

Was macht Theoretische Informatik so schwierig? Ergebnisse einer qualitativen Einzelfallstudie

Felix Kiehn¹, Christiane Frede², Maria Knobelsdorf³

Abstract: Dieser Artikel stellt eine qualitative Einzelfallstudie vor, die als nicht-teilnehmende Beobachtung gestaltet war und in der untersucht wurde, wie drei Informatik Bachelorstudierende einen Reduktionsbeweis zu Unentscheidbarkeit mit Turing-Maschinen entwickeln. Für die Datenanalyse wurde eine angepasste Version der Interaktionsanalyse sowie qualitative Inhaltsanalyse eingesetzt. Die beobachteten Studierenden hatten zwar das grundsätzliche Konzept eines Reduktionsbeweises verstanden, konnten diesen jedoch nicht im Detail umsetzen sowie die gemeinsam besprochenen Ergebnisse und Ansätze formal korrekt niederschreiben. Die Ergebnisse geben damit einen detaillierten Einblick, wie sich Bachelorstudierende der Informatik mit einem Teilthema der Theoretischen Informatik auseinandersetzen und welche Schwierigkeiten sie dabei haben. Dieses Ergebnis zeigt, dass häufig genutzte Annahmen über die Gründe, warum Studierende Schwierigkeiten mit Theoretischer Informatik haben, wie z.B. ein mangelndes Interesse durch das abstrakte und theoretische Erscheinungsbild dieser Fachrichtung, hinterfragt und neu untersucht werden müssen.

Keywords: Qualitative Einzelfallstudie, nicht-teilnehmende Beobachtung, qualitative Inhaltsanalyse, Interaktionsanalyse, Theoretische Informatik, Turing-Maschinen, Reduktionsbeweis.

1 Einleitung

In der Studieneingangsphase der Informatik-Bachelorstudiengänge in Deutschland stellen die mathematischen und theoretischen Fachanforderungen in den korrespondierenden Einführungsveranstaltungen nach wie vor eine große Herausforderung dar. So brachen 45% der Studierenden des Jahrgangs 2010/11 ihr Studium u.a. aufgrund von Leistungsproblemen ab [He17]. Auch international ist das Thema relevant und die Situation ähnlich wie in Deutschland. Hier finden sich für die Eingangslehre der Theoretischen Informatik verschiedene fachdidaktische Ansätze, die sich dieser Problematik annehmen, z.B. [CGM04, Hä04, Ko07, Si07]. Die Ansätze arbeiten mit teils sehr unterschiedlichen didaktischen Konzepten und Methoden, sind jedoch sehr erfolgreich in der Gestaltung einer konstruktivistischen Lehr-Lern-Umgebung. Insgesamt liegt allen Ansätzen direkt oder implizit die Annahme zugrunde, dass die Studierenden

¹ Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, kiehn@informatik.uni-hamburg.de

² Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, frede@informatik.uni-hamburg.de

³ Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, knobelsdorf@informatik.uni-hamburg.de

wenig intrinsisch motiviert sind, sich mit der Theoretischen Informatik auseinander zu setzen und zudem Schwierigkeiten haben, die Fachkonzepte kognitiv zu durchdringen. Diese Annahmen wurden bisher jedoch kaum empirisch untersucht. Wenig ist darüber bekannt, welche Schwierigkeiten die Studierenden tatsächlich haben und welche sich dabei signifikant auf den Lernerfolg negativ auswirken. Als eine der wenigen Ausnahmen haben Parker und Lewis (2014) in einer qualitativen Einzelfallstudie einen Studenten beobachtet und interviewt, wie er eine Aufgabe zu Laufzeitanalyse mit der O-Notation bearbeitet [PL14]. Hierbei stellen die Autorinnen fest, dass der Student vor allem Probleme mit Eigenschaften mathematischer Funktionen, jedoch weniger mit der Laufzeitanalyse selbst, hatte. Auch in einer von uns durchgeführten Studie haben mehrere Studierendengruppen bei der Bearbeitung einer Aufgabe zu NP-Vollständigkeit beobachtet [KF16]. Die Ergebnisse unterstreichen die Probleme der Studierenden mit mathematischen Grundlagen und mit der Entwicklung von Beweiskompetenzen anstelle des generellen Verständnisses der Konzepte.

In einer typischen Einführungsveranstaltung in die Theoretische Informatik stehen Lehrenden eine Vorlesung und betreute Übungsgruppen zur Verfügung, um die Studierenden zu unterstützen. Zusätzlich gibt es durch wöchentliche Übungsaufgaben, die die Studierenden bearbeiten und in der Regel auch schriftlich abgeben müssen, die Möglichkeit den Leistungsstand der Studierenden nachzuverfolgen. Die Klausurergebnisse am Ende einer Veranstaltung leisten einen zusätzlichen Einblick, inwiefern spezifische Fachkompetenzen entwickelt wurden. Aus diesen Datenquellen lässt sich jedoch nur bedingt schließen, wie der Bearbeitungsprozess der entsprechenden Übungsaufgaben in den selbstorganisierten Studierendengruppen selbst erfolgt ist und insbesondere wie es zu fehlerhaften Ergebnissen kommt oder warum z.B. ganze Aufgaben ausgelassen wurden. Qualitative Einzelfallstudien wie die oben genannte Arbeit von Parker und Lewis (2014) geben hier einen detaillierten Einblick, wie sich Lernende mit einem Thema im Verlauf ihres Lernprozesses auseinandersetzen und tragen so zu einer empirisch fundierten Hypothesenbildung über konkrete Schwierigkeiten der Studierenden bei. Aufbauend auf der von uns durchgeführten Studie zu NP-Vollständigkeit (2016) wurde das gleiche Forschungsdesign verwendet, um eine weitere Beweisaufgabe zu einem anderen Themenbereich der Theoretischen Informatik zu untersuchen. In der durchgeführten Einzelfallstudie wurde beobachtet, wie drei Informatik-Bachelorstudierende der Universität Hamburg einen Reduktionsbeweis zu Unentscheidbarkeit mit Turing-Maschinen entwickeln. Für die Datenanalyse wurde eine angepasste Version der Interaktionsanalyse nach Krummheuer [Kr12] sowie qualitative Inhaltsanalyse nach Mayring [Ma10] eingesetzt.

