols, and the meaning and definition of variables.

Another line of research suggests that in addition to developing proof, students should explicitly learn to read and interpret given proof by evaluating fellow students and assessing their own written proof in peer reviews. For example, Powers et al. (2010) pursued the idea that asking students to validate proofs can improve their own proof-writing skills [PCG10]. To investigate this idea, the authors divided a group of 40 mathematics students. One group was taught a proof validation activity once a week, and the other group was the control group. The results of three examinations were compared. One finding was that the validation group performed significantly better on the proof-writing tasks on the final exam after the control group had performed even better on the second exam. This result underscores the authors' experience that the benefits began later in the course and were not evident initially. Ernst et al. (2014) provide another example by using student peer review as a pedagogical method [EHS15]. Students had to submit proofs twice during the semester, which were peer-reviewed by other students (including students from another class) and returned to the author. A survey was administered at the end of the course to determine student perceptions. Overall, the process of writing peer reviews was perceived to be more helpful than receiving the reviews. Many students indicated that they could better evaluate their own proofs as a result of the peer review.

Overall, it should be noted that mathematicians are more likely to understand the benefits of mathematical and formal proofs for their futures than is the case for CS students who take far fewer mathematical and formal courses. Furthermore, how I compiled before, using such formalised mathematical symbols and methods is only one tradition of CS (see Section 2.1.1) but a core of mathematics. Therefore, the results from these two disciplines are only comparable to a limited extent. Nonetheless, this research can likely be mutually beneficial if consciously consulted and adapted to these considerations.

3.5. Summary

Overall, there are two general assumptions about the reason for why students struggle with ToC stem. First, many of the approaches presented assume that students are often unsure of the relevance of theoretical topics to their further studies and careers. Some approaches suggest how to modify and extend existing approaches by linking theoretical concepts to other CS courses or practical examples. However, these approaches have not gained acceptance and are often not sufficiently evaluated. Possible reasons for this could be the need for teachers from different courses to work together or the lack of prior knowledge of specific methods used. The structure of the curriculum can also be an obstacle here.

Another common assumption is that students lack motivation and interest in abstract or mathematical topics. Various software, systems, and tools have been developed for use in teaching the topics more interactively and thereby increasing motivation and engagement. However, even this use has limitations: On the one hand, it would be desirable that the software, systems, and tools could be used in the long term, but this requires reliable updates and support. On the other hand, these tools should not depend on the respective teacher, so there should be a sufficiently broad level of expertise available within educational institutions. Furthermore, there is also the question of how far the software, systems, and tools can be adapted to the needs of the students in own courses, if this is necessary. In addition, software, systems, and tools are usually targeted to specific individual theoretical topics or types of proofs, such that several approaches must be used for different topics. Therefore, the effort expended for only one or two courses in the CS curriculum is quite high. In addition, the approaches must not be for a specific platform, as this would exclude students.

All the described approaches offer elaborated pedagogical solutions for engaging CS students for ToC and lowering attrition and failure rates by incorporating not only different ways of teaching and learning ToC but also alterations in course content. Nevertheless, the approaches have not gained widespread acceptance for several reasons, as noted. Furthermore, most of the existing approaches were developed with the assumption that students' difficulties with ToC are mainly caused by a lack of interest, motivation, or inability to understand the relevant concepts and theorems due to the abstract and formal nature of computation. These assumptions are often based on anecdotal oral feedback or on surveys that provide feedback after the courses are completed. This leads to the fact that none of the assumed student difficulties have been empirically validated in ways that would inform pedagogical approaches with detailed insights about the actual nature of student difficulties.

In the studies that look more intensively at the difficulties of individual topics, it is also difficult to trace how the topics were arrived at and whether it is just based on subjective observation and teacher experience. Only one study went so far as to look at all the topics and assignments of a course. Although this study found that students were particularly challenged with complexity proofs, it only looked at one course and did not include the full range of ToC concepts and associated proof assignments – especially FLAT. Furthermore, the study did not further disaggregate the results – that is, questioning whether all students or only those who failed final exams had experienced difficulties with proof assignments.

Also, in mathematics education, it is experienced that students have particular problems with proofs. Various suggestions have been made over the years as to how student understanding and performance can be improved, but here, too, no single approach has prevailed. Nevertheless, it should be noted that mathematicians are more likely to understand the benefits of mathematical and formal proofs for their futures than is the case for CS students who take far fewer mathematical and formal courses and where these topics are only one of the underlying traditions. Therefore, the results from these two disciplines are only comparable to a limited extent.

Overall, I found a lack of additional studies as a call to action for CS education research in this field. I argue that a student-oriented research approach that conducts a detailed investigation into student difficulties with ToC will provide more sustainable information than the general assumptions that form the basis of current pedagogy. Furthermore, to decidedly focus pedagogical actions in ToC courses on students' requirements, I found that differentiation among student performance is particularly relevant.

Part II.

Student Performance in Final Exams and Homework

4. Study Design

In the following, I provide background on the study design, starting with the research questions and information about the course setup, the data sources and quality. After presenting the research methods, I explain how the data analysis was conducted.

4.1. Research Questions

As presented in Section 3, pedagogical approaches and tools have been introduced to improve undergraduate education of Theory of Computation (ToC). Most of these approaches were developed with the assumption that students' difficulties with ToC are mainly caused by lack of interest, motivation, or inability to understand the relevant concepts and theorems due to the abstract and formal nature of computation.

Nonetheless, a few single case studies (e.g., [GET16] [KF16] [PL14]) provide detailed insights into students' issues with ToC and indicate that students do not seem to lack engagement and interest but rather training in formal methods. In these studies, the choice of topics is not necessarily comprehensible or based on subjective teacher feedback. To provide data about topics which students actually have the most difficulties with, Enström (2014) provided an extensive quantitative study within an Algorithms, Data structures, and Complexity (ADC) course. However, this study considered only one course and did not include the full range of ToC concepts and related proof assignments, especially Formal Languages and Automata (FLAT) (cf. [HMU01] [Sip12]). To disaggregate such results about student performance further and discover whether all students or just those failing final exams showed difficulties with proof assignments, I intend to reinvestigate with a broader basis of data what is known so far about student difficulties in ToC, focusing explicitly on FLAT. For that matter, I conducted a study analyzing student performance in an introductory FLAT course. In particular, I examined students' potential difficulties across all ToC assignments covered in the course, disaggregating results according to student performance in final exams. Therefore, I started my doctorate project with the following two research questions:

- **RQ1: What kind of assignments usually covered during an ordinary, undergraduate introductory FLAT course are causing students the most difficulties?**
- **RQ2: Are there differences among low and high performing students, especially regarding these potentially different assignments as questioned in RQ1?**

4.2. Course Setup

As mentioned in Section 2.2.2, the pedagogical approach in ToC courses traditionally consists of regular lectures and tutorial sessions accompanied by homework assignments that students are supposed to work on together in small study groups without the support of the teacher/tutor [KKB14, pp. 68 – 69]. For this reason, I wanted the course I was analyzing to also include these components.

The chosen course introduced FLAT and is offered annually at the Technische Universität Berlin in Germany. This course is mandatory within the Computer Science (CS) undergraduate program and is usually attended by about 500 students, most of whom are CS majors in their first year. While the first half of the course is concerned with mathematical topics (i.e., sets, logic, and functions), the second half covers actual ToC topics (i.e., mappings, word, grammars, and automata). For a detailed list of all topics, please see Table 5.7. Within the CS undergraduate program, students are required to attend an introductory mathematics course on the topics of analysis and linear algebra, and most do so in the first year of their major. Overall, the FLAT course consists of the following components (which have already been partly described in Section 2.2.2):

- A 90-minute optional lecture per week. A lecturer presents the course topics, central concepts, algorithms, and their proofs and illustrates them with examples using slides, live annotation, and a blackboard in a lecture hall. All material is available online.

- A 90-minute optional tutorial session per week. In this session, a teaching assistant solves practice assignments, visible for everyone on the blackboard, with the help of about 30 students. Afterwards, sample solutions for the presented assignments are given to the students. Usually, a larger number of assignments (and corresponding sample solutions) are available than what can be discussed in the tutorial sessions. Therefore, the teaching assistants focus on covering every topic and discuss various topics and assignment types.
- Students can receive additional support by attending consultation hours with teaching assistants and using available learning videos.

To pass the course, students must collect the so-called portfolio points that sum up to their final grade (with 100 as the maximum portfolio points and 50 needed to pass the course). Portfolio points can be gained in the following three components:

1. Students can submit four sets of homework assignments during the entire course, with each set containing a different number of assignments (30 – 31 assignments in total). These submissions are made in small self-organized study groups, consisting of two to four members. Every homework set makes up to five portfolio points, adding up to 20 portfolio points in total for all homework assignments. All homework in total represents 20% of the overall course grade.
2. An additional 30 portfolio points can be gained in an optional online multiple choice test in the middle of the course, covering the mathematical basics from the first half of the course. The test represents 30% of the overall course grade.
3. The final exam at the end of the course represents 50% of the overall course grade and mainly covers the topics from the second half of the course.

4.3. Data Sources and Quality

For the first half of my dissertation project, I specifically chose to work with the same type of data available to teachers each year – homework and final exam results. Student homework in question provides formative information about student performance during the course, while the final exam results serve a summative function. Overall, I used data from three student cohorts in 2016, 2017, and 2018 (around 1500 students in total) to compare performance and validate any findings. The data was only available in paper form, so I traveled to Berlin several times and went through each document. I entered the assignment points into an spreadsheet of the software Microsoft Excel[1]. Thereby, I anonymized the names of the students but made the results of the homework and the final exams assignable to each other.

In the three years of data gathered from the course, there were two minor changes from the 2016 course presented in Section 4.2:

1. **Lecturer:** In 2017, another person gave the lecture than in 2016 and 2018. The lecturer in 2016 and 2018 has already held the course several times before at Technische Universität Berlin, while the lecturer in 2017 has held a similar course once at another university in 2014.

2. **Assignments:** Compared to 2016, one of the assignments covering functions and mappings was removed in 2017 and 2018.

In Table 4.1, I present the overall numbers of students attending and passing the course in 2016, 2017, and 2018. In the following, I give the statistical data in more detail:

- For the year 2016, independent from the homework submissions, 419 students attended the final exam at the end of the course. In sum, the course suffered from a dropout rate of 25%. Among the 419 students attending the final exam, 296 (70%) passed it with more than 49% of the possible exam points. Regarding their portfolio points, 339 students passed the entire course, which amounts to 59% of the original 571 students starting the course.

- For the year 2017, 407 students attended the final exam. The course suffered from a dropout rate of 29%. Around 303 students (74%) passed the final exam with more than 49% of the possible exam points. Overall, 365 students passed the entire course of the originally 580 students starting the course, which totals 63%.

[1] https://www.microsoft.com/de-de/microsoft-365/excel, last visit on 10 October 2021

- For the year 2018, 399 students attended the final exam. The course suffered from a dropout rate of 23%. Around 73% of all students passed the final exam with more than 49% of the possible points. Overall, 66 % of the originally 521 students passed the course.

At this point, it should be explained why Table 4.1 shows more students who have passed the course than there are who have passed final exams. As described in Section 4.2, this course has a portfolio exam. This means that students have to achieve 50% through the three components of homework, online multiple choice test, and final exam to pass the course. Therefore, they can also get the required 50% through homework and the online multiple-choice test and do not have to reach 50% in the exam itself. Since in usual scenarios a final exam is understood as passed if 50% of the points are achieved, I have listed these numbers here.

Table 4.1.: Statistical data about student participation in homework and final exam

	2016	2017	2018
Submitted first set of homework	571	580	521
Submitted last set of homework	497	468	443
Attended final exam	419	407	399
Passed final exam	296	303	293
Passed entire course	339 (59%)	365 (63%)	345 (66%)

Nevertheless, as is evident from these statistics, all numbers were quite similar each year. Even if students did not submit every assignment within one set of homework, I still considered their data in my analysis as long as they submitted every homework set. This consideration was necessary because only a minority of students submitted all assignments from all four sets of homework, and I do not have reliable insights into why students did not work on all assignments. Possible reasons could be, for example, that students did not understand an assignment, did not know what to do to solve it, or lacked time, interest, or motivation.

Discussing how representative the data is, it should first be emphasized that, to the best of my knowledge, no data about student performance in FLAT courses on a larger scale (nationwide or worldwide) have been published yet. According to the corresponding FLAT course's long-standing lecturer, the overall numbers from Table 4.1 (especially the passing rates) correspond to past performances in the particular ToC course at Technische Universität Berlin. Moreover, the teaching experiences of myself and my colleagues as well as the exchange with different ToC teachers from different universities confirm that passing rates in introductory ToC courses of undergraduate CS programs at German universities tend to be relatively low (less than 50% are not unusual). Among the CS undergraduate students at Technische Universität Berlin that were examined from 2016 to 2018, there was no additional preselection for the CS undergraduate program. Furthermore, lectures, homework assignments, tutored study sessions, and final exams are typical pedagogical elements of ToC courses used in CS undergraduate programs at German as well as many European and Anglo-Saxon universities (cf. Section 2.2.2). Hence, the ToC course at Technische Universität Berlin can be regarded as representative of those undergraduate CS programs at universities sharing the described framework.

4.4. Research Methods

In this subsection, I present the research methods I used during the quantitative analysis. Accordingly, I describe Explorative Data Analysis as well as Qualitative Content Analysis. Furthermore, I present the statistics software SPSS and the statistical evaluations that were conducted.

4.4.1. Explorative Data Analysis

In data analysis, three basic tasks of statistics can be stated: *Describing* (Descriptive), *Searching* (Exploration), and *Inferring* (Induction). Each of these tasks corresponds to a subfield of statistics. Thus, descriptive statistics is dedicated to the description and presentation of data. Exploratory statistics deals with finding structures, questions, and hypotheses, while inductive statistics provides methods to draw statistical conclusions using stochastic models [FHK$^+$16, p. 10].

For my research design, I used Tukey's (1977) model for *Exploratorative Data Analysis (EDA)* [Tuk77], but also provide descriptive information about the data when necessary. Similar to qualitative approaches like grounded theory [BM15], EDA explores a specific research field or observed phenomenon when there are no hypotheses, models, or broad insights available. The key difference between

EDA and qualitative approaches is that EDA uses numerical data to conduct "numerical detective work" [Tuk77] and summarizes several (though not final) ways to explore data. Typically, a readily available sample is used to support answering open questions instead of hypothesis testing. In contrast to limiting the analysis of data, which would be necessary to analyze a hypothesis, EDA techniques are used to provide a clear picture of the whole data set as completely as possible [BDYL12].

One way to begin EDA is to summarize the data, clarify the general data structure, and generate ideas for further analysis steps [Cha86] [Mor09]. Therefore, descriptive information (e.g., mean, deviation) can be calculated for the whole data set and subgroups. Moreover, data plotting is a valuable way to give an overview of the data and make anomalies visible. Commonly used diagrams are histograms and scatterplot diagrams, depending on the data and questions. This combination of different kinds of exploration techniques can be used to build theories and hypotheses by a detailed overview of the research topics, thus enabling further analyses. Therefore, EDA helps to create an overview of a broad research field and may be appropriate for getting closer to this corresponding research field [DB16].

4.4.2. Qualitative Content Analysis

As an established method in German-speaking countries, *Qualitative Content Analysis (QCA)* was used for my research, as developed by Mayring [May14].

In general, Mayring (2016) developed QCA as a coding method that is used to systematically process different data sources (e.g., texts) and to make them describable and verifiable by assigning so-called "categories" to the data material (also called "coding" or "assigning codes"). Such a categorization must be defined beforehand by a selection criterion and level of abstraction; both of these depend on the focus of the research question(s). The next step is to work through the material line by line. Once a suitable passage has been found, a category is constructed for it. An abstracted term or sentence, formulated as closely as possible to the material, is used for the category name. Further suitable text passages are then assigned to this category. If there is a text passage that does not fit the category, a new category is created. After going through the material, it is necessary to check that the categories fit the research focus and do not overlap. If changes are made to the category system, the material must be reviewed again [May16, pp. 115].

In *deductive* category applications, coding is done by explicitly defining categories in advance or applying existing categories, and in *inductive* category formation, inductive categories are grouped into main categories. In the end, the categories represent the core aspects in a short form and are captured in a category system. The categories thus created can also be processed quantitatively if necessary – for example, by the number of occurrences [May20, p. 497].

In general, there exist three different QCA procedures with different objectives for applying or creating a category system [DB16, p. 542]:

1. **The summarizing QCA:** The aim is to inductively reduce and categorize more extensive documents by their core statements.
2. **The explicative QCA:** The aim is to make unclear text passages understandable by looking at the direct text environment and context. Therefore, while summarizing QCA reduces the source material, explicative QCA partially expands it.
3. **The structuring QCA:** Similar to quantitative data analysis, numerical data are obtained with the help of a previously defined category system.

For the analysis of journals, newspaper articles, transcripts, guided interviews, or group discussions, summarizing QCA is recommended, as it is a structured way to condense texts to their core statements. Due to the amount of textual and mathematical documents that I have used during my doctoral project, I also decided to use a summarizing QCA and, therefore, to present the procedure in the manner described below. Usually, four steps are followed [DB16, p. 542]:

1. Paraphrasing: Text passages that are important to the content are reformulated into a short form.
2. Generalization: The paraphrases are brought to a previously defined level of abstraction.
3. First reduction: The relevant paraphrases are chosen by deleting paraphrases that have the same meaning or are unimportant.
4. Second reduction: The remaining paraphrases are bundled and integrated, thus finally creating new paraphrases that summarize the main contents.

4.4.3. Statistical Evaluations

This section provides an overview of the statistical procedures and calculations used during the explorative data analyses.

Analysis of Variances (ANOVA)

As mentioned earlier, three cohorts were studied, which made it possible to compare and validate the results. I used a statistical data analysis to determine if the data samples were statistically significantly different to obtain a robust result. For two data sets, one can use the so-called *t-test* to check whether the means of two groups are statistically significantly different (see [RFHN14a, p. 34]). However, if more than two data sets should be compared, as I do, *Analysis of Variance (ANOVA)* should be chosen because it can overcome the disadvantages of the t-test.

More precisely, ANOVA can compare multiple means within one test, and the test power is many times higher than that of the individual t-tests [RFHN14b, p. 4]. Since I wanted to compare three consecutive years, ANOVA was the appropriate method. The method's name comes from the fact that the simultaneous mean comparison is performed by looking at different variances. For the statistical and mathematical background of the method, I refer to the numerous introductory literature because by using the software SPSS Statistics the calculations were performed automatically after entering the data. In the following, I explain a few terms that I use during the analysis [RFHN14b, pp. 19]:

- **Independent variable:** The variable according to which the subjects are assigned to the different groups (also *factor*).
- **Dependent variable:** The variable to be measured. The variable is also used to check whether the requirements for a statistical procedure are met.
- **Interval scaled:** There are different types of scales. While the *nominal scale* can only make a statement about the equality or difference of the values, the *ordinal scale* can be used to make a greater – smaller statement (without it being possible to say how great the differences are). With the *interval scale*, the distances between the expressions are specified. It can therefore be assumed that a certain distance (interval) always reflects the

same difference between the expressions. An interval scale can always be divided into equal sections [RFHN14a, pp. 7].

- **Normal distribution:** For the normal distribution (also *Gaussian distribution*), about two-thirds of all measured values are within one standard deviation of the mean.
- **Mean:** The mean value represents the average of all measurements. Mathematically, the mean value is the sum of all values divided by their number. The calculation of the mean value requires interval scaled data, since information about the distances between the values is included [RFHN14a, pp. 11].
- **Median:** The median is the value from which all other values deviate the least on average. Mathematically, this means that the median halves the distribution, i.e. there are just as many measured values above as below the median [RFHN14a, p. 11].
- **Variance:** Like the mean value, the variance can only be calculated for interval scaled data. The variance is calculated from the sum of the squared deviations of all measured values from the mean value, divided by the number of all measured values minus one. As a result, the greater the deviation of the measured values from their mean value, the greater the variance [RFHN14a, pp. 14].
- **Standard Deviation:** If the positive square root is taken from the variance, then the standard deviation is obtained. It shows how much the measured values scatter around the mean value; that is, it is an indicator of the width of the normal distribution [RFHN14a, p. 15].
- **Homogeneity of variances:** The variances in each group are approximately equal.
- **P-value:** The p-value indicates the probability that a difference observed in the sample, or a larger difference arose by chance.

Since I had only one factor (the year) I used the one-way ANOVA – two-way ANOVA would be used if there were two factors [FHK+16, pp. 477]. Altogether, to conduct a one-way ANOVA in the first place, several prerequisites must be satisfied [RFHN14b, pp. 30]:

1. **Interval scaled dependent variable:** The dependent variable has to be at least interval scaled.
2. **Normal distribution:** The dependent variable is approx. normally distributed for every data set.
3. **Homogeneity of variances:** There exists a homogeneity of variances.
4. **Independence of measurements:** The measurements of one group are independent of measurements of another group.
5. **Few Outliers:** It should be checked if there are many outliers in the data sets to prevent bias in the results.

When prerequisites 2 and 3 are not met, the more robust ANOVA variant, Welch's ANOVA, is recommended [Mod10].

Post-hoc test

After performing a one-way ANOVA, a single p-value exists that indicates whether any of the groups are statistically significantly different from each other – but not which groups. To find out which groups these are, post-hoc tests can be calculated [RFHN14b, p. 18]. In principle, a post-hoc test makes group-wise comparisons and allows a statement about which means differ significantly in which direction.

When there is a homogeneity of variances, the *Tukey-test* can be used. Otherwise, the *Games Howell post-hoc test* is used [Fie13]. Both tests compare all possible group combinations and identify which combinations are statistically significantly different from each other.

Correlation

A rank coefficient can be used to find correlating values in existing data sets. If the data has no normal distribution, the *Spearman's rank coefficient* is used [dWGP16]. If the data has a normal distribution, the *Pearson correlation coefficient* is used. Instead of calculating the correlation between the data points themselves, the Spearman correlation calculates correlation with the help of order through ranks.

4.5. Data Analysis

In this section, all steps of the actual data analysis are presented using the methods described in Section 4.4. First, I illustrate the software used. In the following, I describe how I used QCA and applied the idea of EDA to gain an overview of the lecture content as well as the data structure. Then, I show which statistical calculations I performed and how I present the results.

4.5.1. Statistics Software SPSS

I used the software IBM SPSS Statistics[2] for the statistical calculations. As one of the world's leading statistical software, SPSS Statistics enables a vast range of data analysis possibilities. Furthermore, SPSS Statistics has an intuitive and easy-to-understand user interface. As a first step, the so-called "variable view" has to be used to define the needed variables (in my case, the students and the number of points for each assignment; Figure 4.1). In a second step, the points for every defined assignment in the so-called "data view" can be inserted or imported (Figure 4.2). Then statistical calculations can be performed.

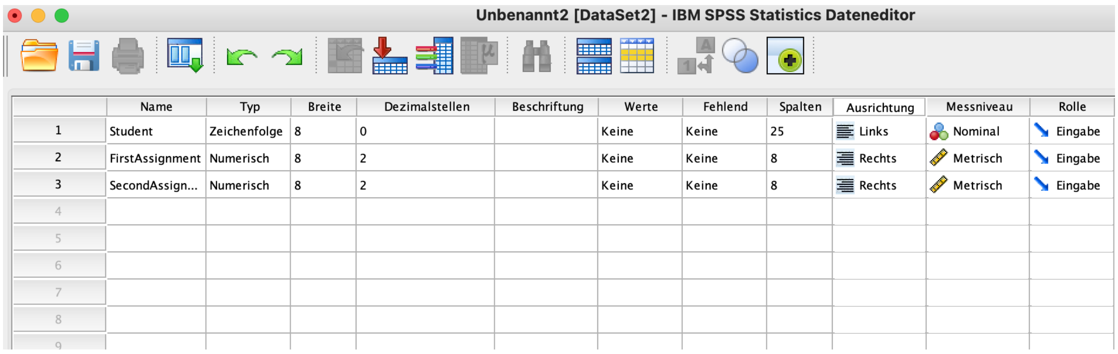

Figure 4.1.: Variable view in SPSS with the variables *student*, *FirstAssignment*, and *SecondAssignment* and the respective settings.

[2]https://www.ibm.com/products/spss-statistics, last visit 16 July 2021

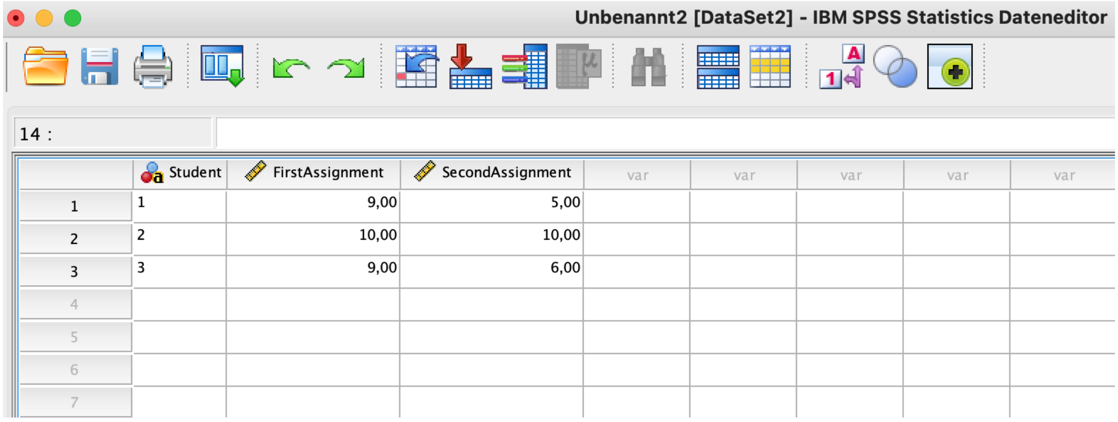

Figure 4.2.: Data view in SPSS with example values for the defined variables.

4.5.2. Overview of the Topics and Assignments

As a first step in my data analysis, I wanted to get an overview of the course topics and assignments. Therefore, I mapped all homework topics to their superordinate topic from the course formulation. In addition, I used a summarizing QCA (cf. Section 4.4.2) to categorize the homework assignments according to general assignment types. In this way, I obtained an overview of the content distribution of the data.

4.5.3. Overview of the Data Structure

Because of its exploratory nature, there is no explicit theoretical background framing this research field and my research questions. At the beginning of this research focus, there are purely statistical terms without restrictions to keep the possibility open of gathering all anomalies and observations. For the type of analysis undertaken here, I had to make two preparations:

1. Since I focused on gaining a deeper understanding specifically of the high and low performing students, I divided all students into "grading groups," depending on their grade in the final exam. Table 4.2 shows the underlying grading system used for the student distribution.

2. Since each assignment had a different total score, I normalized the homework and final exam scores to compare the scores. For this, I used percent-

ages by assigning the full score of the points as 100% and calculated the average of the reached points for a certain assignment. I used the normalized scores for all calculations.

Table 4.2.: Distribution of grades in year 2016 to 2018

Grade	Points
A	100 – 81
B	80 – 70
C	69 – 59
D	58 – 50
F	< 50

As the proper second step of my data analysis and according to EDA (see Section 4.4.1), I summarized the different data sets of homework and final exam scores. Therefore, I created tables and diagrams with Microsoft Excel to get a first overview of the data structure and to identify patterns.

4.5.4. Performance Analysis and Significant Differences

Before conducting the statistical analyses to examine student performance and significant differences as the third step of my data analysis, I have first transferred the data from Microsoft Excel to the software SPSS Statistics (see Section 4.5.1). I have generated a new SPSS file for each grading group to perform the calculations for each group separately. Figure 4.3 presents an example of how the data looked in SPSS (both original and normalized).

Checking ANOVA prerequisites: Before I could evaluate whether the performance differs significantly between the years, I had to check the prerequisites for the one-way ANOVA to made decisions for the correct methods (see Section 4.4.3):

1. **Interval scaled dependent variable:** The assignment points are interval scaled data, since a given distance between points reflects the same difference each time.

2. **Normal distribution:** With SPSS Statistics, I checked if the data set had normal distribution – Figure 4.4 illustrates how to perform the check in SPSS. I found that not every data set had a normal distribution. Figure 4.5 shows a sample histogram given out by SPSS. Nonetheless, since the total number of individual sets was larger than 30, I could assume (according to the central limit theorem) that the distribution approached normal as the sample size grew larger [Sac12].

3. **Homogeneity of variances:** I calculated the homogeneity of variances with Levene's Test for every data set (see Figure 4.6; left screenshot, check mark at "Test auf Homgenität der Varianzen"). Figure 4.7 shows an example of how SPSS outputs Levene's test results. When the test is significant, there is no homogeneity of variances (I refer to the appendix for the results of Levene's test for every assignment.)

4. **Independence of measurements:** Since the values of one year do not contain information about the values of another year, the data is independent of each other. Therefore, the measurements are independent.

5. **Few Outliers:** I used boxplot diagrams to inspect the number of outliers. Since no more than five outliers were found for every assignment, it can be assumed that outliers have no impact. Figure 4.8 shows an example boxplot for the performance in a homework assignment for group A.

After checking the prerequisites, not every data set had a normal distribution and homogeinity of variances. Therefore, I decided to use Welch's ANOVA and the Games-Howell post-hoc test.

Calculating Welch's ANOVA: During Welch's ANOVA, I used the normalized homework and final exam scores as dependent variables and the years as factors. Figure 4.6 shows the setup as well as the settings to calculate the ANOVA and the post-hoc test using the example of the first assignment from the final exam. I also calculated the descriptive statistics (e.g., mean, median). Figure 4.9 shows an example of how SPSS produces out Welch's ANOVA for the final exam assignment E1 to E6 for group A, divided by years.

During the next section, I present the results in a more concise form as directed by

	E1A	E2A	E3A	E4A	E5A	E6A	Jahr	E1GA	E2GA	E3GA	E4GA	E5GA	E6GA
1	19,00	12,00	22,00	10,50	10,50	7,50	2017,00	100,00	75,00	100,00	61,76	95,45	50,00
2	17,50	16,00	21,50	14,00	9,00	15,00	2017,00	92,11	100,00	97,73	82,35	81,82	100,00
3	19,00	15,00	21,00	7,50	9,00	11,50	2017,00	100,00	93,75	95,45	44,12	81,82	76,67
4	15,50	15,50	18,50	13,50	5,50	12,50	2017,00	81,58	96,88	84,09	79,41	50,00	83,33
5	18,00	14,50	22,00	14,00	10,50	13,50	2017,00	94,74	90,63	100,00	82,35	95,45	90,00

Figure 4.3.: The SPSS table shows scores before and after normalizing the data for five students from 2017. E1A through E6A are the original scores that the students achieved, and E1GA through E6GA are the normalized scores, with 100% representing the full score.

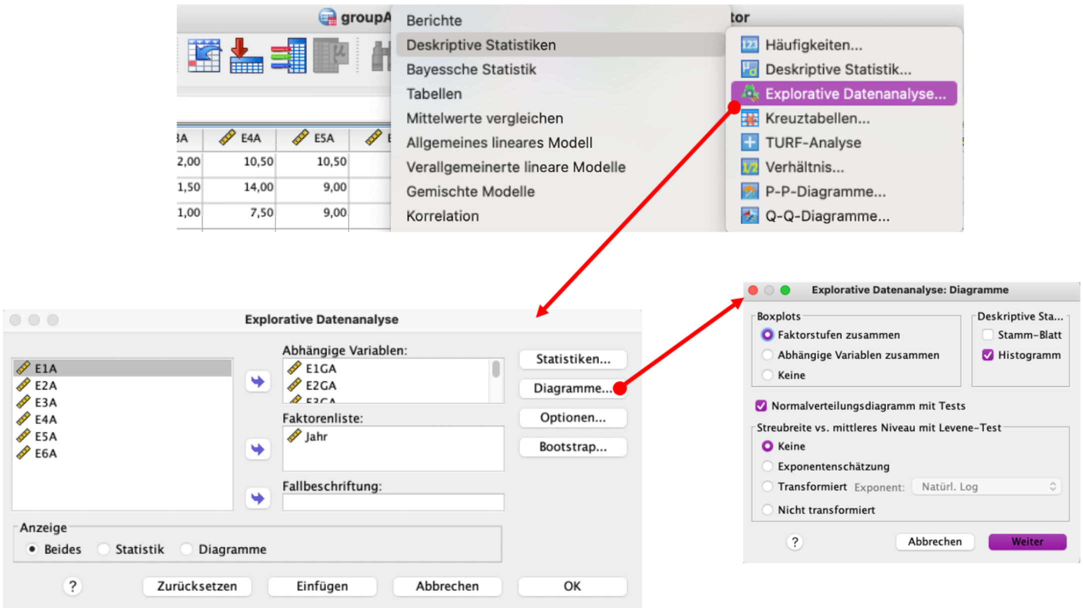

Figure 4.4.: The chosen options and settings for checking the normal distribution using the example of the final exam scores of group A.

APA styles[3]. For better understanding I use the terms "Statistik[a]", "df1", "df2", and "Sig." visible in the table of Figure 4.9 (other analyses may use different but equivalent wording):

$$\text{F(df1, df2) = Statistik}^{a4}\text{ , p = Sig.}^{5}.$$

In the case of the performance analysis, it is sufficient to know that the p-value is used to calculate the significance. All other numbers are used to calculate the p-value. As there was a large number of parallel tests, I decided to use $p < .001$ for all statistical tests.

Examining post-hoc tests: In addition, due to the large amount of data, the results of the Games-Howell post-hoc tests are presented only in summary form. Figure 4.10 presents an example SPSS output of the Games-Howell post-hoc test results. (The detailed results can be found in the appendix.)

Calculating Correlations: After calculating which values differed significantly from each other over the three years, I inspect how homework and final exam performances correlate. Since the data had no normal distribution, I used Spearman's rank coefficient [Bry16, p. 344].

[3]https://apastyle.apa.org/6th-edition-resources/sample-experiment-paper-1.pdf, last visit 03 July 2021

[4]F-Value

[5]Significance

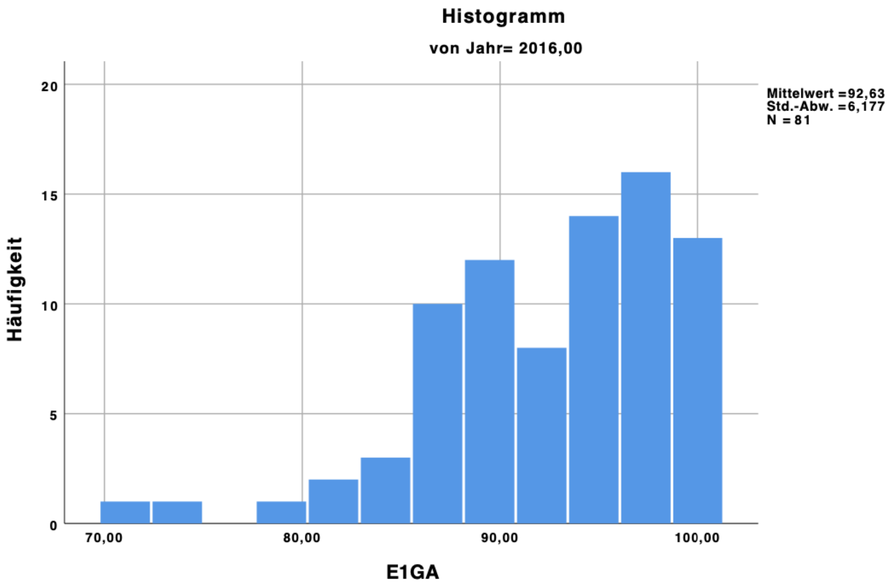

Figure 4.5.: A histogram without a normal distribution for the final exam assignment E1 for group A.

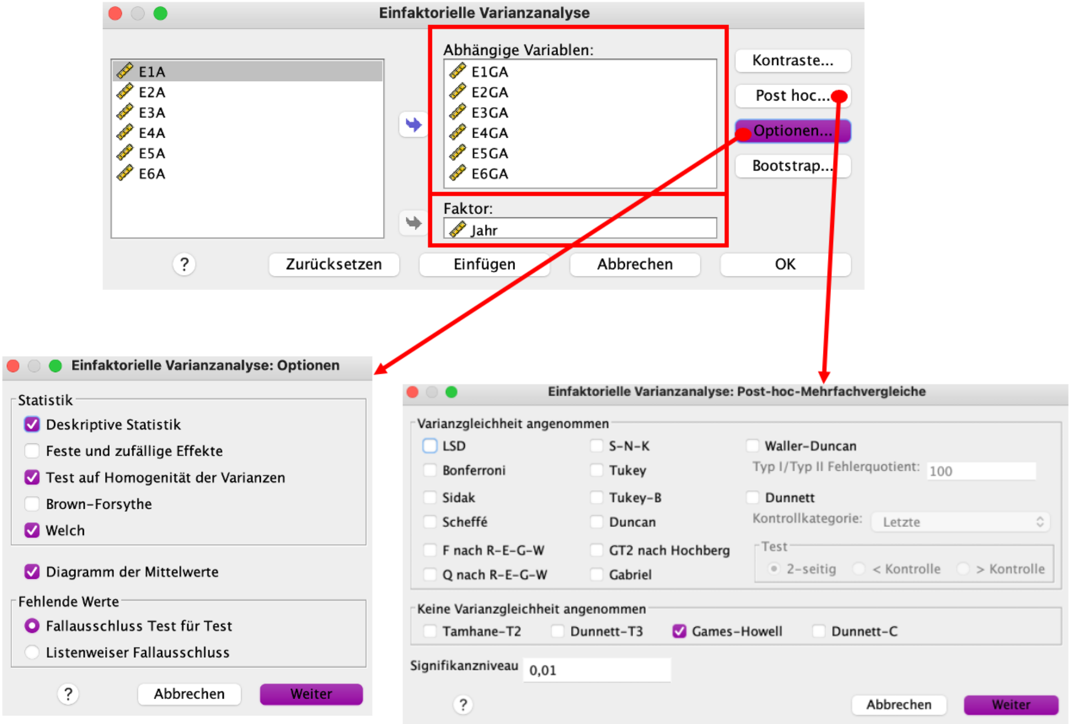

Figure 4.6.: Independent and dependent variables used in ANOVA using the example of the final exam scores of group A. The chosen options and settings for the ANOVA and post-hoc tests can be seen.

		Levene-Statistik	df1	df2	Signifikanz
E1GA	Basiert auf dem Mittelwert	6,951	2	240	,001
	Basiert auf dem Median	4,234	2	240	,016
	Basierend auf dem Median und mit angepaßten df	4,234	2	220,701	,016
	Basiert auf dem getrimmten Mittel	6,163	2	240	,002

Figure 4.7.: An example how SPSS gives out Levene's test results for the final exam assignment E1 for group A.

