Universität Rostock

Traditio et Innovatio

Automatische Sprachübersetzung von LaTeX-Dokumenten

Name:	Hendrik Theede
Matrikelnummer:	221201256
Abgabedatum:	02.12.2025
Betreuer und Gutachter:	Prof. Dr. rer. nat. habil. Clemens H. Cap Universität Rostock Fakultät für Elektrotechnik und Informatik

Bachelorarbeit am Lehrstuhl für Informations- und Kommunikationsdienste

Fakultät für Elektrotechnik und Informatik

Abstrakt

placeholder

Inhaltsverzeichnis

1 Einleitung

1.1 Hintergrund

Der erste Satz dieses Werkes befindet sich noch in Arbeit. Vermutlich wird dieser mit einem Wort beginnen, welches seinerseits den Anfangsbuchstaben „a“ trägt (bsps. „Ausgehend (von)“ oder „Anders (als . . . behaupten)“).

Herkömmliche Software zur Übersetzung von menschlicher Sprache auf TEX-Quellcode anzuwenden, erzeugt schnell Dokumente, welche entweder nicht vollständig übersetzt wurden oder sich nicht mehr kompilieren lassen. Mit Hilfe von Google Translate lassen sich wesentliche Gründe hierfür finden und wie sich diese äußern. Beispielsweise führt eine Übersetzung von `hello wor\textit{ld}` nicht zu `Hallo We\textit{lt}`, sondern zu `hallo wor\textit{ld}`. Abgesehen von der Frage, wo die kursive Hervorhebung im eigentlichen String erfolgen soll, werden Leser eines kompilierten Dokumentes das Wort „Welt“ erkennen. Zuvor beschriebene Zeichenkette wird von TEX zu „hello wor*ld*“ aufgelöst, in welcher das Wort „world“ für einen menschlichen Leser als das englische Wort für „Welt“ erkenntlich bleibt. Fehlt die Kenntnis über eine der Sprachen (DE,EN), würde einem monolingualen Leser Teil der Wortkette geraubt werden. Selbstverständlich sind die Wörter „world“ und „Welt“ einander sehr nahe und auch eine Formulierung der Art „Hallo Welt“ lässt Vermutungen gegenüber eines größeren Kontexts zu.Anders wäre dies, wenn das Auslassen von auch nur einem Wort keine Rückschlüsse mehr auf einen größeren Kontext mehr zulässt. `$\mathbb{P}$robability density function` wäre ein denkbarer stilistischer Weg bereits in z.B. einem Folientitel bereits eine Notation für eine Wahrscheinlichkeitsdichtefunktion einzuführen. Hierbei würde der Verlust des Wortes „probability“ den stochastischen Kontext aufheben. Der Verlust des Wortes „density“ würde einen Kontext innerhalb der Stochastik verändern und ohne das Wort „function“ ist fraglich, wovon die Rede ist. Vor allem in größeren Dokumenten könnten hierdurch Logikbrüche entstehen.

Eine Betrachtung eines „Übersetzers“ als Konzept veranschaulicht die Problematik auf abstrakterer Ebene. Sollte der Kontext des Dokumentes unbekannt sein, werden sich unausweichlich semantische Fehler einschleichen. Bereits das gezeigte Beispiel könnte für z.B. eine Folie einer Lesung den restlichen Kontext der Seite entfernen und dadurch die Möglichkeit bieten umgangssprachliche Bedeutungen in Wörter zu interpretieren, anstatt einer Mathematischen (bspw. „ungerade“ könnte im Englischen „crooked“, statt „odd“ produzieren). Noch weitere sprachliche Beispiele finden sich schnellig durch Wörter mit zeitlichem/räumlichen Bezug. Der Satz `Morgen wird es regnen.` könnte ohne das Wort „morgen“ als Frage mit unzureichend eingehaltener deutscher Grammatik interpretiert werden. (*Wird es regnen?*). Hierbei verliert man eine getroffene Aussage über das Wetter, welches bekanntlicherweise nur schwer vorhergesagt werden kann.