In den nächsten beiden Abschnitten stellen wir das Forschungsdesign der Studie und die konkrete Aufgabe mit Musterlösung vor, welche die Studierenden im Verlauf der Datenerhebung bearbeitet haben. Die Ergebnisse der Datenauswertung werden in Abschnitt 4 vorgestellt und diskutiert. Der Artikel schließt mit einem Fazit in Abschnitt 6.

2 Forschungsdesign

Ziel der hier vorliegenden Studie war es, zu verstehen, warum Theoretische Informatik für die Studierenden so schwierig ist. Zur Beantwortung dieser Frage wurden Beweisaufgaben aus der Theoretischen Informatik betrachtet, welche Teil vieler Aufgaben in der Theoretischen Informatik sind. Im Rahmen einer Einzelfallstudie wurden drei Informatik-Bachelorstudierende aus der Veranstaltung „Formale Grundlagen der Informatik I“ (FGI1) der Universität Hamburg nicht-teilnehmend dabei beobachtet [Ba09], wie sie einen Reduktionsbeweis zu Unentscheidbarkeit mit Turing-Maschinen entwickeln. Durch die Gruppenarbeit fand in der Regel ein sog. *lautes Denken* statt, das eine nicht-teilnehmende Beobachtung sehr gut ermöglicht. Die Arbeitssitzung der Studierenden wurde von uns beobachtet, ergänzend protokolliert und auf Audio aufgenommen. Außerdem wurden Fotos der Mitschriften und der Arbeit am Whiteboard gemacht. Videografie war am Standort nicht möglich, da die Studierenden an einer Studie mit Videoaufnahmen nicht teilnehmen wollten. Am Ende der Beobachtung wurde die Gruppe ergänzend zur Sitzung im Rahmen eines Leitfaden-Interviews befragt.

Die Audiodaten wurden transkribiert und zunächst mit qualitativer Inhaltsanalyse nach Mayring [Ma10] kodiert, wobei wir hierfür das Kategoriensystem aus unserer ersten Studie genutzt haben [KF16]. Für die weitere Interpretation der Daten wurde die Interaktionsanalyse nach Krummheuer angewendet [Kr12] und für die Kombination mit Mayring angepasst. Nach vorab definierten Kriterien und mittels des kodierten Materials wurden sieben ausgewählte Stellen gezielt für die Analyse ausgewählt [Ki17].

Die Interaktionsanalyse nach Krummheuer besteht aus fünf Schritten, die den Interpretationsvorgang strukturieren. Die Quintessenz der Methode liegt im dritten und vierten Schritt: In Schritt drei erfolgt eine ausführliche Analyse der Einzeläußerungen. Dabei wird jede Einzelaussage gemäß ihrer sequentiellen Ordnung analysiert. Besonderes Augenmerk liegt hier in der Erarbeitung unterschiedlicher Sichtweisen durch das Erstellen verschiedener Deutungsalternativen durch den Forschenden. Hierfür obliegt es ihm sich von eigenen ersten Alltagsinterpretationen, z.B. durch gedankliche Kontextvariationen, zu lösen. Die Kategorisierung nach Mayring hilft durch die erarbeiteten Kenntnisse über den untersuchten Gruppenprozess dabei, die Deutungsalternativen sowohl zu erstellen als auch zu diskutieren und so den Grad an Subjektivität beim Deuten zu verringern. Die erstellten Deutungsalternativen werden in Schritt vier im Kontext der darauffolgenden Handlung betrachtet (vgl. [Kr12] S. 3). Diesen Schritt bezeichnet Krummheuer als Turn-by-Turn-Analyse, da die vorher erstellten Deutungsalternativen sich im Kontext der darauffolgenden Handlung, Gesprächszug um Gesprächszug bewähren müssen. Eine vorausgesagte Handlung bzw. Deutung kann als Indiz für eine korrekte Deutungsalternative gesehen werden. Durch die Turn-by-Turn-Analyse können Alternativen auch wegfallen oder gänzlich neue notwendig werden. In Tab. 1 findet sich ein Ausschnitt unserer Datenauswertung, der auch das Vorgehen nach der Interaktionsanalyse illustriert.