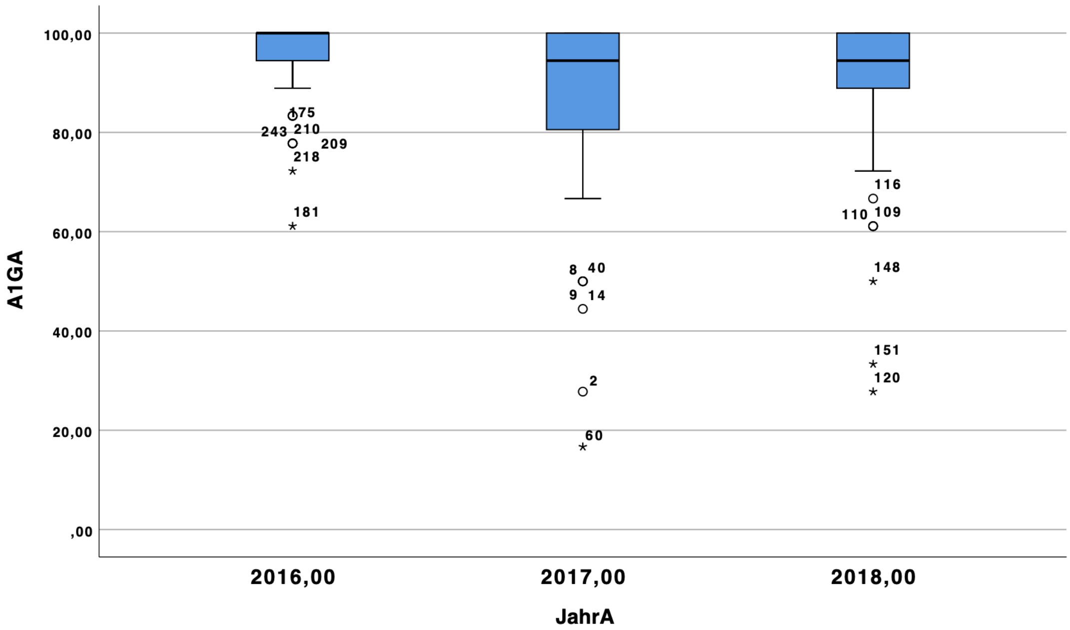

Figure 4.8.: Boxplot diagram for homework assignment A1 for group A divided by years.

		Statistik[a]	df1	df2	Sig.
E1GA	Welch-Test	2,107	2	157,240	,125
E2GA	Welch-Test	6,988	2	156,188	,001
E3GA	Welch-Test	,981	2	159,704	,377
E4GA	Welch-Test	4,151	2	159,353	,017
E5GA	Welch-Test	2,535	2	158,820	,082
E6GA	Welch-Test	12,596	2	155,245	,000

Figure 4.9.: An example how SPSS gives out Welch's ANOVA for the final exam assignment E1 to E6 for group A divided by years.

Abhängige Variable	(I) Jahr	(J) Jahr	Mittlere Differenz (I-J)	Std.-Fehler	Signifikanz
E1GA	2016,00	2017,00	-1,81571	,88324	,103
		2018,00	-1,02736	1,03758	,584
	2017,00	2016,00	1,81571	,88324	,103
		2018,00	,78835	,95626	,689
	2018,00	2016,00	1,02736	1,03758	,584
		2017,00	-,78835	,95626	,689

Figure 4.10.: An example how SPSS gives out Games-Howeel post-hoc test results for the final exam assignment E1 for group A. The plus/minus signs indicate whether an improvement or worsening of the values has occurred (in this case not significant).

5. Analysis of Homework and Final Exam Performance

In presenting the results, I start by giving background information about the analyzed cohorts. Afterwards, I present and discuss students' performance on the final exam assignments. Moreover, I discuss each assignment's topic and how this might be related to students' performances. This is followed by an Analysis of Variance (ANOVA) to determine significant changes in student performances over the three years. I continue this kind of evaluation with students' performances in their homework assignments. In the end, I investigate potential connections between final exam and homework performance and develop hypotheses based on the results. As this is an exploratory data analysis, I decided to present one step of analysis at a time and discuss it immediately afterwards before continuing with the next step.

5.1. The Cohorts

Table 5.1 presents the number of students from the years 2016, 2017, and 2018. For the calculations of mean and median, I used the normalized percentages scores as described in Section 4.5.3. I also present how many students achieved which grade (A – F) in each year.

For 2016, it can be seen that 296 students (70%) of all students attending the exam received a grade higher than F (i.e. received more than 50% of the points). The data also shows that 80 students (20%) scored an A, while the majority of students – that is around 50% – reached an average grade between B – D. The mean and median are in the lower middle range here (C and D). In the following years, the numbers are similar. In 2017, 74% of all students received a grade

Table 5.1.: Distribution of grades in year 2016 to 2018

	2016	2017	2018
Number of Results	419	407	399
Mean	60%	61%	61%
Median	50%	51%	54%
A	80	80	81
B	67	81	72
C	88	80	80
D	61	62	60
F	123	104	106

higher than F with 20% scoring an A; in 2018, 78% scored higher than F, and 20% received an A.

At this point, there already are different narratives regarding how to interpret students' performance as displayed in Tables 4.2 and 5.1. It is not uncommon to explain high scores as well as high failure rates with so-called high and low student performance in the whole course. From that point of view, it is acknowledged that some students are simply very smart and hardworking, while others do not master an understanding of Theory of Computation (ToC) and either do not work hard enough to score better or lack have what is intellectually required to be a top student in ToC. With regard to the research questions, I started questioning this black-and-white picture and wondered whether high- or low-scoring students can actually be related to an overall strong or poor performance. Furthermore, I was questioning whether there are certain assignments that cause all students to gain far less or far more points than other assignments, therefore preventing all students from gaining more points (i.e., a better grade or even passing the course). To understand whether there are certain assignments with a high difficulty for most of the students, I looked into their actual performance on each of the six final exam assignments, as discussed below.

5.2. Final Exam Assignments

In this section, I present the final exam results from the 2016, 2017, and 2018 cohort. Since I started the analysis with the 2016 data first, the respective data

Table 5.2.: Percentage of achieved points for each exam assignment differentiated by grading groups from year 2016

Group	A	B	C	D	F
E1	91%	86%	78%	74%	43%
E2	95%	90%	82%	79%	44%
E3	95%	89%	81%	78%	44%
E4	74%	52%	29%	16%	5%
E5	89%	64%	43%	27%	10%
E6	76%	54%	41%	32%	14%

is presented and interpreted in more detail. The analyses of the 2017 and 2018 data are follow-up studies used to validate the results and determine if there were significant changes.

5.2.1. Final Exam Results from the First Analysis Year (2016)

Table 5.2 shows the values per grading group and assignment number where E1 to E6 corresponds to the six exam assignments. Upon a first glance, the data in Table 5.2 shows how group A performed best in every assignment, confirming their "high performance". This indicates that they have gained "an overall strong understanding" about all ToC topics, while group B and C had average scores, and the performances of group D and F related to "low performing" students. A remarkable observation is that when examining groups and assignments more closely, the assignments E4, E5, and E6 have a lower performance for every single grade group. The lowest performance for every grading group occurs in assignment E4. If visualizing these numbers as a line chart, this pattern becomes immediately visible (see Figure 5.1).

While the low performance for group A is a maximum of around 20% below the highest performance and for group B a maximum of around 30%, the gap is around 55% for group C and 65% for group D. The low performance for group F is a maximum of 40%. Thus, there is a noticeable lower performance for each of the five grading groups. At this point, it seemed questionable that students achieving an A or B were comprehending ToC better than the remaining student groups as they all had trouble with the same assignments. Regarding my first

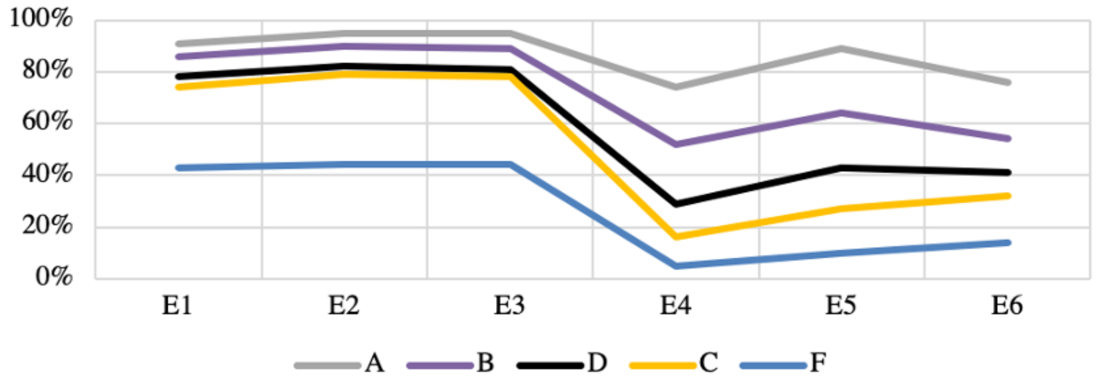

Figure 5.1.: Performance in the final exam assignments (2016) [FK21].

research question, I started questioning what might be different about E4, E5, and E6 in comparison to the other assignments and what might cause all students to perform worse on these. For this part of our analysis, I first summarize each assignment's topic and required activity:

- **E1 Regular Languages:** Specify a non-deterministic automaton (NFA) for a language. Specify a type-3 grammar. Derive words from a grammar.
- **E2 Automata:** Construct a deterministic automaton (DFA) from an NFA. Specify languages.
- **E3 Minimization:** Decide if a state is not reachable. Minimize a DFA with the table-filling algorithm. Specify equivalence classes. Visualize a DFA. Specify languages.
- **E4 Regular Languages:** Prove a language is not regular using the pumping lemma. Specify all equivalence classes of the Myhill-Nerode relation.
- **E5 Context-Free Languages:** Specify a type-2 grammar. Construct a pushdown automaton (PDA).
- **E6 Context-Free Languages:** Give derivations from a PDA. Prove a language is not regular using decidability properties.

As I examined differences between the assignments E4, E5, and E6 and the first three assignments, I noticed that E4 and E6 require students to create a formal proof, among other subtasks. On the other hand, E1, E2, and E3 require students to specify and construct automata and grammars by applying specific algorithms.

Table 5.3.: Percentage of achieved points for each exam assignment differentiated by grading groups from year 2017

Group	A	B	C	D	F
E1	94%	85%	78%	69%	49%
E2	93%	87%	83%	77%	57%
E3	97%	92%	85%	74%	53%
E4	69%	42%	30%	17%	10%
E5	84%	65%	40%	28%	18%
E6	86%	68%	57%	48%	28%

E4, the assignment with the worst performance, is also the assignment requiring students to perform the most formal proof (i.e., pumping lemma with the language $L_4 = \{a^{\mathrm{i}}b^{\mathrm{j}}c^{\mathrm{k}}d^{\mathrm{l}} | i, j, k, l \in N \wedge j < k + l\}$ with $\Sigma = \{a, b, c, d\}$). The data indicate that all students had issues with this assignment, regardless of how they performed on the final exam.

5.2.2. Final Exam Results from the Second Analysis Year (2017)

Table 5.3 presents student performance in 2017. First, it must be mentioned that the assignments in 2017 dealt with the same topics in the same order as in 2016, and they only differed in values. When visualizing the data and comparing 2016 (Table 5.2 and Figure 5.1) and 2017 (see Figure 5.2), it can be seen that all grading groups performed the worst on E4 once again. Additionally, group A performed best with a gap of a maximum 30% between the lowest and highest performance. Group B had a maximum gap of 50%, group C 55%, group D 60%, and group F 50%. Again, group C and group D have the largest gaps.

Furthermore, the performance in E6 is continuously better for every grading group in 2017, indicating a possible change in the assignment compared to 2016. I would imagine that it was due to the respective Pushdown Automaton (PDA) that the students were supposed to use. Before examining this phenomenon in more detail, I elaborate on whether this change is significant in Section 5.2.4.

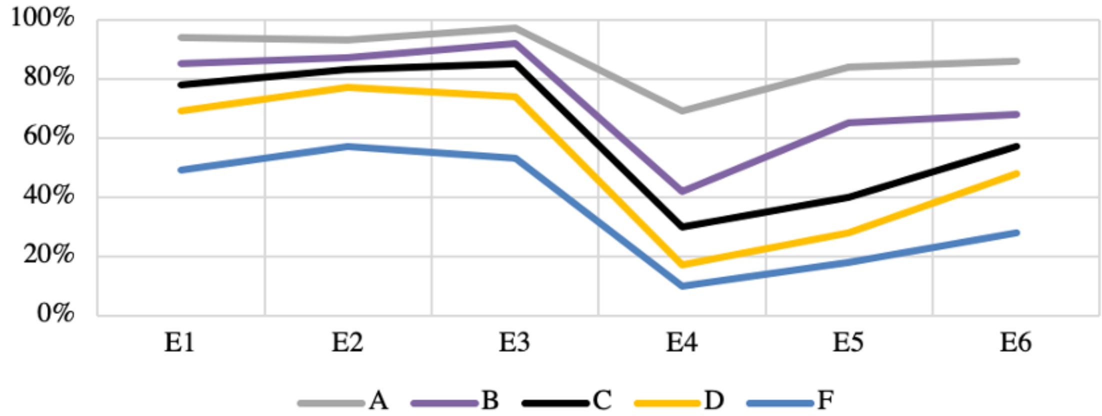

Figure 5.2.: Performance in the final exam assignments (2017) [FK21].

Table 5.4.: Percentage of achieved points for each exam assignment differentiated by grading groups from year 2018

Group	A	B	C	D	F
E1	92%	86%	71%	64%	36%
E2	93%	91%	84%	77%	50%
E3	96%	95%	87%	73%	46%
E4	75%	48%	27%	10%	4%
E5	88%	61%	44%	20%	12%
E6	78%	57%	49%	36%	18%

5.2.3. Final Exam Results from the Third Analysis Year (2018)

For 2018, the assignments again differed only in values but covered the same topics and concepts in the same order. As presented in Table 5.4 and Figure 5.3, students' performance in E4 is consistently the worst. In this year, the low performance for group A is again a maximum of around 20% below the highest performance. Group B had a maximum of 50%, group C 55%, group D 70%, and group F 50%. Compared to 2017, the performance in E6 alternates again.

Altogether, I found similar performances in all three years, indicating that the results of 2016 are not reducible to students of that year.

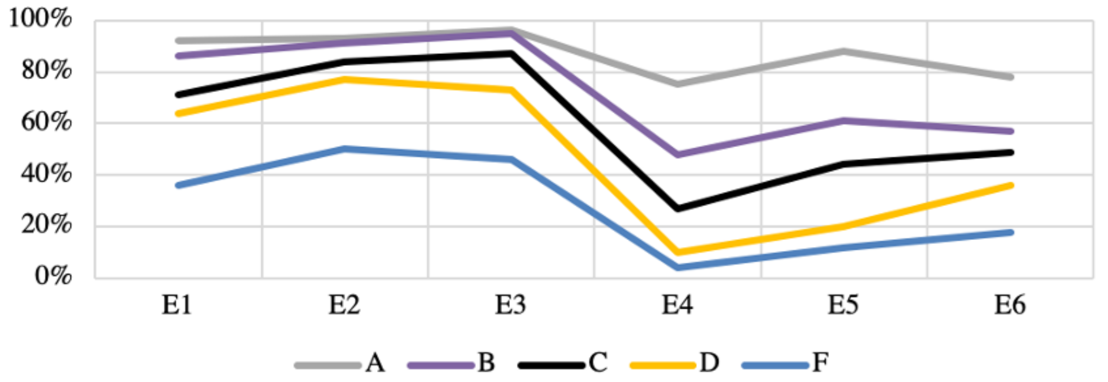

Figure 5.3.: Performance in the final exam assignments (2018) [FK21].

5.2.4. Analysis of Variance (ANOVA)

Continuing from the comparison of the diagrams, I examined if there were assignments with significantly different performance over the years. In doing so, I wanted to understand whether specific wording or certain values in the assignments could lead to performance changes. For this, I conducted one-way Welch's analysis of variances (Welch's ANOVA) to assess the different performances in the grading groups between the years. As mentioned before, I decided to use $p < .001$ as alpha level for all statistical tests.

The values in Table 5.5 show that there were indeed assignments with statistically significant differences in some assignments and grading groups (written in bold and marked with * in the Table). To detect in which years the significance actually occurs, I used Games-Howell tests for post-hoc analysis with the following results. For completeness, I add the mean (M) and standard deviation (SD) of the group performances (The detailed test results can be found in the appendix):

- **Group A:** The Games-Howell post-hoc analysis revealed that for E6, the 80 students of 2017 (M = 86.25, SD = 10.29) performed better than the 81 students from 2016 (M = 76.79, SD = 15.61) and the 81 students from 2018 (M = 79.63, SD = 12.97) .
- **Group B:** For E6, the 81 students of 2017 (M = 67.90, SD = 13.61) performed better than the 60 students of 2016 (M = 55.23, SD = 16.28) and the 71 students of 2018 (M = 57.04, SD = 14.65).

Table 5.5.: Welch's ANOVA of the final exam assignments for every grading group (* p < .001)

G.	A	B	C	D	F
E1	$F_{2,\,157.24}$=2.11, p=.125	$F_{2,\,136.38}$=.678, p=.510	$F_{2,\,154.52}$=1.90, p=.153	$F_{2,\,119.97}$=1.94, p=.148	$F_{2,\,224,22}$=6.52, p=.002
E2	$F_{2,\,156.19}$=6.99, p=.001	$F_{2,\,126.38}$=1.84, p=.164	$F_{2,\,151.67}$=1.73, p=.180	$F_{2,\,115.60}$=10.9, **p<.001***	$F_{2,\,202.92}$=.008, p=.992
E3	$F_{2,\,159.70}$=.981, p=.377	$F_{2,\,125,60}$=7.37, p=.001	$F_{2,\,154.39}$=3.94, p=.021	$F_{2,\,119.54}$=1,83, p=.165	$F_{2,\,222.89}$=.051, p=.951
E4	$F_{2,\,158.50}$=4.15, p=.017	$F_{2,\,132.50}$=2.00, p=.139	$F_{2,\,140.05}$=.278, p=.758	$F_{2,\,90.12}$=1.54, p=.219	$F_{2,\,140.17}$=4.88, p=.009
E5	$F_{2,\,158.820}$=2.54, p=.082	$F_{2,\,136.34}$=.577, p=.563	$F_{2,\,140.42}$=2.64, p=.075	$F_{2,\,96.90}$=1.65, p=.198	$F_{2,\,137.65}$=.258, p=.773
E6	$F_{2,\,155.25}$=12.6, **p<.001***	$F_{2,\,130.76}$=16.5, **p<.001***	$F_{2,\,152.72}$=8.84, **p<.001***	$F_{2,\,109.13}$=11.5, **p<.001***	$F_{2,\,185.45}$=4.87, p=.009

- **Group C:** The 82 students of 2016 performed worse in E6 (M = 44.31, SD = 13.61) compared to the 72 students of 2017 (M = 53.21, SD = 12.87).
- **Group D:** The 60 students of 2018 (M = 85.61, SD = 8.78) performed significantly better in E2 than the 61 students from 2016 (M = 78.38, SD = 11.89) and the 62 students from 2017 (M = 76.97, SD = 16.04). In E6, the 62 students of 2017 (M = 47.53, SD = 15.88) performed better than the 54 students of 2016 (M = 33.83, SD = 14.90).
- **Group F:** The analysis revealed no significant changes for group F.

In total, the post-hoc analyses support the previous observation that the performance in E6 alternates across some years. First, E6 had higher performances in 2017 than in 2016. The analyses revealed that the increase was significant in every grading group except for group F. Second, 2017 had a significant higher performance than 2018 only in group A and group B. (For the exact percentages, please refer to Tables 5.2, 5.3, and 5.4). Nonetheless, I compared the E6 assignments across the years and I made two observations:

1. The PDA in 2017 was less complex than the two of 2016 and 2018 in the sense that it had at most three transition functions on one edge. In 2016 and 2018, it was a maximum of four. I could imagine that this may have had an

influence on the results, but I have no experience concerning the extent to which different PDAs could lead to different results.

2. In a subtask, the students were asked to prove that a language is not regular with the help of closure properties and the language itself differs between the years with $L = \{w \in \Sigma^+ || w|_\mathrm{a} = |w|_\mathrm{b}\}$ in 2016 and $L = \{a^\mathrm{m} b^\mathrm{n}, b^\mathrm{n} a^\mathrm{m} | n, m \in N \wedge n + 1 = m\}$ in 2017. At first glance, the 2016 task seems more abstract than the 2017 task and assumes that students understand the concept of word length. However, a closer look at the 2018 assignment ($L = \{a^\mathrm{n} b^\mathrm{m} | n, m \in N \wedge n < m$) revealed that the language is similar to the language in 2017. Nevertheless, there was a significant difference between 2016 and 2018 for only two grading groups in the study.

Since the PDAs and languages are the only noticeable differences in the assignments for 2016 and 2017, I encourage further analysis of these kinds of subtasks and topics so as understand why students' performance changes. At this point, it should be noted that 2016 and 2018 were taught by the same lecturer, which could explain the similar performance on E6. However, not for all grading groups a significant difference was found between 2017 and 2018, although the lecturer was different.

Apart from this, the post-hoc analyses showed no significant changes in any assignment for more than two grading groups at the same time. Importantly, there were no significant differences in performance on E4, which shows that students performed poorly on this proof assignment in all years for all groups. Moreover, it should be noted again that E6 had only a short proving subtask.

5.2.5. Additional Information

At this stage, I wondered whether other formal or less content-related factors might be influencing student performance. More precisely, I found and provided two additional pieces of information about the assignments of 2016 (see Table 5.6):

- **The level of difficulty as assessed by the teachers:** A star means the students have to reproduce knowledge, two stars describe application tasks (e.g., algorithms), and three stars expect them to combine their existing knowledge in a new way. However, except for assignments E4 and E5, the difficulty of the subtasks was usually between one and three stars. This information gave no further indication of why performance differed with E6 or why the students performed better in the first three assignments.

- **The number of points students could score on each assignment:** The number of points depends on the duration or size of the solution. If examining the distribution of points, the first three assignments totaled 57 points, and the second three totaled 43 of the 100 total points, which was roughly half. Questioning whether fewer students completed the last three assignments, I found in the most common case the same number of student solutions in all six assignments for groups A, B, and C. In groups D and F, the fewest students completed E4, with 15 and 50 students, respectively. For this reason, I do not think point distribution itself encouraged students to prioritize the first three assignments. However, this additional information implies that groups D and F may have had more problems, too little time, or even inhibitions about starting the last three assignments at all.

Nothing had changed in 2017 and 2018 concerning the difficulty and distribution of points, or the decrease in the number of students who have completed the last three assignments. Overall, these results support the assumption that students seem to have difficulty with the Pumping Lemma (PL) in their final exam, which I derived from students' performance in 2016. Therefore, performance was not dependent on the one assignment in that specific year.

Table 5.6.: Difficulties and points to reach for the exam assignments of 2016

Assignment	Difficulty	Points
E1	* - ***	19
E2	** - ***	16
E3	* - ***	22
E4	***	17
E5	**	11
E6	* - ***	15

5.3. Homework Assignments

Building on the insights from the previous analysis regarding the final exams, I wondered whether formal proof assignments also challenged students during their homework assignments. To answer this question, I took a closer look at their performance on the homework assignments. I start this section with an overview of homework assignments and provide detailed analysis of the homework performances in 2016, 2017, and 2018.

5.3.1. Description of the Assignments

The general topics and assignment types in all three years were the same and only differed in the values. Concerning the assignments, there were only the following differences:

- **2017:** a clue was added in A19.
- **2017 and 2018:** a subtask was added in A13 to prove if a relation is a subset. Another assignment about representations systems existing in 2016 was deleted. (I call this assignment A14b in Table 8.)
- **2018:** a subtask was added in A2 and A9, but they had the same assignment type and were similar to the existing subtasks.

Additionally, I assigned all assignment topics to their superordinate topic drawn from the course formulary as described in Section 4.5.2. Here, I derived the following assignment categories as an abstract summary of possible tasks:

- **Proving**: assignments required to develop a proof
- **Specifying**: assignments required to specify – for example, derivations, languages, relations, or grammars from a corresponding given element
- **Constructing** assignments required to construct an automaton
- **Calculating**: assignments required to calculate – for example, sets

At this point, it should be noted that I do not offer additional information about the homework assignments as I did for the final exam assignments (see Section 5.2.5). Likely because of the group work, no strong decrease in the number of students submitting specific assignments of the same homework set occurred. I suppose the time aspect plays a less important role in homework than in the exam. As I could find no connection between difficulty as defined by the teachers and low performance in the final exam, I found that in this case, the information requires no further detail or investigation.

Table 5.7.: Description and categorization of homework assignments

	Assignments	Topic	Category
A1	Calculate given sets	Sets	Calculating
A2	Prove specific properties	Sets	Proving
A3	Prove with the help of Truth tables	Propositional Logic	Proving
A4	Prove with the help of equivalent transformations	Propositional Logic	Proving
A5	Prove specific variable assignments	Propositional Logic	Proving
A6	Prove a given statement for two predicates	Predicate Logic	Proving
A7	Specify the first step of a contradiction	Predicate Logic	Specifying
A8	A mathematical induction over a checksum of a set of numbers	Sets	Proving
A9	Specify properties of relations	Relations/Orders	Specifying
A10	Develop a proof about orders	Relations/Orders	Proving
A11	Specify properties of relations	Functions/Mappings	Specifying
A12	Prove cardinality	Functions/Mappings	Proving
A13	Specify relations and equivalence classes	Relations/Orders	Specifying[1]

[1]Proving in 2017 and 2018

A14	Specify representation systems	Relations/Orders	Specifying
A14b	Prove that two variables form a set are equivalent	Relations/Orders	Proving
A15	Specify mappings	Functions/Mappings	Specifying
A16	Specify words	Words/Languages	Specifying
A17	Specify languages and regular expressions	Words/Languages	Specifying
A18	Specify the order of words	Words/Languages	Specifying
A19	Proof by induction about a given alphabet and words	Words/Languages	Proving
A20	Specify deductions for words and languages	Grammars/ Chomsky	Specifying
A21	Specify a grammar	Grammars/Chomsky	Specifying
A22	Specify grammars and their types of the Chomsky hierarchy	Grammars/Chomsky	Specifying
A23	Develop a proof with the pumping lemma	Regular Languages	Proving
A24	Specify deductions of words and decide if they are accepted by the automata	Automata	Specifying
A25	Construct a DFA to a given language	Automata	Constructing
A26	Specify deductions and languages	Automata	Specifying
A27	Construct a DFA of a given NFA	Automata	Constructing
A28	Specify all equivalence classes of the Myhill-Nerode relation regarding a language A	Regular Languages	Specifying
A29	Specify equivalence classes of a Myhill-Nerode relation	Regular Languages	Specifying

A30	Specify equivalence classes of a Myhill-Nerode relation and specify the automata	Automata	Specifying

5.3.2. Homework Results from First Analysis Year (2016)

For the homework analysis, I performed the same point distribution analysis of each of the 31 homework assignments, as I did for the final exam assignments in Section 5.2.

Table 5.8 contains the average percentages for each grading group for every homework assignment. As a first step, I examined the abstract assignments categories to see how the groups performed on average in each of them (see Table 5.9). At first glance, it can be seen that there is no value below 50%; this means that if there are assignments with very low performance within a category, there are also assignments with such a high performance that it balances the category out. Therefore, not all assignments of a category generally cause difficulties for the students. Nonetheless, the *proving* assignment performances are the lowest in each grading group, while calculating is the best.

However, this summary does provide no outlook on which individual assignments the students are performing low. Therefore, I visualized how each grade group scored in their entire homework spectrum in the same way I did for the final exam assignments, building on the data from Table 5.8. The results can be seen in Figure 5.4 for the first half of the homework assignments and Figure 5.5 for the second half of the assignments. Again, group A scored highest for each assignment, and B and C were second this time. The difference between the groups is less remarkable than it was in the final exam results. Nonetheless, I also find assignments with a low performance, and this holds again for every grading group. For the homework analysis, I consider the assignments as "low performance assignments" if group A scored 80% or less. Since all other groups scored lower than usual on the same assignments as group A, this approach seems to be a good compromise to divide the assignments.

Low Performance Assignments: The lowest performance for the first half of the homework assignments can be found for assignments A2, A5, A6, A7, A8,

Table 5.8.: Percentage of achieved points for each homework assignment differentiated by grading groups from year 2016

Group	A	B	C	D	F
A1	95%	94%	91%	90%	86%
A2	77%	69%	69%	57%	70%
A3	92%	86%	87%	83%	90%
A4	85%	85%	83%	82%	79%
A5	80%	69%	63%	51%	56%
A6	50%	46%	46%	40%	41%
A7	69%	59%	64%	53%	56%
A8	79%	79%	72%	65%	68%
A9	92%	92%	84%	83%	85%
A10	77%	72%	69%	65%	68%
A11	86%	80%	74%	72%	71%
A12	87%	82%	69%	70%	63%
A13	97%	96%	85%	88%	87%
A14	84%	74%	64%	58%	63%
A14b	66%	46%	45%	43%	35%
A15	83%	70%	67%	64%	59%
A16	98%	94%	89%	89%	91%
A17	90%	77%	75%	66%	68%
A18	99%	91%	90%	84%	87%
A19	72%	62%	60%	49%	50%
A20	77%	64%	64%	54%	58%
A21	92%	82%	79%	71%	71%
A22	82%	78%	73%	63%	66%
A23	82%	70%	65%	52%	63%
A24	97%	86%	86%	80%	80%
A25	89%	78%	77%	72%	76%
A26	99%	91%	88%	79%	85%
A27	99%	93%	92%	88%	50%
A28	76%	62%	57%	49%	30%
A29	86%	75%	67%	47%	35%
A30	99%	93%	93%	82%	50%

Table 5.9.: Average percentages per assignment category from year 2016

Category	A	B	C	D	F
Calculating	95%	94%	91%	90%	86%
Proving	77%	70%	60%	60%	56%
Specifying	89%	78%	74%	68%	66%
Constructing	95%	86%	85%	80%	63%

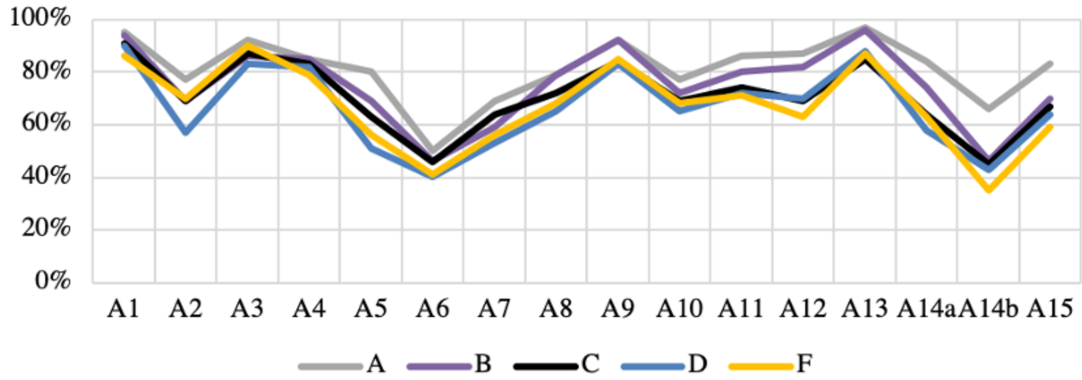

Figure 5.4.: Performance in A1 - A15 (2016) [FK21].

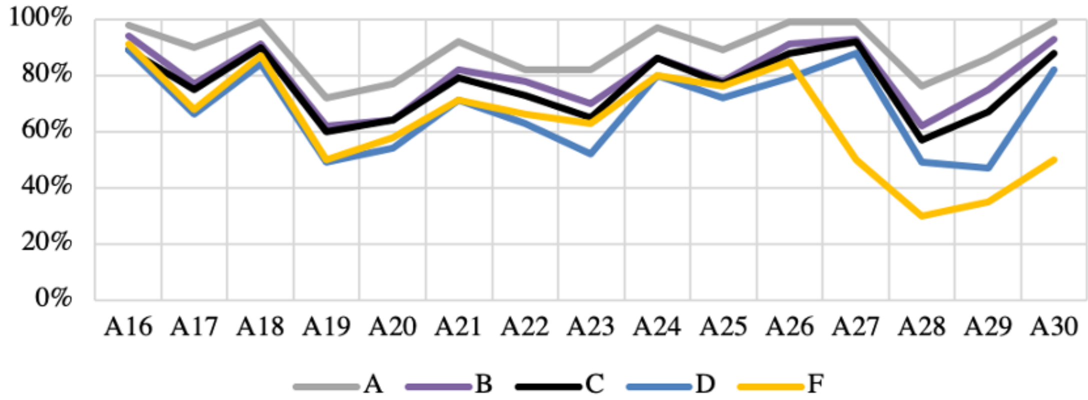

Figure 5.5.: Performance in A16 - A31 (2016) [FK21].

A10, and A14b with a performance from 15 – 50% compared to the remaining assignments (Figure 5.4). Figure 5.5 shows the distribution for the remaining assignments A16 – A31, where the gap between grading group A and the other groups seems to increase. The grading groups show again a lower performance in the same assignments: assignments A19, A20, and A28, with a difference of 10 – 40% from the usual average performance.

After comparing the 10 assignments with low performance, I found that seven out of 10 homework assignments required the students to develop a proof (all but A7, A20, A28). To gain more insight into those proof assignments with the lowest performance and to find commonalities and differences, I provide more detailed information about the respective low performance assignments already briefly given in Table 5.7. More specifically, I give examples of the assignments, indicate what knowledge is particularly necessary here, and suggest how a solution should be approached:

- **A2:** A set of given properties could be used, such as associative property, i.e. ((A or B) or C) = (A or (B or C)). Building on the given properties, the students should prove or disprove, for example $((A \cup C) \setminus B) \cup (C \cap B) = (A \setminus B) \cup C$. In this way, the students had to rearrange the part before the equal sign step by step and with the help of the given properties until they arrived at what is after the equal sign. The students had to understand set properties and the notation of sets.
- **A5:** The students should develop a proof or disprove with arguments about one or more variable assignments, for example, that $\neg (p \leftrightarrow (q \wedge r)) \wedge (p \rightarrow (q \wedge r))$ is contradictory. Accordingly, the students had to select a fitting assignment, while understanding and applying the basics of propositional logic.
- **A6:** The students should prove that $(\forall x.\ P_1(x) \vee P_2(x)) \rightarrow (\forall y.\ (P_1(y) \rightarrow P_2(y)) \rightarrow P_2(y))$ for the one-digit predicates P_1 and P_2. Therefore, the students had to know and apply the basics of first-order logic and had to develop the proof step by step by choosing the correct assumptions.
- **A8:** The students had to develop a proof by induction –that is, following the specific scheme of induction including base case (show it for *n*) and inductive step (show it for *n+1*). More specifically, they had to prove a function that can be used to calculate a number's checksum of a specific set of numbers. In this case, the students needed mathematics knowledge

about checksums, modulo operation, and the previously mentioned induction scheme.

- **A10:** The students had to prove that a specific relation R is a partial order. Therefore, they were given a specific set, for example, $C = \{1, 2, 3, 4\}$. For R : (C, C) and the relation $R = \{(1,1), (2,2), (3,1), (3,3), (3,4), (4,4)\}$, the students had to know and show the properties of a partial order for the proof – that is, reflexivity, antisymmetry, and transitivity.
- **A14b:** The students had to prove that if there exists a representation system of X/R (where X is a set and R is an equivalence relation on X) and (a,b)$\in$R, that a = b. Accordingly, they had to choose several different assumptions and subgoals in a step-for-step manner to finally show a = b. Therefore, the students must have knowledge about sets and representation systems.
- **A19:** The students had to develop a proof by induction over words. For example, they had an alphabet Σ and must show that $\forall w \in \Sigma^*.\ \forall x \in \Sigma.\ |w|_x \leq |w|$. Again, they must use the induction scheme.

Overall, it can be seen that the assignments are different – not only regarding their topic but also their expected kind of proof. Often, the students just had a general scheme in the sense that they are allowed to use a fixed number of properties to use any number of assumptions to show any number of subgoals. In the future, a more detailed examination of the mistakes students make on the particular or similar assignments could be done. Nonetheless, A6 was the assignment showing the lowest performance for group A (50%). Since A7 (as a non-proof assignment) also had a low performance and both assignments are from the same topic, perhaps another approach would be to analyze how students deal in other ways with first-order logic assignments.

High Performance Assignments: Beyond the previously discussed low performances, I also questioned what the remaining assignments with higher performances had in common and what might explain students' higher scores on these. I found that most assignments without low performance required *specifying* activities (see Table 5.7). Nonetheless, there were still four proof assignments with a higher performance (A3, A4, A12, A23). In the following, I also reexamined these four assignments regarding the nature of proof required:

- **A3:** The students had to prove or disprove a statement from propositional logic, e.g., $\neg (p \rightarrow q) \wedge (p \leftrightarrow q \vee \neg p)$. Therefore, they must use a truth table instead of a formal proof.
- **A4:** The students had to show the logical equivalence of two statements from propositional logic, for example, $(p \wedge q) \vee \neg r$ and $(\neg p \rightarrow \neg r) \wedge (r \rightarrow q)$. Therefore, the students must use several properties to change the statement step by step, such as implication and distributivity.
- **A12:** The students had to prove or disprove the cardinality of two sets, for example, card(M) = card (N), when $M = \{n \in N^{+} \mid n \bmod 5 = 0\}$. Therefore, the students need to give a bijection and show step by step the equality.
- **A23:** A23 was a proof that expected students to use the PL. This time, the languages had numbers instead of only letters: $\Sigma = \{0,1\}$ and $L_5 = \{0^n 1^m 0^{n+m} | n,m \in N\}$ and $L_6 = \{11w || 11w|_1 < |11w|_0 \wedge w \in \Sigma^*\}$

As presented, assignments A3 and A4 had schematic instructions for the development of a proof (e.g., specific usage of truth tables and equivalence transformations from propositional logic). A12 is the only proof within the thematic block "functions and mappings", but there was no striking abnormality I could detect in the corresponding kind of proof. Interestingly, A23 was a PL assignment proof like the final exam assignment E4 (see Section 5.2), which students scored lowest in their final exam. However, such a low performance could not be observed for A23.

Low performance assignments	**Middle to high performance assignments**
A2 **A5** **A6** A7 **A8** **A10** **A14b** **A19** A20 A28	A1 **A3** **A4** A9 A11 **A12** A13 A14 A15 A16 A17 A18 A21 A22 **A23** A24 A25 A26 A27 A29 A30

Figure 5.6.: Assignments divided by performance (2016). The assignments written in bold are the proof assignments.

I have summarized the assignments division in Figure 5.6. Altogether, I found 10 assignments with noticeably lower performance and seven of these assignments required formal proofing. Furthermore, I could not detect a specific topic or only one kind of proof in the low performance assignments. Two proof assignments with a higher performance had schematic and less formal instructions for the development of a proof, while one required a schematic and formal proof (pumping lemma). In conclusion, I could not detect patterns, specific topics, or only one kind of proof that explained why students exhibited higher performance on these four proof assignments. Summarizing the described analysis step in this section, I conclude that the majority of students' low performance in proof assignments on the final exam was also evident in homework assignments.

5.3.3. Homework Results from the Second Analysis Year (2017)

To validate the results from 2016, I examined into homework performance for 2017. Tables 5.10 and 5.11 present an overview of the data and category. In this year, assignment A14b was deleted so there were only 30 assignments in total. As A14 and A14b had the same topic, no topic was left out. Overall, the summary is similar with the proving assignments having the lowest performance but no value under 50%. Again, the proving assignments had the lowest performance, and the calculating assignments had the highest.

Figures 5.7 and 5.8 present the visualization of the performance in the homework assignments in 2017. Again, grading group A performed best in most assignments while the other grading groups overlapped or switched orders. Nevertheless, the gap between grading group A and the others increased for most assignments in the second half of the assignments. Interestingly, grading group F performed best in assignment A30 (98%). Figure 5.9 shows the division of assignments for year 2017. Overall, 14 assignments had a lower performance (eight proof assignments), while three proof assignments had a higher performance. At this point, I will not be discussing the difference between the low and high performances in 2016 in more detail. Instead, I refer to the ANOVA, which directly indicates only the statistically significant changes that occurred compared to the 2016 data (see Section 5.3.5).