1.2 Anforderungen

Genauso wie das Fehlen einzelner Wörter die sprachliche Bedeutung für einen Menschen brechen kann, treten ähnliche Probleme auch in TEX auf. Einzelne LATEX Makros *nicht* zu übersetzen ist unbedeutend, da sie ihre Bedeutung für einen TEX Compiler behalten. Alleine einzelne Wörter eines Makros zu übersetzen kann dazu führen, dass größere Inhalte (im Sinne: Menge an Worten) nicht mehr in einem kompilierten Dokument vorzufinden sind, was auf eine Fähigkeit von TEX zurückführbar ist. Die Möglichkeit in bestimmten Fällen eine Dateiendung auszulassen, führt beim Einbinden von anderen `.tex` Dateien in einem TEX Dokument zu fehlerhaften/fehlenden Ressourcenangaben. `\include{clock}` zu `\include{Uhr}` zu übersetzen (wie bspw. Google Translate am 06.10.2025) würde nun nicht mehr zu `\include{clock.tex}` aufgelöst werden, sondern zu `\include{Uhr.tex}` (bei welchem nicht davon auszugehen ist, dass diese Datei zur Kompilierzeit im System zwingend vorliegt).

Daher muss nach einer Lösung gesucht werden, welche diese technischen und sprachlichen Hürden überwinden kann. Neben solchen rein technischen Details, darf die Perspektive des Lesers (wörtlich) nicht missachtet bleiben und keine Übersetzungsprozesse dürfen zu versteckten Inhalten im Dokument führen. Diese Verbergung resultiert aus verschiedenensten Layouting-Problemen, ähnlich wie bei der Skalierung von Boxen auf Webseiten (Anhang A.1) und ist abhängig von einzelnen Sprachen dazu in der Lage unbemerkt verdeckte textliche Inhalte zu provozieren.[1]Wünschenswert ist neben vorigen Aspekten auch Möglichkeiten für den Endnutzer zu erlauben, sollte dieser spezielle Übersetzungen oder Kontexte für einige Wörter wünschen, welche jedoch nicht aus dem Dokument selbst hervorgehen. Außerdem sollte ein möglichst hoher Support für sowohl verschiedene menschliche Sprachen, aber auch verschiedene LATEX-Pakete gegeben sein, wobei Letzteres nur ein Bonus ist, sollten Systeme wie Ti*k*Z, bzw. `pgfplots` oder BibTEX innerhalb LATEX (zusammen mit TEX) nutzbar bleiben.

[1]Zwei adjazente Textfelder müssen sich zwangsläufig überlagern, wenn eines unabdingbar größer werden muss, da z.B. die Textgröße nicht verkleinert werden kann und das Wachstum eines Textfeldes nur in das Gebiet eines Anderen stattfinden kann. Dies wäre z.B. mit Hilfe von Inhaltsangaben auf Lebensmitteln vorstellbar. Sollten diese Wörter übersetzt werden und dadurch mehr Textfläche nach rechts benötigen, würden sie in den tabellarischen Bereich der eigentlichen quantitativen Angaben des z.B. Brennwertes, der Makro- sowie der Mikronährstoffe, hereinragen, wodurch das Risiko besteht, dass diese verdeckt werden.

2 Problemfälle

Uneindeutigkeiten in der Sprache sind für einen Leser oft schwer nachzuvollziehen. In TEX muss allerdings zu mindestens einem Zeitpunkt die Information über das Aussehen des entgültigen Dokumentes in einer modellartigen Form vorliegen. Diese Information kann in LATEX verborgen sein oder aber bei einem Übersetzen verloren gehen. Die Probleme unterteilen sich in verschiedene Fälle und werden hinsichtlich des Kontextes dieses Informationsverlustes in translative (beim Übersetzen), technische (LATEX), spezifische technische (in Kombination mit TEX nutzbare Programme) und sprachliche Probleme. Sprachliche Probleme verursachen teilweise dillematische Probleme,welche als „Schwierigkeiten“ und nicht als zu lösende Probleme dargestellt werden. Einzelne aufgeführte Beispiele zur Veranschaulichung beschriebener Probleme sind mit Hilfe von nicht spezifischer Software erzeugt (Google Translate), um zu zeigen, dass jeweilige Situation in *einer* Software zur Übersetzung von menschensprachlichen Inhalten entstehen *könnten*.