| | |
|---|---|
| <i>S3: „Vielleicht ist das so eine tricky Aufgabe und es ist eigentlich unentscheidbar.“</i> | |
| <i>Alternative 1: S3 vermutet eine Fangfrage, da er die Aufgabe als schwierig empfindet und kommt zum Schluss, dass die Aufgabe nicht gemäß der Aufgabenstellung zu lösen ist.</i> | <i>Alternative 2: S3 macht eine ironische Aussage, welche er nicht ernst meint.</i> |
| <i>Turn-by-Turn-Analyse: S3s Stimme am Ende der Aussage klingt bei der Aufnahme belustigt. Gefolgt wird diese heitere Stimmung von einem Lachen von der darauffolgenden Aussage von S1. Diese Interaktion spricht für die zweite Deutungsalternative. Ein weiteres Indiz dafür ist, dass S3 bereits vorher in der Sitzung ironische Aussagen gemacht hat.</i> | |

Tab. 1: Beispiel für eine Interaktionsanalyse aus der eigenen Datenauswertung

3 Aufgabe und Musterlösung

In der ausgewählten Übungsaufgabe geht es darum, die Unentscheidbarkeit einer gegebenen Sprache L mit Hilfe von Turing-Maschinen (TM) per Widerspruch und mit Hilfe einer Reduktion zu beweisen. Konkret lautet die Aufgabe:

Sei $L = \{ \langle M \rangle \mid M \text{ ist eine TM und } L(M) \text{ ist endlich} \}$. Zeigen Sie, dass L unentscheidbar ist.

Bevor diese Aufgabe gestellt wurde, wurden an zwei Vorlesungsterminen alle für die Aufgabe relevanten Konzepte vorgestellt, wie z.B. Turing-Maschinen, Reduktion, Sprachklassen und Entscheidbarkeit. Das Vorlesungsmaterial basierte dabei auf Hopcroft et al. (2001) [HMU01], welches auch als Literatur zur Vorlesung empfohlen wurde. Vor der Teilnahme an diesem Modul haben zwei der Studierenden das einführende Mathematikmodul „Diskrete Mathematik“ gehört und der dritte Student „Mathematik für Physiker I“ und „Formale Grundlagen der Informatik II“⁴. Im Folgenden stellen wir eine Musterlösung für die Aufgabe vor.

Es wird angenommen, dass die Sprache L entscheidbar sei. Dann muss eine TM M_L existieren, die L entscheidet. Für die Reduktion, also die Konvertierung von Probleminstanzen eines bekannten, unentscheidbaren Problems in Instanzen von L , wird eine Konstruktion beschrieben, die das leistet. Aus einer beliebigen TM A und einem beliebigen Wort w wird eine TM $M_{A,w}$ konstruiert, die A auf w simuliert, wenn sie mit einem Wort aus der Menge $\{a\}^*$ auf dem Band gestartet wird. Wenn A hält, wechselt $M_{A,w}$ in einen akzeptierenden Zustand. Wenn A nicht hält, simuliert $M_{A,w}$ weiter und hält nicht. Wenn das Wort auf dem Band nicht Element der Menge $\{a\}^*$ ist, wechselt $M_{A,w}$ in einem nichtakzeptierenden Zustand und hält. Ist die TM $M_{A,w}$ so konstruiert, akzeptiert sie die Sprache $L(M_{A,w}) = \{a\}^*$ genau dann, wenn A auf w hält. Wenn A auf w nicht hält, ist $L(M_{A,w})$ leer und nur in diesem Fall endlich. Sei nun M_H eine TM, die das Halteproblem

⁴ FGI2 hat wenig Überschneidungen mit FGI1 und behandelt Parallelität und Nebenläufigkeit

entscheiden soll. Falls $M_H \langle A, w \rangle$ als Input erhält, konstruiert M_H die beschriebene TM $M_{\langle A, w \rangle}$. Die erstellte Maschine wird dann als Input an M_L übergeben. Abb. 1 illustriert die beschriebene Konstruktion. Aus dem Input A, w von M_H (links) wird die $M_{\langle A, w \rangle}$ konstruiert und dient als Input der TM M_L . Es liegen nun die folgenden beiden Fälle vor: Wenn M_L akzeptiert, lehnt M_H ab, und wenn M_L ablehnt, akzeptiert M_H . Durch diese Konstruktion mit M_L , M_H und $M_{\langle A, w \rangle}$ entscheidet M_H das Halteproblem, da M_H immer hält und akzeptiert, wenn $L(M_{\langle A, w \rangle}) = \{a\}^*$. Die Bedingung dafür ist, dass A auf w halten muss.

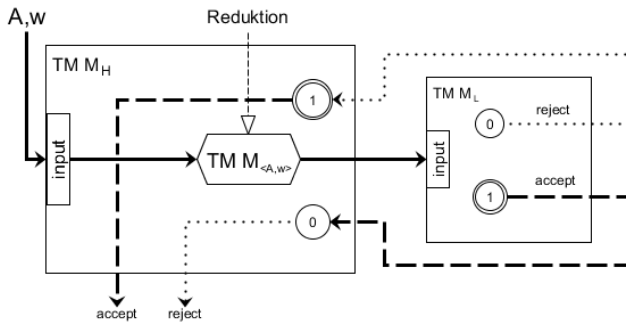


Abb. 1: Darstellung der Turing-Maschinen für den Reduktionsbeweis, die gestrichelten Linien stellen einen akzeptierenden Output und die gepunkteten Linien einen ablehnenden Output dar.