Table 5.10.: Percentage of achieved points for each homework assignment differentiated by grading groups from year 2017

Group	A	B	C	D	F
A1	88%	88%	85%	82%	79%
A2	65%	53%	64%	57%	47%
A3	90%	83%	84%	82%	81%
A4	71%	58%	61%	52%	58%
A5	70%	59%	56%	51%	45%
A6	74%	69%	61%	62%	61%
A7	67%	65%	55%	60%	55%
A8	79%	74%	63%	69%	66%
A9	87%	83%	81%	82%	81%
A10	73%	71%	65%	69%	65%
A11	81%	78%	79%	77%	75%
A12	82%	75%	71%	78%	73%
A13	58%	54%	55%	51%	51%
A14	65%	58%	49%	56%	55%
A15	78%	67%	64%	68%	61%
A16	97%	95%	96%	90%	88%
A17	91%	78%	76%	72%	72%
A18	91%	89%	90%	88%	87%
A19	83%	69%	66%	73%	65%
A20	87%	83%	79%	75%	72%
A21	86%	80%	69%	70%	72%
A22	78%	61%	56%	57%	56%
A23	70%	53%	49%	49%	46%
A24	94%	87%	78%	82%	82%
A25	82%	76%	65%	62%	64%
A26	91%	85%	80%	76%	78%
A27	97%	95%	93%	92%	93%
A28	67%	55%	53%	57%	53%
A29	79%	65%	57%	56%	56%
A30	94%	89%	92%	83%	98%

Table 5.11.: Average percentages per assignment category from year 2017

Category	A	B	C	D	F
Calculating	88%	88%	85%	82%	79%
Proving	74%	65%	63%	63%	60%
Specifying	83%	76%	72%	72%	71%
Constructing	90%	86%	80%	77%	79%

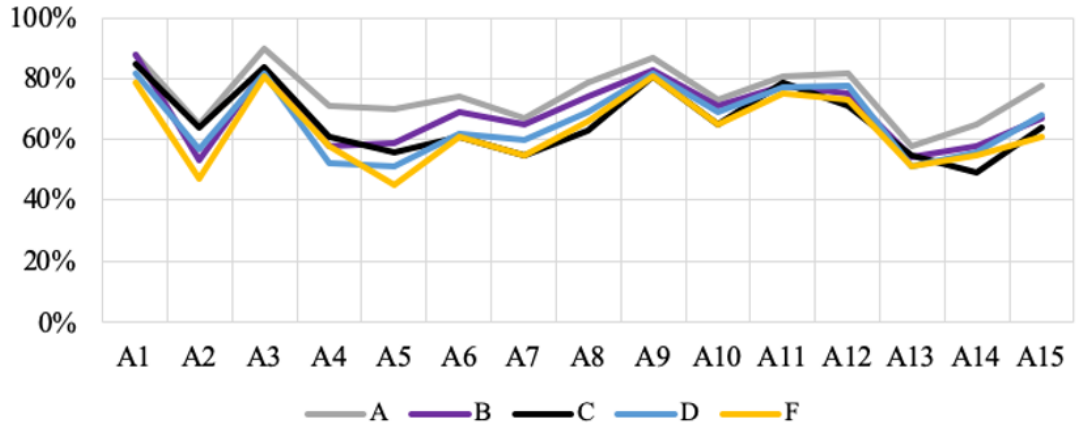

Figure 5.7.: Performance in A1 - A15 (2017) [FK21].

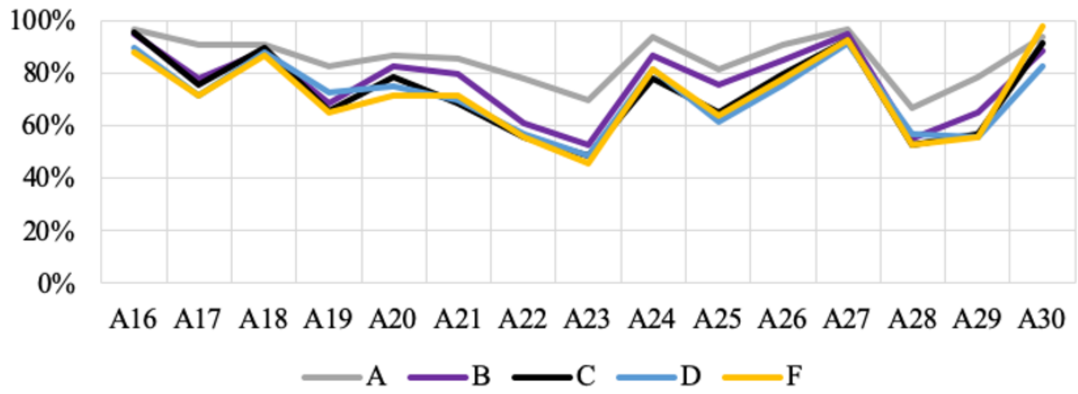

Figure 5.8.: Performance in A16 - A30 (2017) [FK21].

Low performance assignments	Middle to high performance assignments
A2 **A4** **A5** **A6** A7 **A8** **A10**	A1 **A3** A9 A11 **A12** A16 A17
A13 A14 A15 A22 **A23** A28	A18 **A19** A20 A21 A24 A25
A29	A26 A27 A30

Figure 5.9.: Assignments divided by performance (2017).

5.3.4. Homework Results from the Third Analysis Year (2018)

The order and topics of the assignments in 2018 stayed the same as in 2017, and no other assignment was deleted. The students' performance can be seen again in Table 5.12, with a summary in 5.13. In general, no major changes can be seen there compared to the values of 2016 and 2017, which are very similar. The values remain above 50% and the proving assignments still had the lowest performance.

The data visualization for the first half of the assignments is shown in Figure 5.11. The lines proceed similarly again; however, the gap this time between grading group A and the other groups did not increase in the second half of the assignments (Figure 5.12). Figure 5.10 presents the division of assignments for year 2018. This time, 16 assignments had a lower performance (nine proof assignments), while two proof assignments had a higher performance. Again, I refer to the ANOVA, which directly indicates the statistically significant changes that occurred compared to the 2016 and 2017 data, instead of discussing the differences (see Section 5.3.5).

Low performance assignments	Middle to high performance assignments
A2 **A4** **A5** **A6** A7 **A8** **A10**	A1 **A3** A9 A11 **A12** A16 A17
A11 **A13** A14 A15 **A19** A20	A18 A21 A24 A25 A26 A27
A22 **A23** A28	A29 A30

Figure 5.10.: Assignments divided by performance (2018).

Table 5.12.: Percentage of achieved points for each homework assignment differentiated by grading groups from year 2018

Group	A	B	C	D	F
A1	91%	93%	88%	90%	83%
A2	72%	70%	67%	71%	67%
A3	90%	93%	91%	92%	87%
A4	79%	81%	78%	78%	77%
A5	67%	62%	63%	62%	63%
A6	65%	60%	59%	55%	59%
A7	67%	68%	67%	65%	63%
A8	76%	71%	71%	66%	67%
A9	88%	89%	87%	83%	74%
A10	74%	70%	69%	70%	61%
A11	74%	72%	72%	68%	59%
A12	81%	77%	73%	64%	63%
A13	62%	58%	54%	47%	46%
A14	61%	54%	52%	45%	46%
A15	74%	58%	58%	59%	50%
A16	93%	94%	97%	84%	84%
A17	83%	79%	72%	66%	66%
A18	94%	95%	86%	84%	79%
A19	64%	59%	58%	49%	50%
A20	78%	77%	68%	68%	59%
A21	81%	82%	79%	70%	64%
A22	77%	73%	67%	62%	55%
A23	69%	70%	55%	56%	50%
A24	87%	91%	83%	73%	73%
A25	84%	89%	74%	74%	66%
A26	90%	89%	82%	75%	73%
A27	97%	96%	94%	82%	83%
A28	73%	71%	62%	56%	58%
A29	87%	82%	73%	73%	69%
A30	94%	90%	92%	84%	78%

Table 5.13.: Average percentages per assignment category from year 2018

Category	A	B	C	D	F
Calculating	91%	93%	88%	90%	83%
Proving	73%	70%	67%	65%	63%
Specifying	81%	79%	75%	70%	66%
Constructing	91%	93%	84%	78%	75%

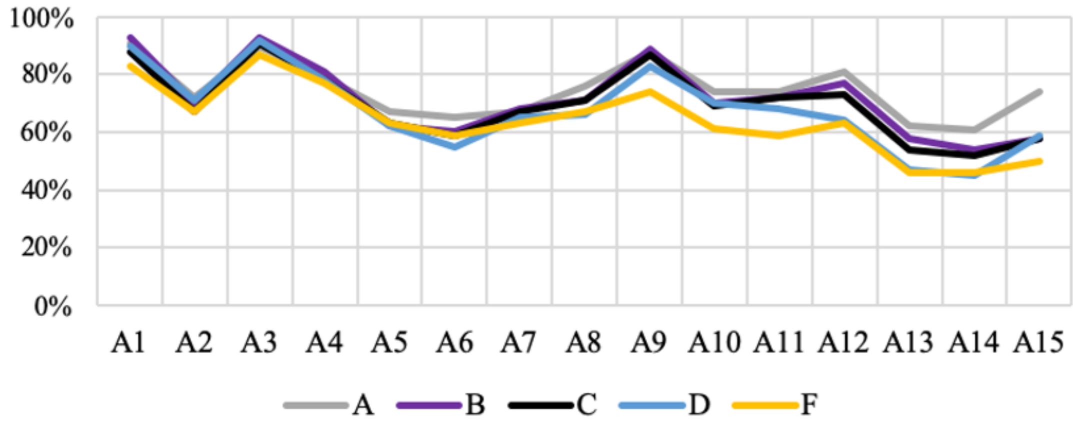

Figure 5.11.: Performance in A1 - A15 (2018) [FK21].

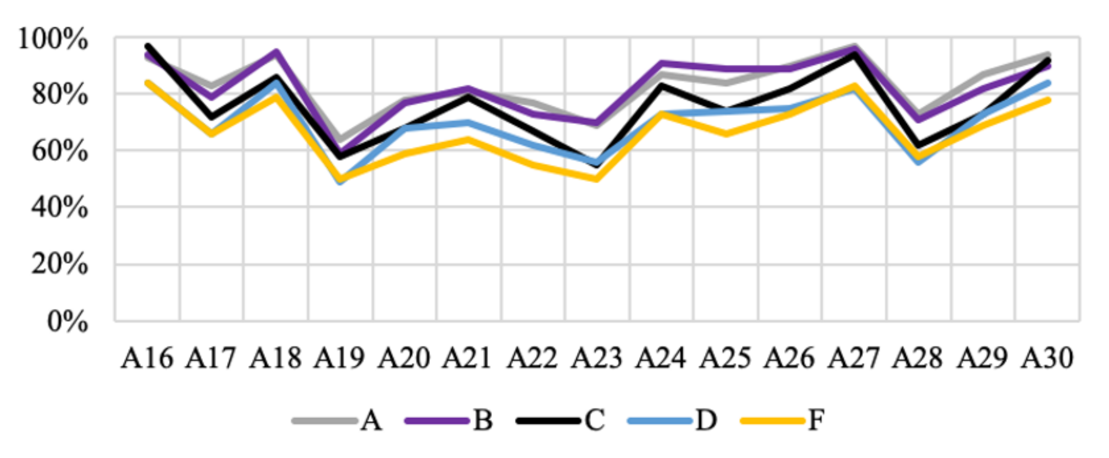

Figure 5.12.: Performance in A16 - A30 (2018) [FK21].

Table 5.14.: Welch's ANOVA of the homework assignments for every grading group (* p < .001)

Group	A	B	C	D	F
A1	$F_{2,\ 140.60}$=7.89, p=.001	$F_{2,\ 137.15}$=2.58, p=.079	$F_{2,\ 145.22}$=.09, p=.048	$F_{2,\ 109.82}$=7.66, p=.021	$F_{2,\ 190.25}$=3.31, p=.039
A2	$F_{2,\ 156.76}$=5.90, p=.003	$F_{2,\ 133.15}$=10.31, **p<.001***	$F_{2,\ 146.49}$=13.66, **p<.001***	$F_{2,\ 113.17}$=7.66, p=.001	$F_{2,\ 201.41}$=21.81, p=.**p<.001***
A3	$F_{2,\ 146.31}$=1.96, p=.145	$F_{2,\ 131.85}$=3.06, p=.050	$F_{2,\ 146.50}$=2.47, p=.088	$F_{2,\ 110.46}$=3.79, p=.026	$F_{2,\ 182.43}$=6.29, p=.002
A4	$F_{2,\ 152.761}$=9.77, **p<.001***	$F_{2,\ 123.13}$=.18.42, **p<.001***	$F_{2,\ 144.23}$=11.14, **p<.001***	$F_{2,\ 111.96}$=17.66, **p<.001***	$F_{2,\ 200.92}$=14.25, **p<.001***
A5	$F_{2,\ 153.29}$=4.48, p=.013	$F_{2,\ 135.41}$=.1.17, p=.312	$F_{2,\ 144.72}$=.927, p=.398	$F_{2,\ 112.48}$=2.26, p=.110	$F_{2,\ 201.82}$=10.25, **p<.001***
A6	$F_{2,\ 156.55}$=22.6, **p<.001***	$F_{2,\ 130.09}$=.15.22, **p<.001***	$F_{2,\ 143.34}$=8.05, **p<.001***	$F_{2,\ 111.19}$=77.9, p=.001	$F_{2,\ 203.59}$=15.11, **p<.001***
A7	$F_{2,\ 157.05}$=.086, p=.917	$F_{2,\ 128.83}$=1.27, p=.284	$F_{2,\ 144.16}$=3.12, p=.047	$F_{2,\ 107.40}$=1.80, p=.170	$F_{2,\ 201.32}$=1.89, p=.154
A8	$F_{2,\ 156.46}$=.612, p=.544	$F_{2,\ 133.97}$=1.50, p=.227	$F_{2,\ 143.52}$=1.07, p=.346	$F_{2,\ 108.94}$=.661, p=.518	$F_{2,\ 201.03}$=.138, p=.871
A9	$F_{2,\ 150.87}$=3.04, p=.051	$F_{2,\ 132.82}$=7.95, p=.001	$F_{2,\ 145.22}$=4.37, p=.014	$F_{2,\ 104.52}$=7.58, p=.001	$F_{2,\ 198.33}$=3.27, p=.040
A10	$F_{2,\ 153.00}$=.617, p=.541	$F_{2,\ 129.24}$=.081, p=.922	$F_{2,\ 145.94}$=.514, p=.599	$F_{2,\ 105.16}$=3.50, p=.034	$F_{2,\ 201.49}$=1.02, p=.361
A11	$F_{2,\ 150.72}$=7.45, p=.001	$F_{2,\ 130.87}$=2.29, p=.106	$F_{2,\ 144.10}$=2.06, p=.131	$F_{2,\ 103.26}$=1.22, p=.301	$F_{2,\ 197.62}$=1.76, p=.174
A12	$F_{2,\ 150.71}$=1.22, p=.298	$F_{2,\ 127.53}$=.1.21, p=.303	$F_{2,\ 145.83}$=.157, p=.855	$F_{2,\ 96.36}$=1.33, p=.269	$F_{2,195.63}$=.626, p=.536

Table 5.14.: Welch's ANOVA of the homework assignments for every grading group (* p < .001)

Group	A	B	C	D	F
A13	$F_{2,\ 122.73}$=150.5, **p<.001***	$F_{2,\ 127.33}$=.69.20, **p<.001***	$F_{2,\ 143.30}$=40.9, **p<.001***	$F_{2,\ 102.15}$=47.72, **p<.001***	$F_{2,\ 199.02}$=48.56, **p<.001***
A14	$F_{2,\ 150.68}$=14.23, **p<.001***	$F_{2,\ 124.49}$=7.39, p=.001	$F_{2,\ 143.86}$=5.26, p=.006	$F_{2,\ 101.64}$=.614, p=.543	$F_{2,\ 195.09}$=3.53, p=.031
A15	$F_{2,\ 151.3}$=18.58, **p<.001***	$F_{2,\ 124.19}$=2.77, p=.066	$F_{2,\ 137.44}$=1.19, p=.306	$F_{2,\ 101.34}$=.402, p=.670	$F_{2,\ 198.35}$=.319, p=.727
A16	$F_{2,\ 135.84}$=1.41, p=.247	$F_{2,\ 113.09}$=.081, p=.922	$F_{2,\ 125.07}$=3.21, p=.044	$F_{2,\ 96.68}$=.050, p=.951	$F_{2,\ 192.05}$=.977, p=.378
A17	$F_{2,\ 144.89}$=4.25, p=.016	$F_{2,\ 116.76}$=.156, p=.856	$F_{2,\ 136.54}$=.672, p=.513	$F_{2,\ 98.75}$=.601, p=.550	$F_{2,\ 194.68}$=397, p=.673
A18	$F_{2,\ 143.88}$=1.63, p=.200	$F_{2,\ 117.75}$=.3.89, p=.923	$F_{2,\ 135.51}$=.776, p=.462	$F_{2,\ 110.33}$=.713, p=.493	$F_{2,\ 193.33}$=.024, p=.976
A19	$F_{2,\ 143.47}$=14.60, **p<.001***	$F_{2,\ 121.31}$=1.54, p=.218	$F_{2,\ 132.54}$=1.34, p=.266	$F_{2,\ 91.37}$=10.50, **p<.001***	$F_{2,\ 182.36}$=3.94, p=.021
A20	$F_{2,\ 142.99}$=10.52, **p<.001***	$F_{2,\ 115.91}$=.13.80, **p<.001***	$F_{2,\ 135.86}$=9.73, **p<.001***	$F_{2,\ 97.15}$=9.14, **p<.001***	$F_{2,\ 193.38}$=7.20, p=.001
A21	$F_{2,\ 145.38}$=3.22, p=.043	$F_{2,\ 123.34}$=.181, p=.834	$F_{2,\ 137.25}$=.164, p=.849	$F_{2,\ 95.00}$=.486, p=.617	$F_{2,\ 184.53}$=.023, p=.977
A22	$F_{2,\ 147.05}$=.076, p=.927	$F_{2,\ 124.63}$=10.78, **p<.001***	$F_{2,\ 132.67}$=2.42, p=.093	$F_{2,\ 92.27}$=2.35, p=.101	$F_{2,\ 188.32}$=2.28, p=.105
A23	$F_{2,\ 147.94}$=3.14, p=.046	$F_{2,\ 123.94}$=9.93, **p<.001***	$F_{2,\ 140.14}$=2.83, p=.063	$F_{2,\ 99.58}$=1.82, p=.168	$F_{2,\ 190.22}$=10.43, **p<.001***
A24	$F_{2,\ 143.34}$=5.80, p=.004	$F_{2,\ 117.76}$=.1.96, p=.145	$F_{2,\ 136.47}$=1.44, p=.241	$F_{2,\ 93.87}$=.999, p=.372	$F_{2,\ 200.58}$=.089, p=.915

Table 5.14.: Welch's ANOVA of the homework assignments for every grading group (* p < .001)

Group	A	B	C	D	F
A25	$F_{2,\,140.20}$=.483, p=.618	$F_{2,\,119.74}$=.7.32, p=.5001	$F_{2,\,134.82}$=.240, p=.787	$F_{2,\,98.76}$=5.12, p=.008	$F_{2,\,186.37}$=4.11, p=.018
A26	$F_{2,\,127.04}$=17.83, **p<.001***	$F_{2,\,127.59}$=4.66, p=.011	$F_{2,\,131.26}$=7.40, p=.001	$F_{2,\,103.81}$=12.8, p=.283	$F_{2,\,185.36}$=4.59, p=.011
A27	$F_{2,\,149.44}$=.317, p=.729	$F_{2,\,105.68}$=.772, p=.465	$F_{2,\,143.68}$=.709, p=.494	$F_{2,\,97.58}$=.849, p=.431	$F_{2,\,203.87}$=3.79, p=.024
A28	$F_{2,\,144.042}$=.364, p=.695	$F_{2,\,123.14}$=.3.27, p=.041	$F_{2,\,137.43}$=.991, p=.374	$F_{2,\,101.17}$=1.64, p=.199	$F_{2,\,181.62}$=2.66, p=.073
A29	$F_{2,\,1148.71}$=2.24, p=.110	$F_{2,\,123.53}$=6.08, p=.003	$F_{2,\,140.15}$=2.22, p=.112	$F_{2,\,102.41}$=10.63, **p<.001***	$F_{2,\,180.81}$=12.17, **p<.001***
A30	$F_{2,\,125.56}$=.991, p=.374	$F_{2,\,120.25}$=2.51, p=.086	$F_{2,\,138.69}$=1.22, p=.299	$F_{2,\,99.04}$=5.96, p=.004	$F_{2,\,198.14}$=.050, p=.952

5.3.5. Analysis of Variance (ANOVA)

As I had previously done for the final exam assignments, I conducted one-way Welch's ANOVA for the homework assignments in order to understand if there were significant performance changes over the years. Through this process, I hoped to find out what distinguishes these tasks so I could understand what is causing problems for the students. For completeness, I present all the ANOVA calculations in Table 5.14, but I summarize the results and their meaning in the following.

Table 5.15.: Assignments with significant changes in at least one grading group

-	A	B	C	D	F
2016 better as 2017	A1, A4, A13, A14, A15, A26	A2, A4, A9, A13, A22, A23	A2, A4, A13	A4, A13	A2, A4, A13, A23
2016 better as 2018	A11, A13, A14, A26	A13	A13	A13	A13
2017 better as 2016	A6, A19, A20	A6, A20	A20	A6, A19, A20	A6, A20
2017 better as 2018	A6, A19, A20				
2018 better as 2016				A29	A6, A29
2018 better as 2017		A2, A4, A23	A2, A4	A4	A2, A4, A29

Table 5.15 summarizes the results of Welch's ANOVA (Table 5.14) and the Games-Howell post-hoc test by providing a summary of all assignments showing significant changes in at least one grading group. For clarity, I do not explicitly report the number of students, the mean, and the standard deviation, but refer instead to the appendix for all values. (The full results and values for the post-hoc tests can also be found there.) For further analysis, I only consider the assignments with significant changes that occurred in all or almost all groups (A2, A4, A6, A13, and A20). For a summarizing overview, I include tables with the average points achieved in the cases mentioned (see Table 5.8, 5.10, and 5.12).

Table 5.16.: Percentages of the assignments having significant changes in three or four grading groups

Number	Year	A	B	C	D	F
A2	2016	-	69%	69%	-	70%
	2017	-	53%	49%	-	47%
	2018	-	70%	67%	-	67%
A6	2016	50%	46%	-	37%	39%
	2017	85%	79%	-	61%	60%
	2018	64%	60%	-	55%	59%

Changes in three or four grading groups

- **A2:** As Table 5.16 shows, 2017 was significantly worse for the proof assignment A2 for grading groups B, C, and F compared to 2016 and 2018. As already mentioned in Section 5.3.2, a set of properties is given in these assignments (e.g., associative property), but only 2017 contains $\perp$ as a special symbol for a so-called "false statement." I suggest that a more in-depth look at this assignment could reveal whether students particularly struggle with this or other abstract symbols.
- **A6:** 2017 was significantly better in the proof assignment A6 for A, B, D, and F compared to 2016 and 2018. For A6, I could not detect any noticeable difference in the assignments. It could be possible that the significant differences may also result from other factors besides the assignment values themselves, such as the different lecturer in 2017 or different practice assignments in the tutorial sessions.

Changes for every grading group

- **A4:** Every grading group exhibited worse performance on the proof assignment A4 in 2017 compared to 2016 (see Table 5.17); A4, however, was almost the same in all three years.
- **A13:** In A13, every grading group had a lower performance in 2017 compared to 2016 and also to 2018. Furthermore, the performance was up to 40% lower and was thus the lowest of the three assignments with changes for every grading group. In this case, I could even see a clear difference in A13: A subtask was added in 2017 and 2018 to prove an equivalence

Table 5.17.: Percentages of the assignments having significant changes in all grading groups

Number	Year	A	B	C	D	F
A4	2016	85%	86%	80%	76%	77%
	2017	71%	58%	61%	52%	58%
	2018	79%	81%	78%	78%	78%
A13	2016	97%	96%	86%	88%	84%
	2017	59%	54%	55%	51%	51%
	2018	63%	58%	55%	51%	54%
A20	2016	75%	64%	64%	55%	59%
	2017	87%	83%	79%	75%	72%
	2018	78%	77%	68%	73%	67%

relationship. A13 is added to the proof assignments from now on, resulting in a total of twelve proof assignments.

- **A20:** The students performed better on the non-proof assignment A20 in 2017 compared to 2016; once again, however, I could not find a noticeable difference in the assignments. All assignments had the form of "Give a derivation for the word" and a different order for *a*s, *b*s, and *c*s.

There were also assignments with no significant changes. Overall, no significant changes were apparent in any grading group for 14 assignments (four proof assignments). Another 11 assignments exhibited significant change for only one or two grading groups (three proof assignments). This missing significance means these assignments with a high or low performance in 2016 had almost the same level of performance in subsequent years. Altogether, performance did not differ significantly across seven proof assignments for any of the grading groups.

To sum up the ANOVA results, I was able to find significant increases or decreases in peformance in some assignments. For most of the proof assignments, the same or even a decreased performance was evident, which supports the assumption that students have the most problems with formal proof assignments. The fact that every grading group performed worse in A13 in 2017 and 2018 after adding a proving subtask could also suggest that lower performance resulted from this additional proof assignment. However, it is not always possible to find differences in the assignment that would explain these changes. More research, therefore, should be conducted and additional research methods implemented in order to understand why students particularly struggle with these assignments. Nevertheless, this analysis offers starting points for deciding which assignments would more likely lead to a richer understanding of this problem.

5.4. Comparing Final Exam Results with Homework Assignments

After identifying overall low performance in proof assignments on both final exams and homework, I wondered if these two findings were related to each other. I questioned whether students performing high or low in the final exam had high or low scores overall in the homework.

To investigate the connection between final exam results and homework assignments, I worked with an incomplete data set. This necessity is explained next. As described in the section on course setup (see Section 4.2), students were allowed to submit their homework in groups (by two to four people); it is not unlikely that students could have submitted a correct homework assignment, due to the overall group performance, that they might not have been able to create on their own. To account for this, I used only data where all members of one group performed comparably in the final exam (i.e., the same or one grade difference). Students that did not participate in the final exam were omitted, but their group members still remained in the data set used here. I assume that it is unlikely that a student who has done all the work throughout the homework assignments would not participate in the final exam. In doing so, I prevent the results from being distorted by the fact that students with poor outcomes in the final exam only achieved good outcomes in the homework using a joint solution with a more high-performing student.

After the described data cleansing, I had performance information about 246 (2016), 104 (2017), and 89 students (2018). It should also be emphasized that not all students submitted a solution for every homework assignment and, therefore, the number for every submitted assignment differs. These numbers were considered for data analysis while missing submissions were ignored. I decided to work with the incomplete data set because I did not know anything about the students' reasons and, therefore, had no valid argument not to do it. I assumed that these omissions resulted from lack of time, understanding, or any other reason students why might omit specific assignments. Due to the overall high number of submissions for each assignment, I had sufficient data to conduct an analysis that could create valuable insights.

Next, I analyze the predictive power of the overall homework performance for all three years. Before I present results, I would like to address the limitations of the Spearman's rank correlation coefficient. I chose to work with this coefficient because the data had no linear distribution. Since the Spearman correlation calculates correlation with the help of order through ranks, high homework and final exam scores have a high rank, and low scores have a low rank. A difference between the intervals of the scores is not considered. Furthermore, due to the sample sizes, I cannot completely rule out errors. Schönbrodt and Perugini (2013) demonstrated that a sample size of 250 is necessary to prevent possible errors from affecting the values [SP13, pp. 10]. Because the sample size of 2016 was almost as large as this value, I can rule out errors here and expect similar results for 2017 and 2018 – despite the smaller sample size. Nevertheless, a correlation among all homework and final exam scores in total means that I cannot provide any information about the respective individual homework in which the students have achieved their scores. Furthermore, due to the small number of students, I am also unable to provide information about which grading group these students come from.

When I calculate the coefficient with the presented limitations in mind, there is at least a significant positive medium correlation (r = .434) in 2016. This value means that for around 18% of all analyzed students, a similar high or low rank in homework score and final exam score was found. Therefore, it can be predicted how these students would score in the final exam based on their homework score. Due to the small sample size, I treat the data from 2017 (r = .475) and 2018 (r = .532) with caution, but a similar correlation would not be surprising considering the similarity in the previously presented performance analyses. The corresponding p-values show that the probability that this medium correlation is random is less than 1%. Therefore, it is not necessarily possible to predict the final exam results from the average performance on homework. From this, I can conclude homework assignments served their formative purpose and left room for most of the students to practice ToC-related tasks.

5.5. Developing Hypotheses

In this section, I summarize the results in data-driven hypotheses and relate them to the research questions:

- **RQ1: What kind of assignments usually covered during an undergraduate introductory FLAT course are causing students the most difficulties?**
- **RQ2: What differences can be found among low and high performing students, especially regarding potentially different assignments as questioned in RQ1?**

Outcome 1: The analysis indicates that specific assignments lead to a lower student performance. In the final exam of 2016, three assignments showed a lower student performance, of which two assignments required formal proving (at least in part). In the final exam of 2017 and 2018, the formal proof assignments continued to be the ones with lowest student performance. In the homework assignments of 2016, eight of 11 proof assignments had a low performance in each grading group. This student performance stayed mostly the same in the homework of 2017 and 2018. For two homework assignments, I even found a significant decrease in all five grading groups, and I believe that one of these might stem from an additional proof subtask that was added in 2017 and 2018. Through the replication of the results based on the 2016 data, I found evidence again that all students showed performance strength as well as deficiencies, each in the same assignments. More precisely, all students showed the lowest performance scores in formal proof assignments, independent of the particular ToC topic and related characteristics to be proven. This leads, therefore, to the following data-driven hypothesis:

Hypothesis 1: In ToC introductory courses, formal proof assignments in general (i.e., independent from their ToC context) are the most challenging assignments for all students (i.e., independent of their overall final exam performance). In addition, the challenge factor of the proof assignment increases with the level of formalization.

Outcome 2: Ranking the final exam and homework scores according to the Spearman correlation coefficient, the results of this correlation indicate that for the majority of students it is not necessarily possible to predict their final exam results from their average performance on homework. (More precisely, across all grading groups, only 18% of student homework performance correlated significantly with final exam performance.) This means that students performing strong in homework assignments did not necessarily perform comparably strong in the final exam and vice versa. This result shows that homework assignments served their formative purpose and left room for most of the students to practice ToC-related tasks. Building on this, there is no foundation for classifying students into high and low performers in a ToC course as their performance may differ in terms of actual assignments both in homework and in the final exam. Consequently, this leads to the following data-driven hypothesis:

Hypothesis 2: In ToC introductory courses, students' homework performance does not allow for prediction about performance on final exams.

5.6. Further Discussion

The first hypothesis from my quantitative study (see Section 5.5) shows that formal proof techniques are significantly challenging to students in ToC courses, as presented by some of the studies for reduction proofs or complexity proofs in Section 3.3 (e.g., [Arm09] [GET05] [PL14] [Ens14]). Unfortunately, there were no reduction proofs or complexity proofs within the assignments of the Formal Languages and Automata (FLAT) course. Nonetheless, there were also completely different kinds of proof assignments which the students had to solve. Here, the less formal the proof, the better the performance. This underscores the previous assumption that students are particularly challenged when they must use formal reasoning methods. Since no subject showed particularly low performance, I also support the argument that abstract mathematical language is more of a problem than the theoretical concepts themselves.

At this point, I cannot say whether these results contradict the assumptions of other works in Section 3.1. On the one hand, a general performance analysis cannot indicate whether students are not motivated or interested (e.g., [CnGM04] [HK04]), or whether they have difficulty grasping the relevance of ToC for their future work (e.g., [HHP03] [Sig07]). However, as previously stated, these assumptions were not extensively empirically examined and validated until now. On the other hand, the works in Section 3.2 often have the goal of creating a connection between practice and formal abstraction with the assumption that students will also find it easier to understand the formalisms once the visualization has become more apparent to them. Moreover there is not enough work that explicitly deals with evaluating how the transition between visualization and formalism works for the students. It is also not clear how students return to developing their own proofs without tool support – for example, through Interactive Theorem Provers (ITP) (see Section 3.2), and whether they are then more able to develop formally correct proofs.

In addition to the result that students have low performance in many proof assignments, another result can be presented: proof techniques present a challenge for all students (high-performing and low-performing as well). Building on this finding, the second hypothesis raises questions about the bimodality of student performance in ToC (see Section 1.1). I can refute the argument that students with high final exam scores displayed an overall strong performance for all assignment types covered during the course. In contrast, students with low scores on the final exam also showed mixed performance depending on assignment types. With these results, I reinforce the argument that ToC courses need to reconsider their traditional pedagogy model, and I discuss one specific potential approach to how this can be implemented.

As mentioned before, Gal-Ezer and Trakhtenbrot (2016; see also 2013) analyzed students' misconceptions about reduction proofs [GET16] [Tra13]. Since the authors developed a series of exercises addressing students' potential misconceptions proactively, they aimed to help students to identify their misconceptions. Therefore, the authors could better support them in building an understanding of reduction proofs. As this proactive approach runs counter to traditional pedagogy in ToC courses, I want to underline the importance of Gal-Ezer's and Trakhtenbrot's research approach. They have closely analyzed their students' problems with a specific topic and have attempted to address the problem so as to improve their students' situation directly. Furthermore, one option for a schema-based proof like the PL would be not to generally try to make the proof development more interactive or practical, but to focus instead on teaching it in smaller increments. At this point, it should be noted again that ITP scaffolds students' proof development and provides immediate feedback (see Section 3.2). Nonetheless, I already provided information about the possible limitations of such approaches (see Section 3.5).

As already described in Section 3.4, mathematics undergraduates also face difficulties with formal proofing and abstraction just as Computer Science (CS) undergraduates do. To conclude this chapter, I would like to note two existing studies of mathematics students that parallel my present findings. Anapa and Şamkar (2010) found that even students generally successful in mathematics do not trust their proving abilities, but they made no statement how these students really perform [AŞ10]. On the other hand, Stylianou et al. (2015) made the statement that "high-performing students tended to hold a more positive and active stance with respect to their beliefs about proof than their low-performing counterparts" [SBR15]. In this case, "performance" is determined by the multiple choice test explicitly designed for their study. However, the high-performing students here seem to have fewer problems dealing with proofs, but how they write them is not explicitly investigated. Some of the answers of the high-performing students in the multiple-choice test are particularly interesting. For example, the statement "I enjoy the challenge posed to me when doing proofs" received on average only 50% agreement. "I think it is important for assessments to include constructing mathematical proofs" received only 35%. The statement "I think I would benefit from classroom instructions that involved working in groups with my classmates to discuss how to prove mathematical statements" received a 45% response. On the last item, the low-performing students had a minimally higher agreement with 50%. This study shows that while some of the students studied may have been able to implement or understand proofs, they did not enjoy learning them. Overall, it is unclear whether the high-performing students would also score comparably high in the actual assignments in class. Therefore, the study results are only comparable with ours to a limited extent. However, I would like to emphasize that even mathematics students who perform highly or are successful in specific (or even most mathematical) topics do not necessarily develop an equally deep understanding of proof methods or its importance.

5.7. Summary

To get an overview of the topics and assignments in an introductory ToC course that students have the most difficulty with, I conducted a quantitative study in a FLAT course. In addition, to examining whether student performance in the course is bimodal, I explicitly chose to separate students by their final exam grade and group them on that basis.

Using Exploratorative Data Analysis (EDA) as an approach and various statistical methods, I investigated the final exam performance and the homework assignment performance of around 500 students. I found that student performance was lowest on most proof assignments. Overall, the low performance was independent of the final exam grade. More specifically, the lowest student performance on the final exam was on a proof about regular languages using pumping lemma, while the corresponding homework assignments had higher performance.

To validate the results, I conducted the same analysis for the following two years (i.e., for about 1000 additional students) and obtained similar low performances for proof assignments and the discrepancy in the pumping lemma assignments. In addition, I analyzed the three cohorts considered for significant differences. One notable finding was significantly poorer performances on a task in each grading group after adding a subtask that required a proof.

Lastly, I also calculated whether a correlation between final exam and homework performance could be found, according to which final exam results could be predicted based on homework performance. Here, a correlating performance was found for only one-fifth of the students.

In total, I summarized the results in two hypotheses describing how formal proof assignments are the most challenging for all students independent of their overall final exam performance, and homework performances does not necessarily allow for prediction about final exam performance.

Part III.

Collaboration and Learning within Student Groups

6. Theoretical Background

As mentioned in Section 1.2, students in collaborative groups can build up shared knowledge that goes beyond their own understanding as indicated by [Ros92]. In order to examine collaborative groups, I first take a closer look at the concept of collaboration. Furthermore, it appears likely that, as a counterpart to shared knowledge, there also exists individual knowledge that belongs to only to a single person. Before I approach these different types of knowledge and use them for my analysis of students' learning processes in collaborative groups, I illustrate how my understanding of learning is shaped by Distributed Cognition Theory (DCOG).

6.1. Collaboration and Interaction

Considering various possibilities of collaborative learning in different Computer Science (CS) courses as mentioned in Section 1.2, the question arises as to what collaboration means in the first place. Building on several theories that understand learning as a social activity (e.g., [Mea34] [Vyg80]), Roschelle and Teasley (1995) define collaboration as "a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem" [RT95, p. 70]. In the context of analyzing collaborative problem-solving situations of students, "activity" is understood as "a coordinated effort to solve the problem together" [ibid., p. 70]. Particular attention within this definition is paid to students "coordinating" their activities (i.e., deciding on how they arrange their activities together and deciding on a shared conception of the problem) and working "synchronously" while being in the same room at the same time. In this setting, a "problem" can be any kind of task, such as a homework assignment, while the "shared conception" of a problem is the goal of the activity, such as a homework solution.

Existing work investigated how students' critical thinking was enhanced through collaborative situations by encouraging students to engage both with the material and other group members and learn how to formulate and defend their arguments (e.g., [Esp18], [KSLF13]). Moreover, studies report significant advantages of collaborative work between students in general skills, such as communication [TCC$^+$01] and social behaviors [TK04]. On the other hand, studies in this field indicate that different learning techniques are beneficial either for individual learners (e.g. studying examples) or for collaborative learners (e.g. learning from solving problems) [KPKJ11]; moreover, Jeong and Hmelo-Silver (2016) have drawn attention to the fact that "the final outcome [of collaboration] is not a static aggregation of individual contributions and may go beyond the understanding of its individual members (Stahl, 2006)" [JHS16, p. 255].

Since most of the presented work about collaboration (outside of CS) used quantitative data or surveys to measure student learning in working groups, a "focus on individual contributions and individual outcomes leaves out much of the richness of student interactions [within a collaboration]" [Bar00, p. 406]. In order to gain deeper insights into how students work in collaborative learning situations, I consider it necessary to shift the focus of the analysis away from the individual learning success to the unit that makes collaboration possible in the first place – that is, the *interaction* within the group. To this end, I rely on Barron's (2000) definition of interaction as "the dynamic interplay in meaning-making over time in discourse between participants, what they understand, the material resources they have available and choose to utilize, the type of contributions that they make and how those are taken up in a given discourse" [Bar00, p. 406]. Building on this definition then, interaction among students in a study group is understood as the actual actions and reactions between two or more people while using material and resources within the overarching activity – that is, within the collaboration.