2.1 Translativ

2.1.1 Sonderzeichen

Original	Übersetzung
Korrekt	
`1 \label{problem:encounter:solve}` TEX Code 1: Test	`1 \label{Problem:Begegnung:Lösung}` TEX Code 2: Test
Unerwünscht	
`1 \section{example}` TEX Code 3: Test	`1 \Abschnitt{Beispiel}` TEX Code 4: Test

Tabelle 1: Fehler in einem Token

Beschreibung und Begründung Der für einen TEX Compiler relevante Befehl `\label` bleibt unverändert, allerdings `section` wird fälschlicherweise als `Abschnitt` übersetzt. Warum `label` nicht erfasst werden sollte,

wenn die folgenden drei Wörter übersetzt wurden, wirft Fragen auf. Zu sehen ist ein String, welcher menschliche Sprache mit Sonderzeichen vermischt. Da insbesondere Klammern in (vielen) sprachlichen Kontexten hilfreich sind, werden deren Inhalte selbst zunächst nach zusammenhängenden Worten durchsucht, welche ihrerseits durch : getrennt sind, oder als Klammern betrachtet werden können. Ein Entfernen der Klammern lässt lässt in erstem Beispiel `\label problem encounter solve` und in zweitem Beispiel `\section example` stehen. Zu sehen ist hier also bereits, dass Google Translate bei einer, wenn man es so interpretieren möchte, Vernestung zweiten Grades scheitert, jedoch einfache Vernestungen noch erkennt[2].

Takeaway Teile der TEX-Syntax lassen sich anhand von `\`, `{`, `}`, `[`, `]`, `$`, `$$` oder `\%` erkennen und müssten daher ausgeschlossen werden. Anders als in mathematischen Formeln zeigen sich Sonderzeichen jedoch nicht paarweise auf, sodass sie nicht paarweise ignoriert werden können. Man kann sich diese Art von Fehlern wie 0-dimensionale Fehler vorstellen, wobei die nullte Dimension hierbei bei einem einzelnen Wort beginnt (welche als Punkte zu verstehen sind).

2.1.2 Leerzeichen

Original	Übersetzung
Korrekt	
`\usepackage[urlbordercolor=red]{ »hyperref}` TEX Code 5: Test	`\usepackage[urlbordercolor=red]{ »hyperref}` TEX Code 6: Test
Unerwünscht	
`\usepackage[ urlbordercolor = red ]{ »hyperref}` TEX Code 7: Test	`\usepackage[ urlbordercolor = rot ]{ »hyperref}` TEX Code 8: Test

Tabelle 2: Fehler in einem einzeiligen Dokument

[2] *Vernestung*: Klammern in Klammern, wie in der Mathematik

Beschreibung und Begründung Die Optionen innerhalb eckiger Klammern lassen auch Whitespace zu. Dies kann jedoch für die Nutzung einiger Funktionen in z.B. wichtigen Paketen wie `hyperref` dazu führen, dass falsche Wörter übersetzt werden, die ein Kompilieren des Dokumentes verhindern.

Abstrahierung Teile der TEX-Syntax lassen sich nicht nur anhand der zuvor beschriebenen Zeichenketten erkennen, sondern lassen sich auch in Zeilen wiederfinden. Diese Art von Fehlern bahnt den Weg zu einer Dimension, wodurch nicht nur innerhalb eines Wortes (Punktes), sondern auch zwischen verschiedenen Punkten Fehler entstehen könnten (also innerhalb einer Zeile).