Die beschriebene Konstruktion ist so in der Lage das Halteproblem zu entscheiden. Da aber bewiesen ist, dass das Halteproblem unentscheidbar ist, kann eine solche Konstruktion eigentlich nicht existieren. Unsere Annahme, dass L entscheidbar ist, muss entsprechend falsch sein und L ist somit unentscheidbar. \square

4 Ergebnisse

Um das Prozesshafte der einzelnen, kategorisierten Textstellen in ihrem Zusammenwirken zu visualisieren, haben wir deren prozentualen Anteil pro Zeiteinheit von fünf Minuten mit einem Balkendiagramm visualisiert (siehe Abb. 2). Jeder Balken stellt dabei eine feste Zeiteinheit dar und die Balkenanordnung von links nach rechts die chronologische Abfolge der Arbeitssitzung. Das Tortendiagramm bezieht sich auf den zeitlichen Umfang, in welchem sich kategorisierte Tätigkeiten in der gesamten Sitzung darstellten. Diese Diagramme geben einen Überblick über die konstant wechselnden Aktivitäten eines Gruppenarbeitsprozesses. So verbringen die Studierenden mehr als 50% der Gesamtzeit damit, die Aufgabe, Definitionen, und mögliche Lösungsansätze zu klären und erst spät (ca. ab Minute 35) gehen sie dazu über, eine Lösung zu entwickeln und aufzuschreiben.

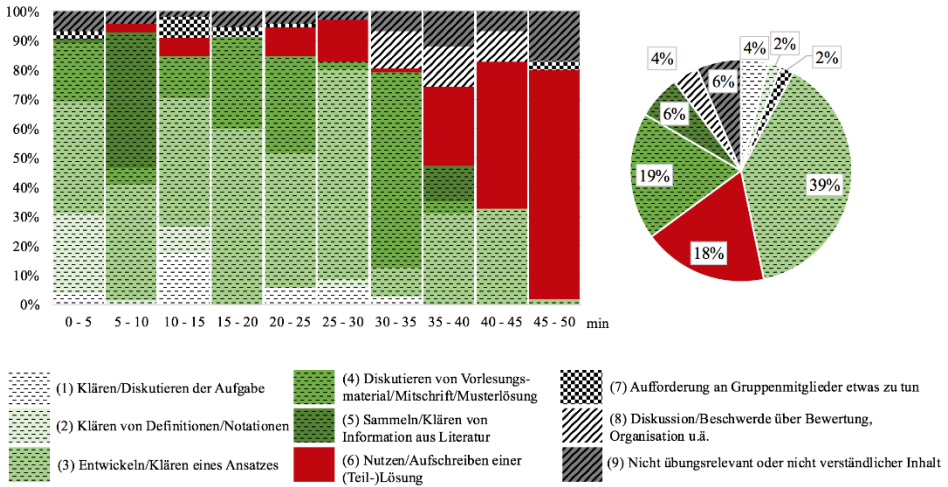


Abb. 2: Prozentualer Gesamtanteil an Kategorien im Verlauf der Gruppenarbeit (links), Prozentualer Anteil während der gesamten Gruppenarbeit (rechts)

Im Folgenden wird der Ablauf der Studie zusammenfassend nach den primär auftretenden Kategorien beschrieben. Letztere sind durchnummeriert in Abb. 2 aufgelistet, auf welche sich im Weiteren durch ihre korrespondierende Nummer in Klammern bezogen wird.

Minute 0 – 5: Die Studierenden versuchen vorrangig das Konzept der Turing-Maschine und Entscheidbarkeit zu verstehen (2) und nachzuvollziehen wie die Turing-Maschinen aus der Aufgabenstellung funktionieren soll (3). Dafür nutzen sie ihre Mitschrift aus der Übung und versuchen sie auf die neue Aufgabe anzuwenden (4). Am Ende der ersten fünf Minuten sind sie sich einig, dass sie eine angenommene Entscheidbarkeit aus der Aufgabenstellung zum Widerspruch führen müssen, sind sich aber nicht ganz sicher wie sie dabei vorgehen sollen. **Minute 5 – 10:** Ein Großteil der nächsten Zeit wird darauf verwendet ein Beispiel aus einem Buch zu besprechen, das Unentscheidbarkeit mit Hilfe eines Paradoxons beschreibt (5). Die Studierenden machen ihre Unschlüssigkeit darüber deutlich, ob ihnen dieses Beispiel dabei weiterhilft, Entscheidbarkeit richtig zu verstehen. Die Studierenden diskutieren darüber, wie die Turing-Maschinen verschachtelt werden sollen und wie genau ihre Funktionsweisen bzw. Eingaben sind (3). **Minute 10 – 15:** Auf der Suche nach der Lösung, beginnen die Studierenden darüber zu diskutieren, wie sie das Verfahren in der Übung verstanden haben, aber dass es hier nicht anwendbar zu sein scheint (4). Sie können nicht mehr nachvollziehen, wieso sie in der Übung das Halteproblem verwendet haben und zu einem Widerspruchsbeweis gekommen sind (4). Die Studierenden beschließen erst einmal einen Schritt zurück zu gehen und prüfen jetzt, ob sie die Aufgabenstellung korrekt verstanden haben (1). Weiterhin versuchen sie eine Definition des Begriffs Entscheidbarkeit zu finden. Um sich zu versichern, was sie schon wissen, definieren sie Entscheidbarkeit an dem Konzept der regulären Sprachen (2). Ihre Feststellung, dass sie eine Turing-Maschine haben, die eine Sprache akzeptiert und es eine