To focus on interaction as the central aspect of collaboration, Barron (2000) exemplified how analyzing interaction can provide valuable insights into students' working processes. By analyzing two triads of high-performing sixth-graders, she showed how students interacted in noticeably different ways to solve problems. One group had a "high degree of mutuality and consistent joint attention, and all members' efforts were directed towards sharing the work of problem solving" [Bar00, p. 432]. On the contrary, the other group had problems "[reaching] a common ground," but "one partner's persistence to have his ideas acknowledged, coupled with the other partner's momentary discarding of his own unproductive solution path, led to a moment in which [...] common ground reestablished" [ibid., p. 432]. Furthermore, Deitrick et al. (2015) observed how a group of younger students interacted while programming a computer music system [DSA$^+$15]. It became apparent, that one student's extensive musical background advanced the overall problem-solving process.

6.2. Distributed Cognition Theory

Cognitivism can be seen as the predominant learning theory during the latter half of the twentieth century [TK14]. The theory is focused on the term cognition that traditionally "refers to the brain's mechanisms for information processing, error correction, memory, perception, and communication" and is "located solely within the skull" of individuals [DSA$^+$15, p. 52]. Cognitivism, therefore, became a counterpart to the learning theory of behaviorism, which only considers observable behavior as the subject of psychological studies.

By conducting a cognitive ethnography of navigation aboard U.S. Navy Ships, Hutchins (1995) challenged the traditional cognitive perspective [Hut95]. Through analyzing the navigation, he discovered that "the very notion of distributed cognition and the need for cognitive ethnography arose from the observation that the outcomes that mattered to the ship were not determined by the cognitive properties of any single navigator, but instead were the product of the interactions of several navigators with each other and with a complex suite of tools" [HHK00, p. 183].

By considering human cognition as distributed, DCOG is extending Activity Theory, which has its roots in Russian cultural-historical psychology: "Vygotsky and colleagues (1978) analyzes human activity as having three fundamental characteristics; first, it is directed toward a material or ideal object; second, it is mediated by artifacts; and third, it is socially constituted within a culture" [BB03, p. 299]. Thereby, the artifacts such as tools and languages "are often metaphorically referred to as tools or *mediational means* [34]." [KF16, p. 74]. Similar to Activity Theory, Hutchins' research (1995) considered cognitive processes as not just individual and mental but distributed "across the member of a social group", between "internal and external (material or environmental) structure", and over "time in such way that the products of earlier events can transform the nature of later events" [HHK00, p. 176]. In sum, "the cognition process is understood to be distributed between, several agents, between agents and tools, and/or through time and space ([15], p. 176)." [KF16, p. 74].

For the case of observing groups, it is especially "important to come to an understanding of the ways in which cognitive properties of groups may differ from those of individuals" [Hut95, p. 177]. Overall, Hutchins' research (1995) did not explicitly focus on the learning process, but rather on the collaboration and the understanding of individuals' cognitive processes. For that matter, he described how individuals contributed through interaction to achieve a common outcome they could not have achieved on their own.

6.3. Individual and Shared Knowledge

Recently, DCOG has been used as a framework for analyzing learning in collaborative groups (e.g., [DSA+15] [KF16]). As such, learning is extended beyond the individual; it involves "no longer just change in individuals' conceptual models (a constructivist take on what learning is) or behavior (a behaviorist take), but also includes changes in the relationships between individuals and in their individual and joint relations to tools and settings, which can also be modified over time" [DSA+15, p. 53]. I follow this approach by understanding the concept of learning within the whole cognitive system of which our students are part (e.g., their self-organized study groups and the interplay between mathematical inscriptions, theoretical objects, and mental and collaborative processes [Kno15]). Now, in light of DCOG, the "inscriptions and visualizations used in lecture notes embody not only the factual knowledge that is usually paid attention to, but also knowledge for how to use these tools in order to create further factual knowledge, e.g., when working on an assignment" [KF16, p. 74].

Under the presented conditions, students' learning is understood as internalizing knowledge – that is, the internal reconstruction of external situations embodied by the respective environment and the use of various tools. I refer to this internalized knowledge as *individual knowledge*. As *shared knowledge*, I understand the knowledge that arises when individuals in collaborative groups externalize their (not necessarily) disjunctive individual knowledge through different tools to a shared body of knowledge that is necessary, for example, to solve tasks [AH14].

Nevertheless, Hutchins (1995) himself briefly linked his description of DCOG to the idea of learning by asking "What happens if we consider adults learning more complicated thinking strategies in more complex social settings where the primary goal of the activity is successful task performance rather than education?" [Hut95, p. 283] By this question, he underlines the aspect that student homework groups may have the aim to solve assignments correctly by contributing parts of their individual knowledge to a shared knowledge instead of building new knowledge or expanding their individual knowledge.

7. Study Design

One of the most notable findings of the quantitative studies presented in Part II was the performance discrepancy between the individual final exam assignment and the Pumping Lemma (PL) homework assignment. I found that all students, regardless of their overall final grade, had an average lower performance on their individual final exam assignment; but the particular homework assignment they worked on as a group demonstrated a higher performance. Building on these findings, I start this section by motivating the research questions for the qualitative study. Then, I present the overall research methods and used software, before illustrating the data collection and providing background information about the student participants. In the end, I present the conducted data analysis. For all the remaining sections, please note that I italicize individual letters, symbols, and numbers needed, so as to distinguish them from the remaining text.

7.1. Research Questions

As illustrated in Section 3.3 there exist several studies providing insight into Computer Science (CS) students' difficulties with proof assignments (e.g., reduction [AGE06] [AGEH06] [Arm09] [GET16] [Tra13]); NP-completeness [KF16]; big-O notation [PL14]). I also identified some studies that are especially focused on students working on PL assignments [Pil10] [GET05]. A more detailed study of PL assignments by Smith and McCartney, investigating students' errors in PL assignments revealing student issues with quantifiers, abstraction, and symbolic formulation [SM14]. As these studies mostly focus on students' written solutions, insights into the actual student working process with PL assignments are not present to complete the overall picture and provide further insights about students' entire learning process in Theory of Computation (ToC) courses. For example, it is not known whether those student errors revealed by Smith and McCartney can be found earlier during homework assignments or are, for example, just the result of the exam situation. Therefore, I began to examine the data first to answer the following research question in Section 8:

- **RQ3: What kind of pitfalls and challenges do students encounter in a self-organized study group when working on a PL homework assignment?**

Having gained an overview of the students' work process and their content pitfalls and challenges through RQ3, I shifted the analysis's focus towards collaboration. As mentioned before, my quantitative analysis revealed that students have the lowest performance in the final exam in PL assignments. At the same time, they perform higher in their homework assignments, which they must solve in group work (see Section 4.2). Although at first glance, it does not seem difficult to guess why students develop better solutions together than alone (e.g. group work, more time), I could also expect that within this group work, students gain a kind of understanding that should enable them to perform comparably in final exams. I followed Barron's [Bar00] and Deitrick's [DSA$^+$15] research that describes student interaction as a valuable source for understanding how student groups actually work together on homework assignments (see Section 6.1). Since this kind of research has not yet been done in the field of ToC, this next step of my analysis pursues a twofold objective. First, my analysis provides insights into student interactions during homework groups using PL assignments as an example; and second, it investigates how these interactions are related to the student performances in similar final exam assignments to uncover further possible reasons as to why students fail ToC courses. Therefore, I explore the following research questions in Section 9:

- **RQ4: What kind of interaction between students can be observed within ToC study groups while they are working on homework assignments?**
- **RQ5: In what way are the observed interaction forms related to the group's assignment performance as well as students' individual final exam assignment performance?**

In the following, I expanded on the previous focus on the interaction process. When analyzing and comparing homework and final exam assignments, it is hard to avoid wondering if, how, and what students learn in the homework that leads to low results at the end of the course. Interaction alone can only reveal visible behavior or communication as the cause of low performance. However, the interactions observed and evaluated during RQ4 and RQ5 also provided information about how students share their knowledge during the interaction and what tools they use. Therefore, I conclude by examining what happens at the level of individual and shared knowledge in light of Distributed Cognition Theory (DCOG). This focus extends the view from how students interact with each other and with the available material to how students manage their knowledge in student groups – that is, how they learn. Based on these concepts, as defined in Section 6, I answer the following research question at the end of Section 9:

- **RQ6: How do students externalize individual knowledge to shared knowledge in their homework groups?**

7.2. Research Methods

The following section presents the development of video data as a research source. Afterwards, I present the research method's videography and video interaction analysis as well as the software that was used (MaxQDA).

7.2.1. Video Data as Research Source

Knoblauch et al. (2014) describe images and photographs as the general antecedents of videos, as they allow emotions and expressions to be captured and discussed. The authors specifically cite Charles Darwin's book "The Expressions of the Emotions in Man and Animals" as important literature because he used photographs to compare the emotions of humans and animals. They consider this work is a famous example of how the development of photographs (or a sequence of photographs and early film) was essential for behavioral science studies [KTS14, p. 29].

Knoblauch et. al (2014) argue that in 1963, the approach known as "context analysis" described the importance of understanding behavior and movement within a context. Among other things, they state how the following principles were recommended: no body moment is without meaning in the context in which it occurs, and body activity can and does influence the behavior of other group members and even has communication functions. Another movement that they understand as primarily relevant to the presentation of visual material is the "ethnological or ethnographic film," which uses ethnological methods. In this film tradition, the authors see the emphasis is on documenting behaviors and expressions in other cultures. They describe how, as film technology improved, the quality of films increased and their production became easier. Following these developments, they report the use of film gained broader recognition in the field of workplace studies [ibid., p. 39].

Overall, Knoblauch et al. (2014) summarize the importance of videos for research as "video recordings [...] enable us to retain visually graspable processes in mimetic form. In addition to the role of language, gestures, facial expressions, posture, and body formations, video recordings also allow us to grasp the role played in interaction analysis by accessories, clothing, speech style, and sounds, as well as setting and social ecology. With video, these elements can be observed synchronically, in their simultaneous interplay, as well as diachronically, over a period of time" [KTS14, p. 41].

Therefore, the authors report how video data are a valuable source for social research, having several advantages, such as the following:

- Everyday situations can be captured – these insights are less manipulated.
- Expressions and behavior can be recorded within their actual context.
- Detailed and uninterrupted pictures of social processes can be captured.
- Situations can be repeated, fast-forwarded, rewound, and viewed in slow motion to gain more detailed insights.
- In contrast to surveys or interviews, the camera can be focused on the situation itself instead of talking about the situations.

The authors also describe how video data reduces three-dimensional space to a two-dimensional screen and is therefore not a perfect copy of the real world [KTS14, p. 44]. Furthermore, they state that it must be considered to what extent people are influenced in their behavior by the presence of cameras. Ultimately, they stress that the camera will always focus on a particular situation and thereby likely neglect other, collateral situations ("subjective camera") [ibid., p. 49].

7.2.2. Videography

In order to analyze the video recordings of students working on their homework assignments, I used video data and chose to use *videography* as meant by Knoblauch et al. (2014) [KTS14]. The authors define videography as a method that specifically links video analysis with ethnography specifically by focusing on data gathered in the context of an ethnographic collection process [KTS14, p. 19]. In the following, the term "ethnography" refers to focused ethnography, which according to the authors differs from conventional ethnography in that it also uses recorded situations and not just field notes and diaries [ibid., p. 72]. They further describe how ethnography can be understood as the process of describing and analyzing a picture of a research field as completely as necessary for the research focus. Thereby, the data is obtained, on the one hand, by conducting interviews and collecting documents, and, on the other hand, by purely observing (and recording) the situations.

The authors summarize the core of videography as follows: "Researchers go 'to the field' and focus the video camera on natural situations in which actors act, and they analyze how they act" [KTS14, p. 20]. Essentially, each part of the sentence describes the specific features of videography:

- **"Researchers go 'to the field'..."**: The authors describe how like ethnographers, researchers physically enter a field. The field represents the location where the observed situation and recordings take place. Unlike ethnographers, the researchers only observe rather than actively participating in the field [ibid., p. 20].
- **"and focus the video camera..."**: The authors argue that video recording is limited because the camera and the duration of the recording focus on specific situations. However, unlike audio-only recordings, video recordings obtained through videography are more detailed, complete, and accurate. They state how the technical possibilities to repeat, stop, and slow down the video recordings make it possible to notice details that are not perceived by the participants themselves in real time. To focus on more than one situation, more cameras can be used [ibid., p. 21].
- **"on natural situations..."**: The authors explain how videography differs from other variants of video analysis because it focuses on "natural" situations which take place in a limited space and time and in which actors are

involved. By "natural" they meant that these interactions represent everyday situations of specific actors that are not intentionally and specifically created for research [ibid., p. 22].

- **"in which actors act, ...":** The authors describe that in the case of videography, researchers assume that social reality is created by the everyday actions of actors. Accordingly, the *how* of the actions is of interest [ibid., pp. 23 – 25].
- **"and they analyze how they act.":** The authors argue that before a precise analysis can take place, the data collected with videography must be made accessible. They stress that since even a few minutes generate a large amount of data that needs to be processed for analysis, it is necessary to identify relevant segments of the video material that can be subjected to detailed analysis [ibid., p. 26].

7.2.3. Video Interaction Analysis

Since videography refers to a comprehensive research process including the capturing of data, Knoblauch et al. (2014) developed video interaction analysis (VIA) as a technique for the actual analysis of interaction that is depicted in video recordings [KTS14, p. 53]. Accordingly, VIA builds on three analytical characteristics of interaction [ibid., pp. 69 – 70]:

1. Methodicity: "the actors themselves already organize their actions in a systematic way."
2. Orderliness: "in everyday actions, the actors create meaningful sequences."
3. Reflexivity: sequences "can be reproduced systematically [...] and simultaneously made understandable in terms of their execution."

In the following, I summarize the steps of Video Interaction Analysis (VIA). Overall, the steps in the analysis process can be carried out in different orders or with different degrees of detail. This depends on the amount of data, the quality of the data, or the research focus. I arranged them as described below to increase comprehensibility (see Figure 7.1).

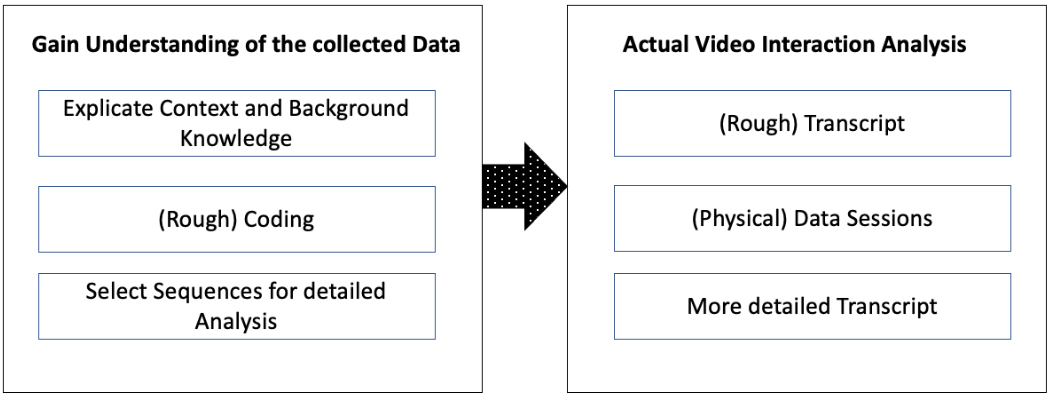

Figure 7.1.: Video Interaction Analysis steps

The starting point of any VIA is not the analysis itself, but the attempt to understand the data collected. In order to achieve this understanding, the following steps can be taken, depending on the data material and the research focus:

1. **Explicate context and background knowledge:** The authors state that as a prerequisite for using VIA in the first place, researchers should have sufficient information of the research field and understand the context in which they want to collect data [KTS14, p. 93]. To embed the recorded situation in this context, the authors stress that it is also useful to know what relevance the observed situation has in the context beyond the filming session itself (e.g., for the actors) [ibid., pp. 94]. To make the analysis and interpretation comprehensible for a potential reader, they recommend to explicate this context and the background knowledge of the field in a results presentation [ibid., p. 96].

2. **(Rough) coding:** Depending on the length and type of data, the authors do not always see it as reasonable or possible to analyze whole sessions. In this case, a first (rough) coding can help the researcher structure and provide an overview of the data [ibid., p. 86].

3. **Select sequences for detailed analysis:** The authors stress that during the analysis, the researcher is faced with the problem of how to select the sequences that are relevant for further investigation [ibid., p. 88]. Selecting the relevant sequences cannot be easily formalized; for example, the corresponding research question and the data quality play a decisive role in the

selection decision. The authors describe two principles as helpful in guiding the selection of relevant sequences: "First, the researchers are guided by the relevance of the observed actors" (i.e., *reflexivity*). "Second, the researchers pay attention to recurrent features of the interaction that is taking place, features that perhaps have not been noticed by the actors themselves" [ibid., p. 94].

After understanding the data and deciding on relevant sequences, the actual video analysis can follow:

4. **(Rough) transcript:** The authors state that the sessions or selected sequences should be transcribed. This transcript can serve as an orientation that structures the flow of the sequences for the researchers [ibid., p. 102]. In addition, the transcript provides an overview of the order of the sequences (i.e., *orderliness*). At this step, the transcript may still be a draft and serve as a working document for the (physical) data sessions.
5. **(Physical) data sessions:** The authors recommend to meet with other researchers to discuss chosen sequences and derive interpretations. These (physical) meetings tend to improve the understanding of "what" and "how" of the observed interactions (i.e., *methodicity*). Furthermore, such meetings help to further validate the outcome of the data analysis [ibid., p. 97].
6. **More detailed transcript:** If the transcript was previously a draft, the authors recommend a more detailed transcript of the sequences to be created after the data sessions [ibid., p. 110]. This detailed transcript no longer contains only the conversations, but also visual observations discussed in the data sessions.

7.2.4. Transcription and Coding Software MaxQDA

In order to analyze the sessions, I used the coding software MaxQDA[1]. Figure 7.2 shows the interface and design of the software. The window "List of documents and data sources" contains the video and audio files that I loaded into the program. I created a folder for each group and its data (with the names A, B, and C). In addition, a transcription could be linked directly to the data, so that the time stamps stored in each case are directly accessible. The window "List of codes"

[1]https://www.maxqda.com, last visit 21 July 2021

contains any number of user-defined codes to which a color can be assigned. In addition, their frequency of the codes is displayed on the right. By clicking twice on a code, a list of all passages in all documents that have been assigned to this code is displayed in an additional window. In the window "Coded transcript", the codes can be drawn on the individual parts of the transcripts.

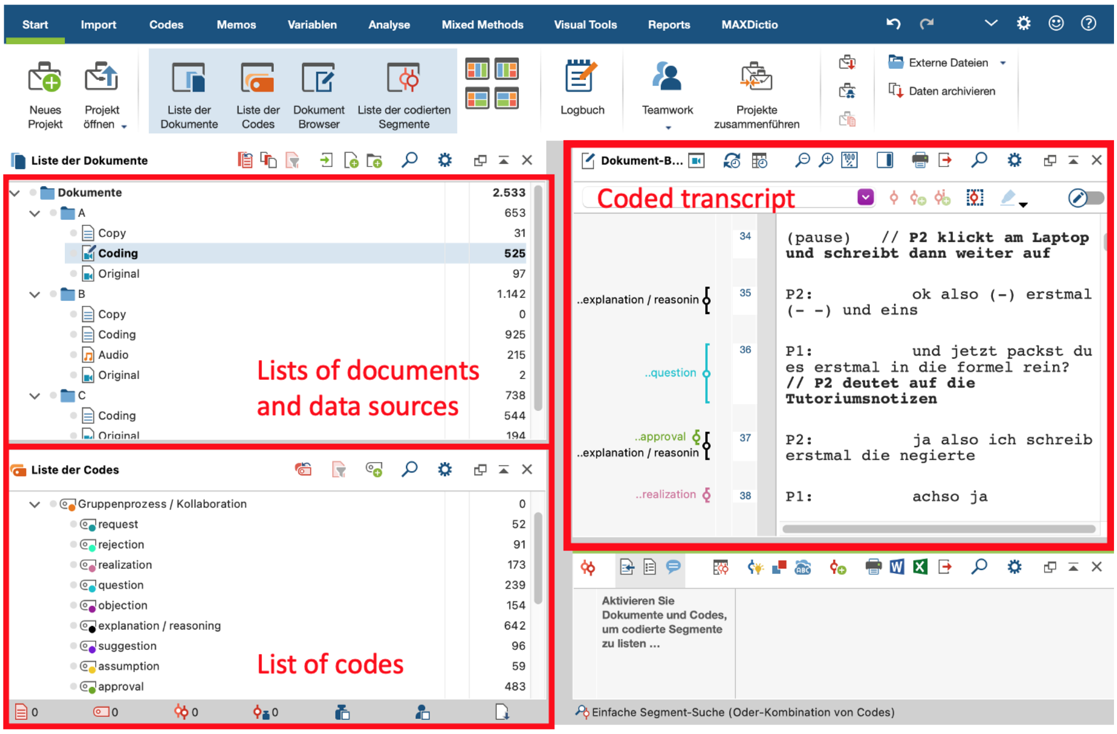

Figure 7.2.: The interface of the coding software MaxQDA.

I transcribed the students' homework groups based on the GAT-2 rules [SABW11]. In general, I used the following rules for a first transcript (The complete transcripts can be found in the appendix.):

- All text is written in lower case and without punctuation
- Depending on their length, pauses are given with (.), (-), (–), (—), or (pause)
- Information about how something is said are given within < < and > >

- Incomprehensible passages are represented by (unv.)
- Simultaneous text passages are represented by [ ] in the respective lines
- Ambiguous text passages are represented by (*text*?)
- Behavior is indicated with //

7.3. Setup and Data Collection

In order to get to know the course design as well as the assignments and topics for the data collection beforehand, I had an intensive exchange with the lecturers and instructors. The data was gathered in week 16 of the FLAT course described in Section 4.2 during the winter term 2019. One week before the data collection, I visited all 26 tutorial sessions, where I explained the purpose and the aim of my research and emphasized that I was not involved in grading, homework design, or any other teaching activities of the FLAT course. Furthermore, I stated that this study was not meant to test students but rather to understand better how students work on such assignments and gain insights for improving pedagogical approaches. (For my complete announcement, see the appendix.) Afterwards, I asked students for permission to attend, observe, and record their homework group sessions. Three groups volunteered.

I provided the students a seminar room with a round table, chairs, power plugs, and a whiteboard. Furthermore, a consent form for the use of the video data was signed by the students and me. (The template can be found in the appendix.) Observing the sessions, I did not participate or communicate with the students about their assignments or solutions ("direct non-participant observation", [CBÖ18, p. 44]). Because the students worked in groups, their dialogues were natural think-out-loud sessions without interference from my side. The two observed assignments the students had to solve were part of their course homework so that they did not have to put in any extra effort due to additional assignments. (The observed PL assignments are presented in Section 2.2.2.) With the aim of not only capturing the students' voices and conversations but also comprehending what they wrote down and how they interacted while working on their assignments, I decided to use three cameras and an audio device for the recording. As shown in Figure 7.3, one camera filmed the students from the front, one from the side (Panasonic HC-V777EG-K), and one from the ceiling (GoPro Hero5); the audio device was placed on the edge of the table (Boundary microphone Sennheiser E 901 and audio recorder Zoom H4n Pro).

After the students decided to end the session, I was allowed to photograph their notes. Furthermore, I conducted interviews to get more information and background knowledge about the students in our study. More precisely, I collected data about the students – for example, the semester they are in, which study program they are attending, and information about their working process, such as why they decided to solve certain things in a certain way or whether this was a typical working process for them. (Refer to the appendix for the complete interview guide.) Furthermore, a few weeks later, I also had the opportunity to analyze the pumping lemma assignments from the observed students' final exams. Since I could not communicate with the students again after the final exam and ask them specific questions about their solutions, I was able to only use the final written solutions for further analysis. In doing so, I noted how the teaching assistants assessed the solutions and which mistakes prevented them from getting a higher score.

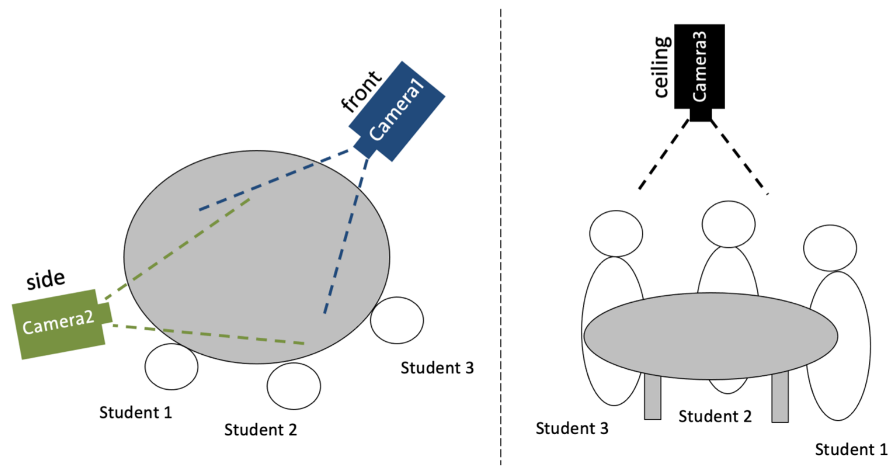

Figure 7.3.: Position of cameras and table in the provided seminar room (left image from above, right image from the side).

7.4. Study Participants

Within my study, I name the observed groups A, B, and C and enumerate the participating students from left to right, according to their seating arrangement. As seen in Figure 7.4, A2 had a writing pad in front of her; during the session, she was the only one writing on it. A1 had A2's existing notes from the tutorial session in front of him. In group B, all students had sheets of papers in front of them, but B2 was the one writing the solution down (Figure 7.5). B1 used his sheet to copy B2's final solutions, and B3 used his sheet for short notes. In group C, the students worked independently on the solutions on their own writing pads (Figure 7.6). During the session, they discussed some ideas. In all groups, laptops and smartphones were only used to display the actual assignment sheet, the course formulary, or sample assignments and solutions from the tutorial sessions. Through interviews, I gained further statistical data that are presented in Table 7.1

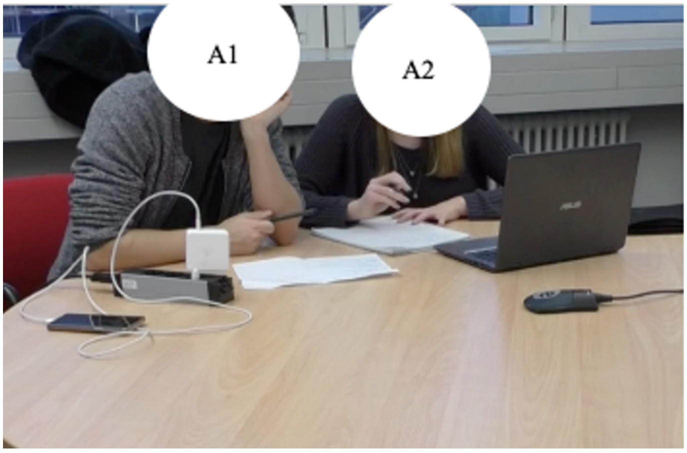

Figure 7.4.: Seating distribution of group A.

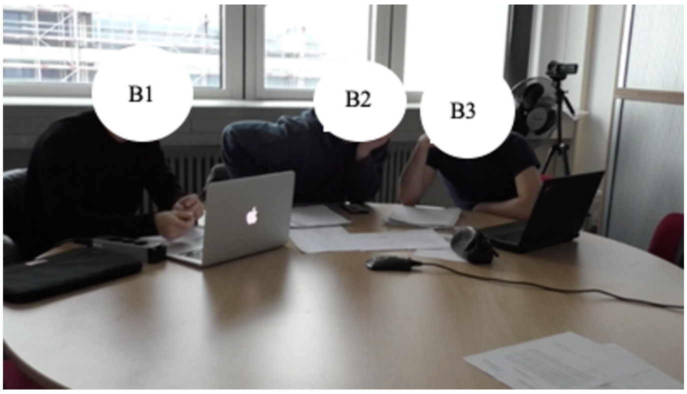

Figure 7.5.: Seating distribution of group B.

Figure 7.6.: Seating distribution of group C.

Table 7.1.: Statistical Data about the observed students including performances in the PL assignments (FE = Final exam, PL = Pumping Lemma, HW = Homework)

Student	Lecture Attendance	Tutorial Session Attendance	HW PL Assignment Score	FE PL Assignment Score	FE Errors	FE Score
A1	Never	Always	10/12	0.5/11	Wrote only a gap text	C
A2	Always	Always	10/12	7.5/11	Formal errors	B
B1	Never	Never	8/12	-	-	-
B2	Mostly	Always	8/12	10.5/11	Missed writing down conditions	A
B3	Mostly	Always	8/12	10.5/11	Missed writing down one assumption	B
C1	Mostly	Mostly	6/12	7/11	No reasonings for correct decisions	B
C2	Mostly	Mostly	6/12	1/11	Lack of case distinctions	D
C3	Mostly	Mostly	6/12	1.5/11	Lack of case distinctions	B
C4	Mostly	Mostly	6/12	1.5/11	Lack of case distinctions	B

8. Analysis of Pitfalls and Challenges

To answer RQ3[1], I decided to collect content-related information to gain a deeper insight into the working processes of the students and possible difficulties with Pumping Lemma (PL) assignments. This chapter explicitly also serves to provide background information on the observed group processes and to establish a basis for the following qualitative analysis. This section starts with the respective data analysis steps before presenting and discussing the results.

8.1. Data Analysis

The data analysis was based on Qualitative Content Analysis (QCA) (see Section 4.4.2) using a summarizing and interpretative technique. I used the software MaxQDA (see Section 7.2.4) and performed the following steps during my analysis process:

1. **Background Knowledge:** I acquired context and background knowledge through intensive exchange with teaching assistants and lecturers.
2. **Transcription:** Since the three sessions were each no more than two hours long and I was interested in the whole working process, I chose to transcribe the whole sessions. Besides transcribing the verbal statements, I also took notes of which students or which material the students interacted with.
3. **Analysis of Pumping Lemma Steps:** After transcribing the observed study groups, I used the PL scheme presented in Section 2.2.2 to structure the

[1] "What kind of pitfalls and challenges do students encounter in a self-organized study group when working on a PL homework assignment?"

transcripts for further analysis. In this way, I aimed to identify those sequences which students actually worked on in the PL assignments. To achieve this, I transferred the PL scheme to codes that relate the single steps of the scheme to activities observable in the group sessions. I coded the data with these codes using summarizing QCA (see Section 4.4.2). In addition, I used the PL steps to represent the group activities in two graphical ways to analyze how much time each group spent on each activity: (1) Pie chart: I calculated an approximate time share of each step of the entire work process in each group. For that matter, I counted the number of lines in the transcripts coded with one step and calculated the percentage of total lines in each session. This does not correspond to the exact amount of time each activity was related to one step but is a sufficient approximation for my purpose because two student groups were constantly speaking, while only one group had a few silent phases. (2) Line chart: I split the written transcript into five-minute blocks and counted first the overall number of lines within one block creating a benchmark for a time frame of each line. Next, I counted the number of lines belonging to one category within the block in order to determine how much time students spent on that activity.

4. **Analysis of Pitfalls and Challenges:** With the sequences related to the steps of the PL scheme, I focused on potential pitfalls and challenges, considering all sequences in which students, asked questions, felt uncertain, missed relevant information to proceed with, or made incorrect assumptions and decisions. To determine this, I consulted sample solutions for the assignments. In this data analysis step, I defined a coherent statement per person as the smallest data unit, which could be longer than one sentence. The result of this data-driven analysis was a summary of all pitfalls and challenges students encountered during their working sessions.

5. **Analysis of Origins and Overcome:** To provide more information about the students' pitfalls and challenges, I further evaluated the results from the second step questioning the cause of students' pitfalls and challenges as well as their ability to overcome them. To do this, I re-examined all the recorded sequences with pitfalls and challenges and explicitly focused on finding the origin.

8.2. Results

In this section, I first present the three study sessions through single sequences in which students worked on the PL assignments. In order to get a complete picture of the overall session, I also add descriptive information about what material the students used to make sure their solution was correct. Second, I introduce the pitfalls and challenges that students encountered during their work process, identify the actual reasons for students' difficulties, and whether they were able to solve them. The section ends with a discussion of my findings and observations in light of the existing research presented in Section 3.3.

8.2.1. Working with the Pumping Lemma Scheme

In this section, I give insights into each group's process of solving the given assignments (see Section 2.2.2). As described in Section 8.1, I transferred the PL scheme to codes that relate the single steps of the scheme to activities observable in the group sessions. I name these codes *steps*:

- **Step 1**: Understanding the assignment and corresponding formal language
- **Step 2.1**: Choosing the correct word w
- **Step 2.2**: Justifying the own choice of w
- **Step 3**: Choosing a distribution xyz that has the required properties
- **Step 4.1**: Choosing a natural number k that helps in creating a contradiction
- **Step 4.2**: Inserting k in the word w and calculating the correct outcome
- **Step 4.3**: Providing the final reasoning why the language is not regular in the conclusion of the proof

The pie charts in Figures 8.1, 8.2, and 8.3 illustrate the percentage share of the total working process it took the groups to complete the corresponding PL step. Overall, it is already apparent that group A was the only group that carried out all steps in both assignments; group B was the only group ending assignment 2 after step 3; and group C did not justify their choice of w (step 2.2.) in any assignment and, therefore, completely missed a crucial formal step. Group A spent half of

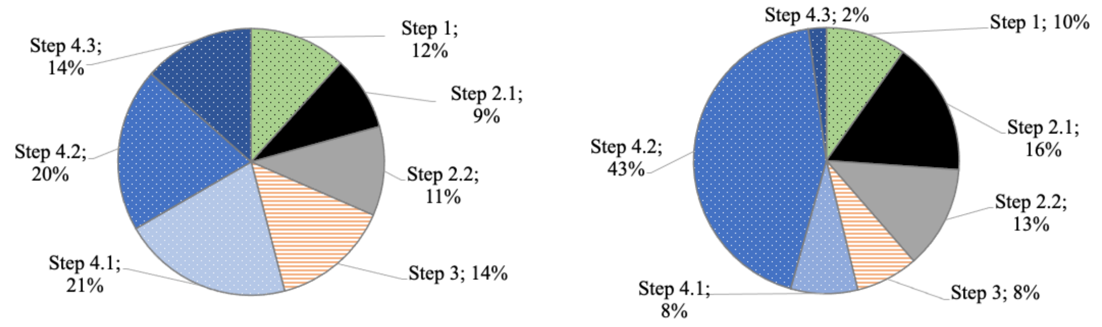

Figure 8.1.: Distribution of PL steps of group A in assignment 1 and 2.

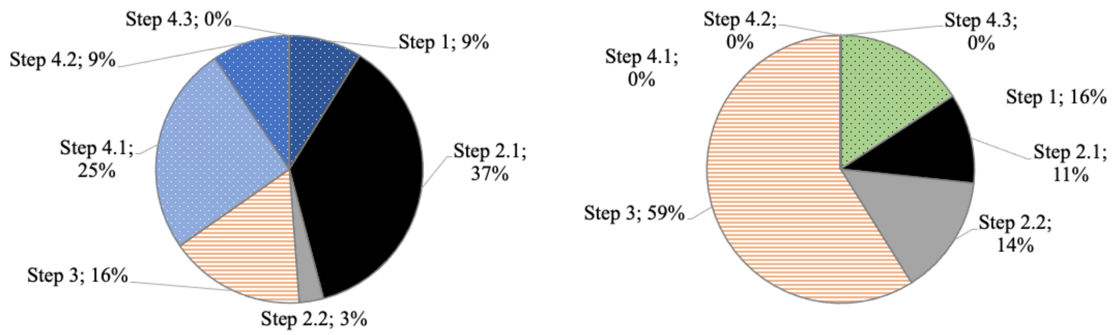

Figure 8.2.: Distribution of PL steps of group B in assignment 1 and 2.

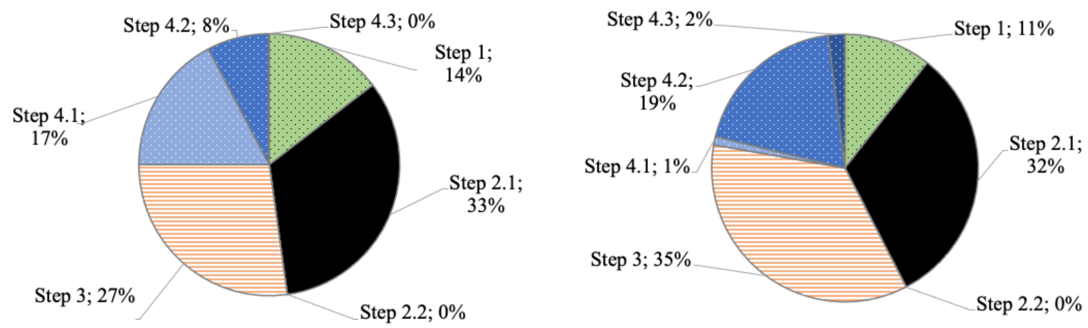

Figure 8.3.: Distribution of PL steps of group C in assignment 1 and 2.

the time with steps 4.1 to 4.3, while group B and C spent almost a quarter of time also with step 3.

Figures 8.4 to 8.9 show the line charts for each of the three groups observed. The order and time sequence in which the activities were performed is also presented. Overall, group A and B worked longer on the second than the first assignment. However, for group A and C this difference amounts to only five minutes for each of them, while group B needed 15 minutes more than they did for the first assignment.

Considering the diagrams and PL steps, I continue with a description of observable insights within the working processes and present additional information about how students assured themselves that their approach was correct:

- **Group A:** Group A followed the PL scheme closely, and the corresponding PL steps in the diagrams show this quite well. Until the students had developed their own solution for assignment 1, they did not use any materials other than A2's notes from the tutorial. At the end, they also checked whether their solution was similar to the sample solution from the tutorial. In short, they solved the first assignment without further difficulties. As for the second assignment, they spent more time on choosing a correct word (steps 2.1 and 2.2) and for the final calculation (step 4.3). As the formal language in the second assignment was more complex than that in assignment 1, this might explain the time delay. Nonetheless, they also had no major issues with assignment 2 and did not even consult the sample solution from the tutorial session again to assure themselves that their solution was correct.

- **Group B:** Group B had problems following the PL scheme in the first assignment and switched back and forth instead. They also only began to work on the assignment after a few minutes (recognizable by the empty part in the diagram 8.6, which is not assigned to any PL step). During this time, B2 explained the PL scheme (presented in Section 2.4.2) by using an example from the tutorial session. In assignment 1, the group spent more than half of their time on understanding the language and the assignment (step 1) as well as on choosing a word w (step 2.1). For that, they also consulted the course formulary and sample solutions from the tutorial session. After they had decided on an xyz (step 3), they inserted a k and tried to calculate the outcome (steps 4.1 and 4.2) but adjusted their distribution at the end. They have not attached the final justification (step 4.3) for that

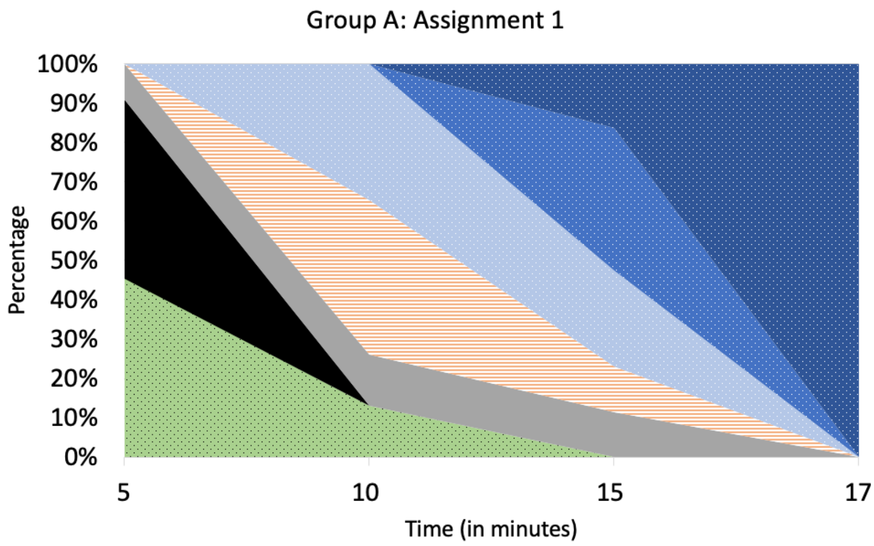

Figure 8.4.: The order and temporal sequence of occurrence of the PL steps for assignment 1 of Group A.