2.1.3 Zeilenbrüche

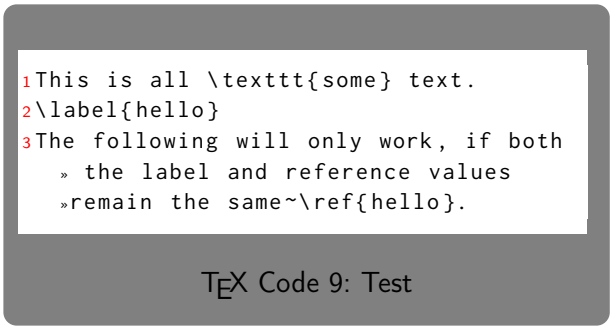
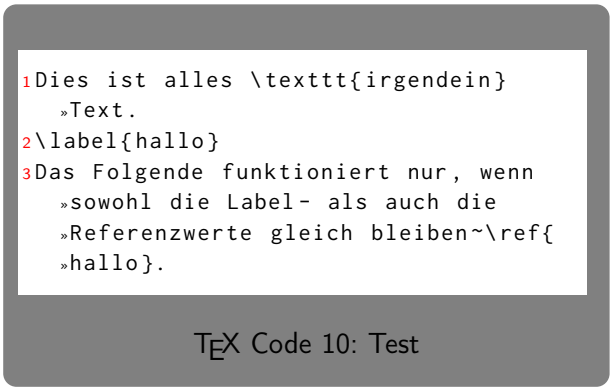
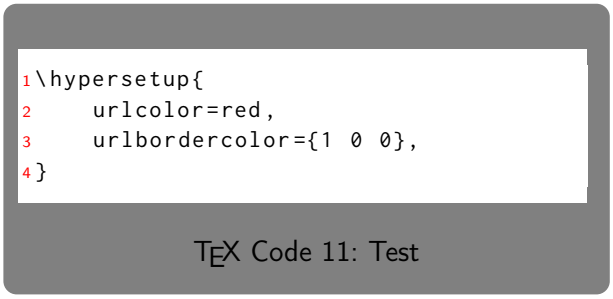
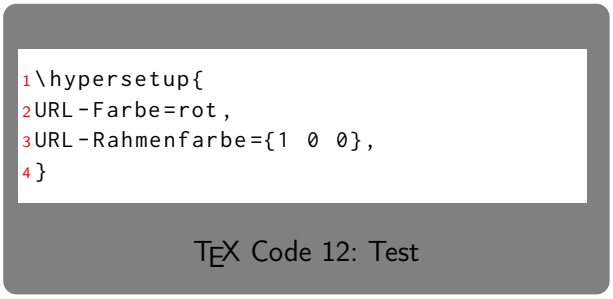
Original	Übersetzung
Korrekt	
	
TEX Code 9: Test	TEX Code 10: Test
Unerwünscht	
	
TEX Code 11: Test	TEX Code 12: Test

Tabelle 3: Fehler in einem einzeiligen Dokument

Beschreibung

Abstrahierung Teile der TEX-Syntax lassen sich nicht nur anhand von einzelnen Zeilen oder Zeichenketten erkennen, sondern könnten sich auch auch in verschiedenen Zeilen wiederfinden lassen. Diese Art von Fehlern kann 2-dimensional betrachtet werden, wodurch Fehler auch zwischen Zeilen entstehen können.