andere Turing-Maschine gibt, die eben diese Turing-Maschine akzeptiert (3), halten sie am Whiteboard fest (7). **Minute 15 – 30:** Die Studierende diskutieren in dieser Zeit ausführlich ihre Notizen aus der Übung und versuchen weiterhin die Lösung aus der Übung zu verstehen. Sie glauben, dass die Funktionsweisen ihrer Turing-Maschinen das unlösbare Halteproblem lösen (4). Um diesen Ansatz nachzuvollziehen, nutzen sie das Whiteboard und verdeutlichen hieran die Ein- und Ausgaben ihrer Turing-Maschinen (3). **Minute 30 – 35:** Sie vergleichen ihren Ansatz mit der Lösung aus der Übung, um zu verstehen, ob ihre vorher gewählte Begründung bezüglich der Lösung des Halteproblems genügen würde (4). Anschließend diskutieren die Studierenden, wie sie ihre Beweisidee richtig verschriftlichen könnten, wobei sie auf die online verfügbare Musterlösung der Präsenzaufgabe zurückgreifen, da die Lösung in der Übung für sie im Nachhinein nicht verständlich aufgeschrieben war (4). Als die Studierenden die Musterlösung finden, stellt sich heraus, dass einer der Studierenden seit Beginn die Mitschrift zu einer anderen (in diesem Fall eine ähnliche, aber nicht passende) Aufgabe angeschaut hat als seine Mitstudierenden (4). **Minute 35 – 40:** Die Studierenden versuchen die Musterlösung nachzuvollziehen und sind sich ihrer eigenen Lösung dabei relativ sicher, so dass sie diese verschriftlichen wollen (6). Sie wollen es anschließend nicht so aufschreiben wie der Übungsgruppenleiter, da sie der Meinung sind, dass es den Anforderungen des Korrigierenden nicht genügt (6). Sie diskutieren, ob sie mehr Text oder die Zeichnung hinzufügen sollten, meinen aber, dass auch das dem Übungsgruppenleiter nicht reichen wird (8). **Minute 40 – 50:** Sie stellen fest, dass sie nicht genau wissen, wie formal sie beim Aufschreiben sein müssen (8). Letztendlich entscheiden sie, dass ihre Lösung nicht schlechter sei das, was sie aus der Übung mitgenommen haben und ihr "Grundgerüst" richtig sei (6). Sie wollen die Zeichnung zusammen mit einem erläuternden Text abgeben, sind sich mit der schriftlichen Art der Lösung aber noch nicht ganz einig (6). Schließlich beenden sie die Sitzung und einer der Studierenden ist nicht ganz zufrieden, möchte aber die Tage weiter darüber nachdenken.

Um den Arbeitsprozess der Gruppe verstehen und einordnen zu können, wurde zusätzlich ihr Vorgehen während der Studiensitzung und ihr Lösungsvorschlag mit der Musterlösung aus Abschnitt 3 verglichen. Darauf wollen wir nun genauer eingehen. Binnen der ersten fünf Minuten haben sich die Studierenden gemäß der Methodik aus Vorlesung und Übung für einen Widerspruchsbeweis mit einer Reduktion entschieden. Sie treffen auch die für den Widerspruchsbeweis wichtige Annahme, dass eine Turing-Maschine als entscheidbar angenommen werden muss. Allerdings nimmt nur ein Student das Problem L aus der Aufgabenstellung als entscheidbar an, während die übrigen Studierenden erstmal andere Turing-Maschinen als entscheidbar annehmen. In Minute 5-10 entnehmen die Studierenden der Präsenzaufgabenlösung die Idee, zwei Turing-Maschinen zu verknüpfen. Hier sind sie mit der Musterlösung konform. Die Verknüpfung besteht allerdings darin, dass eine Turing-Maschine die Eingabe für die andere ist. Dass die Turing-Maschine für L gemäß Musterlösung als Unteroutine für die Turing-Maschine eines unentscheidbaren Problems, wie z.B. des Halteproblems, dienen soll, fehlt hier allerdings. In Minute 10-15 wollen die Studierenden für ihren Beweis das Halteproblem nutzen, von dem sie wissen, dass es unentscheidbar ist. Aber erst im Zeitblock zwischen