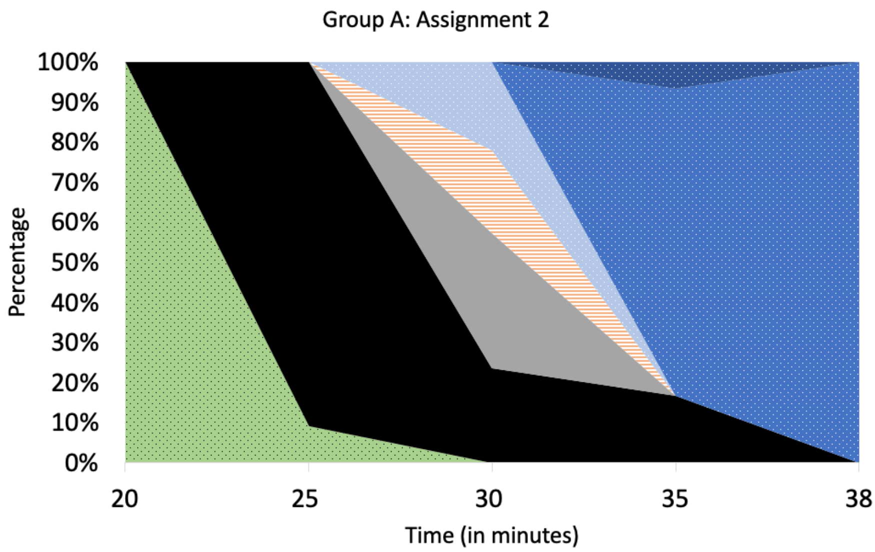

Figure 8.5.: The order and temporal sequence of occurrence of the PL steps for assignment 2 of Group A.

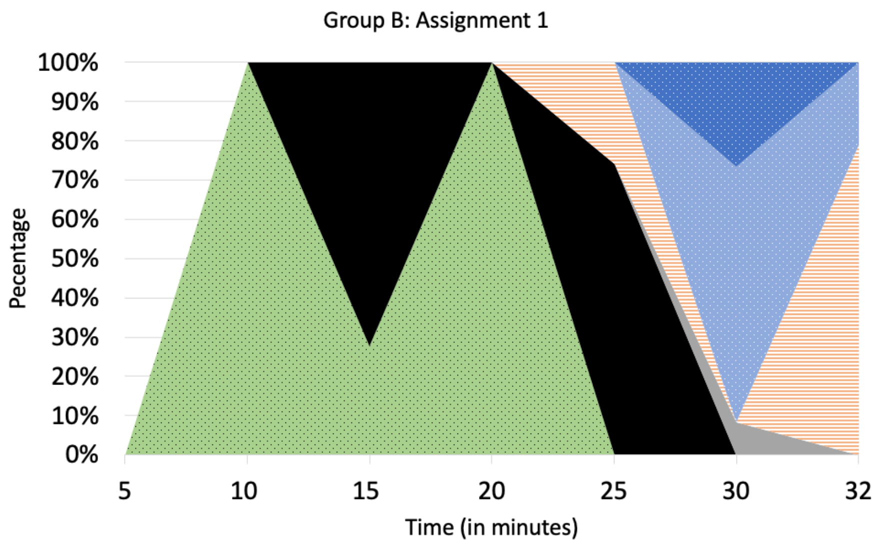

Figure 8.6.: The order and temporal sequence of occurrence of the PL steps for assignment 1 of Group B.

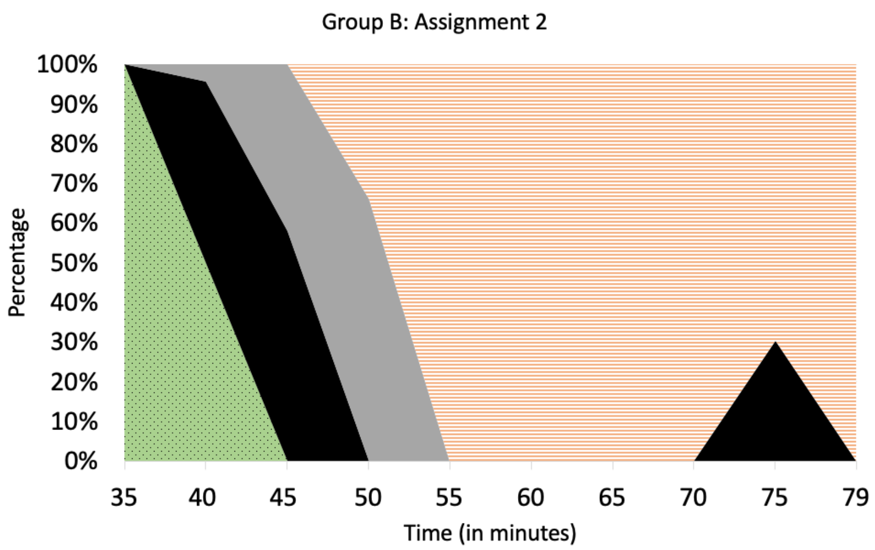

Figure 8.7.: The order and temporal sequence of occurrence of the PL steps for assignment 2 of Group B.

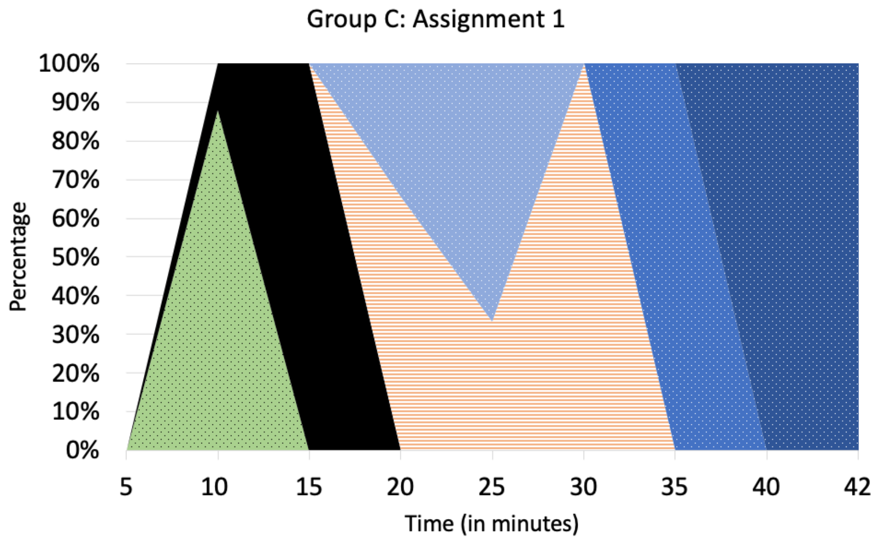

Figure 8.8.: The order and temporal sequence of occurrence of the PL steps for assignment 1 of Group C.

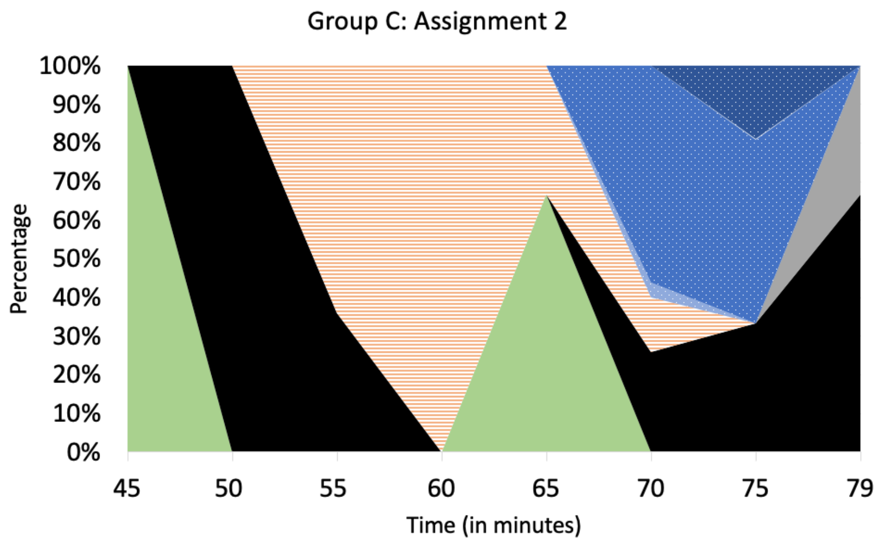

Figure 8.9.: The order and temporal sequence of occurrence of the PL steps for assignment 2 of Group C.

assignment, despite having investigated several sample solutions from the course material on a laptop. Furthermore, the group started assignment 2 in a more structured way, could follow the scheme, and spent less time trying to understand the language and choose a word *w* (steps 1 and 2). However, they could not decide on a final choice of distributing *xyz* (step 3). They tried unsuccessfully to compare their solutions with the sample solutions; but in the end, they decided that B3 would ask her teaching assistant the next day in their tutorial session.

- **Group C:** Group C mostly followed the scheme in assignment 1. They only started working on the assignments after a few minutes as they read the other assignments on the assignment sheet first. In the first PL assignment, they spent most of their time choosing a distribution for *xyz* (step 3) and deciding on an appropriate *k* (step 4.1). They missed justifying their choice of the *w* (step 2.2), although they did compare their final solution with a sample solution from the tutorial session. However, the process for assignment 2 was different: After they followed the scheme closely until step 3 (apart from 2.2, which they skipped again), they returned to the step 1 and the step 2.1. and even justified their choice of the *w* in the end (step 2.2). The discussion about the correct word continued until the end, even though a calculation with the *k* inserted had already been carried out. The reason why some steps were repeated in group C and the diagram for the second assignment looks as if they had started over is because the students in this group were working on their solutions independently on their own writing pads. Therefore, when they asked for support from their group members, they were at different steps at different times in assignment 2.

Overall, this analysis step provides an overview of the working process of each group and thus acts as the basis for all further analyses.

8.2.2. Capturing Students' Pitfalls and Challenges

Based on the seven PL steps (see Section 8.2.1), I analyzed each step for the potential pitfalls and challenges that students encountered. Table 8.1 summarizes the pitfalls and challenges I identified and which I subsume under the term "difficulty". Furthermore, the table shows the specific assignments in which different groups encountered these pitfalls and challenges.

Table 8.1.: List of student difficulties capturing their pitfalls and challenges with the PL assignments

Reason for Difficulty	Difficulty related to	Examples	Assign.1	Assign.2	Ability to overcome difficulty
Wrong understanding	Attribute dependency	Misunderstanding that x and y have to be chosen with the maximum length of n.	C	B/C	partly
		Misunderstanding that w needs to be dependent from n.	B	B/C	yes
	Formal notation	Misunderstanding that equal exponents can be represented differently.	A/B	A	yes
	Attribute choice	Misunderstanding that any word can be chosen if it lies within the language.	B/C	A/B/C	yes
		Misunderstanding that y cannot be empty.	C	B/C	partly
Missing knowledge	Correlation	Lack of knowledge that the chosen word needs to be valid for all n.	C	A/B/C	yes
	Definition	Lack of knowledge that it is sufficient to find <u>one</u> counterexample.	-	C	yes
	Attribute choice	Lack of knowledge which k is an appropriate choice.	A/B/C	C	yes
Missing ability	Formal notation	Missing ability to express/phrase basic mathematical notation.	-	A/B/C	yes
		Missing ability to read basic mathematical notation.	-	A/C	yes

Overall, all groups were challenged by incorrect or missing understanding about attribute choice, attributes dependencies, correlations, definitions, and a lack of formal notation abilities. These difficulties occurred more often in assignment 2, even if the corresponding difficulty did not occur in assignment 1. Essentially, I have found three potential reasons for student's difficulties: (1) wrong understanding, (2) missing knowledge, (3) missing ability. Furthermore, the analysis revealed the following insights:

- **Group A**: Group A was the only group challenged with representation of exponents again in assignment 2.
- **Group B:** Group B encountered a wrong understanding about the attribute dependencies, precisely between w and n in assignment 1 and x and y and n in assignment 2. Furthermore, group B was also challenged with their attribute choice in assignment 2 (precisely keeping the condition that $|y| \geq 1$). This misunderstanding prevented them from solving assignment 2 during the recorded group session.
- **Group C:** Group C was the only group challenged with wrong understanding about the attribute dependency of x and y and n already in assignment 1. Moreover, group C was also challenged by missing knowledge about attribute choices in both assignments (precisely k), but they were able to overcome this difficulty in the second assignment immediately (precisely because they remembered from their tutorial session that they should try $k = 2$ if $k = 0$ was not working).

The results indicate that the students were missing relevant understanding and abilities when working on the PL assignments. Nevertheless, only group B could not find a correct distribution of xyz that did not violate the conditions. Nonetheless, they were able to notice this and spent much time trying to solve it (see Figures 8.2 and 8.7). Furthermore, they assumed correctly at what point in their solution this issue occurred.

Otherwise, I found that the student groups were mostly able to notice, discuss, and solve these difficulties before deciding on a final solution. Therefore, no difficulty led to a wrong solution in the end. Solving most difficulties occurred because the students relied on individual group members or sample solutions and a formulary from the tutorial session that was attended beforehand. Sometimes, the students just needed assurance about their ideas and received it from another student in the group or noticed their wrong understanding for themselves while explaining.

Since some of the difficulties were due to missing knowledge, students were able to benefit from the group work as at least one person could add the missing knowledge with information gained from the tutorial session or lecture. In this way, the homework group concept ensured that the student groups could develop the general proof.

Nonetheless, even if the difficulties were recognized and solved in principle, it was not guaranteed that a student's answer or explanation was complete or correct in every respect. Although their assignment processing and solution approach was structured and scaffolded by the predefined scheme to solve PL assignments, the students' final solutions missed formal justification and reasoning of their decisions, which led to point deductions in their submission. Therefore, this kind of working process could also lead to incorrect or incomplete definitions and explanations.

Table 8.2.: Concepts by Smith and McCartney [SM14]

Concepts	Final exam errors
Quantifiers	Students made errors regarding conditions and statements with more than one quantifier.
Symbolic formulation	Students were inconsistent in their usage of symbols and applied formulations without understanding them.
Abstraction	Students were not able to abstract from the given symbols and connect the formulation to a graphical representation.

8.3. Discussion

To discuss the study results and answer the research question, I compared them with related work discussed in Section 3.3. Smith and McCartney (2014) analyzed students' written solutions to PL assignments in a final exam focusing on potential errors and relevant knowledge required for correct solutions [SM14]. Table 8.2 illustrates the main concepts Smith and McCartney identified as well as the corresponding final exam errors students made in their study [ibid., p.

1675]. Since their results are based on the written final exam assignments and my results relied on the actual working process, I compared both outcomes for further discussion. Consequently, I discuss my identified difficulties with regard to the main concepts Smith and McCartney have identified as the main reasons for students' errors:

- **Quantifiers:** I found that my observed students were uncertain about correlations indicating a lack of knowledge regarding quantifiers in the PL scheme. Furthermore, students encountered difficulties due to attribute choice and dependencies resulting in one group ending the session without a proof for assignment 2.
- **Symbolic Formulation:** I found that my observed students had difficulties understanding and expressing formal notation. Furthermore, they were not able to define every symbol correctly or understand its meaning completely.
- **Abstraction:** I found that although my observed students discussed the meaning of y when combined with the graph of an automaton, only one group really tried to use an actual visualization. Furthermore, no group was able to completely justify why it does not make sense to choose $k = 1$.

Overall, the study's student difficulties roughly correspond with the main concepts presented by Smith and McCartney. Although the related data is different, this indicates similar difficulties students encountered in both the homework and final exam assignments. However, it needs to be noted that all but one student group solved the difficulties described during the homework session by themselves. In contrast, students could not solve their errors in the final exam that Smith and McCartney analyzed. As I had previously identified PL assignments as those with the lowest performance in final exams but with a higher performance in homework assignments [FK21], I had therefore assumed that students would be able to solve their difficulties during a group session. I do not find it surprising that they did.

In my analysis process, I also noticed that no more than two people were involved in the working process, while the other students were left behind at an early stage of the working process. Due to this, they could not ask specific questions, recognize errors, or attend to the solution process more than being physically present. If the students had taken the time or tried to involve all students in the working process, they would probably have had to explain their solution approaches more clearly and understandably, leading to a more thorough understanding of the topic by all students.

8.4. Summary

As the first step for my qualitative study of student groups working on PL assignments, I collected content-related information about the working processes of the students and their difficulties with PL assignments. Therefore, I transferred the PL scheme to single steps of the scheme to activities observable in the group sessions. The students mostly went through the steps but sometimes forgot some or had to go back to previous steps. In the end, only one group did not find a solution for assignment 2 in the session. Based on the overview, I analyzed the pitfalls and challenges encountered by the students. The result is that most of the reasons for difficulties are *wrong understanding*, *missing knowledge*, and *missing ability*.

Overall, I found similar difficulties in the homework groups to those previously found by other researchers in similar solved final exam problems. However, although these difficulties were often overcome in group work, this analysis also revealed some possible weaknesses of group work. Group work can ensure that individual students do not need to understand the entire solution process if each group member contributes individual parts to the common solution. Furthermore, it may happen that not all students participate in the group work process and are left behind at an early stage of the working process. Due to this, they could not ask specific questions, recognize errors, or attend to the solution process more than being physically present.

9. Analysis of Interactions and Knowledge

Based on the previous step of the qualitative study (see Section 8), I extend the analysis to the collaborative group in terms of the students' work with each other and with their environment, answering RQ4[1] and RQ5[2]. By expanding and summarizing these findings, I provide an answer on RQ6[3]. In the following, I describe the conducted data analysis, before presenting the results. The section ends with a discussion and summary.

9.1. Data Analysis

In my research study, I made small adaptations of Video Interaction Analysis (VIA) steps (see Section 7.2.3), which resulted in the following scheme that was used within the data analysis (see Figure 9.1):

1. **Background Knowledge:** As described in Section 8.1, I had acquired context and background knowledge through intensive exchange with teaching assistants and lecturers (corresponding to step 1 of VIA).
2. **Transcription:** For the previous analysis step, I had already transcribed all three sessions (corresponding to step 4 and 6 of VIA).

[1] "What kind of interaction between students can be observed within Theory of Computation (ToC) study groups while they are working on homework assignments?"
[2] "In what way are the observed interaction forms related to the group's assignment performance as well as students' individual final exam assignment performance?"
[3] "How do students externalize individual knowledge to shared knowledge in their homework groups?"

3. **First Coding and Selection of Sequences:** I re-examined the coding presented in Section 8.2.1 and marked a few sequences that from my perspective showed unexpected situations between the students (corresponding to steps 2 and 3 of VIA). In order to assess this, I have compared such situations with my experiences and those of my colleagues and the perceptions of the students working together. The goal was to hold physical data meetings about those situations so that I can gain further understanding of VIA and narrow down how I want to further analyze the data.

4. **Physical Data Session:** During two physical data sessions with other researchers, I discussed some situations between the students as noticed in the captured data (corresponding to step 5 of VIA). While we were interpreting the selected sequences with regard to my research questions, I gained further insights into how to continue with the data analysis.

5. **Second Coding:** The overview of the Pumping Lemma (PL) steps provided information about how long the group worked on every PL step (see Section 8.2.1); however, it did not allow me to draw any conclusions about how each group member contributed to the group work. Consequently, I decided to carry out a second coding. In order to get a more objective view of the interactions, I used inductive Qualitative Content Analysis (QCA; see Section 4.4.2) with the summarizing technique as another method within my VIA. I defined one coherent statement per person as the smallest data unit. This statement could be longer than a sentence but could also be just a word or a sound. A student could have made several coherent statements in a row. I began with only one session by summarizing the coherent statements on an abstract level, differentiating a new code from the existing codes, and explicating them as categories. Once one session was coded, I coded the remaining sessions using the category system, adding new categories as needed. I also counted all the codes observed in the sessions to analyze how often an interaction occurred. The result of this data-driven coding was a category system that provided an overview of how often students interacted with each other in a particular way.

6. **Clustering:** In the next step, I clustered the identified interaction patterns into roles that I observed during the group sessions. For this purpose, I re-examined the sessions to find the sequences with recursive interaction patterns that indicate a specific role distribution in the respective session.

7. **Externalizing and Internalizing of Knowledge:** As a last summarizing step, I focused on identifying and describing the interplay between individual and shared knowledge and whether I can focus on specific individual interactions categories. Since I have explicitly chosen that I will observe PL assignments, I focused on how individual knowledge necessary to solve PL assignments was externalized and contributed to shared knowledge through specific interaction situations and structures, and how the development of shared knowledge then affected the individual knowledge gained. On this basis, I re-examined the following factors for the assessment of individual and shared knowledge:

   - To identify which individual knowledge was externalized, I paid attention to students' discussions, their oral justifications, their usage of tools and material, and their written solution during the collaborative working process.
   - For insights into what knowledge was internalized during the homework, I considered the following two sources: (1) I observed two similar assignments directly after each other, and (2) I analyzed the thematically similar assignments from the observed students' final exam.

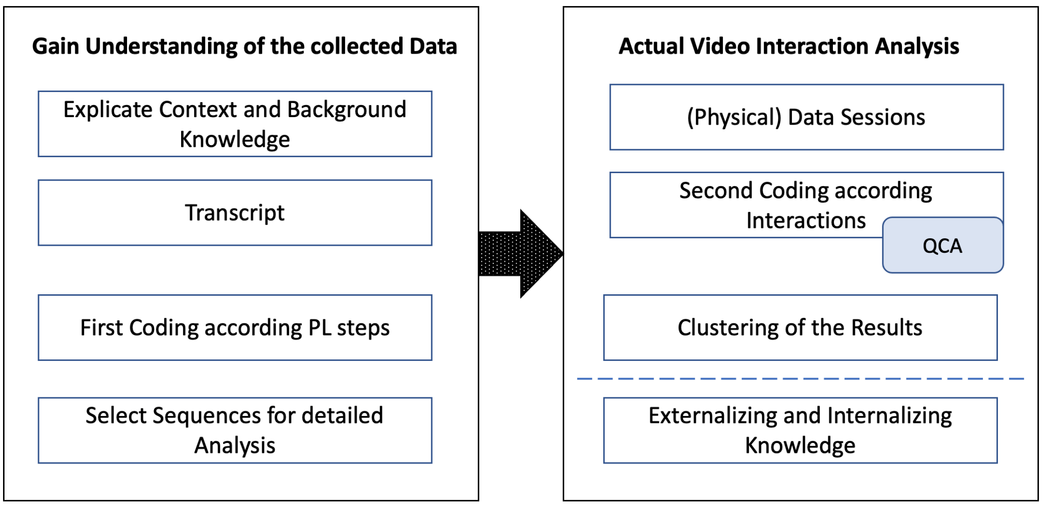

Figure 9.1.: My adaption of the Video Interaction Analysis steps.

Table 9.1.: Category system of student interactions in study groups of ToC

No.	Name & Definition	Examples
1	**Approval:** Someone approves a statement made by a group member	"yes"; "< <approving> uhm>"; "exactly"; "that's correct"; "ok"; "absolutely"; "I would also say"; "yes, x can be empty; "yes, I think so"
2	**Assumption:** Someone makes an assumption about a specific step or choice of variables	"so, let's say this is empty."; "if n would be zero"; "assuming we would use the word *abc*..."; "we probably have the same problem"; "I think it is also in the formulary"; "that means we can also simply say that we don't care at all"; "I assume that choosing one would be unclever"; "I think that makes it a little more pleasant"; "I think then j would have to be at least two"

Table 9.1.: Category system of student interactions in study groups of ToC

No.	Name & Definition	Examples
3	**Explanation / Reasoning:** Someone explains or justifies part of the assignments, the used concepts, or the solution (often followed by an approval)	"we kind of have this thing where if b is in x, it's not in y"; "because you are allowed to draw the exponents together"; "z is then c"; "because xy is less than or equal to n" "yes that is just the (predicate logic) proof actually;" "what you have to pay attention to is that you somehow choose a word (-) that is part of the language (-)"; "you have a language in which the number of zeros in the word are equal to the number of ones in the word"; "we can take n here because this is only valid for x and y"; "and you can see that a is not as big as b"
4	**Objection / Correction:** Someone objects or corrects a specific step or statement often providing some new information to support the objection or correction (often introduced with "but")	"but it must be because otherwise y would be empty"; "if you write it down like that, there has to be a plus"; "but it is actually not every word"; "we would have to take in any case what has to do with n"; "but that would also mean that you have the same number of as and the same number of bs"; "but this it actually not every word that you can form"; "I think that is a bit different"; "but we did say that must be dependent"; "but we have to show that it is part of the language"

Table 9.1.: Category system of student interactions in study groups of ToC

No.	Name & Definition	Examples
5	**Question:** Someone asks a question (usually, but not always, introduced by a question word)	"well, what could be a smart word right here?"; "similar to first-order logic?"; "so why at all?"; "that doesn't make it better, does it?"; "where does the plus one just come from?"; "how are we supposed to do that?"; "what do you mean?"; "what would happen if we choose two?"; "but does it really matter which k I choose?"
6	**Realization:** Someone becomes aware of a part of the assignment or solution approach that he/she did not know, that was not clear to him/her before, or that he/she understood differently before	"oh, yeah, it's true, otherwise it doesn't make much sense"; "that somehow does not quite look like it's going to work"; "oh right we did that yesterday in the tutorial session, too"; "ah that must be the same"; "oh yes logical"; "right! you are right"; "oh right, we have to make it dependent on n"; "ah ok well I did not know that"; "oh because you summarized it right above"

Table 9.1.: Category system of student interactions in study groups of ToC

No.	Name & Definition	Examples
7	**Rejection:** Someone rejects a statement (often followed by an objection/correction)	"no"; < <denying> uhm>; "no that makes no sense";
8	**Request:** Someone is requested by a group member to do something	"just a moment ..."; "take a look"; "take care"; "insert two in k"; "show me your calculation"; "finish the calculation please"
9	**Suggestion:** Someone makes a suggestion to the group	"or we could just write it down really quickly"; "well, we can leave it blank for now"; "let's try to find out first"; "or we could consult the tutorial solution"; "well I would say that we just say m is one"; "then we can try it out for $k = 2$;" "let's calculate"; "let's do this without b"

9.2. Results

The presentation of the analysis results is organized as follows:

1. First, I present the developed category system of student interaction.
2. Second, I provide information about the interactions that took place. In doing so, I also offer a deeper insight into the sessions with examples from the transcripts of all three groups.
3. Third, I present how I clustered the interaction patterns into roles.
4. Fourth, I present possible relations between student roles and final exam performances.
5. Fifth, I summarize how the observed students externalize individual knowledge and externalize shared knowledge.

9.2.1. The Category System

I have found nine forms of interaction capturing students' working sessions in the study groups. Table 9.1 presents the names and definitions of the categories and provides examples from the transcripts.

9.2.2. The Interactions

The presentation of the interactions is organized as follows:

1. First, I present the frequency of the observed student interactions in the study groups. I start by describing the interactions among all groups and then go into more detail regarding the interactions of individual group members.
2. Second, I go into more detail about the group sessions and illustrate identified recurring interaction patterns with exemplary excerpts from the transcripts

Occurrence of Student Interactions

Table 9.2 gives the data for Figures 9.2 and 9.3, showing the interaction occurrences in all groups and in each of the assignments. Here, each group showed a comparable number of frequencies in which each interaction occurred. Furthermore, all groups used most of their time for *explanation* and *approval*. This indicates what I could observe in each session: All groups established a working process in which at least one person was *explaining* (e.g., topics, concepts, or approaches to solutions), and at least one person was *approving* statements made by the others.

Table 9.2.: Occurrences of interaction categories in all groups

Category	Gr. A		Gr. B		Gr. C	
	Ass.1	Ass.2	Ass.1	Ass.2	Ass.1	Ass.2
Approval	61	84	94	141	49	68
Assumption	7	10	3	13	12	7
Explanation	102	92	113	207	82	62
Objection	10	14	36	50	83	42
Question	25	28	36	71	55	31
Realization	26	37	32	47	19	13
Rejection	10	12	5	32	23	21
Request	2	4	3	17	9	17
Suggestion	15	19	16	27	14	11
Sum	258	300	338	605	290	272

In order to achieve a more detailed overview of the individual interaction frequency, I broke down the relationship between interactions and session time down to the level of each group member and assignment. Tables 9.3 to 9.5 present the number of occurrences for every group divided by group members, while Figures 9.4 to 9.9 visualize the data. In the following, I summarize the findings divided by groups:

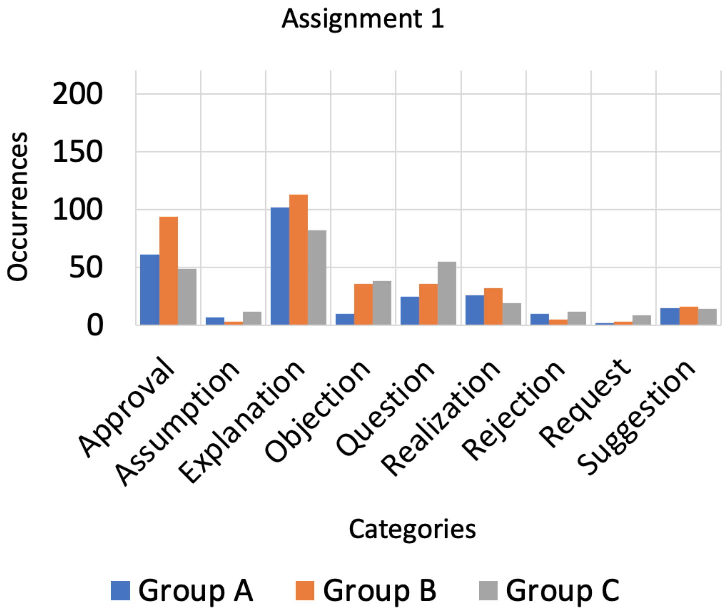

Figure 9.2.: Occurrences of interaction categories in all groups in assignment 1.

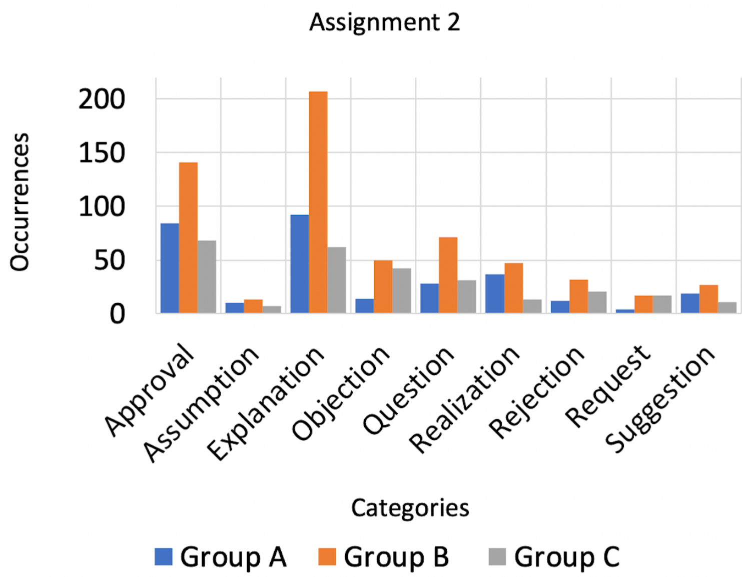

Figure 9.3.: Occurrences of interaction categories in all groups in assignment 2.

Table 9.3.: Occurrences of interaction categories for group A divided by members

Category	A1		A2	
	Ass.1	Ass.2	Ass.1	Ass.2
Approval	25	39	36	45
Assumption	3	5	4	5
Explanation	45	30	57	62
Objection	5	4	5	10
Question	20	24	5	4
Realization	25	34	1	3
Rejection	1	5	9	7
Request	1	3	1	1
Suggestion	8	3	7	14
Sum	133	149	125	151

Table 9.4.: Occurrences of interaction categories for group B divided by members

Category	B1		B2		B3	
	Ass.1	Ass.2	Ass.1	Ass.2	Ass.1	Ass.2
Approval	3	3	38	67	53	71
Assumption	0	0	0	11	3	2
Explanation	2	4	87	138	24	65
Objection	0	0	18	28	18	22
Question	3	3	1	20	32	48
Realization	1	0	2	13	29	34
Rejection	0	0	5	20	0	12
Request	0	1	0	6	3	10
Suggestion	1	1	7	12	8	14
Sum	10	12	158	315	170	278

Table 9.5.: Occurrences of interaction categories for group C divided by members

Category	C1		C2		C3		C4	
	Ass.1	Ass.2	Ass.1	Ass.2	Ass.1	Ass.2	Ass.1	Ass.2
Approval	18	30	10	9	21	29	0	0
Assumption	7	4	3	1	2	2	0	0
Explanation	33	32	19	7	28	21	2	2
Objection	23	18	1	5	13	19	1	0
Question	31	13	7	5	13	13	0	0
Realization	12	6	2	2	5	5	0	0
Rejection	8	10	1	3	3	7	0	1
Request	7	16	1	0	1	1	0	0
Suggestion	9	6	0	1	5	4	0	0
Sum	148	135	44	33	91	101	7	3

- **Group A:** Table 9.3 and Figures 9.4 and 9.5 present the interaction occurrences for students A1 and A2. Altogether, both students had a similar number of overall interactions. However, the total number of interactions is divided into different categories, for example, in the assignment 1, both students took the most time for *explanations* and *approvals*, even though A1 had far more *realizations* and *questions* than A2. The distribution of interaction occurrences did not become more balanced in assignment 2, resulting in A2 *explaining* even more and A1 having even more *questions* and *realizations*. In essence, this division and development leads me to assume that A2 was the driving force behind the development of the solution, while A1 took a more passive stance, at least in the assignment 2.
- **Group B:** Table 9.4 and Figures 9.6 and 9.7 present the interaction occurrences for students B1, B2, and B3. B2 and B3 share approximately the same number of interactions in assignment 1; in assignment 2, B2 has a larger number of interactions than B3. During the whole session, B1 had almost no interactions. In assignment 1, B2 gave the most *explanations*, while B3 asked the most *questions* and had the most *realizations*. In assignment 2, B2 and B3 interacted more with each other and even had a slightly more balanced interaction distribution than in assignment 1. Looking at the division and development, it is certain that B2 was the main person responsible for the solution development; however, B3 was more involved in the actual solution development in assignment 2.

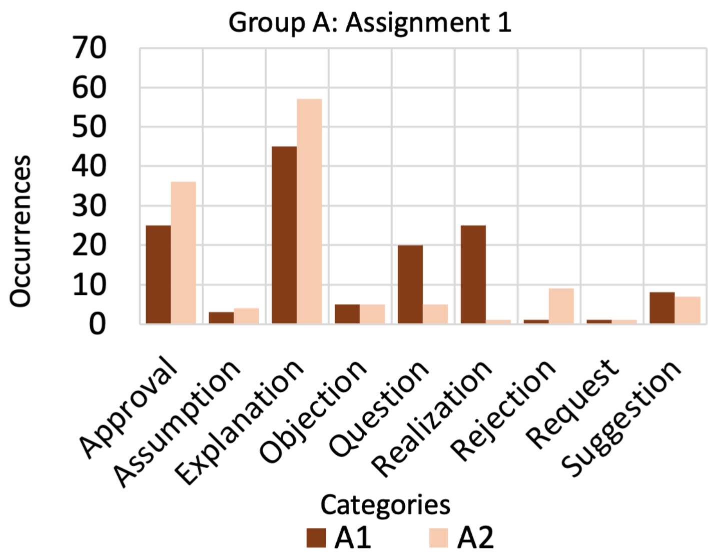

Figure 9.4.: Occurrences of interaction categories in group A in assignment 1 divided by group members.

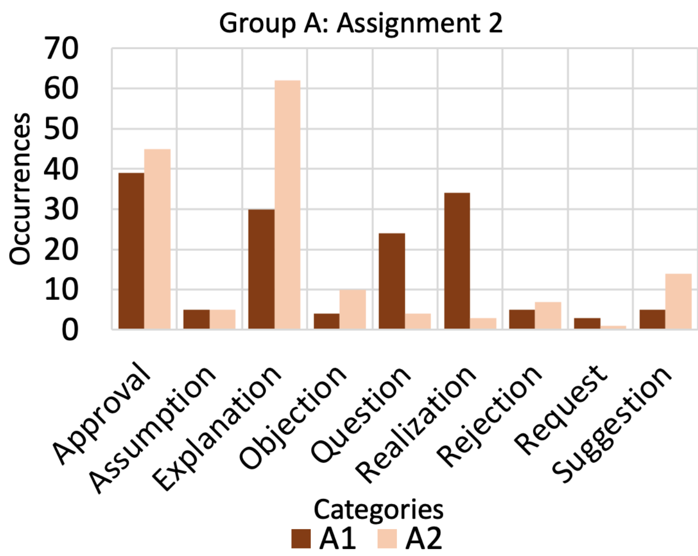

Figure 9.5.: Occurrences of interaction categories in group A in assignment 2 divided by group members.

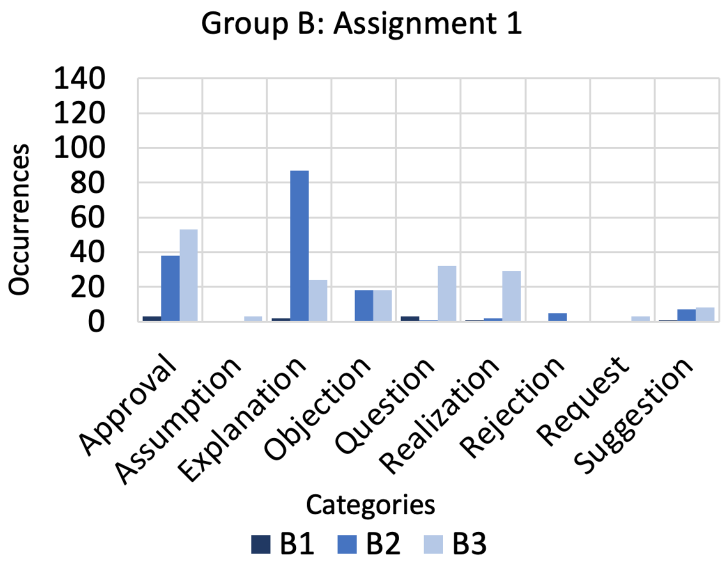

Figure 9.6.: Occurrences of interaction categories in group B in assignment 1 divided by group members.