2.1.4 Dokumentenbrüche

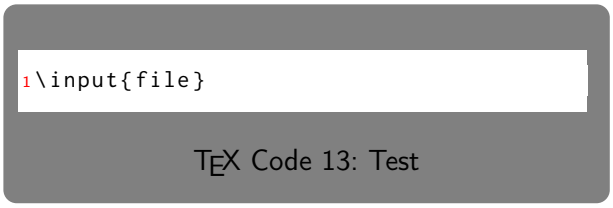
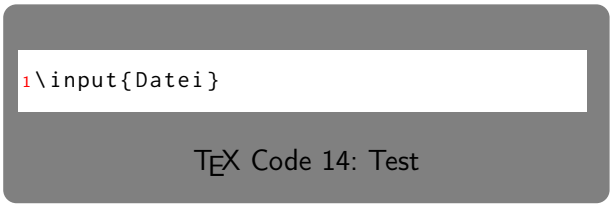
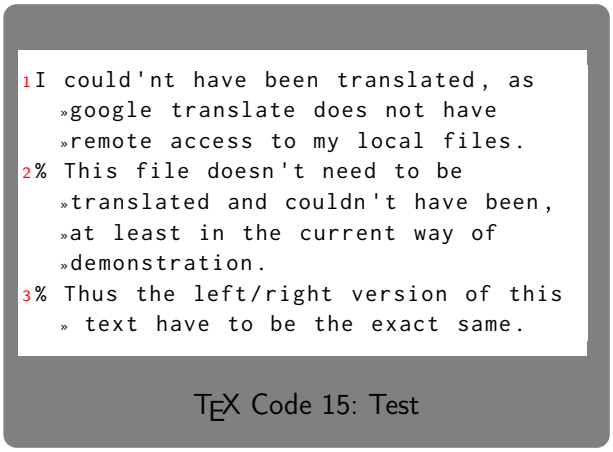
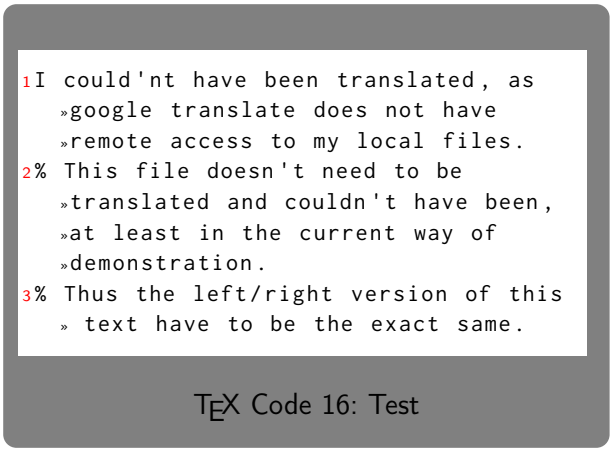
Original	Übersetzung
Korrekt	
	
TEX Code 13: Test	TEX Code 14: Test
Unerwünscht	
	
TEX Code 15: Test	TEX Code 16: Test

Tabelle 4: Fehler in einem einzeiligen Dokument

Beispiel

Beschreibung Die Übersetzung von Datei x, welche Datei y via `input` oder `include` führt zwar dazu, dass Datei x übersetzt wird, aber Datei y nicht.

Abstrahierung Teile der TEX-Syntax müssen nicht zwingend in einer Datei vorliegen, sondern könnten auch in verschiedenen Dateien integriert sein. Die Klassifizierung simpler Probleme gelangt in TEX hier bereits in der dritten Dimension an, weswegen sich fortan bereits mit „fortgeschrittenen“ Problemen beschäftigt werden muss (welche sich Teils über mehrere „Dimensionen“ erstrecken). Eine vierte Dimension existiert physikalisch nicht,

ist jedoch mathematisch formulierbar[3] und äußert sich in diesem Kontext auf eine Erhöhung von Laufzeitkomplexitäten.

2.2 Technisch

2.2.1 Referenzen

Beispiel Mittels `cite` wird auf ein Werk verwiesen, in welchem ein Kontext für eine Übersetzung gesetzt wird (bspw. kann eine Referenz auf den Euklid einen mathematischen Kontext setzen). Jedoch produziert nur die Kenntnis, das eine Referenz auftritt und das Wort (z.B.) „ungerade“ die Übersetzung: „crooked“, statt „odd“.

Beschreibung Bereits das Vorstellen von Problemen, welche eine Entstehung (ein Kompilieren) eines LaTeX Dokumentes verhindern könnten, führt bis in die dritte Dimension. Da eine physikalische Vorstellung hier nicht weitergeführt werden kann, wird auf eine zeitliche Schilderung umgeschwenkt. Sie eignet sich an dieser Stelle, wenn man als „Zeit“ den Entstehungspunkt der folgenden Probleme innerhalb des LaTeX Dokumentes betrachtet. Ähnlich wie solche Vorfälle, die ein erneutes Kompilieren provozieren, zeigen sich Hürden, welche zwar ihrerseits keine neue Übersetzung provozieren müssen, jedoch eine *andere* Übersetzung (wörtlich) erzeugen *müssten* (falls diese Fälle den Kontext eines Teiles des Dokumentes ändern).