Minute 15 und 30 gelangen sie zu einem Punkt, an dem sie glauben das eigentlich unentscheidbare Halteproblem mit ihrer Annahme und ihrem Beweis zu lösen und somit einen validen Widerspruch zu haben. Ihr Lösungsansatz definiert eine Turing-Maschine M,w , die als Eingabe $L(M)$ bekommt und 1 beim Anhalten und 0 beim Nichtanhalten ausgibt, wobei hierbei weiter simuliert werden soll. M,w dient dann als Input für die Turing-Maschine H (Halteproblem), welche dann entweder 1 (entscheidbar) und 0 (nicht entscheidbar) ausgibt. Die Studierenden liefern in ihrem Lösungsvorschlag allerdings keinen wirklichen Reduktionsbeweis. So kommt das eigentliche Problem L nur im Lösungsvorschlag von $P1$ vor (und das eigentlich nur in der Annahme) und die Turing-Maschinen, die sie verwenden, sind nur grob grafisch umrissen. Bei ihnen werden nur Namen, Input und Ausgabeverhalten definiert. Eine Reduktion, also die Umwandlung von Probleminstanzen von einer Turing-Maschine für L zu einer Turing-Maschine für das Halteproblem, fehlt völlig. Den Widerspruch verorten die Studierenden in der Tatsache, dass ihre TM M,w nun das Halteproblem löst. Den Widerspruch beim Halteproblem zu vermuten ist richtig und so auch in der Musterlösung zu finden. Allerdings wird aufgrund mangelnder Erklärung der Turing-Maschinen und ihrer Funktionsweise nicht klar, warum die Turing-Maschine dies tun soll. Durch die fehlende Reduktion fehlt zudem eine korrekte Verknüpfung zwischen den Maschinen. Die Studierenden weichen, bis auf ihre Absichtserklärung L als entscheidbar anzunehmen und einen Widerspruchsbeweis mit L und H führen zu wollen, stark von der Musterlösung ab. Ausgehend von den präsentierten Lösungsvorschlägen wäre dies durch einen Tutor wohl mit 0–0,5 von 4 möglichen Punkten bewertet worden.

5 Diskussion

Die Interaktionsanalyse ergab insgesamt fünf Kernbeobachtungen, die die Arbeitssitzung der Studierenden besonders charakterisieren. Diese wollen wir nun zusammenfassend vorstellen.

Verständnisschwierigkeiten hinsichtlich Turing-Maschinen: Verschiedene Indizien im Verlauf der Sitzung deuten darauf hin, dass die Studierenden das Konzept der Turing-Maschine und ihre Eigenschaften nicht vollständig durchdrungen haben. Dies zeigt sich zuerst in Minute 5-10, wenn die Studierenden diskutieren, wie die Turing-Maschinen in ihrem Beweisansatz interagieren sollen. Die Studierenden erkennen dabei nicht, dass die Akzeptanz einer Turing-Maschine nicht das Gleiche ist, wie ihr Output auf Maschinenband. So verwechseln sie die Ausgabe 0 der Turing-Maschine für Nichtakzeptanz mit der Ausgabe 0 als auf dem Band geschriebenes Ergebnis. Zudem versuchen die Studierenden für ihren Reduktionsbeweis nicht einen Reduktionsalgorithmus zu finden, sondern beschäftigen sich primär damit, wie sie zwei Turing-Maschinen (eine für das Halteproblem und eine für ihr neues Problem L) verknüpfen können. Die beiden von ihnen skizzierten Turing-Maschinen betrachten sie nur oberflächlich, indem sie ihnen eine Bezeichnung geben und Input und Output definieren. Wie die Maschinen intern funktionieren und wie ihr Laufzeitverhalten

aussieht, wird nicht besprochen oder aufgeschrieben.

Defizite im Erstellen formal korrekter Beweise: Sowohl in den mündlichen Formulierungen als auch in ihren schriftlichen Notizen, nutzen die Studierenden kaum mathematische Sprache, um einen formal korrekten Beweis zu entwickeln. Die Studierenden behelfen sich stattdessen mit umgangssprachlichen Beschreibungen für ihren Beweis. Die Studierenden scheinen aber verstanden zu haben, dass sie einen Widerspruchsbeweis führen müssen und dafür eine Reduktion benötigen und dass sie hierfür eine Annahme stellen und Turing-Maschinen miteinander verknüpfen müssen. Auch sind sie in der Lage, vorangegangene Kenntnisse auf die Problemstellung zu übertragen. Dies demonstrieren sie z.B. in Minute 10-15, als sie zur Klärung des Begriffs „Entscheidbarkeit“ einfachere, ihnen bereits bekannte Automatenmodelle und Sprachfamilien (DFAs, NFAs und reguläre Sprachen) heranziehen und erkennen, dass diese entscheidbar und durch Turing-Maschinen darstellbar sind. Es fehlen ihnen allerdings Fachkompetenzen, um darauf basierend einen formal korrekten Beweis zu erstellen.

Verbesserte Gruppenarbeit durch gemeinsame Arbeitswerkzeuge: Eine weitere Beobachtung, die während der Analyse auffällt, ist die Art und Weise, wie die Gruppe ein Whiteboard ab Minute 10-15 in ihren Arbeitsprozess einbaut. Die Nutzung zeigt positive Effekte auf die Gruppenarbeit. So nutzten die Studierenden das Board, um Skizzen ihrer Turing-Maschinen zu präsentieren, und beziehen sich in den darauffolgenden Gesprächen stets auf die Whiteboard-Notizen, die sie regelmäßig aktualisieren und ergänzen. Auch für die finale, schriftliche Version ihres Beweises nutzen die Studierenden ihre am Whiteboard erstellten Notizen. Im Interview gaben die Studierenden an, sonst immer ohne Tafel oder Whiteboard ihre Übungsaufgaben zu arbeiten (Die Beobachtungsstudie wurde in einem Raum mit Whiteboard durchgeführt). Hier merkten sie ergänzend an, das Whiteboard für die Gruppenarbeit als hilfreich empfunden zu haben, da sie Fehler ohne großen Aufwand korrigieren konnten, alle Gruppenmitglieder auf dem gleichen Stand sein konnten und jeder nachvollziehen konnte, was jeweils die Person macht, die etwas aufschreibt.