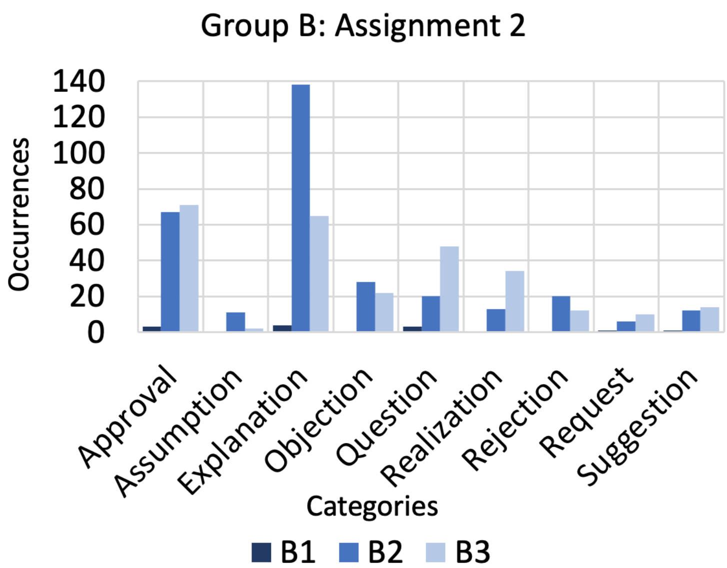

Figure 9.7.: Occurrences of interaction categories in group B in assignment 2 divided by group members.

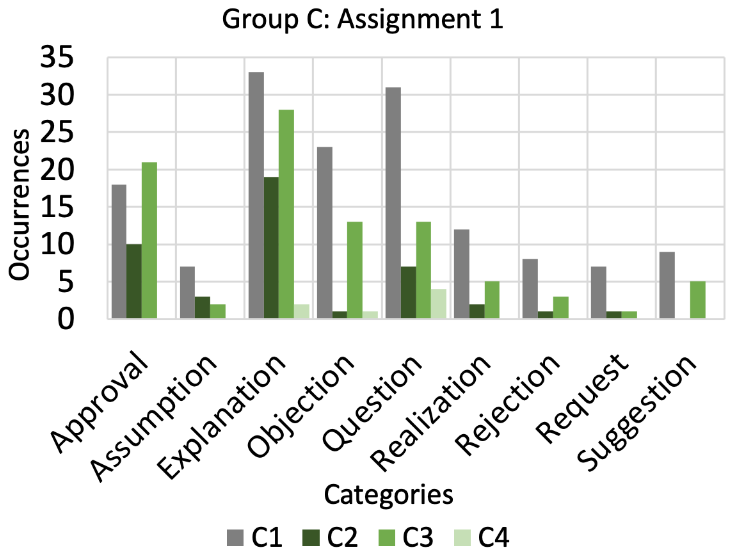

Figure 9.8.: Occurrences of interaction categories in group C in assignment 1 divided by group members.

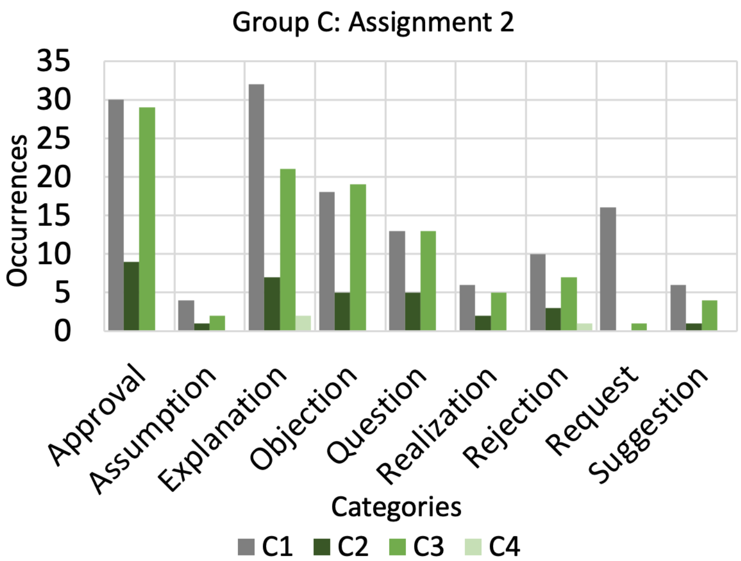

Figure 9.9.: Occurrences of interaction categories in group C in assignment 2 divided by group members.

- **Group C:** Table 9.5 and Figures 9.8 and 9.9 present the interaction occurrences for students C1, C2, C3, and C4. C1 had the highest number of interactions in assignment 1 and assignment 2. Overall, C2 had a low participation, while C4's contribution was almost non-existent. In assignment 1, C1 had the most interactions of *explanations* and *questions*, whereas C3 had a high number of *explanations* and *approvals*. In assignment 2, C1 had a similar number of *explanations*, but fewer *questions* and more *approvals*; C3 had far more *approvals* and a few more *objections*. Nevertheless, it is safe to say that while C3 was the person responsible for the solution development in assignment 1, C1 also offered support. In assignment 2, both students seem to be equally active in developing the solution.

Analysis of the Interactions

After giving an overview of the interaction occurrences within the group sessions, I now present the occurring interactions in more detail. In this way, I elaborate on how interactions between members in groups and the available material are shaped and whether patterns can be found. Therefore, I describe several typical interactions for each group which are then illustrated with transcript excerpts. During the description, I specifically refer to the PL steps (see Section 8.2.1) to connect the working process with the interaction sequences.

I have adjusted the excerpt transcripts (see Section 7.2.4) to make it easier to read – that is, I added punctuation marks and upper and lower case. Additional information about how something is said is still shown within << and >>, while pauses are shown with (-), and additional information needed for understanding or the formally correct writing of a verbal explanation is written in [ ]. I continue to italicize letters, symbols, and numbers to distinguish them from the surrounding text, and I also italicize any actions of the students within round brackets and behind a vocal statement.

Interaction in Group A

Assignment 1. The session started with A2 writing down the PL Scheme (see Section 2.2.2) on her sheet of paper. Accordingly, she left blanks within the scheme to be filled in by the group with the chosen variables relevant for the proof. Focusing on the interaction between A1 and A2, the excerpt in Table 9.6 shows how A1 wanted to discuss the whole scheme first, while A2 already wanted to put the scheme into practice. The table also exemplifies how A1 asked *questions* and relied on A2 and her *explanations*.

Table 9.6.: Excerpt 1 from Group A at [00:03:04]

Student	Content
A1	So (-) uhm (-) I know parts of the procedure.
A2	I think it is also written down in the formulary. Ok, so first the resolution (-) here. *(A2 points to the tutorial notes)*
	So, [n is] arbitrary but fixed. *(A2 writes and A1 reads what is being written)*
A2	< <agreeing> uhm> Yes, then we have to choose a word.
A1	< <agreeing> uhm> (-) from the language? I mean L_1.
A2	Yes. Do we already want to choose the word because that's what it all comes down to?
A1	What does it come down to?
A2	Well, we have to justify it [the word] here and we have to choose k and then -. *(A2 points to the tutorial notes)*
A1	Well, let's leave it blank for now. (-) How should we continue?
A2	Well, then we choose k (-).

While working on the assignment, A2 provided A1 with necessary information from the tutorial session, even though both had attended it. The excerpt in Table 9.7 exemplifies further the dynamic between the two students and shows how A1 recalled a hint from his teaching assistant about what to choose as k (PL step 4.1) only after A2 addressed it.

Table 9.7.: Excerpt 2 from Group A at [00:07:34]

Student	Content
A2	Ok, now we have to choose k.
A1	< <approving> mmh> This is actually the difficult part, isn't it?
A2	Well, we should always take 2 or 0 [for k].
A1	Oh, yeah right that was a hint from him [teaching assistant].

While working on assignment 1, A2 wrote down the solution the group developed together. During this process, A1's questions often referred to something A2 had previously said or written down. The excerpt in Table 9.8 illustrates this further during the PL step 4.1.

Table 9.8.: Excerpt 3 from Group A at [00:12:30]

Student	Content
A1	So wait (- - -) so I just got lost between the word and the definition. *(A1 points from A2's solution to the laptop)*
A2	Uhm ok so (-) where exactly?
A1	I understand that we have to say that the word has to be in this language. I think it would be better if we just continue with k.
A2	Ok.
A1	Let's just choose k and calculate it because I don't understand it yet. I guess that would be the next step.
A2	Actually, it doesn't matter what we choose as k.
A1	Oh ok <laughing>. So, we can just choose 2.

A1 had no more questions about the solution that A2 had largely developed on her own. In the end, they compared their solution with the sample solution from the tutorial. Hereby, A1's statement suggests that he had more confidence in a comparison with the tutorial solution as a decision for the correctness of the solution more than his own understanding or in assuming that A2 got it right on her own (Table 9.9). After assignment 1, the students started directly with the second assignment.

Table 9.9.: Excerpt 4 from Group A at [00:16:53]

Student	Content
A1	Choose k (-) and now, we need the last sentence.
A2	Yes.
A1	Yes otherwise the proof is (.) I think right (.) then we have it just like in the tutorial session.

Assignment 2. Table 9.10 shows an excerpt from the beginning of working on the second assignment. It shows how A1 and A2 reacted differently to the language of assignment 2. A1 was overwhelmed, while A2 suggested that they first understand the language. This reaction also shows that he is not necessarily used to the formal way of writing and that it intimidated him in a certain way.

Table 9.10.: Excerpt 5 from Group A at [00:17:13]

Student	Content
A1	Yeah, [following the PL scheme and developing a PL proof] this is totally doable. (- -) and now, the second assignment *(both students start to read assignment 2).* Okay, I shouldn't have said that this [the PL proof] is doable. <laughing>
A2	So, we have *b* -. *(writes the language on a sheet of paper)*
A1	< <slightly shocked> My god>. *(both of them continue reading the assignment on the laptop and A1 is overwhelmed by the notation and conditions)*
A2	Let's first try to find out what -.
A1	- what this means?
A2	Yes.
A1	< <laughing> good idea.>

The dynamic between both students continued to be the same as during the first assignment, even as additional problems with the formal notation emerged in the course of the task. Once, A1 was asked to write his idea down (during the PL step 1). This happened because A2 directly asked for it as A1 realized that he could not put his idea into words (Table 9.11). However, A1 was clearly aware of his lack of knowledge in formal writing.

Table 9.11.: Excerpt 6 from Group A at [00:23:39]

Student	Content
A2	What do you think is the best way to write down this justification that our chosen word is in the language?

A1	Good question <laughing> uhm. (-) So, here we just said that -. (-) *(A1 points to the solution of assignment 1)*. Uhm then, we just say that x is element of (.) a and b to the power of n is - (-) and that is also element of a to the power of - so any number (- -) I cannot put this into words.
A2	Can you write it down?
A1	< <laughing> Uhm> I actually meant (-) x is an element of - I'm not quite sure if you write it like this - a and b to the power of n [$\{a,b\}^{\mathrm{n}}$] and that's just an element of (-) a and b to the power of any number [$\{a,b\}^*$] (.) so that you can say that the two things are the same *(In total, A1 wrote $x \in \{a,b\}^n \in \{a,b\}^*$ on the tutorial notes, while looking at the laptop)*
A2	But this is not directly in line with our assignment because here they can also put a before b and we want b *(A2 points to A1's solution and what they wrote down until that point)*

At the end of the session, A1 could no longer follow A2's calculations in PL steps 4.1 and 4.2. and left it to A2 to solve the problem alone without further questions. The sequence in Table 9.12 took place at the end of the session and shows A1's uncertainty after A2 was able to calculate a part of the result in her head. When A2 tried to explain to him how she came up with it, he waved her off, saying that she should finish her calculations first and then he would look at it.

Table 9.12.: Excerpt 7 from Group A at [00:28:40]

Student	Content
A2	I think this time it would be better if we really choose these 2 [as k] because if I think we have to choose a 0 here (-) then (-) yes, then the y completely disappears. *(A2 looks at her solution)*
A1	Yeah, and we don't want that.
A2	Yes, and then it is also true that the word is still in the language.
A1	You can already see that? < <laughing> holy...> ok
A2	Well, because then that's just the n. *(A2 points to her solution)*
A1	I'll just trust you. We'll take 2, and then we'll calculate, and then I'll for sure see that it fits.
A2	Well, imagine we are removing a lot of *b*s right now.
A1	< <approving> Uhm.>

A2	Then we take away a few less *b*s than *as* then it remains (-) uhm what is bigger here (-) uhm becomes even bigger than *m*. [...]
A1	No, everything is fine, so we can just calculate it first, and then I'll see for sure.

To sum up, A2 was the driving force behind the development of both solutions as indicated by her frequency of interactions (especially *explanations*) within the group (see Figures 9.4 and 9.5). Furthermore, A2 explained to A1 the PL and the developed solutions as well as the necessary concepts, while A1 had a more passive role by repeating *explanations* or asking follow-up *questions*. This kind of role distribution remained the same during the entire session.

Interaction in Group B

Assignment 1. The session started with B3 requesting B2 to explain the whole PL scheme (see Section 2.2.2) as B3 had not yet attended the tutorial session. In order to explain the PL scheme, B2 used an assignment from the tutorial session. Table 9.13 shows how B2 and B3 dealt with B1 being five minutes late to the session after they had already started discussing the PL scheme. It can be seen how B1 was not properly integrated into the group process even at the beginning, but the excerpt also shows that his group members already expected that he wasn't in the tutorial session anyway.

Table 9.13.: Excerpt 8 from Group B at [00:02:48]

Student	Content
B3	I assume you have not been in the tutorial session this week.
B1	Nope.
B3	Ok, then this will be fun (-) uhm (- - -) yes, how should we do it then? I would say you [B2] continue explaining and then I will explain to you [B1] the beginning or so. <laughing>

However, B3 did not explain the beginning of the scheme, but rather started working on assignment 1 with B2. Despite also not attending the tutorial session, B3 came prepared to the session. The excerpt in Table 9.14 shows that he knew that they could get the missing information from the formulary while trying to understand the language (PL step 1).

Table 9.14.: Excerpt 9 from Group B at [00:14:54]

Student	Content
B2	You can combine a to the power of *m* [a^{m}] and *a* to the power of *l* [a^{l}]
B3	Exactly, but that is only possible with the concatenation.
B2	Yeah.
B3	We also had something [about concatenation] in the formulary, I think (-) *(everyone looks at the formulary on their laptops or smartphones).*

When they were not sure whether they could apply the concatenation from the formulary in their case, they also mentioned a possible point deduction during their tutor's evaluation of their solution. At this point, B3 suggested that they should look into the tutorial assignments and solutions to see if there was a similar assignment that could help them answer their questions (Table 9.15).

Table 9.15.: Excerpt 10 from Group B at [00:17:30]

Student	Content
B2	I think that will lead to a point deduction again <laughing>
B3	Probably.
B2	But I don't know. I don't need every point.
B3	What we could do (-) Uhm we could look into the tutorial assignments again to see whether there was a similar case.
B2	< <approving> uhm>
B1	The only one is assignment 1g.)
B3	1g?
B1	Yes.

When using the tutorial assignment as an example did not help them either, it was again B3 who suggested that he could explicitly ask the tutor again tomorrow (Table 9.16).

Table 9.16.: Excerpt 11 from Group B at [00:19:36]

Student	Content
B3	I think that's a bit different (-) I mean what we could do - uhm I'm still planning to attend the tutorial session tomorrow. We can just start this and I could ask the tutor I'm with tomorrow if you have to prove this again somehow.
B2	Yes.

During all of assignment 1, B2 guided the solution development by writing down the solution step by step and giving explanations at the same time. B3 had suggested not only that they should use the available course documents, but he also asked a lot of questions during the working process to fully understand how the PL scheme works. Table 9.17 illustrates the situation during the PL step 3.

Table 9.17.: Excerpt 12 from Group B at [00:26:46]

Student	Content
B3	So, in this case what would *xy* be and what would *z* be?
B2	Well, *y* would be either (-) *a* or (-) uhm.
B3	Or nothing.
B2	Or nothing. *y* would be (-) it can only be *a*.
B3	<agreeing sound>
B2	Yes, *y* can only be *a* because *xy* has to be less than or equal to *n* [$\|xy\| \leq n$]. *n* would be 1 so *x* would have to be lambda and *z* would then just be *cb*.
B3	And for *z*, there are no restrictions?
B2	No, *z* is actually just the rest.
B3	Actually, this is how you do it? First, you think about what would *y* be and the one before that is *x* and the one after that is *z*.
B2	Basically yes.

Most of the time B3 only wrote down short notes to clarify his questions; however, he also wrote down a whole calculation once. During PL step 4.1, he wanted to understand whether they could also choose 0 as *k* instead of 2 and what the result of the calculation looked like (Table 9.18). Unlike B2, he could not calculate and see the result in his head.

Table 9.18.: Excerpt 13 from Group B at [00:28:46]

Student	Content
B3	Just a question. (-) If we had taken 0, it would say that we had *a* to the power of *i* [a^{i}]. That means in the end, it would say *a* to the power of *i* and *a* to the power of *n* minus *i* minus *j* [$a^{i}a^{n-i-j}$]?
B2	So, *a* to the power of *n* minus *j* [a^{n-j}]. Then the plus *j* simply becomes minus *j*. And that is also not part of the language.
B3	Just one moment, I have to write it down at my age and cannot do it in my head.

Instead of explaining the PL scheme to B1 in the beginning as promised in Table 9.13, B2 and B3 turned to him only after they had decided on the final solution. The excerpt in Table 9.19 shows how B2 and B3 answered B1's questions together. However, B3 could only reproduce the *explanations* B2 gave him during the assignment; when it came to giving a more detailed explanation, B3 had to rely on B2 again.

Table 9.19.: Excerpt 14 from Group B at [00:29:36]

Student	Content
B3	Ok (-) do you have any idea what we just did? *(addressing B1)*
B1	I didn't quite understand the part with this *k*.
B3	Well, that's why this is called pumping. So, in the sense of (- -). I honestly don't know how he [the lecturer] explained it in the lecture *(B3 turns to B2)*
B1	You are allowed to choose *k* yourself?
B3	Yes.
B2	If you consider it as an automaton, it is somehow a loop.
B3	< <agreeing> uhm> So, you can choose the *k* by yourself and we chose in this case 2.
B1	Ok.
B3	With 0 it would have been a little less writing effort (-) but now there is a word (-) in this case it is *a* to the power of *n* plus *j* [a^{n+j}](-) *cb* to the power of *n* [cb^{n}](-) and we chose this word *(B3 points to the solution)*

B1	Yes, up until this point I understood it, but with the k I had to think about it for a moment, whether you choose it yourself.

After B1's question was answered, he began to copy paste B2's solution, while B2 and B3 started on the assignment 2.

Assignment 2. B2 and B3 continued their dialogue in assignment 2, but this time B3 tried to be more active in developing the solution together with B2. He still needed corrections and support from B2 several times. For example, the excerpt in Table 9.20 shows how B3 had problems with the formal notation of regular expressions during PL step 2.1. and how he was corrected by B2.

Table 9.20.: Excerpt 15 from Group B at [00:33:53]

Student	Content
B3	This should be a comma here. *(B3 wrote $b(a,b)^*c^m$)*
B2	Uhm.
B3	You can also use a comma, can't you?
B2	No, for a regular expression it must be a plus.
B3	Wait a moment. Can you explan this to me?
B2	Well, this is a set (-). There are elements in it (-) This means you can select an element from x and write it down as often as you like. So, you can choose as often as you like from x.
B3	<agreeing sound>
B2	But if you write a comma in a regular expression (-) then that would mean that you write *ab* and you can write it as many times as you want. But only *ab ab ab*.
B3	Oh ok.
B2	So if you put a plus in between, it means you can write a or b as many times as you want.
B3	Ok. It has to be a plus then. *(B3 wrote $b(a+b)^*c^m)$)*

After both students understood during PL step 2.1. how many *a*s and *b*s are possible in their word, B3 called himself a "child" indicating a hierarchy in their working session (Table 9.21).

Table 9.21.: Excerpt 16 from Group B at [00:35:59]

Student	Content
B2	The order [of *a* and *b*] is random.
B3	It could also be *baabba*?
B2	Yes. It can be like every possible combination of *a*s and *b*s.
B3	Right, but that also means that we cannot have more *b*s than as in any case.
B2	Yes. <both are laughing>
B3	Yeah, ok, that just sounded like a child figured out what three plus four is. <laughs>

As the following extract shows, B3 stuck to the hierarchy as in the previous excerpt and even asked B2 for permission to try to distribute *xyz* completely on his own (PL step 3). Nonetheless, B2 immediately provided help as B3 did not seem sure (Table 9.22).

Table 9.22.: Excerpt 17 from Group B at [00:48:38]

Student	Content
B3	Wait a minute, can I try this [distributing *w* into *xyz*]? *(addressing B2)*
B2	Sure *(B2 leans back)*
B3	Uhm.
B2	I have a concrete idea of how to do it.
B3	Ok, if my idea is not the same as your idea, then -
B2	No, you can certainly do it in many ways.
B3	Uhm ok. Anyways (- - -)
B2	It doesn't differ much from the first assignment except that it has one more *a* at the beginning and -. I mean a *b*.
B3	So basically, you just write a *b* at the beginning and then how many as you want.
B2	Yes, and then, we are already out of the area of *x* and *y*.

As for B1's interactions in the assignment 2, he was once again unsuccessful in the attempt to get back into the discussion. The following excerpt shows how at the moment B1 asked a question regarding the distribution of *xyz* (PL step 3), B2 and B3 were still in the middle of a discussion and did not involve B1 in their ongoing conversation. However, at this point, both B2 and B3 knew what their problems were and reacted to B1's question simultaneously. This actually demonstrates that B2 and B3 also worked together as equals on some occasions (Table 9.23).

Table 9.23.: Excerpt 18 from Group B at [00:55:31]

Student	Content
B1	Didn't you say now that l can never be 0 or - (-)?
B2/B3	Not quite yet.
B2	We are still thinking about how to do it.
B3	The problem is just - (-).
B2	Because l can be 0 if x is lambda and we have a b in y. So, if i is 0 - (-).
B3	The problem is that y can never be 0 [$\|y\| \geq 1$] and if you allow that l can be 0 [$l = 0$] (.) then you have to guarantee at the same time that if l is 0 that you do not put i equal to 1 [$i = 1$] by any means.
B1	Then (-) uhm if a is equal to i plus l [i+l] (-) then it would never be 0, that way we would always have something in z.
B3	Yes, but if z is always only the residue, so z can also be empty (-) x can be empty, z can be empty, but there must always be something in y.
B1	I see.

Despite B2 and B3 working as equals on certain occasions in assignment 2, there were still situations when B3 could not always follow B2's explanations. The next excerpt shows an example of B3 not being able to follow B2's idea of distributing xyz anymore (PL step 3).

Table 9.24.: Excerpt 19 from Group B at [01:02:46]

Student	Content
B3	So, the way I understood it is that it should be valid for every j and for every l and every i.
B2	That's right! That's right! Because if we would just say j plus l is less than or equal to n [$j + l \leq 1$], then we would forget the b in the beginning.

B3	<agreeing sound>
B2	This means we have to say j plus - uhm l is one less than n (-) so, we can add this b and we do that by just saying j plus l plus 1 is less than n [$j+l+1 \leq n$] but - .
B3	Ok, I just believe you now <laughing> you can tell me -. (-) I will give it a look again once we have finished this. (-)
B2	Yes, I think that is right.
B3	I believe you that it is smaller than n.

To sum up, I find again one person (B2) to be the driving force in developing the solution while another group member (B3) was supporting it. In fact, B3 was not only occupying a supporting role but also relied on B2 as a sort of "personal tutor" in order to get the PL scheme explained at his pace during solving the first assignment. This resulted in B2 explaining both the whole PL scheme as well as his solution ideas. In the second assignment, B3 was actually more active in helping B2 develop the solution.

Interaction in Group C

Assignment 1. The session started with C1 encouraging C3 to lead the group work, as she explicitly stated that C3 was capable of developing proofs (Table 9.25).

Table 9.25.: Excerpt 20 from Group C at [00:03:02]

Student	Content
C1	Oh, that is this strange proving. C3, you know how to do it! <laughing> (- -) Oh well, you are the only one of us that can do it.

C1's assumption that C3 understands proof assignments better than she did marked the beginning of a dialogue between the two. The dialogue occurred whenever C1 wanted to reassure herself of her ideas, while building her solution using a sample solution from the tutorial session as a template. The excerpt in Table 9.26 shows the dynamic between C1 and C3 while working on PL step 2.1. In this situation C1 had an initial idea and started a dialogue with C3; however, C3 objected to her idea. Nevertheless, C1 could not initially understand C3's explanation and revisited it with the help of a tutorial assignment. Although C2 tried to help her she did not actually engage with him.

Table 9.26.: Excerpt 21 from Group C at [00:10:07]

Student	Content
C1	We now have to add this - this at the beginning of the curly brackets [C1 means $a^{\mathrm{m}}a^{\mathrm{l}}cb^{\mathrm{m+l}}$] in *w*.
C3	Uhm but (-) only if it is dependent of *n*.
C1	No, how? (-) You cannot take two letters and then say it should only be dependent on one.
C3	But it can only be dependent on *n*.
C1	But there are TWO letters
C3	<laughing>
C1	Why is it only dependent on *n*? Am I not able to put there a second letter? Why not?
C3	<laughing> because it does not work (- - -)
C1	Why not?
C3	As I said before: it must be dependent on *n*. Because you (-) you check if it is greater than or equal to *n* [$\geq n$] (-) Therefore, you can always compare it with *n* and if - oh, and *n* can be arbitrary - then, it grows with *n*. But if you choose some constant that has nothing to do with *n*, then you can choose a very big *n* and it has no impact. (-) so it must be somehow dependent on *n*
C1	Uhm but we have two letters.
C3	Yes.
C2	So, we have *a, b,* and *c* but *cb* doesn't matter? Because we have *w* -
C1	No, I am talking about the *m* and the *l*.
C2	Ok.
C1	Maybe, we could only take *n* in the tutorial assignment because only *n* was in the formula *(C3 flips through her notes)*.
C2	It said *w* equals 0 -
C1	That's not an *n* at all.
C2	But *n* means (-) what does *n* means again?
C1	I have no idea.

C2	So, *n* says the number?
C1	Ok, I choose the word 0 to the power of *n* [0^n] and 1 to the power of *n* [1^n] and that is actually just a lot of zeros and behind that a lot of ones.
C4	Yes.
C1	But actually not every word that you can form with that language. This means we can also say that we don't care at all and just add (-) *n*, *m*, and *l* equal to n and that's just how it is?
C3	Yes.

Another example of how C3 explained the content to C1 can be seen in Table 9.27. Nonetheless, C3 explained the meaning superficially here as well, focusing on how to choose *k* during the PL step 4.1 instead of understanding *k*.

Table 9.27.: Excerpt 22 from group C at [00:15:05]

Student	Content
C1	Why should I chose *k* now? (.) What is *k* again?
C3	*k* is (.) uhm how often you can do "the pumping thing" (-) so that you can repeat this part in the middle several times (-) you can make a loop (-) and the *k* is just how often you repeat it and he said in the tutorial session that *1* does not work.
C1	1 does not work?
C3	Most of the time, uhm wait (-) no, 1 works never but most of the time *0* and 2 works (-) so just try 0 and 2.
C1	I still don't understand the whole concept.

During assignment 1, all students worked independently on their solutions. As a result, their decision for variables and solutions differed or were not on the same level at the same time. This leads to C1 and C3 being the only ones who ended up with two final solution approaches. Although C1 always relied on C3's explanations during the working process, she decided to pick her own solution instead of the one C3 came up with, since her own solution was more like the tutorial solution. This indicates that the tutorial sample solution is trusted more than C3's knowledge (Table 9.28).

Table 9.28.: Excerpt 23 from Group C at [00:30:23]

Student	Content
C1	So, we have two solutions that both make sense somehow (- -) uh (- -) I personally would rather choose the sample solution from the tutorial session.

However, C1 wanted C3's confirmation of her solution, before the group decided to use it. Although C3 stated that she understood C1's choice of variables, she did not mention that C1's written solution was not formally correct, and that she did not properly justify her decisions for each variable. In the end, C1 assured that she was not the one who had to submit their solution (Table 9.29). At this point, C1 mentioned her problems with formal writing.

Table 9.29.: Excerpt 24 from group C at [00:41:01]

Student	Content
C1	Who actually has to submit the solution?
C4	I have to submit
C1	<relieved sigh> (–) I would not have been able to bring my solution into this formal format properly.

Assignment 2. The students continued to work independently on a solution for the assignment 2. As the number of C2 and C4's interactions continued to decrease (Figures 9.8 and 9.9), C1 and C3 continued the dialogue. Overall, C1 needed less support from C3; however, she still relied on her on several occasions. For example, Table 9.30 illustrates how C1 needed a correction while choosing xyz during the PL step 3 because she violated the restriction that $|xy| \leq n$ (see Section 2.2.2).

Table 9.30.: Excerpt 25 from Group C at [00:58:37]

Student	Content
C1	C3, how willing are you to ignore (-) what our teaching assistant said?
C3	Umh.
C1	Imagine (-) you do not choose the first x and y equal (-) wait (-) x equals b to the power of n plus 1 [$x = b^{\mathrm{n+1}}$] (-) all of our bs (–) and a to the power of i [a^{i}]. So, x is b to the power of n plus 1 and a to the power of i [$x = b^{\mathrm{n+1}}a^{\mathrm{i}}$], y is a to the power of j [$y = a^{\mathrm{j}}$], z is -
C3	But that's (.) not possible
C1	Why not?
C3	We have this small part here, so that the amount of xy must be less than or equal to n [$\|xy\| \leq n$] and if you take b n plus 1 [$b^{\mathrm{n+1}}$] then- *(C3 holds up her solution for C1 to see)*
C1	Why?
C3	Because it's like that in the proof.
C1	Ah I don't like the proof!

Eventually, C1 changed her seat to be able to communicate exclusively with C3 and compare their approaches (Table 9.31). She again ignored C2's statement that he was also not making any progress or was having problems.

Table 9.31.: Excerpt 26 from Group C at [01:06:05]

Student	Content
C1	C3, would you like to come to me?
C3	I don't want anything anymore <laugh>
C2	Oh man, this assignment is really stupid.
C1	Shall I give you my opinion on this? *(C1 addresses C3)* Can we just swap seats so that C3 and I can talk -. (-) Thank you. *(C1 addresses C2)*

The following excerpt illustrates how C1 and C3 worked more on the same level during assignment 2 (PL step 4.1). They both had a problem with the final calculation since they chose $k = 0$. When it occurred to them to choose $k = 2$ instead, their problem was instantly solved (Table 9.32).

Table 9.32.: Excerpt 27 from Group C at [01:07:38]

Student	Content
C3	Then I said k is 0 (-) and then it is this 1 here and then 0 blah blah who knows (.) and then I only have this here (.) because the one there is omitted.
C1	I have exactly the same, C3.
C3	Yes, and when I say two n plus 2 - (-)
C1	I have the same problem and want to talk to you about it.
C3	Ok <laughing>
C1	Try to choose 2 as k.
C3	I was also thinking if I should do this <laughing>.
C1	It has just come to my mind.
C3	Yes, it has also just occurred to me.

However, as in assignment 1, C1 was not able to formally write down the reasons for her decisions. Table 9.33 shows that she was even aware of this and left this work to her group member, whose turn it was to submit their solutions and who did not participate in the group process (C4).

Table 9.33.: Excerpt 28 from Group C at [01:10:46]

Student	Content
C1	So, I have the solution. You are welcome to write it down in a proper way, C4.
C2	But the justification is still missing.
C1	Yes, but I cannot do that.
C2	So why can you say that it is not regular but you cannot justify why it is not regular?
C1	I just cannot write it down formally so that we don't get points deducted.

To sum up, one person (C3) was the driving force behind the correct solution of assignment 1, while C1 relied heavily on her confirmation. However, C1 was also very active in further developing and correcting her own ideas. In the second assignment, both students were equally active in developing the solution together.

Table 9.34.: Roles derived from observed interaction patterns and students who inhabited the role during the study group. Students who not only (but mostly) interacted in one specific role are put in brackets.

Role Name	Role Description	Related Student
Teacher	(1) explains the content and the topics. (2) answers most of the questions. writes down most part of the whole solution approach. (3) (sometimes) is pushed into this position by pupil-students who rely on him or her and explicitly keep asking them questions. (4) explains every step of their solution, several times if necessary. (5) points out the mistakes of their group members, so that they could understand what they had done wrong. (6) is not necessarily able to solve the tasks 100% correctly.	A2, B2, (C3)
Pupil	(1) has many (follow-up) questions and realizations about the topic, concepts, and the solution approaches. (2) is more passive in writing down and interacting in the group, in some cases even slightly aggressive, if it happened that he or she misunderstood something again. (3) his or her explanations and ideas are often (partly) incorrect. (4) keeps repeating something the teachers had already explained. (5) sometimes the teacher's solution approaches are not even questioned anymore, but simply accepted.	A1, (B3), (C1)
Silent Observer	hardly shows any interaction	B1, (C2), C4

9.2.3. Interaction Patterns and Roles

I observed in every group that a maximum of two students were truly working together on the assignments while reenacting a *teacher-pupil* relationship. This happened regardless of the overall number of students in a group. Furthermore, in both groups consisting of more than two students (specifically groups B and C) at least one student barely or not at all interacted with the other group members and behaved as what I call a *silent observer*.

I found that the number and quality of an explanation (i.e., not just repeating *explanations* already mentioned by other group members or wrong *explanations*) and the number of asked *questions* is particularly crucial to highlighting the *teacher* and *pupil* role (see Figures 9.4 to 9.9). In Table 9.34, I summarize these three identified roles that were derived from the interaction patterns.

Overall, in every group, the *teacher*-student was the one to explain the PL scheme once again and share further relevant information with the group through written notes and oral *explanations*. Consequently, *teacher*-students acted as a kind of "personal tutor" by not only answering most of the *pupil*-students' *questions* but also assessing their suggested solution ideas. Although I found a *teacher-pupil* role distribution in every group, this was not consistently the case through the entire working sessions. Some students (B3, C1) who started the second assignment as *pupils*, ended the sessions the same, or almost the same, level as the *teacher*-student regarding *explanations* of the topics or correctness of their ideas. In these cases, the role distribution of *teacher* and *pupil* became more balanced, indicating a learning process for the *pupil*-student.

Nonetheless, even though one *teacher*-student explained or calculated most of the solutions, this does not mean that other group members necessarily understood the solution or learned the concepts and topics thoroughly. Instead, there were still moments towards the end of the respective group sessions when the *pupil*-students simply gave up trying to understand the solution, preferring to wait for the *teacher*-students to write it down completely.

In my study, only the students of group C used several sheets to create their own copy of the group work including their own individual solutions and approaches while working together. (This was true even for the *silent observers*.) Nonetheless, the *silent observer*-students did not use the opportunity to interact with their peers and discuss their different solutions. Moreover, they did not take the opportunity to have their mistakes corrected by their group members so that similar mistakes would not occur in the future. This lack of follow-up questions or interactions with the *teacher*-students indicated that the *silent observer*-students, on the one hand, did not seem to intend to fully understand the solution process at this point (or even feel responsible for doing so). The *teacher*-students, on the other hand, did not feel responsible to integrate their fellow students or to take care of it as a real tutor would do.

9.2.4. Performance in the Final Exam relating to Roles

Table 7.1 presents additional information about the students' homework and final exam performance that I obtained through student interviews and exchanges with teaching assistants. With the aim of identifying whether students' role distributions during their group work (see Table 9.34) were related to students' performance in the final exam, I grouped the students based on the following two criteria:

1. **The performance in the final exam and in the PL assignment:** I considered whether the students scored at least 50% in the final exam PL assignment, achieved less than 50%, or did not participate.
2. **The observed interactions as represented by the roles:** I classified whether the students delivered most of the *explanations*, had the most *questions*, or had almost no interaction during group work.

The results for assignment 1 and assignment 2 are shown in Table 9.35. I have marked in bold the students whose cluster has changed.

It is apparent that most *teacher*-students and *pupil*-students had a high to middle performance on the final exam. On the contrary, all *silent observers* achieved a low performance or did not participate in the final exam. Furthermore, C3 (as *teacher*) and A1 (as *pupil*) performed low. So, even though the amount of interaction noticed in groups can be helpful in estimating the performance in final exams assignments in some cases, there are also students who perform differently than expected.

Table 9.35.: Clustering of students based on their performance in the final exam PL assignment and the observed roles and interactions

	Teacher-student: Highest number of explanations during group work	*Pupil-student*: Highest number of questions within a group	*Silent observer*: Almost no interaction with the other group members during group work
Assignment 1			
High to middle performance in the FE PL assignment	A2, B2	B3, C1	
Low performance in the FE PL assignment	C3	A1	C2, C4
No participation in the FE			B1
Assignment 2			
High to middle performance in the FE PL assignment	A2, B2, **C1**	B3	
Low performance in the FE PL assignment		A1, **C3**	C2, C4
No participation in the FE			B1

9.2.5. Extension to Individual and Shared Knowledge

At this point, my interaction categories from Section 9.2.1 have provided an overview of the interaction patterns in each session and the overarching interaction frequency and structure (see Section 9.2.2). In the final step, I extend the results around how students learn in light of Distributed Cognition Theory (DCOG) (see Section 6.2) Since I have explicitly chosen which homework I observed, the individual and shared knowledge are to be understood in the context of the topic of the PL assignments. Therefore, I examine student sessions from the following two perspectives as described in Section 9.1:

1. The externalization of individual knowledge
2. The internalization of shared knowledge

Since the transcript excerpts in Section 9.2.2 illustrate exemplary situations typical for the respective session, I also refer to these excerpts in the following findings. Because I focus on the results occurring in all groups, I present the results in summary form rather than separately by groups.

Externalizing Individual Knowledge

The students interacted with their group members, tools, and material in different ways to externalize individual knowledge while working on PL assignments. In the following, I present five noticeable similarities between all groups:

1. **Usage of material.** To externalize knowledge through answering *questions* and *explaining* knowledge, the students relied heavily on the material from the tutorial session. Since the tutorial session was the first time an example assignment was solved step by step by their teaching assistant, the students tried to reconstruct the external situation of the tutorial session. For this purpose, they used assignments and solutions from this same tutorial session or additional assignments and its solutions that were available online shortly after the tutorial session in different situations:
    - Right at the beginning of the sessions, each group brought or discussed solutions and notes from the tutorial session as an example for their own PL proof. One group even wrote the general PL pattern down first as a gap text and then began to fill in the gaps (e.g., Table 9.6).

Therefore, the sample solutions were on the table or available digitally via laptops or smartphones.

- The students repeatedly used tips and *explanations* their teaching assistant gave them in the tutorial sessions (e.g., Table 9.7).
- Perhaps because the assignments and sample solutions from the tutorial session were already available to them on the table or digitally, they used it first to solve open questions when none of them knew the answer directly (e.g., Table 9.15).
- The students compared their solution to the sample solution from the tutorial session. They did this to check whether their solution approach was correct or to decide for one solution if different approaches were available (e.g., Tables 9.9 and Table 9.28).
- When group B had open questions after consulting assignments and solutions for sources, they decided to ask the teaching assistant directly in an upcoming tutorial session instead of finding the answer in another way – for example, fellow students, books, websites (e.g., Table 9.16).