2.2.2 Laufzeiten

Beschreibung Zudem müssen einige Instanzen bedacht werden, in welchen zwar nicht eine Übersetzung selbst stattfinden muss, aber Texte in einem Dokument verändert werden müssen, nachdem diese bereits kompiliert wurden, bzw. in einer PDF vorliegen, welche ihrerseits angepasst werden müsste, was sich nur mit einem erneuten Kompilieren ändern lässt (da logische Änderungen innerhalb des Dokumentes auftraten).

2.2.3 Unerreichbare Informationen

Beispiele

Beschreibungen Ein Dokument erwähnt ein Werk, in welchem es um die C-Programmierung geht. Rein aus den im System vorliegenden Dateien ist kein Kontext für das Wort „String“ erkennbar, sodass ein Zugriff auf eine externe Ressource unabdingbar ist.

Abstrahierung Einfache Cloud-Architektur. Ein Client möchte auf ein beliebiges Wissen einer Webseite (bzw. dem Server und den beanspruchten Speicherplätzen in einem (beliebigen) Rechenzentrum[4] zugreifen).

[3] physikalisch: die vierte Dimension ist die Zeit, wenn man eine nicht-euklidische 3-dimensionale Bewegung verlangt (Teleportation)

[4] Hierbei ist nicht von Festspeicher zu reden. Aus Sicherheitsgründen sei davon auszugehen, dass sich die physischen Adressen des wissensrepräsentierenden Speichers regelmäßig und unvorhersehbar ändern

Original

Dokument

```
\documentclass{standalone}
\usepackage{natbib}
\bibliographystyle{agsm}
\begin{document}
~\cite{salomon_c} %Kapitel 5
refers to character arrays as a string.
\bibliography{example_original.bib}
\end{document}
```

Bibliothek

```
% Bad Citation, as the title is missing. All necessary information can be garnered by accessing the book behind the isbn.
@misc{salomon_c,
    author={{Prof. Dr.-Ing. habil. Ralf Salomon}},
    year={2013},
    title={Siehe ISBN: 978-3-00-042684-1},
}
```

Übersetzung

```
\documentclass{standalone}
\usepackage{natbib}
\bibliographystyle{agsm}
\begin{document}
~\cite{salomon_c} %Kapitel 5
bezeichnet Zeichenarrays als Zeichenfolge.
\bibliography{example_original.bib}
\end{document}
```

```
% Falsche Zitierung, da der Titel fehlt. Alle notwendigen Informationen finden Sie im Buch hinter der ISBN.
@misc{salomon_c,
author={{Prof. Dr.-Ing. habil. Ralf Salomon}},
year={2013},
title={Siehe ISBN: 978-3-00-042684-1},
}
```

Tabelle 5: Beispiel für einen verpassten literarischen Kontext

2.3 Spezifischer Technologien

Hier wenden wir uns von Problemen einer Übersetzung ab und widmen uns denen eines Lesers. Alle textlichen Inhalte eines Dokumentes zu übersetzen, als auch eine kontextuelle Fachsprache zu bewahren scheint aus abstrakterer Perspektive ausreichen, kann allerdings zu Situationen führen, in welchen Informationen verloren gehen, da diese vom Endnutzer nicht mehr gesehen werden können.

2.3.1 Kommentare

Beispiele

Beschreibungen Wohingegen sich 2.3.1 nicht mit anderen, in Kommentaren referenzierten, Dateien beschäftigt, soll sich hier auf solche Fälle konzentriert werde.