Nutzung von bereitgestellten Materialien: Zur Erarbeitung ihres Lösungsvorschlages nutzen die Studierenden während der ersten 30 Minuten nur das, woran sie sich ohne Unterlagen aus der Vorlesung und der Übung erinnern. Als einzige externe Quelle kommt das Lehrbuch „Theoretische Grundlagen der Informatik“ [So08] zum Einsatz, welches einer der Gruppenmitglieder zusätzlich zur Vorlesung und Übung nutzt. Aus diesem Buch wird ein Beispiel für Unentscheidbarkeit zitiert (sog. Barbierparadoxon) und die Gruppe versucht dieses auf ihren Lösungsvorschlag zu übertragen. Das Beispiel passt insgesamt jedoch nicht zu einem Reduktionsbeweis und verwirrt die übrigen zwei Studierenden. Die Gruppe nutzt keine weiteren Materialien oder Quellen für ihren Beweisansatz. Die vom Dozenten bereitgestellten Vorlesungsfolien, Vorlesungsaufzeichnungen, das Online-Skript sowie das in der Vorlesung empfohlene Buch von Hopcroft et. al. [HMU01] werden während der gesamten Gruppensitzung nicht erwähnt und auch nicht genutzt. Das einzige Material, welches vom Lehrenden stammt und während der Sitzung eingesetzt wird, ist

die Musterlösung zu einer Aufgabe, die ebenfalls einen Reduktionsbeweis nutzt und in der Übung sowohl bearbeitet als auch besprochen wurde. Diese Musterlösung soll den Studierenden im Sinne einer „Blaupause“ helfen, ihren eigenen Beweis zu führen. Die Musterlösung wird jedoch erst nach Minute 30 aufgeschlagen, um zu klären, wie der eigene bisher besprochene Lösungsansatz aufgeschrieben werden sollte.

Nichterkennen von Fehlern: Im Hinblick auf die Musterlösung ist die durch die Studierenden entwickelte Lösung mangelhaft, insbesondere was die Reduktion betrifft. So spielt die erste und zweite Beobachtung hier eine entscheidende Rolle. Die Studierenden scheinen sich aber während der Arbeitssitzung nicht dessen bewusst zu sein, dass sie Fehler machen. So äußert ein Student in Minute 10-15, dass er den Beweis in der Übung verstanden hat und gemäß dem Verfahren in der Vorlesung vorgegangen ist. Ein weiterer Student bestätigt das⁵. Auch die Motivation der Studierenden sich mit der Musterlösung zu befassen, fußt nicht auf der Idee ihren Ansatz an sich zu verbessern, sondern sich an der Schreibweise und schriftlichen Struktur der Musterlösung zu orientieren. Bei der Auseinandersetzung mit der Musterlösung kommt die Gruppe zu dem Schluss, die Aufgabe aus ihrer Sicht gut gelöst zu haben⁶ und beginnt anschließend mit der Verschriftlichung. Sie erkennen in der Musterlösung nicht jene relevanten Stellen, an denen die Beweisarbeit geleistet werden muss und sehen entsprechend nicht, dass in ihrer Lösung essentielle Beweisbestandteile fehlen.

Die genannten Beobachtungen deuten insgesamt darauf hin, dass die Studierenden Schwierigkeiten damit hatten, ihren Ansatz korrekt zu beurteilen und dessen Qualität richtig einzuschätzen. Sie konnten zwar mit den Grundbegriffen der Reduktion arbeiten und hatten eine grobe Idee, wie der Beweis über die Konstruktion miteinander arbeitender Turing-Maschinen umgesetzt wird, erarbeiteten den eigentlichen Reduktionsalgorithmus jedoch nicht. Über das Fehlen der Reduktion in ihrem Ansatz schienen sie sich nicht bewusst zu sein und interpretierten die Weitergabe des Outputs einer Turing-Maschine an eine andere als die eigentliche Reduktion und Verknüpfung der auftretenden Maschinen (siehe Musterlösung in Abschnitt 2). Dies deutet erneut daraufhin, dass die Studierenden allgemein Probleme mit der formalen Beweisentwicklung haben. Als einen möglichen Lösungsvorschlag für diese Probleme haben wir an der Universität Hamburg angefangen, interaktive Beweisassistenten zu nutzen, um Bachelor-Studierende in der Entwicklung von Beweisaufgaben zu unterstützen. Die Studierenden haben mit dem interaktiven Beweisassistenten Beweisaufgaben aus den Themenbereich „Logik“ und „Datenstrukturen“ mit dem Beweisassistenten Coq gelöst, um sich intensiver mit der Herangehensweise der Beweisentwicklung und deren Struktur zu beschäftigen [KFB17]. Ein solcher Ansatz stellt eine Möglichkeit dar, die in dieser Arbeit beobachteten Probleme der Studierenden durch fachdidaktische Maßnahmen anzugehen. Auch in der Mathematikdidaktik werden Ansätze diskutiert, die sogenannte intelligente Tutorensysteme einsetzen oder verstärkt auf Kompetenzentwicklung

⁵ „Ja, in der Übung hatte ich das so verstanden, dass wir irgendwann benutzen, dass TM H halt das Haltproblem quasi ist und wir das deswegen nicht lösen können, weil wir halt nicht wissen, ob die irgendwann anhält oder nicht.“ –Student 2

⁶ „Ja, also eigentlich haben wir das/ ist das schon relativ richtig.“ –Student 2

fokussieren [ADS12, SDB08, SN04].