2. **Exclusion of group members.** I found that not all students interacted in the group, and not all students externalized knowledge in the homework group. In each of my three observed groups, a maximum of two people worked intensively together regardless of the actual number of group members. For this point, group A can be neglected because it consisted of only two students. For groups B and C, some students were left behind at an early stage of the working process and exerted little effort to become involved. Their *questions* were often of a general nature and were sometimes ignored or only briefly answered because the other students discussed their own problems intensively (e.g., Table 9.23). Another clear illustration of this situation was when C1 even swapped places with her seat neighbor C2, who expressed his difficulties with the assignment. However, C1 wanted to sit closer to C3, with whom she was discussing exclusively during assignment 2 (see Table 9.31).
3. **Distribution of interactions.** Thirdly, all three groups had a similar distribution of interactions in that one person frequently *explained* while one person frequently *questioned* during both assignments (see Table 9.1). Only in group C did the interaction distribution become balanced for the second assignment after both involved students had solved the assignment on their

own sheet of paper (Figures 9.8 and 9.9). Furthermore, whether they had their own paper sheet available or not, the students aimed for one joint solution available on one of their paper sheets. In group A and B, the students who *explained* the most information were also the students who provided or corrected the solution approach to solve both assignments.

4. **Verbal paraphrases.** In two groups, only one student wrote the solution down during the actual development (A2 and B2), while every student in group C had his or her own sheet of paper. However, most *questions* were formulated verbally, or it was pointed to the available solution instead of illustrating the idea or problem on an extra sheet. Once in groups A and B, the verbal discussion reached its limits, and the students who did not write down the solution used a sheet of paper for at least externalizing their ideas as notes. The difference here was that one student was explicitly asked to write it down, while one student did so on his own initiative since he did not find words sufficient. Nevertheless, both students wrote their ideas down in a formally incorrect way and were corrected by their group members (e.g., Tables 9.11 and 9.20).

5. **Quality of explanations, answers, and solutions.** In all groups, some aspects of the solution remained unclear to some students, even though attempts were made to answer their questions. Sometimes the students waved off that they would certainly understand it in the end when the finished calculation was there (e.g., Tables 9.12 and 9.24). This shows that the explanation and answering by students is of a different quality than that of a tutor/teacher would have and could be superficial (e.g., Table 9.27). For this reason, it was not possible for some students to thoroughly understand the topics and solutions in that kind of group session. Furthermore, two out of three groups also mentioned that they expected point deductions from their tutor/teacher. Thus, C1 does not even try to solve their difficulties with formal reasoning in the group. In group A they even say that they do not need every point.

Internalizing of Shared Knowledge

The analysis of internalization of shared knowledge was more limited than the previous analysis. Since I could not talk to the students after evaluating the sessions, I could not reproduce in detail what knowledge students actually internalized. For indicators of whether shared knowledge internalized (or whether an attempt was made to do this), I decided to pay attention to situations that illustrate whether students accept and understand new knowledge for them. As an indicator, I used the following situations: whether students wrote the solution or new knowledge down for their personal use; whether students verified that their personal solution was correct when they wrote a solution down; whether the interaction distribution became more balanced in assignment 2 than in 1 in terms of the category *explanations* (see Table 9.1). Following these considerations, I identified four noticeable findings:

1. **Understanding the pumping lemma.** Overall, the students' focus was placed more on copying an existing sample solution as much as possible instead of understanding every step or symbol. This can be seen at several points – for example, not all students were involved in the solution process (e.g, Table 9.23), and some questions were answered superficially and only to the extent necessary to solve the assignment (e.g., Table 9.27). Furthermore, the students used tutorial assignments and solutions intensively (e.g., Tables 9.9, 9.15, and 9.28).

2. **Writing down and verifying of personal solution or new knowledge for personal use.** Regarding writing down, I need to distinguish between group A and B (no parallel working on the assignment) and group C (parallel working on the assignment):

    - In groups A and B, only one solution existed in the end, and this solution was written down by the one person answering the most *questions*. As mentioned before, almost all questions were handled verbally except for two cases (e.g., Tables 9.18 and 9.20).

    - Group C had four solutions that were more or less finished but not always similar solutions at the end. Every student worked on their own sheet of paper, but only two students discussed their solutions and ideas with each other (C1 and C3). Therefore, two students did not discuss their idea or solutions, so these were not corrected or finished. Before ending the session, group C confirmed who was going to submit the solution (see Table 9.29). Their system of rotation as to who has to submit the solutions led to C4 being the next. Such a working process is problematic since the final solution is C1's idea that she created with the help of C3. No one explained the approach again and, in the end, C4 got no formally correct solution, but only an approach. Nevertheless, he also did not ask any further *questions* or put his own solution up for discussion.

3. **Interaction differences between assignments 1 and 2.** For assignment 2, the distribution of interactions was not more balanced instead of a fixed question-answer situation between only two students (see Figures 9.4 to 9.8). A change of such an interaction distribution only occurred in group C. After C1 was insecure initially, she was convinced of her solution at the end of assignment 1 and asked far fewer questions in assignment 2. However, many questions came up repeatedly in the other groups, and the interaction of A1, B1, C2, and C4 even decreased, indicating a lack of internalized shared knowledge.

4. **Final exam results.** As can be seen in Table 7.1, of the nine students observed, eight participated in the final exam. While reviewing the final exam solutions, I made two discoveries, as follows:

   - The transfer of habits from the group work: I noted that A1 only wrote a gap text in the final exam. This could mean that he memorized the PL pattern group A used intensively during the group work. Nonetheless, A1 was unable to fill it in with the right choices. I speculate here (also based on the lower interaction in assignment 2 that can be seen in Figure 9.5) that A1 only took reproducible knowledge in the form of the pattern but by no means achieved the knowledge to apply the PL to unknown languages.

   - The problem with not annotated errors: As mentioned before, C3 confirmed C1's solution to assignment 1. However, C3 did not point out that C1 did not formally justify her decisions for the variables. These are explicit steps during a PL proof (see Section 2.2.2). C1 also omits this formal justification in the final exam.

9.3. Discussion

In this section, I discuss research questions 4 to 6. First, I analyze the observed interaction patterns and role behavior in light of existing research. Second, I discuss the results of the relationships among observed interaction patterns, derived roles, and the performances in the homework groups and final exams. In a final step, I discuss the implications of students externalizing and internalizing of knowledge.

RQ4: What kind of interaction between students can be observed within ToC study groups while they are working on homework assignments?

My results describe the kind of interaction patterns that occurred while three student groups were solving two PL assignments together. By analyzing the interaction patterns, I was able to derive two observations regarding the roles that students inhabited during their group sessions: (1) One student turning into a *teacher* and one student turning into his or her *pupil*, independent of the overall group size, while (2) remaining students become *silent observers*. Next, I explain these three roles in more detail.

Every *teacher*-student was the driving force behind the group's progress. He or she was the person responsible for the final solution of the assignment given. For this reason, I assume that without this student, developing a solution would have taken much longer or would even have been unlikely. Deitrick et al. (2015) came to a similar conclusion in their study, where a group of students could only solve a programming task because one student had a musical background [DSA+15]. As with this conclusion, I observed that not all of my students came to the study group with the same level of preparation, as some students had not attended the lecture and the tutorial session. Therefore, at least the *pupil*-students required further tutoring and explanation from the *teacher*-students since they had not fully understood all the necessary information to be capable to solve the PL assignments given.

My second observation concerning the *silent observer*-students in the groups supports the existing research results that in student groups with more than two members, not everyone participates (equally). For example, research about into group participation in programming courses has already emphasized that "small teams of four, occasionally, five members" [Pre05, p. 41] often split into separately working pairs within their teams, which seems "more beneficial than working in groups of four" [ibid., p. 41]. Further research into introductory CS1 courses supports my results as well. In groups of three or four students, not every member is "actively engage[d] in the group" [GVG00, p. 98]. The situation when students in collaborative working settings participate without any fair share of contribution to the group's goal is also known as "free riding". Recently, Tenenberg (2019) stated that despite the awareness of free riders in engineering education, there are no specific theories that can be used for understanding and approaching the phenomenon [Ten19]. In his extended review, he suggests combining several theoretical insights from research in sociology, economics, and psychology to be used by engineering educators to structure group work with the goal of avoiding free-riding students. For example, I found the following two guidelines surprisingly simple to incorporate into CS education: (1) "make teamwork rules and expectations explicit" in order to encourage students to try to learn during the session in such a way that they are able to solve the final exam individually; (2) encourage early discussions of "what are 'fair' contributions" to the group work and how a missing contribution will be handled so that silent students (or "free riders") also know they have to prepare before the session in order to contribute their share to the solution process [ibid., p. 1717]. Other research provides more reasons for the silent behavior – for example, "issues of learning preferences, motivation, preparation for the session, cultural literacy, language, concerns with face and group dynamics," and "a [...] lack of learning" [RCH08, p. 211]. Recent research also stress that "the tensions underlying group communication may be challenging for quiet students" and this is "an area which may require further attention from educators wishing to use collaboration techniques in the higher education classroom" [MU20, p. 253].

RQ5: In what way are the observed interaction forms related to the group's assignment performance as well as students' individual final exam assignment performance?

My study confirms that students accomplished the group work not necessarily with the same amount and quality of individual understanding as the results of the group work may imply [Ros92]. Although the collaboration I observed in the study groups resulted in final solutions most of the time, the group work did not guarantee that all students understood the approach and the solution in such a way that they were able to solve similar tasks independently – for example, in the final exam.

In general, I expected that the *teacher*-students – who answered their group members' *questions*, had a high frequency of *explanations*, and were the driving force behind the joint solution – would mostly be able to perform well on a similar PL assignment on final exam. Furthermore, the performance of the *silent observer*-students on the final exam PL assignment did not surprise me either. The students with no interaction in the group or any other contribution to the homework solution either did not take the final exam or received almost no points on the PL assignment of the final exam.

What I found particularly interesting was the performance of the three *pupil*-students (A1, B3, C1) and the way their final exam performances on the PL assignments related to their group performance, providing potential explanations for the former. Moreover, when examining A1's forms of interactions (mostly follow-up *questions* and repetitions of *explanations* provided by his *teacher*-student), I was not surprised that he did not solve the final exam PL assignment on his own. *Pupil*-students B3 and C1 also required a lot of *corrections* and *explanations* by their group's *teacher*-student (see Figures 9.6 to 9.9). But unlike A1, they were able to discuss with their *teacher*-student on an equal footing through the second assignment reaching a quality of exchange that can be interpreted as the kind of desired collaboration found in the education literature (e.g., [Ros92] [RT95]).

Overall, I wondered how it could be that B2 (*teacher*-student) and especially B3 (*pupil*-student), got the highest final exam assignments score of all the students I observed (see Table 7.1), despite not being able to solve the assignments in the group session. I propose two reasons: First, B2 and B3 had a long discussion while trying to solve the problem that remained unsolved during the session. I observed that they tried to understand the PL and the individual symbols in detail and worked on their solutions more thoroughly than the other groups. Moreover, before the students ended their session, B3 suggested they should turn to a teaching assistant for help in the next few days. This additional exchange with the teaching assistant might have ensured that the students in group B were able to ask more specific questions and build a more thorough understanding of the assignments and solutions, resulting in high performance in the final exam assignment. This kind of renewed intensive opportunity for discussion and consultation with the teaching assistant is similar to the flipped classroom idea where students gather the topics for themselves before trying to solve assignments under supervision in a classroom [BV13].

RQ6: How and why do students externalize individual knowledge to shared knowledge in their homework groups?

In the course I analyzed, which follows the traditional pedagogical approach (see Sections 2.2.2 and 4.2), it is noticeable that students rely heavily on the tutorial session. By following the PL scheme from the tutorial session, the students externalize their internal representation from their tutorial session to contribute individual knowledge to a shared knowledge. Following DCOG as the theoretical background, when the students use the example assignments and sample solutions from the tutorial session, these documents not only embody the factual knowledge about PL assignments but also the knowledge for how to use these documents to solve similar assignments. Considering, for example, the superficiality of their solutions, preferring the possibility to formulate questions verbally instead of writing it down, and the exclusion of group members, I establish the actual goal of the students: It is not about each of them understanding the solution down to the last detail. Instead, it is about somehow finding a solution that earns as many points as possible from the teaching assistant, even if the students already expect point deductions. Accordingly, the students do not necessarily try to understand the meaning of PL or the different variables used but simply copy the tutorial solution as far as possible.

A possible explanation for the students focusing on the tutorial session and their tutors was already provided by Nespor (2014). In his research, he conducted several interviews, observed class and study sessions, collected course material, and simply talked with physics and management students to offer a detailed picture of several factors connected to student learning. He found that "professors were the definers of relevance" for learning material [Nes14, p. 90]. In my case, the teaching assistants also had this kind of authority since they would decide how many points students would get for their submitted assignments.

What the students now accomplish in their homework group is to contribute and merge (more or less) disjunctive knowledge to solve an assignment in group work. Overall, no student solved the assignments totally on their own, but engaged in discussion with at least one other person, who supported the process by giving ideas or even making mistakes. I refer here again to the definition of collaboration as presented in Section 6.1: Collaboration is described as "a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem" [RT95, p. 70]. In this way, students are actually doing what is expected of them in a homework group when they are expected to solve assignments *collaboratively*. At this point, one must distinguish the goals of collaboration and Collaborative Learning (CL) that is extending this theoretical framework by explicitly emphasizing the importance of learning: CL "refers to an instruction method in which learners at various performance levels work together in small groups toward a common goal" [Gok95, p. 22]. Consequently, "the students are responsible for one another's learning as well as their own. Thus, the success of one student helps other students to be successful." [ibid., p. 22]. This definition assumes that students working in groups have different kinds of prior knowledge and background (like the observed students had). While working together towards the correct homework solution, they are supposed to engage in self-organized learning and even be responsible for their group members' learning success.

At this point, I also want to return to Hutchins (1995) and his work about DCOG [Hut95] (see Section 6.2). He also states that it appears that individuals contribute through interaction to achieve a common outcome they could not have achieved on their own. Based on the qualitative analyses I conducted, I am prepared to support Hutchins's assumption: Student groups seem more likely to aim to combine parts of their individual knowledge into a shared knowledge to solve an assignment (i.e., they *collaborate* to generate a solution that will earn them as many points as possible from their tutor/teacher) rather than teaching their peers or learning on their own (as would be expected with *collaborative learning*).

Altogether, when learning is defined as internalizing an internal reconstruction of external situations, the students learn during this homework session how to solve a PL assignment in a group with specific group members and access to specific tools (mostly material from the tutorial session). Never in the course are the students expected to work on PL assignments under final exam conditions – for example, alone and without additional material. Consequently, not all students did learn how to solve an assignment alone in a fixed time and without access to these tools.

9.4. Summary

In the second step of my qualitative study, I focused on examining how students interact with each other and with material while completing PL assignments in group work. Therefore, I video-recorded three student groups, and used VIA and QCA to develop a category system representing nine forms of interactions.

The analysis allowed me to show that the most common interactions in student group work were *explanations*, *questions*, and *approvals*. The category occurrences suggest the desired learning situation in the student group where at least one student explains something and at least one student asks questions. However, this situation only occurred with two students, no matter how many students were in the group. Moreover, in each case, one person was the *explaining* person, and the other was the *questioning* person.

The observation of the interaction patterns allowed me to divide the students into the roles of *teacher* (mostly interacted through explanations), *pupil* (mostly interacted through questions), and *silent observer* (mostly did not interact). By mapping the performance of these student roles onto similar final exam assignments, it was shown that *teacher* performed mostly well and *silent observer* consistently performed poorly. The performance of *pupil* students is different.

Next, I aimed to understand how students share their individual knowledge and what are the reasons why not all of them can use the shared knowledge in the homework group later on in the final exam. Therefore, re-examined the interaction situations and extended the analysis to the actual learning process through externalizing individual knowledge and internalizing shared knowledge in the light of DCOG. On the one hand, while externalizing their knowledge, I found that students relied heavily on information from the tutorial assignments, tutorial solutions, and what their tutor/teacher might want to see. Not all group members were involved, answers were often superficial, and hardly any personal notes or solutions were written down. On the other hand, I found that students lacked internalization of new knowledge by focusing on *one* copy of an example solution as much as possible and not discussing personal solution ideas or problems.

During my analysis, I also realized that a distinction must be made between collaboration and collaborative learning. While some of the students successfully work collaboratively to create a shared solution to a proof assignment, they are not aware that they are also expected to self-learn and teach each other in group work, as would be the case in collaborative learning. The goal of group work should be that each student ends up being able to solve such an assignment on their own, rather than just creating a solution that they could not have achieved on their own. Otherwise, students will not be able to solve similar PL assignments in the final exam.

Part IV.

Concluding Review

10. Summary and Conclusion

This final chapter contains a summary of the work and its results. The scientific contributions are presented, followed by suggestions for possible future work. The thesis concludes with reflections on implications for practice.

10.1. Summary

The present work provided further insights into the difficulties of students in introductory courses of Theory of Computation (ToC). As a result, new insight into the reasons for the rather low performance and high failure rates within these courses were found.

At the beginning, I stated the motivation behind this dissertation project. During my master's thesis, I conducted an observational study of Computer Science (CS) students working on an NP-completeness proof. The origin of this study was based on the fact that failure rates in ToC courses are high, and the final exam results tend to be poor. As a first starting point for understanding students' situation in those courses, the abstract and formal proofs were understood as a possible reason for the low performance. My insights into students' working processes showed how the students had no structure in their work process and were unsure of what they had to prove and what was already given. Moreover, they understood the process of writing down only as something to do for their own work and hardly used the opportunity to record information for all to see. In addition, however, the students also indicated how interesting and exciting the assignment was, thus contradicting the common assumption in related research in the field that students lack interest in the subject matter or the skills to understand the subject. After these previously unknown insights into how students work on proof assignments, I found that student working processes were a previously neglected source for valuable insights concerning how student-oriented pedagogy can foster working proficiency in this field of education. However, since I chose the topic to be observed based on my subjective teaching experience, I first wanted to understand what students' performance on different assignments and topics really is. Based on this quantitative analysis, I chose which assignments I examined in more detail through a qualitative analysis, focusing on student collaboration in their homework groups (Chapter 1).

Before the analyses, I have summarized and discussed related work that emphasizes the importance of ToC as part of CS formation. Together with engineering and sciences, the mathematical ideas that fostered the development of ToC form a tripartite of three traditions. On the one hand, the three traditions have mutually developed and benefited from each other. On the other hand, this tripartite division also led to ambiguities about the objective of a CS university degree and the emphases that should be placed in the academic discipline, since the three traditions pursue different goals, use different methods, and produce different products. While CS curricula covered the various disciplines and traditions that formed the basis for CS, ToC also has an established place in the recommended CS curricula to this day. While the importance of mathematical roots was illustrated in the early curricula, an emphasis in today's curricula is on presenting the learning outcomes and competencies that CS students can and should develop by taking ToC courses. Thereby, it becomes even more apparent how important it is to examine the teaching of ToC more closely in order to prevent students from failing essential ToC topics and potentially dropping out of their CS studies as a result. As a content example, I have explained how Pumping Lemma (PL) for regular languages are taught today and how the historical developments of ToC still have an influence on teaching ToC (Chapter 2).

The current state of research shows how the origins of students' difficulties in ToC courses are generally based on two assumptions. First, many of the approaches presented assume that students are often unsure of the relevance of theoretical topics to their further studies and careers. Therefore, approaches have been offered for modifying and extending existing approaches by linking theoretical concepts to other CS courses or practical examples. Second, another common assumption is that students lack motivation and interest in abstract or mathematical topics. Various software, systems, and tools have been developed and used to make the topics more interactive and increase motivation and engagement. In the studies that look more intensively at the difficulties of individual topics, it is also not sufficiently traceable, for the most part, how they arrived at the topics, or whether it is just based on subjective observation and teacher experience (Chapter 3).

In the first study of the thesis, I investigated how students perform in all homework and final exam assignments of an introductory course in ToC. This study was motivated not just by the lack of empirically investigated assumptions about why students have problems with theoretical topics. In addition, the question arose as to whether there was a bimodal distribution of grades that would prove that some students consistently find the theoretical topics easier than others (see Section 1.1.).

During the analysis, the first two research questions of this dissertation were addressed:

- RQ1: What kind of assignments usually covered during an ordinary, undergraduate introductory Formal Languages and Automata (FLAT) course are causing students most difficulties?
- RQ2: Are there differences among low and high performing students, especially regarding these potentially different assignments as questioned in RQ1?

Due to a lack of insight into the research topic, I conducted an Exploratorative Data Analysis (EDA). Tables and graphs provided an overview of the data on which further statistical analysis could be based. First, I analyzed only the students from the 2016 cohort. I then decided to look at two more almost identical cohorts in terms of structure, content and participant numbers, and failure rates. The core of the analysis was the one-way Analysis of Variance (ANOVA), as this allowed significant differences in the cohorts to be made apparent. In order to see what differences could be found among the students depending on their final exam grades, I divided the students into grading groups for this analysis according to their grades (Chapter 4).

Overall, I was able to gain several interesting insights: All cohorts performed low in comparison to the average performance in the other final exam assignments in one assignment that required a formal proof using the PL. This lower average performance applied to all grading groups, regardless of the overall final exam score. Therefore, I could already question that some students gain an overall strong understanding about all content in ToC course and also question some bimodality to some extent. The homework assignments were examined in the same way to see if the other assignments had low performance and if these also required proofs. Once again, most of the proof assignments had worse performance on average, but the PL assignment performance was not as low as the one in the final exam. Moreover, the different grading groups were worse in the same assignments than the other assignments' performance. Through the ANOVA, I was able to find significant differences in some assignments from the different years. Unfortunately, this type of analysis could not provide accurate reasons for the differences. However, one assignment from 2016 received another subtask in subsequent years that required proof development and the performance then became significantly worse. This underscores the possible difficulties of the students with formal proof assignments (Chapter 5).

For the second part of the thesis, I decided to select an assignment based on the quantitative study results. Through an observational study, I wanted to understand students' problems in these assignments more deeply. Since I was particularly surprised by the discrepancy in the PL assignment between performance on the final exam and the homework, I decided to do a more detailed analysis of PL assignments. It turned out that existing research focuses on the analysis of written solutions. I decided to observe student groups working collaboratively on two PL homework assignments for a more detailed picture, as such an analysis could provide insights into the overall solution process. As a result of this, I have emphasized understanding the collaborative work and the extent to which students contribute to a shared solution. In the course of the analysis, I addressed four research questions through three different analysis focuses (see Section 1.2).

In order to examine collaborative groups, I first explored the concept of collaboration in more detail. Thereby, I motivated interactions as the unit of analysis. In the following, I defined my understanding of learning following Distributed Cognition Theory and considered it performed by students externalizing individual knowledge and internalizing shared knowledge (Chapter 6).

In order to record the whole student working process and to evaluate it in detail, I decided to use video recordings. This allowed me to see more precisely when and at what point the students worked together and what material they used. I analyzed the video data with a mixture of Video Interaction Analysis (VIA) and Qualitative Content Analysis (QCA) (Chapter 7).

As the first step for my qualitative study of student groups working on PL assignments, I worked on the following research question:

- RQ3: What kinds of pitfalls and challenges do students encounter in a self-organized study group when working on a PL homework assignment?

Therefore, I collected content-related information about the working processes of the students and their pitfalls, challenges, and difficulties with PL assignments. The result was that most of the reasons for difficulties are *wrong understanding*, *missing knowledge*, and *missing ability*. Overall, I found similar difficulties in the homework groups to those previously found by other researchers in similar solved final exam problems. However, although these difficulties were often overcome in group work, this analysis also revealed some possible weaknesses of group work. Group work can ensure that individual students do not have to understand the entire solution process if each group member contributes individual parts to the common solution. Furthermore, it may happen that not all students participate in the group work process and are left behind at an early stage of the working process. Due to this, they could not ask specific questions, recognize errors, or attend to the solution process more than being present physically (Chapter 8).

In the second step of my qualitative study, I focused on examining how students interact with each other and with the available material while completing PL assignments in collaborative group work and how they learn in the light of Distributed Cognition Theory (DCOG). The following three research questions framed my work:

- RQ4: What kind of interaction between students can be observed within ToC study groups while they are working on homework assignments?
- RQ5: In which way are the observed interaction forms related to the group's assignment performance as well as students' individual final exam assignment performance?
- RQ6: How do students externalize individual knowledge to shared knowledge in their homework groups?

Therefore, I inductively used QCA to develop a category system representing nine forms of interactions. This allowed me to detect specific interaction patterns which could be classified into the three student roles of *teacher* (mostly interacted through explanations), *pupil* (mostly interacted through questions), and *silent observer* (mostly did not interact). By mapping the performance of these student roles to similar final exam assignments, it became apparent that *teacher* performed mostly well and *silent observer* consistently performed poorly, as I would expect based on their interactional participation in the group work. However, the performance of *pupil* students was different. After studying the students' interactions with each other, the question remained as to how exactly students share their individual knowledge and what are the reasons why not all of them are able to use the shared knowledge in the homework group later in the final exam. On the one hand, while externalizing their knowledge, I found that students rely heavily on information from the tutorial assignments, tutorial solutions, and what their tutor/teacher might want to see. Accordingly, not all group members were involved, answers were often superficial, and hardly any personal notes or solutions were written down. On the other hand, I found that students lacked internalization of new knowledge by focusing on copying an example solution as much as possible and not discussing personal solution ideas or problems. During my analysis, I also found that some of the students do successfully work collaboratively to create a shared solution to a proof assignment. At the same time, they were not aware that they are also expected to self-learn and teach each other in group work. For this reason, I have discussed the difference between the concepts of collaboration and collaborative learning. In the end, the goal of group work should be that each student ends up being able to solve such an assignment on their own, rather than just creating a solution that they could not have achieved on their own. Otherwise, students will not be able to solve similar PL assignments on the final exam (Chapter 9).

10.2. Scientific Contribution

This doctoral project provides scientific contributions in the research field of education, specifically to the research field of ToC education through my student-centered research focus. In what follows, I first show contributions that relate to difficulties in ToC and then briefly address new findings on pedagogical design. At the end, there is also a contribution concerning the selection of data sources.

All students have lower performance in proof assignments, regardless of their overall final exam grade

Until now, the assumption that students have the most difficulty with proof assignments in ToC courses was mostly based on subjective teaching feedback or occasional student surveys. By analyzing around 1500 students during a quantitative study, I was now able to demonstrate that students in their first year in an introductory course of ToC (more specifically FLAT) have low performance in almost all proof assignments. I also found that this phenomenon can affect all students regardless of their final exam grades. This contradicts the principle of bimodality, where it is assumed that some students can gain an overall strong understanding of all ToC topics, while others simply cannot. So far, this assumption has not been supported by an examination into the homework and exam performance.

Students have similar difficulties in pumping lemma homework and pumping lemma final exam assignments

If only the final and written solutions are used for research and studies, the problems students have but also solve while working on the assignment may be overlooked. This becomes particularly relevant when students are supposed to solve their homework in groups but then solve their exams individually. For this reason, I specifically studied student groups working on assignments and not only their final written result. Although students have similar problems in PL homework assignments and similar final exam assignments, students are often able to solve them together in their homework groups. This ensures higher performance in the pumping lemma homework compared to the individual final exam assignment.

Not all students can benefit from working on assignments in groups

Previous work has shown that students improve their general skills, such as communication and social skills, through collaboration. However, there is no work on how ToC topics are handled in such a collaborative group that is supposed to solve homework together. I have now observed that only two students in the group collaborated intensively, regardless of the actual group size. The students who did not collaborate did not show up for the final exam or did poorly on a similar proof assignment. In the group itself, these students either did not try to integrate or were not integrated into the group process by their group members. I hypothesize that students did not explicitly see this group work as an opportunity to learn and practice with their group members. However, if they were deeply engaged in the topic within the group, they would know what questions they could ask their tutor about. Eliminating such ambiguities could then lead to better results in the final exams on the subject. However, if they do not deal with the topic at all, they miss the opportunity to clear up their confusion.

The tutor is the authority that counts

So far, there is no evidence about what materials students really use when working on PL assignments and how they externalize and internalize knowledge. I was able to find out that their work process is shaped by the tutorial session they can attend and the material they receive there. Unlike in the lecture, they receive practice problems and sample solutions in the tutorial session. They then focus in the homework group on finding the practice problems that are most similar to their homework in order to use them as templates for their own solution. One possible explanation for this focus on the tutorial session and material is that the tutor also grades the homework. Overall, the student's goal is therefore to get as many points as possible for the homework by copying their tutor as closely as possible. The current course structure favors this focus, as the points in the homework count toward the final course grade.

Video data provides new insights into how students work on ToC assignments.

Previous assumptions about students' difficulties with theoretical topics have often been based on the experiences of teachers or on lecture evaluations in which students give their subjective assessment of the course at the end of the course. Attempts to obtain objective results have focused on analysis of written outcomes. My performance analyses yielded objective information about certain patterns in student performance. Using these results, observations or video data can be planned to examine these objective low performing assignments in more detail. In this way, it is also possible to identify phenomena and difficulties that students overlook during their work process or problems that do not reappear in written work because they may have been solved. In this respect, video data offer a previously neglected but valuable source for educational research.

10.3. Future Work

The results of my quantitative and qualitative study offer several starting points for further studies, which are briefly outlined below.

Quantitative performance analysis of ToC courses at other universities

Further quantitative performance analyses of students from different ToC courses and universities can provide an attractive basis for deeper analysis of emergent phenomena or patterns about students' difficulties. The study by Enström, for example, already reached the result that complexity proofs cause the most difficulties for students [Ens14]. It would be interesting to find out whether these difficulties also occur for all students regardless of the final exam grade.

Qualitative studies on similar or other assignments

Some significant differences found between assignments other than PL assignments could not be explained by purely quantitative analyses (see Chapter 5). It is not certain whether the qualitative study conducted can be limited to PL assignments or whether it can be considered for other ToC assignments that have to be solved in group work. Therefore, I encourage that further qualitative studies should be conducted. Overall, I present two starting points here to pursue this research further:

- It could be useful to validate and extend the results with observational studies on similar PL assignments from the same or other universities or with other group sizes. Such studies could provide insights into the influence of the respective course setups or discover a possible differentiation for other group sizes.
- It would be interesting to conduct further observational studies to examine whether students in other (proof) assignments collaborate differently or face different challenges and pitfalls than in PL assignments. My analyses provide starting points for other low-performing assignments that are worth investigating.

Analyses to analyze changes in pedagogical design

My results already offer valuable information about creating a pedagogy that better addresses student learning in this highly challenging field of CS. In the next section, I provide suggestions for implications for practice that can serve as the basis for changes in pedagogical design. Based on implemented changes, similar quantitative and qualitative studies can be conducted to assess the impact of the changes on failure rates and student performance.

10.4. Implications for Practice

The conducted quantitative and qualitative studies offer different implications for practice. In the following, I consider the significance of both studies separately.

Implications Based on the Quantitative Study

The first hypothesis from my quantitative study extends the findings that formal proof techniques are strongly challenging students in ToC courses to the fact that proof techniques are challenging for all students independent from their overall final exam performance. I have described in Section 2 that the topics and concepts of ToC are formalized. Only then is it possible to give formal proofs in the way that the roots of the mathematical tradition of CS have used. The fact that all students have difficulties with the proof assignments indicates that it is precisely the formalism that causes problems for the students here. The second hypothesis explicitly challenges bimodality about student performance in ToC. Therefore, I can refute the argument that students with high final exam scores display an overall strong performance for all assignment types covered during the course. In contrast, students with low scores on the final exam also showed mixed performance depending on assignment types. With these results, I reinforce the argument that ToC courses need to reconsider their traditional pedagogy model. I highlighted the approach of Gal-Ezer and Trakhtenbrot (2016; see also 2013) who analyzed misconceptions students developed about reduction proofs [GET16] [Tra13]. Based on these results, the authors developed a series of exercises addressing students' potential misconceptions proactively. This approach tries to put the responsibility for learning back on the side of the teacher and not on the side of the learner because previous attempts to improve the poor situation in ToC courses have not shown any improvement for various reasons. Even if mathematics and CS students are only partially comparable, an exchange with mathematicians can also provide helpful ways of thinking about how to teach formalized concepts and especially, formal proof methods. So far, for example, generic proofs or peer review procedures have not been specifically tested in ToC courses but seem to enable better results in mathematics for some of the students.

Implications Based on the Qualitative Study

The results of my qualitative research have important implications for understanding learning in homework groups. On the one hand, I found that only some students benefit from the possibility of working out the solution for themselves but still use their group as support. On the other hand, students need to be aware of the real goal of a group session – not only to solve the assignment but also to deal intensively with the topics themselves. Therefore, the findings of the qualitative research should also be considered in relation to the pedagogical approach. As implemented in the analyzed ToC course (see Section 1.3.), the traditional pedagogical approach does not promote internalization of knowledge as it would be necessary for performing high on the final exam since only the group submission is controlled. For a desired individual internalization, the incentive would have to be changed – for example, if students had to present a solution individually in the tutorial session, they would prepare and understand the topics and assignments differently. My findings also suggest that a flipped-classroom approach is worth considering (see Section 9.3). Likewise, transparency about the goals of the collaborative groups and the impact of non-participation could have an impact on the group collaboration and outcomes [Ten19]. Overall, the goal should be for students not only to work out a solution but also to really be able – and explicitly know that they are expected – to work through the assignments in such a way that they can solve them on their own in the future. Ultimately, this is the only situation in which they can solve such assignments on their own and talk about them with their teaching assistants before the next topic is introduced. Students should be able to benefit from this situation.

10.5. Concluding Remarks

This work has provided new insights into the reasons for CS students' high failure rates and low final exam performance in ToC introductory courses. Among other things, the use of video recordings of student groups has provided a previously neglected source of data that can serve as a basis for changes to the course structure. Through my findings, I question the assumptions made at the outset that students are generally not motivated or interested in the theoretical topics or find them too abstract or mathematical.

Building on the findings, work can be done to improve the students' situation in ToC introductory courses, especially regarding their difficulties in proof development. This will not only give them a better foundation in their further studies, but it will also ensure that more qualified computer scientists can be trained if ToC courses does not make them to discontinue their studies altogether.

Bibliography

[ACH$^+$68] William F Atchison, Samuel D Conte, John W Hamblen, Thomas E Hull, Thomas A Keenan, William B Kehl, Edward J McCluskey, Silvio O Navarro, Werner C Rheinboldt, Earl J Schweppe, et al. Curriculum 68: Recommendations for academic programs in computer science: a report of the acm curriculum committee on computer science. Communications of the ACM, 11(3):151–197, 1968.

[ADS12] Serge Autexier, Dominik Dietrich, and Marvin Schiller. Towards an intelligent tutor for mathematical proofs. arXiv preprint arXiv:1202.4828, 2012.

[AGE06] Michal Armoni and Judith Gal-Ezer. Reduction–an abstract thinking pattern: the case of the computational models course. In Proceedings of the 37th SIGCSE technical symposium on Computer science education, pages 389–393. SIGCSE, 2006.

[AGEH06] Michal Armoni, Judith Gal-Ezer, and Orit Hazzan. Reductive thinking in computer science. Computer Science Education, 16(4):281–301, 2006.

[AH14] Nicholas Alchin and Carolyn P Henly. Theory of Knowledge Third Edition. Hachette UK, 2014.

[AK01] Lorin W Anderson and David R Krathwohl. A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman,, 2001.

[Alm00] Dennis Almeida. A survey of mathematics undergraduates' interaction with proof: some implications for mathematics education. International Journal of Mathematical Education in Science and Technology, 31(6):869–890, 2000.

[Arm09] Michal Armoni. Reduction in cs: A (mostly) quantitative analysis of reductive solutions to algorithmic problems. Journal on Educational Resources in Computing (JERIC), 8(4):1–30, 2009.

[AŞ10] Pınar Anapa and Hatice Şamkar. Investigation of undergraduate students' perceptions of mathematical proof. Procedia-Social and Behavioral Sciences, 2(2):2700–2706, 2010.

[Bar00] Brigid Barron. Achieving coordination in collaborative problem-solving groups. The journal of the learning sciences, 9(4):403–436, 2000.

[Bau96] Rüdiger Baumann. Didaktik der informatik, 2., vollst. neu bearb. Aufl., Stuttgart ua, 1996.

[BB03] Olav W Bertelsen and Susanne Bødker. Activity theory. HCI models, theories, and frameworks: Toward a multidisciplinary science, pages 291–324, 2003.

[BDYL12] John T Behrens, Kristen E DiCerbo, Nedim Yel, and Roy Levy. Exploratory data analysis. Handbook of Psychology, Second Edition, 2, 2012.

[BEW08] Grant Braught, L Martin Eby, and Tim Wahls. The effects of pair-programming on individual programming skill. In Proceedings of the 39th SIGCSE technical symposium on Computer science education, pages 200–204, 2008.

[BJJ01] Dave Berque, David K. Johnson, and Larry Jovanovic. Teaching theory of computation using pen-based computers and an electronic whiteboard. SIGCSE Bull., 33(3):169–172, June 2001.

[BK13] Rolf Biehler and Leander Kempen. Students' use of variables and examples in their transition from generic proof to formal proof. In Proceedings of the 8th Congress of the European Society for Research in Mathematics Education, pages 86–95, 2013.

[BM15] Melanie Birks and Jane Mills. Grounded theory: A practical guide. Sage, 2015.

[Bow18] Jonathan P Bowen. The impact of alan turing: Formal methods and beyond. In International Summer School on Engineering Trustworthy Software Systems, pages 202–235. Springer, 2018.

[BR07] William Billingsley and Peter Robinson. Student proof exercises using mathstiles and isabelle/hol in an intelligent book. Journal of Automated Reasoning, 39(2):181–218, 2007.

[Bro04] Wayne Brookes. Computing theory with relevance. In Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30, pages 9–13. Australian Computer Society, Inc., 2004.

[Bry16] Alan Bryman. Social research methods. Oxford university press, 2016.

[BV13] Jacob Bishop and Matthew A Verleger. The flipped classroom: A survey of the research. In 2013 ASEE Annual Conference & Exposition, pages 23–1200, 2013.

[CBÖ18] Malgorzata Ciesielska, Katarzyna W Boström, and Magnus Öhlander. Observation methods. In Qualitative methodologies in organization studies, pages 33–52. Springer, 2018.

[CEK13] Pierluigi Crescenzi, Emma Enström, and Viggo Kann. From theory to practice: Np-completeness for every cs student. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education, pages 16–21. ACM, 2013.

[CGG+05] Joshua J. Cogliati, Frances W. Goosey, Michael T. Grinder, Bradley A. Pascoe, Rockford J. ROSS, and Cheston J. Williams. Realizing the promise of visualization in the theory of computing. J. Educ. Resour. Comput., 5(2), June 2005.

[Cha86] Chris Chatfield. Exploratory data analysis. European journal of operational research, 23(1):5–13, 1986.

[Chu36] A Church. An unsolvable problem of elementary number theory. american journal of mathematics, 58, 345-63. Repr. in Davis, 1965:88–107, 1936.

[Chu07] Daniela Chudá. Visualization in education of theoretical computer science. In Proceedings of the 2007 international conference on Computer systems and technologies, page 84. ACM, 2007.

[CLRS09] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms. MIT press, 2009.

[CnGM04] Carlos I. Chesñevar, Maria P. González, and Ana G. Maguitman. Didactic strategies for promoting significant learning in formal languages and automata theory. SIGCSE Bull., 36(3):7–11, June 2004.

[DB16] Nicola Döring and Jürgen Bortz. Forschungsmethoden und evaluation. Wiesbaden: Springerverlag, 2016.

[DDP00] Vladan Devedzic, John Debenham, and Dušan Popovic. Teaching formal languages by an intelligent tutoring system. Educational Technology & Society, 3(2):36–49, 2000.

[DKL03] Helen Drury, Judy Kay, and Warren Losberg. Student satisfaction with groupwork in undergraduate computer science: do things get better? In Proceedings of the fifth Australasian conference on Computing education-Volume 20, pages 77–85, 2003.

[DM15] Peter J Denning and Craig H Martell. Great principles of computing. MIT Press, 2015.