Abstrahierung Hier treffen simple Fehler aus den ersten drei Kategorien (in 2.1.1, 2.1.2 und 2.1.3 geschildert) aufeinander. In die dritte Dimension, also in andere Dateien, wird jedoch (vorerst) nicht traversiert, da auskommentierte Datei-Einbindungen nicht erfasst werden dürften. Ausgehend von **??** wird nun erwartet, dass eine Referenzierung von Dateien erwartet wird, welche sich in Kommentaren verbergen. Dies kann jedoch 2.3.4 beinhalten.

2.3.2 Dilemmatische Makros

Beispiele

Beschreibungen

Abstrahierung

2.3.3 TikZ und Layouting

Beispiele

Beschreibungen

Abstrahierung

2.3.4 Quellmehrsprachigkeit

Beispiele

Beschreibungen

Abstrahierung Quelltexte anderer Quellsprachen (Programmiersprachen) können ihrerseits auf andere Dateien verweisen, oder andere Syntaktik tragen. Das Erkennen dieser ist theoretisch gesehen leicht, jedoch praktisch gesehen schnellig zu übersehen.

2.4 Sprachliche Schwierigkeiten

2.4.1 Glossare und Nomenklaturen

Beispiele

Beschreibungen

Abstrahierung

2.4.2 Weitere

Beispiele

Beschreibungen

Abstrahierung

3 Technologien

3.1 Übersicht

3.1.1 Auflistung

3.1.2 Eingrenzung

3.1.3 Auswertung

3.2 Einschätzung

3.3 Fazit

4 Offene Problematiken

4.1 Verfolgte Ideen

4.2 Gelöste Probleme

4.3 Lessons Learned

4.4 Fazit

5 Fazit

5.1 Zusammenfassung

5.2 Ausblick

5.3 Weiteres

6 Eigenständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt und ohne fremde Hilfe verfasst habe. Dazu habe ich keine außer den von mir angegebenen Hilfsmitteln und Quellen verwendet und die den benutzten Werken inhaltlich und wörtlich entnommenen Stellen habe ich als solche kenntlich gemacht. Ich versichere, dass die eingereichte elektronische Fassung mit den gedruckten Exemplaren übereinstimmt.

Rostock, den 02.12.2025

Hendrik Theede

Literatur

A Anhänge

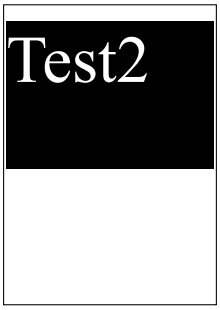
A.1 Fontskalierung auf Webseiten

Beispielsweise produziert die folgende HTML-Notation bei einer Skalierung im Browser von 120 Prozent (Abbildung 2a) und 50 Prozent (Abbildung 2b) jeweilig zwei verschiedene PDF (unter welchen nur Zweitere alle textlichen Inhalte offenbart). Ähnliches kann auch innerhalb TEX geschehen, sollte

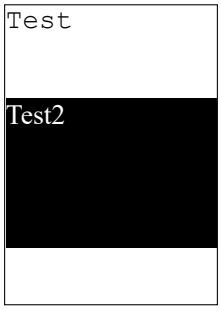
```
<html>
    <head>
        <title>Example</title>
        <style>
            /*formatting options are: none and black*/
            .t{
                font-size:13em;
                height:50%;
            }
            /*formatting option: none = no background, black, courier*/
            .t#none{
                font-family: 'Courier New', Courier, monospace;
            }
            /*formatting option: black = black background, white, serif*/
            .t#black{
                background-color:black;
                color:white;
                margin-top: -2em;
            }
        </style>
    </head>
    <body>
        <div class="t" id="none">Test</div>
        <div class="t" id="black">Test2</div>
    </body>
</html>
```

Abbildung 1: HTML-Beschreibung einer Webseite mit zwei Textflächen

Abbildung 2: Um die Dokumente von der restlichen Papierfläche abzugrenzen wurden schwarze Rahmen mittels Ti*k*Z hinzugefügt.



(a) Zuvorige HTML-Beschreibung liefert bei einer Browser-Skalierung von 120% obige Graphik



(b) Zuvorige HTML-Beschreibung liefert bei einer Browser-Skalierung von 50% obige Graphik