6 Fazit

In diesem Artikel wurde eine Einzelfallstudie vorgestellt, in der drei Informatik Bachelorstudierende beobachtet wurden, wie sie eine Unentscheidbarkeitsaufgabe mit Turing-Maschinen aus der Theoretischen Informatik bearbeiten. Für die Analyse wurde eine angepasste Variante der Inhaltsanalyse nach Krummheuer sowie qualitative Inhaltsanalyse nach Mayring genutzt. Im Ergebnis konnten Verständnisschwierigkeiten von Turing-Maschinen sowie Defizite in der Entwicklung formal korrekter Beweise bei den Studierenden beobachtet werden. Hier wurde besonders deutlich, dass die Studierenden vom Ansatz her, die Aufgabe zwar richtig bearbeitet hatten, aber wesentliche Beweiselemente in ihrer Lösung fehlten. Dieses Ergebnis ist relevant, weil vielfach angenommen wird, dass Informatikstudierenden ausreichend Motivation und Interesse fehlt, um sich erfolgreich mit Theoretischer Informatik auseinanderzusetzen. Die hier vorliegenden Ergebnisse zeigen, dass die Schwierigkeiten differenzierter zu betrachten sind und neben motivationalen Aspekten mathematische Grundlagen im Entwickeln und Aufschreiben formaler Beweise in der Theoretischen Informatik eine Rolle spielen können.

Literaturverzeichnis

- [ADS12] Autexier, S.; Dietrich, A.; Schiller, M. Towards an Intelligent Tutor for Mathematical Proofs. In: Proceedings THedu'11: 1-28, 2012.
- [Ba09] Bachmann, G.: Teilnehmende Beobachtung. In (Kühl, S.; Strodtholz, P.; Taffertshofer, A.; Hrsg.): Handbuch Methoden der Organisationsforschung: Quantitative und Qualitative Methoden, VS Verlag für Sozialwissenschaften, Wiesbaden, 1. Auflage, S. 248-271, 2009.
- [CGM04] Chesñevar, C.; González, M.; Maguitman, A.: Didactic strategies for promoting significant learning in formal languages and automata theory. In: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, ITiCSE '04, S. 7-11, 2004.
- [Hä04] Hämäläinen, W.: Problem-based learning of theoretical computer science. In: Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference, S1H/1-S1H/6, 2004.
- [He17] Heublein, U.; Ebert, J.; Hutzsch, C.; Isleib, S.; König, R.; Richter, J.; Woisch, A.: Zwischen Studierenerwartungen und Studienwirklichkeit. DZHW, Hannover, 2017.
- [HMU01] Hopcroft, J. E.; Motwani, R.; Ullman J. D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 2. Auflage, 2001.

- [KF16] Knobelsdorf, M.; Frede, C.: Analyzing Student Practices in Theory of Computation in Light of Distributed Cognition Theory. In: Proceedings of the Conference on International Computing Education Research (ICER '16). S. 73-81, ACM, 2016.
- [Ki17] Kiehn, F.: Empirische Untersuchung von Studierendengruppen bei der Bearbeitung einer Übungsaufgabe aus der Theoretischen Informatik. Masterarbeit, Universität Hamburg, 2017
- [Kn17] Knobelsdorf, M.; Frede C.; Böhne S.; Kreitz C.: Theorem Provers as a Learning Tool in Theory of Computation. In: Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17), ACM, 2017
- [Ko07] Korte, L.; Anderson, S.; Pain, H.; Good, J.: Learning by Game-building: A Novel Approach to Theoretical Computer Science Education. In: SIGCSE Bull. 03/07, S. 53-57, 2007
- [Kr12] Krummheuer, G.: Die Interaktionsanalyse. In (Heinzel, F., Hrsg.): Methoden der Kindheitsforschung. 2. Auflage, Beltz Juventa, Weinheim, S. 324-247, 2012.
- [Ma10] Mayring, P.: Qualitative Content Analysis. Forum: Qualitative Social Research 2/2010, <http://www.qualitative-research.net/index.php/fqs/article/view/1089>, Abrufdatum: 21.04.2017
- [PL14] Parker, M.; Lewis, C.: What Makes big-O Analysis Difficult: Understanding How Students Understand Runtime Analysis. J. Comput. Sci. Coll., 04/14, S. 164-174, 2014.
- [SDB08] Schiller M.; Dietrich D.; Benzmüller C. Proof step analysis for proof tutoring—a learning approach to granularity. Teaching Mathematics and Computer Science 6.2: 325-343, 2008.
- [Si07] Sigman, S.; Engaging students in formal language theory and theory of computation. SIGCSE Bull. 01/07, S. 450-453, 2007.
- [SN04] Sommer, R; Nuckols, G. A Proof Environment for Teaching Mathematics. Journal of Automated Reasoning, Volume 32(3): 227-258, 2004.
- [So08] Socher, R.: Theoretische Grundlagen der Informatik. Carl Hauser Verlag, München, 3. Auflage, 2008.