[DSA+15] Elise Deitrick, R Benjamin Shapiro, Matthew P Ahrens, Rebecca Fiebrink, Paul D Lehrman, and Saad Farooq. Using distributed cognition theory to analyze collaborative computer science learning. In Proceedings of the eleventh annual International Conference on International Computing Education Research, pages 51–60. ACM, 2015.

[Dwe14] Carol S Dweck. Mindsets and math/science achievement. 2014.

[dWGP16] Joost CF de Winter, Samuel D Gosling, and Jeff Potter. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. Psychological methods, 21(3):273, 2016.

[EHS15] Dana C Ernst, Angie Hodge, and Andrew Schultz. Enhancing proof writing via cross-institutional peer review. Primus, 25(2):121–130, 2015.

[EKN+11] Emma Enström, Gunnar Kreitz, Fredrik Niemelä, Pehr Söderman, and Viggo Kann. Five years with kattisâĂŤusing an automated assessment system in teaching. In 2011 Frontiers in Education Conference (FIE), pages T3J–1. IEEE, 2011.

[Ens14] Emma Enström. On difficult topics in theoretical computer science education. PhD thesis, KTH Royal Institute of Technology, 2014.

[Esp18] Molly Espey. Enhancing critical thinking using team-based learning. Higher Education Research & Development, 37(1):15–29, 2018.

[FHK+16] Ludwig Fahrmeir, Christian Heumann, Rita Künstler, Iris Pigeot, and Gerhard Tutz. Statistik: Der weg zur datenanalyse. Springer-Verlag, 2016.

[Fie13] Andy Field. Discovering statistics using IBM SPSS statistics. sage, 2013.

[Fin14] Aaron Finerman. University Education in Computing Science: Proceedings of a Conference on Graduate Academic and Related Research Programs in Computing Science, Held at the State University of New York at Stony Brook, June 1967. Academic Press, 2014.

[FK18] Christiane Frede and Maria Knobelsdorf. Exploring how students perform in a theory of computation course using final exam and homework assignments data. In Proceedings of the 2018 ACM Conference on International Computing Education Research, pages 241–249. ACM, 2018.

[FK21] Christiane Frede and Maria Knobelsdorf. A differentiated picture of student performance in introductory courses to theory of computation. Computer Science Education, 31(3):315–339, 2021.

[GET05] Judith Gal-Ezer and Mark Trakhtenbrot. Challenges in teaching the pumping lemma in automata theory course. ACM SIGCSE Bulletin, 37(3):369–369, 2005.

[GET06] Judith Gal-Ezer and Mark Trakhtenbrot. Algebraic characterization of regular languages: how to cope with all these equivalences? In Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, pages 325–325, 2006.

[GET16] Judith Gal-Ezer and Mark Trakhtenbrot. Identification and addressing reduction-related misconceptions. Computer Science Education, 26(2-3):89–103, 2016.

[GKL+02] Michael T Grinder, Seong B Kim, Teresa L Lutey, Rockford J Ross, and Kathleen F Walsh. Loving to learn theory: active learning modules for the theory of computing. In ACM SIGCSE Bulletin, volume 34, pages 371–375. ACM, 2002.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. Monatshefte für mathematik und physik, 38(1):173–198, 1931.

[Gok95] Anuradha A Gokhale. Collaborative learning enhances critical thinking. 1995.

[GOMSJVGP08] César García-Osorio, Iñigo Mediavilla-Sáiz, Javier Jimeno-Visitación, and Nicolás García-Pedrajas. Teaching push-down automata and turing machines. SIGCSE Bull., 40(3):316–316, June 2008.

[GVG00] Scott Grissom and Mark J Van Gorp. A practical approach to integrating active and collaborative learning into the introductory computer science curriculum. In Proceedings of the seventh annual CCSC Midwestern conference on Small colleges, pages 95–100, 2000.

[HA20] Geoffrey L Herman and Sushmita Azad. A comparison of peer instruction and collaborative problem solving in a computer architecture course. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education, pages 461–467, 2020.

[Ham04] Wilhelmiina Hamalainen. Problem-based learning of theoretical computer science. In Frontiers in Education, 2004. FIE 2004. 34th Annual, pages S1H–1. IEEE, 2004.

[Hei91] Bettina Heintz. Die Herrschaft der Regel: Zur Grundlagengeschichte des Computers. University of Zurich, 1991.

[HHK00] James Hollan, Edwin Hutchins, and David Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction (TOCHI), 7(2):174–196, 2000.

[HHP03] Margaret Hamilton, James Harland, and Lin Padgham. Experiences in teaching computing theory via aspects of problem-based learning. In Proceedings of the fifth Australasian conference on Computing education-Volume 20, pages 207–211. Australian Computer Society, Inc., 2003.

[HK04] Hashim Habiballa and Tibor Kmet'. Theoretical branches in teaching computer science. International Journal of Mathematical Education in Science and Technology, 35(6):829–841, 2004.

[HMU01] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. Acm Sigact News, 32(1):60–65, 2001.

[Hut95] Edwin Hutchins. Cognition in the Wild. MIT press, 1995.

[HW06] Michael Hielscher and Christian Wagenknecht. Atocc: learning environment for teaching theory of automata and formal languages. In ITiCSE, volume 6, pages 26–28, 2006.

[JHS16] Heisawn Jeong and Cindy E Hmelo-Silver. Seven affordances of computer-supported collaborative learning: How to support collaborative learning? how can technologies help? Educational Psychologist, 51(2):247–265, 2016.

[JTFoCCS13] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Association for Computing Machinery, New York, NY, USA, 2013.

[KAPG07] Laura Korte, Stuart Anderson, Helen Pain, and Judith Good. Learning by game-building: a novel approach to theoretical computer science education. In ACM SIGCSE Bulletin, volume 39, pages 53–57. ACM, 2007.

[KF16] Maria Knobelsdorf and Christiane Frede. Analyzing student practices in theory of computation in light of distributed cognition theory. In Proceedings of the 2016 ACM Conference on International Computing Education Research, ICER '16, pages 73–81, New York, NY, USA, 2016. ACM.

[KFBK17] Maria Knobelsdorf, Christiane Frede, Sebastian Böhne, and Christoph Kreitz. Theorem provers as a learning tool in theory of computation. In Proceedings of the 2017 ACM Conference on International Computing Education Research, ICER 2017, Tacoma, WA, USA, August 18-20, 2017, pages 83–92, 2017.

[KIP13] Antti Knutas, Jouni Ikonen, and Jari Porras. Communication patterns in collaborative software engineering courses: A case for computer-supported collaboration. In Proceedings of the 13th Koli Calling International Conference on Computing Education Research, pages 169–177, 2013.

[KKB14] Maria Knobelsdorf, Christoph Kreitz, and Sebastian Böhne. Teaching theoretical computer science using a cognitive apprenticeship approach. In Proceedings of the 45th ACM technical symposium on Computer science education, pages 67–72. ACM, 2014.

[Kle36] Stephen Cole Kleene. General recursive functions of natural numbers. Mathematische annalen, 112(1):727–742, 1936.

[Kno15] Maria Knobelsdorf. The theory behind theory-computer science education research through the lenses of situated learning. In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, pages 12–21. Springer, 2015.

[KPKJ11] Femke Kirschner, Fred Paas, Paul A Kirschner, and Jeroen Janssen. Differential effects of problem-solving demands on individual and collaborative learning outcomes. Learning and Instruction, 21(4):587–599, 2011.

[KSLF13] Kyoungna Kim, Priya Sharma, Susan M Land, and Kevin P Furlong. Effects of active learning on enhancing student critical thinking in an undergraduate general science course. Innovative Higher Education, 38(3):223–235, 2013.

[KTS14] Hubert Knoblauch, René Tuma, and Bernt Schnettler. Videography. Peter Lang, 2014.

[Lis11] Raymond Lister. Computing education research geek genes and bimodal grades. ACM Inroads, 1(3):16–17, 2011.

[May14] Philipp Mayring. Qualitative content analysis: theoretical foundation, basic procedures and software solution. 2014.

[May16] Philipp Mayring. Einführung in die qualitative Sozialforschung. Beltz, 2016.

[May20] Philipp Mayring. Qualitative inhaltsanalyse. In Handbuch qualitative Forschung in der Psychologie, pages 495–511. Springer, 2020.

[Mea34] George Herbert Mead. Mind, self and society, volume 111. Chicago University of Chicago Press., 1934.

[Mic] Michael Hielscher. Atocc from automaton to compiler construction.

[Mod10] Karl Moder. Alternatives to f-test in one way anova in case of heterogeneity of variances (a simulation study). Psychological Test and Assessment Modeling, 52(4):343–353, 2010.

[Moo94] Robert C Moore. Making the transition to formal proof. Educational Studies in mathematics, 27(3):249–266, 1994.

[Mor09] Stephan Morgenthaler. Exploratory data analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 1(1):33–44, 2009.

[MRFW+12] Juan Pablo Mejia-Ramos, Evan Fuller, Keith Weber, Kathryn Rhoads, and Aron Samkoff. An assessment model for proof comprehension in undergraduate mathematics. Educational Studies in Mathematics, 79(1):3–18, 2012.

[MU20] Ann Medaille and Janet Usinger. âĂIJthatâĂŹs going to be the hardest thing for meâĂİ: tensions experienced by quiet students during collaborative learning situations. Educational studies, 46(2):240–257, 2020.

[MWBF06] Charlie McDowell, Linda Werner, Heather E Bullock, and Julian Fernald. Pair programming improves student retention, confidence, and program quality. Communications of the ACM, 49(8):90–95, 2006.

[Nes14] Jan Nespor. Knowledge in motion: Space, time and curriculum in undergraduate physics and management. Routledge, 2014.

[PBCE19] Elizabeth Patitsas, Jesse Berlin, Michelle Craig, and Steve Easterbrook. Evidence that computer science grades are not bimodal. Communications of the ACM, 63(1):91–98, 2019.

[PBLS13] Leo Porter, Cynthia Bailey Lee, and Beth Simon. Halving fail rates using peer instruction: a study of four computer science courses. In Proceeding of the 44th ACM technical symposium on Computer science education, pages 177–182, 2013.

[PCG10] Robert A Powers, Cathleen Craviotto, and Richard M Grassl. Impact of proof validation on proof writing in abstract algebra.

International Journal of Mathematical Education in Science and Technology, 41(4):501–514, 2010.

[Pil10] Nelishia Pillay. Learning difficulties experienced by students in a course on formal languages and automata theory. ACM SIGCSE Bulletin, 41(4):48–52, 2010.

[PL14] Miranda Parker and Colleen Lewis. What makes big-o analysis difficult: understanding how students understand runtime analysis. Journal of Computing Sciences in Colleges, 29(4):164–174, 2014.

[Pos36] Emil L Post. Finite combinatory processesâĂŤformulation1. The journal of symbolic logic, 1(3):103–105, 1936.

[Pre05] David Preston. Pair programming as a model of collaborative learning: a review of the research. Journal of Computing Sciences in colleges, 20(4):39–45, 2005.

[RBFR06] Susan H. Rodger, Bart Bressler, Thomas Finley, and Stephen Reading. Turning automata theory into a hands-on course. SIGCSE Bull., 38(1):379–383, March 2006.

[RCH08] Louisa Remedios, David Clarke, and Lesleyanne Hawthorne. The silent participant in small group collaborative learning contexts. Active learning in higher education, 9(3):201–216, 2008.

[RFHN14a] Björn Rasch, Malte Friese, Wilhelm Hofmann, and Ewald Naumann. Quantitative Methoden 1: Einführung in die Statistik für Psychologen und Sozialwissenschaftler. Springer-Verlag, 2014.

[RFHN14b] Björn Rasch, Malte Friese, Wilhelm Hofmann, and Ewald Naumann. Quantitative Methoden 2. Springer, 2014.

[Rob15] Borut Robič. The foundations of computability theory. Springer, 2015.

[Ros92] Jeremy Roschelle. Learning by collaborating: Convergent conceptual change. The journal of the learning sciences, 2(3):235–276, 1992.

[RPB17] Fernando J Rodríguez, Kimberly Michelle Price, and Kristy Elizabeth Boyer. Exploring the pair programming process: Characteristics of effective collaboration. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pages 507–512, 2017.

[RT95] Jeremy Roschelle and Stephanie D Teasley. The construction of shared knowledge in collaborative problem solving. In Computer supported collaborative learning, pages 69–97. Springer, 1995.

[SABW11] Margret Selting, Peter Auer, and Dagmar Barth-Weingarten. A system for transcribing talk-in-interaction: Gat 2. 2011.

[Sac12] Lothar Sachs. Applied statistics: a handbook of techniques. Springer Science & Business Media, 2012.

[SBR15] Despina A Stylianou, Maria L Blanton, and Ourania Rotou. Undergraduate studentsâĂŹ understanding of proof: Relationships between proof conceptions, beliefs, and classroom experiences with learning proof. International Journal of Research in Undergraduate Mathematics Education, 1(1):91–134, 2015.

[Sig07] Scott Sigman. Engaging students in formal language theory and theory of computation. In ACM SIGCSE Bulletin, volume 39, pages 450–453. ACM, 2007.

[Sip12] Michael Sipser. Introduction to the Theory of Computation. Cengage Learning, 2012.

[SK07] Carsten Schulte and Maria Knobelsdorf. Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. In Proceedings of the third international workshop on Computing education research, pages 27–38, 2007.

[SM14] Thérèse Smith and Robert McCartney. Computer science students' concepts of proof by induction. Proceedings of the 14th Koli Calling International Conference on Computing Education Research, pages 51–60, 2014.

[SN04] Richard Sommer and Gregory Nuckols. A proof environment for teaching mathematics. Journal of Automated Reasoning, 32(3):227–258, 2004.

[SP13] Felix D Schönbrodt and Marco Perugini. At what sample size do correlations stabilize? Journal of Research in Personality, 47(5):609–612, 2013.

[TCC+01] Patrick T Terenzini, Alberto F Cabrera, Carol L Colbeck, John M Parente, and Stefani A Bjorklund. Collaborative learning vs. lecture/discussion: Students' reported learning gains. Journal of Engineering Education, 90(1):123–130, 2001.

[Ten19] Josh Tenenberg. Factors affecting free riding on teams: Implications for engineering education. International Journal of Engineering Education, 35(6A):1703, 2019.

[TK04] Konstantinos Tatsis and Eugenia Koleza. The effect of students' roles on the establishment of shared meaning during problem solving. International Group for the Psychology of Mathematics Education, 2004.

[TK14] Josh Tenenberg and Maria Knobelsdorf. Out of our minds: a review of sociocultural cognition theory. Computer Science Education, 24(1):1–24, 2014.

[Tra13] Mark Trakhtenbrot. Students misconceptions in analysis of algorithmic and computational complexity of problems. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education, pages 353–354, 2013.

[TS08] Matti Tedre and Erkki Sutinen. Three traditions of computing: What educators should know. Computer Science Education, 18(3):153–170, 2008.

[Tuk77] John W Tukey. Exploratory data analysis, volume 2. Reading, Mass., 1977.

[Tur37] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. Proceedings of the London mathematical society, 2(1):230–265, 1937.

[Ver05] Rakesh M Verma. A visual and interactive automata theory course emphasizing breadth of automata. In ACM SIGCSE Bulletin, volume 37, pages 325–329. ACM, 2005.

[Vyg80] Lev Semenovich Vygotsky. Mind in society: The development of higher psychological processes. Harvard university press, 1980.

[WD05] Michel Wermelinger and Artur Miguel Dias. A prolog toolkit for formal languages and automata. ACM SIGCSE Bulletin, 37(3):330–334, 2005.

[Web01] Keith Weber. Student difficulty in constructing proofs: The need for strategic knowledge. Educational studies in mathematics, 48(1):101–119, 2001.

[Wei01] Franz E Weinert. Concept of competence: A conceptual clarification. 2001.

[Zin08] Daniel Zingaro. Another approach for resisting student resistance to formal methods. ACM SIGCSE Bulletin, 40(4):56–57, 2008.

[Zin14] Daniel Zingaro. Peer instruction contributes to self-efficacy in cs1. In Proceedings of the 45th ACM technical symposium on Computer Science Education, pages 373–378, 2014.

[Zuk16] Olaf Zukunft. Empfehlungen für Bachelor-und Masterprogramme im Studienfach Informatik an Hochschulen (Juli 2016). Gesellschaft für Informatik eV, 2016.

Part V.

Appendix

List of Figures

List of Tables

List of Acronyms

ACM Association for Computing Machinery

AL Algorithms and Complexity

ANOVA Analysis of Variance

ADC Algorithms, Data structures, and Complexity

CL Collaborative Learning

CS Computer Science

DFA Deterministic Automaton

DS Discrete Structures

DCOG Distributed Cognition Theory

EDA Exploratorative Data Analysis

XML Extensive Markup Language

FLAT Formal Languages and Automata

GI Gesellschaft für Informatik e.V.

IEEE Institute of Electrical and Electronics Engineers

ITP Interactive Theorem Provers

NFA Non-deterministic Automaton

PL Pumping Lemma

PDA Pushdown Automaton

QCA Qualitative Content Analysis

SDF Software Development Fundamentals

ToC Theory of Computation

TM Turing machine

VIA Video Interaction Analysis

List of Publications

The following peer-reviewed manuscripts were published during the time of the doctorate:

- Christiane Frede and Maria Knobelsdorf. **A differentiated Picture of Student Performance in Introductory Courses to Theory of Computation**. In *Computer Science Education 31.3*, pages 315-339, 2021. Taylor & Francis.

C. Frede acquired the data, performed the evaluations, and wrote large parts of the manuscript.

- Christiane Frede and Maria Knobelsdorf. **Exploring how Students Perform in a Theory of Computation Course using Final Exam and Homework Assignment Data**. In *Proceedings of the 2018 ACM Conference on International Computing Education Research. ICER 2018, Espoo, Finland, August 13-15, 2018*, pages 241-249, 2018. ACM.

C. Frede acquired the data, performed the evaluations, and wrote large parts of the manuscript.

- Christiane Frede and Maria Knobelsdorf. **Explorative Datenanalyse der Studierendenperformance in der Theoretischen Informatik**. In *Hochschuldidaktik der Informatik - HDI 2018 - 8. Fachtagung des GI-Fachbereichs Informatik und Ausbildung/Didaktik der Informatik, Frankfurt, Germany, September 12-13, 2018*, 135-150, 2018. Universitätsverlag Potsdam.

C. Frede acquired the data, performed the evaluations, and wrote large parts of the manuscript.

- Maria Knobelsdorf, Christiane Frede, Sebastian Böhne, and Christoph Kreitz. **Theorem provers as a learning tool in theory of computation**. *In*

Proceedings of the 2017 ACM Conference on International Computing Education Research, ICER 2017, Tacoma, WA, USA, August 18-20, 2017, pages 83-92, 2017. ACM

C. Frede conducted the observational studies, acquired the data, evaluated the results, and wrote parts of the manuscript.

- Felix Kiehn, Christiane Frede and Maria Knobelsdorf. **Was macht Theoretische Informatik so schwierig? Ergebnisse einer qualitativen Einzelfallstudie**. In *47. Jahrestagung der Gesellschaft für Informatik, Informatik 2017, Chemnitz, Germany, September 25-29, 2017* volume P-275, of LNI, pages 267-278, 2017. GI

C. Frede supported the development of the research design and advised and supported the analysis.

- Maria Knobelsdorf and Christiane Frede, **Analyzing Student Practices in Theory of Computation in Light of Distributed Cognition Theory**. *In Proceedings of the 2016 ACM Conference on International Computing Education Research, ICER 2016, Melbourne, Australien, September 08-12, 2016*, pages 73-81, 2016. ACM

In the course of her master thesis, C. Frede conducted the observational studies, acquired the data, and evaluated the results.

Results of Statistical Calculations

In the following I have attached the unabridged tables with the results of the statistical calculations as obtained when exporting from SPSS Statistics (cf. Section 3.2.3.). More specifically, I give the following four results:

1. The descriptive statistics divided by year (tables with the heading *ONEWAY deskriptive Statistiken*). More precisely, this includes N = the number of cases (in this case the number of students), the mean, standard deviation, standard error, the lower and upper limit for the 95% confidence interval, as well as maximum and minimum.

2. Levene's test (tables with the heading *Test der Homogenität der Varianzen*). When p (*Signifikanz)* < .05, there is no homogeinity of variances.

3. Welch's ANOVA (tables with the heading *Robuste Testverfahren zur Prüfung auf Gleichheit der Mittelwerte*). When p < .001. there is a significant difference between the years.

4. Games Howell post-hoc tests (tables with the heading *Mehrfachvergleiche*). The values marked with an * indicate that a significant change to the positive or negative has taken place. The signs on the values then show the direction.

Final Exam Assignments

First, I provide the final exam assignments starting with Group A and continuing up to Group F. For example, "E1GA" means "Final Exam Assignment 1, Group A". As an example, I provide the results for Group A within this document. The results for the other groups are provided on the CD in the following order:

- Group A: Final Exam Assignments (E1GA - E6GA)
- Group B: Final Exam Assignments (E1GB - E6GB) (only on CD)
- Group C: Final Exam Assignments (E1GC - E6GC) (only on CD)
- Group D: Final Exam Assignments (E1GD - E6GD) (only on CD)
- Group F: Final Exam Assignments (E1GF - E6GF) (only on CD)

ONEWAY deskriptive Statistiken

		N	Mittelwert	Std.-Abweichung	Std.-Fehler	95%-Konfidenzinter. Untergrenze
E1GA	2016,00	81	92,6251	6,17748	,68639	91,2591
	2017,00	80	94,4408	4,97185	,55587	93,3344
	2018,00	82	93,6524	7,04601	,77810	92,1043
	Gesamt	243	93,5695	6,15254	,39469	92,7921
E2GA	2016,00	81	96,2192	4,98320	,55369	95,1173
	2017,00	80	92,5781	7,28012	,81394	90,9580
	2018,00	82	94,5935	5,89437	,65092	93,2984
	Gesamt	243	94,4719	6,27314	,40242	93,6792
E3GA	2016,00	81	95,9315	5,52880	,61431	94,7090
	2017,00	80	96,7330	4,90313	,54819	95,6418
	2018,00	82	97,0898	5,19091	,57324	95,9492
	Gesamt	243	96,5862	5,21617	,33462	95,9271
E4GA	2016,00	81	75,1997	14,69071	1,63230	71,9513
	2017,00	80	68,8603	16,83860	1,88261	65,1130
	2018,00	82	75,4017	15,83943	1,74917	71,9214
	Gesamt	243	73,1808	16,03390	1,02857	71,1547
E5GA	2016,00	81	89,0012	13,38523	1,48725	86,0414
	2017,00	80	83,9205	16,23565	1,81520	80,3074
	2018,00	82	88,2317	14,81099	1,63560	84,9774
	Gesamt	243	87,0688	14,95616	,95944	85,1789
E6GA	2016,00	81	76,7901	15,61200	1,73467	73,3380
	2017,00	80	86,2500	10,28941	1,15039	83,9602
	2018,00	81	79,6296	12,97074	1,44119	76,7616
	Gesamt	242	80,8678	13,68136	,87947	79,1353

ONEWAY deskriptive Statistiken

		95%-Konfidenzinterv...		
		Obergrenze	Minimum	Maximum
E1GA	2016,00	93,9910	71,05	100,00
	2017,00	95,5472	78,95	100,00
	2018,00	95,2006	75,00	100,00
	Gesamt	94,3470	71,05	100,00
E2GA	2016,00	97,3211	78,13	100,00
	2017,00	94,1982	68,75	100,00
	2018,00	95,8886	66,67	100,00
	Gesamt	95,2646	66,67	100,00
E3GA	2016,00	97,1541	68,18	100,00
	2017,00	97,8241	79,55	100,00
	2018,00	98,2304	68,18	100,00
	Gesamt	97,2454	68,18	100,00
E4GA	2016,00	78,4481	38,24	100,00
	2017,00	72,6075	20,59	100,00
	2018,00	78,8820	35,29	100,00
	Gesamt	75,2069	20,59	100,00
E5GA	2016,00	91,9609	45,45	100,00
	2017,00	87,5335	36,36	100,00
	2018,00	91,4860	35,00	100,00
	Gesamt	88,9588	35,00	100,00
E6GA	2016,00	80,2422	23,33	100,00
	2017,00	88,5398	50,00	100,00
	2018,00	82,4977	50,00	100,00
	Gesamt	82,6002	23,33	100,00

Test der Homogenität der Varianzen

		Levene-Statistik	df1	df2	Signifikanz
E1GA	Basiert auf dem Mittelwert	6,951	2	240	,001
	Basiert auf dem Median	4,234	2	240	,016
	Basierend auf dem Median und mit angepaßten df	4,234	2	220,701	,016
	Basiert auf dem getrimmten Mittel	6,163	2	240	,002
E2GA	Basiert auf dem Mittelwert	3,747	2	240	,025
	Basiert auf dem Median	2,720	2	240	,068
	Basierend auf dem Median und mit angepaßten df	2,720	2	220,402	,068
	Basiert auf dem getrimmten Mittel	3,505	2	240	,032
E3GA	Basiert auf dem Mittelwert	,239	2	240	,787
	Basiert auf dem Median	,251	2	240	,778
	Basierend auf dem Median und mit angepaßten df	,251	2	227,950	,778
	Basiert auf dem getrimmten Mittel	,345	2	240	,708
E4GA	Basiert auf dem Mittelwert	,512	2	240	,600
	Basiert auf dem Median	,512	2	240	,600
	Basierend auf dem Median und mit angepaßten df	,512	2	234,312	,600
	Basiert auf dem getrimmten Mittel	,506	2	240	,604
E5GA	Basiert auf dem Mittelwert	1,715	2	240	,182
	Basiert auf dem Median	1,813	2	240	,165
	Basierend auf dem Median und mit angepaßten df	1,813	2	231,190	,166
	Basiert auf dem getrimmten Mittel	2,119	2	240	,122
E6GA	Basiert auf dem Mittelwert	9,606	2	239	,000
	Basiert auf dem Median	7,411	2	239	,001
	Basierend auf dem Median und mit angepaßten df	7,411	2	214,479	,001
	Basiert auf dem getrimmten Mittel	9,204	2	239	,000

Robuste Testverfahren zur Prüfung auf Gleichheit der Mittelwerte

		Statistik[a]	df1	df2	Sig.
E1GA	Welch-Test	2,107	2	157,240	,125
E2GA	Welch-Test	6,988	2	156,188	,001
E3GA	Welch-Test	,981	2	159,704	,377
E4GA	Welch-Test	4,151	2	159,353	,017
E5GA	Welch-Test	2,535	2	158,820	,082
E6GA	Welch-Test	12,596	2	155,245	,000

a. Asymptotisch F-verteilt

Mehrfachvergleiche

Games-Howell

Abhängige Variable	(I) Jahr	(J) Jahr	Mittlere Differenz (I-J)	Std.-Fehler	Signifikanz
E1GA	2016,00	2017,00	-1,81571	,88324	,103
		2018,00	-1,02736	1,03758	,584
	2017,00	2016,00	1,81571	,88324	,103
		2018,00	,78835	,95626	,689
	2018,00	2016,00	1,02736	1,03758	,584
		2017,00	-,78835	,95626	,689
E2GA	2016,00	2017,00	3,64107*	,98442	,001
		2018,00	1,62570	,85456	,141
	2017,00	2016,00	-3,64107*	,98442	,001
		2018,00	-2,01537	1,04221	,133
	2018,00	2016,00	-1,62570	,85456	,141
		2017,00	2,01537	1,04221	,133
E3GA	2016,00	2017,00	-,80142	,82334	,595
		2018,00	-1,15826	,84023	,355
	2017,00	2016,00	,80142	,82334	,595
		2018,00	-,35685	,79317	,895
	2018,00	2016,00	1,15826	,84023	,355
		2017,00	,35685	,79317	,895
E4GA	2016,00	2017,00	6,33942	2,49171	,032
		2018,00	-,20201	2,39249	,996
	2017,00	2016,00	-6,33942	2,49171	,032
		2018,00	-6,54143	2,56979	,032
	2018,00	2016,00	,20201	2,39249	,996
		2017,00	6,54143	2,56979	,032
E5GA	2016,00	2017,00	5,08070	2,34667	,081
		2018,00	,76945	2,21068	,935
	2017,00	2016,00	-5,08070	2,34667	,081
		2018,00	-4,31125	2,44339	,185
	2018,00	2016,00	-,76945	2,21068	,935
		2017,00	4,31125	2,44339	,185
E6GA	2016,00	2017,00	-9,45988*	2,08146	,000
		2018,00	-2,83951	2,25524	,421
	2017,00	2016,00	9,45988*	2,08146	,000
		2018,00	6,62037*	1,84403	,001
	2018,00	2016,00	2,83951	2,25524	,421
		2017,00	-6,62037*	1,84403	,001

Mehrfachvergleiche

Games-Howell

Abhängige Variable	(I) Jahr	(J) Jahr	99%-Konfidenzintervall Untergrenze	99%-Konfidenzintervall Obergrenze
E1GA	2016,00	2017,00	-4,4280	,7965
		2018,00	-4,0943	2,0396
	2017,00	2016,00	-,7965	4,4280
		2018,00	-2,0419	3,6186
	2018,00	2016,00	-2,0396	4,0943
		2017,00	-3,6186	2,0419
E2GA	2016,00	2017,00	,7254	6,5567
		2018,00	-,9006	4,1520
	2017,00	2016,00	-6,5567	-,7254
		2018,00	-5,0981	1,0673
	2018,00	2016,00	-4,1520	,9006
		2017,00	-1,0673	5,0981
E3GA	2016,00	2017,00	-3,2355	1,6326
		2018,00	-3,6416	1,3251
	2017,00	2016,00	-1,6326	3,2355
		2018,00	-2,7011	1,9874
	2018,00	2016,00	-1,3251	3,6416
		2017,00	-1,9874	2,7011
E4GA	2016,00	2017,00	-1,0280	13,7068
		2018,00	-7,2729	6,8689
	2017,00	2016,00	-13,7068	1,0280
		2018,00	-14,1374	1,0546
	2018,00	2016,00	-6,8689	7,2729
		2017,00	-1,0546	14,1374
E5GA	2016,00	2017,00	-1,8598	12,0212
		2018,00	-5,7645	7,3034
	2017,00	2016,00	-12,0212	1,8598
		2018,00	-11,5342	2,9117
	2018,00	2016,00	-7,3034	5,7645
		2017,00	-2,9117	11,5342
E6GA	2016,00	2017,00	-15,6254	-3,2944
		2018,00	-9,5082	3,8292
	2017,00	2016,00	3,2944	15,6254
		2018,00	1,1661	12,0747
	2018,00	2016,00	-3,8292	9,5082
		2017,00	-12,0747	-1,1661

*. Die Differenz der Mittelwerte ist auf dem Niveau 0.01 signifikant.

Homework Assignments

Next, I provide the homework assignments starting with Group A, up to Group F, For example, the abbreviation "A1GA" was used, which means "Homework assignment 1, Group A". The results are provided on the CD in the following order:

- Group A: Homework Assignments (A1GA - A30GA) (only on CD)
- Group B: Homework Assignments (A1GB - A30GB) (only on CD)
- Group C: Homework Assignments (A1GC - A30GC) (only on CD)
- Group D: Homework Assignments (A1GD - A30GD) (only on CD)
- Group F: Homework Assignments (A1GF - A30GF) (only on CD)

Transcripts

In the following I have attached the original transcripts on the CD. I used the transcript rules as described in Section 6.2.3. In the raw transcript version, the participants were labeled P1 to PX according to their seating order from left to right. Only during the data analysis I renamed them according to their group (cf. section 7.4).

- Original transcript Group A (only on CD)
- Original transcript Group B (only on CD)
- Original transcript Group C (only on CD)

Other Documents and Information

In the following, I give further information and documents used before and during the studies.

- Email sent before the study
- Interview Guide
- Declarance of Independence / Consent form for the use of the video data

Email sent before the study

German Version: Liebe Studierende,

Ich bin Christiane Frede, Wissenschaftliche Mitarbeiterin an der Universität Hamburg. Ich promoviere in der Informatikdidaktik zu der Fragestellung, wie die Studierenden beim Erlernen der Theoretischen Informatik unterstützt werden können.

Um aber in diesem sehr wenig erforschten Feld aber voranzukommen und entscheidende Fortschritte zu machen, brauche ich eure Hilfe! Ich möchte gerne Beobachtungsstudien durchführen. Das bedeutet, dass ich eure Gruppen gerne bei der Bearbeitung einer Hausaufgabe beobachten möchte. Außerdem wird dies mit Audio- und Videoaufnahmen begleitet. ABER: Ich kann die Kameras so einstellen, dass man eure Gesichter nicht sieht und werde nirgendwo eure Namen oder Bilder von euch benutzen, auf denen ihr identifiziert werden könnt. Ihr werdet also von niemanden wieder erkennbar sein, falls ich einzelne Aufnahmen im Rahmen meiner Doktorarbeit nutze.

Vorweg sei gleich gesagt: Alle Informationen, die ich durch die Beobachtungsstudien erhalte, werden nicht in eure Benotung einfließen. Die Auswertung findet erst ab Frühjahr statt, also wenn dieses Modul schon abgeschlossen ist und wird komplett anonymisiert durchgeführt. Da diese Hausaufgaben die Phase darstellen, in denen ihr die Themen erlernt, geht es auch gar nicht darum, wie gut ihr die Aufgaben löst, sondern nur darum, den Lernprozess besser zu verstehen.

Was habt ihr davon? Dafür, dass ich euch beobachten darf (auch wenn das etwas seltsam klingt, ist es gar nicht so schlimm - Versprochen!) stellen wir euch einen Arbeitsraum am Campus zur Verfügung. Wir treffen uns dort und ihr könnt in Ruhe solange an der Aufgabe arbeiten, wie ihr möchtet oder bis ihr sie gelöst habt. Anschließend werde ich euch evtl. noch kurz interviewen und dann geht ihr einfach wieder. Ihr müsst also die Aufgaben nicht per Chat verteilen und alleine arbeiten oder euch irgendwo treffen, wo ihr vielleicht nicht die nötige Ruhe habt - so lernt ihr vielleicht auch etwas über euch selber und wie ihr am besten die theoretischen Themen lernen könnt.

Außerdem können diese Daten einen entscheidenden Hinweis darauf geben, wie wir die Lehre der Theoretische Informatik verbessern können, wovon ihr und alle nachfolgenden Studierenden profitieren werden!

Ich werde vom 08.-11.01.2019 in die Tutorien kommen und euch meine Idee erneut darstellen und auch für Fragen zur Verfügung stehen. Anschließend bekommt ihr die Möglichkeit euch in eine Tabelle einzutragen und daraufhin mit mir einen Termin zu suchen, der euch passt.

Bei weiteren Fragen könnt ihr euch gerne vor oder nach meinem Besuch in den Tutorien bei mir melden.

English Translation: Dear students,

I am Christiane Frede, a research associate at Universität Hamburg. I am doing my PhD in computer science education on the question of how students can be supported in learning Theory of Computation.

However, to make progress in this still very under-researched area, I need your help! I would like to conduct observational studies —that is, I would like to observe your groups working on a homework assignment. This will also be accompanied by audio and video recordings. BUT: I can set the cameras so that your faces will not be seen, and I will not use your names or pictures of you where you can be identified. So you will not be recognizable to anyone should I use individual shots as part of my dissertation.

Let me say this upfront: Any information I get from the observational studies will not be used in your assessment. The evaluation will not occur until spring — that is when this module has already been completed and will be completely anonymous. Since these homework assignments are the phase in which you learn the topics, it is not at all about how well you do the assignments but only about a better understanding of your learning processes.

What do you get out of it? In return for letting me observe you (even if that sounds a bit strange, it is not that bad - I promise!), I provide you with a workroom on campus. We will meet there, and you can work on the tasks as long as you like or until you have solved them. After that, I may have a short interview with you, and then you just leave. This way, you do not have to distribute the assignments via chat and work alone or meet somewhere where you may not have the peace and quiet you need — this way, you may also learn something about yourself and how to learn the theoretical topics best.

In addition, this data may provide a crucial indication of how we can improve the teaching of Theoretical Computer Science, which will benefit you and all subsequent students!

I will come to the tutorials from 08-11/01/2019, present my idea to you again, and be available for questions. Afterward, you will have the opportunity to sign up in a table and thereupon find a date with me that suits you.

If you have any further questions, please feel free to contact me before or after my visit to the tutorials.

Interview Guide

- Why did you solve something in a certain way?
- What exactly did you mean when you said ...?
- How did you come up with it? Have you seen similar proof elsewhere?
- Was this a typical course of events overall, or does it otherwise go differently? If so, how does it go otherwise?
- Do you usually work together physically as a group?
- Time for more details...
- Do you regularly attend the FLAT lecture?
- What materials do you use besides the lecture slides?
- How important is the tutorial for you? The lecture?
- How do you do in the mathematics modules? Better / worse than here?
- In which year are you?
- Which study program do you attend?
- Did you have computer science in school? If yes, to what extent and which topics were taught?
- Which advanced courses or courses with higher requirements did you take?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Fakultät für
Mathematik, Informatik
und Naturwissenschaften

Einverständniserklärung

HIERMIT ERKLÄRE ICH MICH EINVERSTANDEN, MICH IM RAHMEN EINER EMPIRISCHEN STUDIE VON CHRISTIANE FREDE BEIM BEARBEITEN VON ÜBUNGSAUFGABEN AUS DER VERANSTALTUNG „FORMALE SPRACHEN UND ALGORITHMEN", WINTERSEMESTER 2018/2019, BEOBACHTEN UND IM ANSCHLUSS INTERVIEWEN ZU LASSEN UND DIES DURCH AUDIO- SOWIE KAMERAAUFNAHMEN (SO WEIT WIE MÖGLICH WIRD AUF DAS FILMEN VON GESICHTERN VERZICHTET) BEGLEITEN ZU LASSEN.

ICH BIN DAMIT EINVERSTANDEN, DASS DIE AUDIOAUFNAHMEN FÜR EINE TRANSKRIPTION VERWENDET WERDEN (ÜBERTRAGEN DER AUDIOAUFNAHME IN EINE SCHRIFTLICHE FORM) UND DIE AUFNAHMEN NACH BEENDIGUNG DER STUDIE GELÖSCHT WERDEN, WOBEI DIE ANONYMISIERTE TRANSKRIPTION ARCHIVIERT WIRD.

ICH BIN DAMIT EINVERSTANDEN, DASS MITARBEITENDE DER STUDIE MIT DER TRANSKRIPTION DER AUDIOAUFNAHME SOWIE DEN KAMERAAUFNAHMEN AUSSCHLIESSLICH IN ANONYMISIERTER FORM ARBEITEN WERDEN.

ICH BIN DAMIT EINVERSTANDEN, DASS AUSSCHNITTE AUS DER TRANSKRIPTION UND DEN KAMERAAUFNAHMEN IM RAHMEN DER DOKTORARBEIT VON CHRISTIANE FREDE SOWIE MÖGLICHEN WEITEREN PUBLIKATIONEN (Z.B. KONFERENZ- ODER ZEITSCHRIFTEN-ARTIKEL), EINER DIDAKTISCHEN FORTBILDUNG ODER EINEM VORTRAG (Z.B. KONFERENZVORTRAG, FORTBILDUNG, ETC.) VERÖFFENTLICHT WERDEN, WOBEI NUR AUSSCHNITTE VERWENDET WERDEN, AUS DENEN NICHT AUF MEINE IDENTITÄT GESCHLOSSEN WERDEN KANN.

VORNAME:

NACHNAME:

DATUM, UNTERSCHRIFT

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, den 29.11.2021

Vorname, Nachname