

Wissenschaftskommunikation in der Informatik

Vorlesung, Wintersemester 2022

Johannes Waldmann, HTWK Leipzig

31. Januar 2023

Einführung

Ziele der LV, Anwendung

- – mit englischsprachiger Fachliteratur umgehen,
 - eigene Ergebnisse und Ausführungen gemäß den in der Informatik üblichen Konventionen verschriftlichen,
 - Paper anderer Autoren begutachten,
 - eigene Ergebnisse in Form einer englischsprachigen Präsentation halten.

aus: Modulbeschreibung C902 (gesehen am 15. 10. 22)

<https://modulux.htwk-leipzig.de/modulux/modul/6337>

- Anwendung: Planung, Recherche, Aufschreiben, Verteidigen der Bachelor-Arbeit (im 6. Semester)

Beispiele für schriftliche Arbeiten

- beachte: Titel, Zusammenfassung, Struktur (Gliederung), Einleitung, Diskussion/Ausblick, Literatur, Behauptungen, Begründungen, Veranschaulichungen, (durchgehende) Beispiele,
- Marcellus Siegburg: *REST-orientierte Refaktorisierung des E-Learning-Systems Autotool*, Masterarbeit, HTWK Leipzig, 2015 <http://www-ps.informatik.uni-kiel.de/~msi/thesis/thesis.pdf>
- Johannes Waldmann: *Matrix Interpretations on Polyhedral Domains*, RTA 2015, <https://drops.dagstuhl.de/opus/volltexte/2015/5205/>

Vorgehensweise für unsere LV

- Grundlagen (VL): Funktionen, Formen, Abläufe der wissenschaftlichen Publikation (in der Informatik)
- Übungen: zu einer konkreten Frage der Informatik
 - Frage nennen, motivieren (durch ein Beispiel)
 - relevante Quellen finden (wenigstens zwei!), zitieren, zusammengefaßt darstellen, vergleichen
 - ggf. Beispiel durchrechnen, offene Fragen nennen, Bewertungskriterien für Antworten entwickeln, Arbeitsplan vorschlagen (aber *nicht* durchführen!)
- Prüfung: in einem Textdokument (deutsch, LaTeX) aufschreiben, Dokumente von anderen begutachten, in einem Vortrag (englisch) präsentieren und verteidigen

Organisation

- bis Ende KW 43: Arbeitsgruppen bilden (je 3 Personen), Repo anlegen, Thema wählen (meine Vorschläge folgen)
- bewertet werden Teilleistungen zu Meilensteinen:
 1. Dokument- und Repo- Struktur, 2. Recherche und Literaturangabe, 3. Text- und Formelsatz, 4. Gutachten sowie das schließlich resultierende Dokument.
(Im Teil *Englisch* wird der Vortrag bewertet.)
- primäre Quelle für Informationen zu meinen LV: <https://www.imn.htwk-leipzig.de/~waldmann/lehre.html>
dort Links zu Skripten und Arbeitsanleitungen.
Fragen an mich per Mail, nur von Hochschul-Adressen.
Abgaben (Meilensteine) nur durch Commits (keine Mail).

Themen (Grundsätzliches)

- grundsätzlich: eine *konkrete* Fachfrage der Informatik
Bezug zu einer LV ist nützlich (dort wird die Frage eingeordnet und passende Literatur wird genannt).
- folgende Themen und Themensammlungen sind *zu allgemein*, Sie müssen das *konkretisieren*!
(z.B. ein Puzzle, ein „Gesetz“ auswählen)
- folgende Quellenangaben sind *ungenau, unvollständig*
(z.B. nur URL, es fehlen: Autor, Titel, Ort, Jahr, Datum des Aufrufs)
einige Quellen sind gar *nicht zitierfähig*
(unwissenschaftlich — wurden nicht begutachtet)
Sie müssen selbst recherchieren!

Themenbereich: Informatik und Gesellschaft

- Verantwortung der Informatiker,
<https://blog.fiff.de/>,
<https://gewissensbits.gi.de/>,
- **Aufmerksamkeits- und Überwachungswirtschaft**
[https://www.imn.htwk-leipzig.de/~waldmann/edu/ws21/inf/folien/#\(206\)](https://www.imn.htwk-leipzig.de/~waldmann/edu/ws21/inf/folien/#(206))
digitale Selbstverteidigung (z.B. Browser-Einstellungen)
- **Recht auf informationelle Selbstbestimmung**
(„Datenschutz“) <https://www.saechsdsb.de/>,
<https://www.bfdi.bund.de/>

Themenbereich: Grundlagen

- **mechanische Computer und Puzzles** <https://www.imn.htwk-leipzig.de/~waldmann/edu/modul/seminar/mech/>
- **kombinatorische Optimierungs-Aufgaben** <https://erich-friedman.github.io/mathmagic/>
- zu VL Modellierung, Digitaltechnik: Beziehungen zw. Hausaufgaben (auch autotool) zu Logik, Automaten
- **ausgewählte Aspekte von esoterischen Programmiersprachen** (<https://www.ioccc.org/>, <http://www.99-bottles-of-beer.net/>)
- **Softwaretechnik-„Gesetze“** <https://www.netmeister.org/blog/software-engineering-laws.html>

Themenbereich: Anwendungen und Aktuelles

- „aktuell“ = die Anwender haben sich endlich entschlossen, oder konnten überredet werden, die längst bekannten Grundlagen nun tatsächlich zu benutzen
- Bsp: JEP 431 Sequenced Collections; Lossless compression with Asymmetric Numeral Systems; sichere Identifizierung, Authentifizierung mit Shibboleth
- Quellensammlungen: z.B. <https://queue.acm.org/>
Journalismus, z.B.: <https://arstechnica.com/>,
<https://www.heise.de/>, <https://www.theregister.com/>
ohne jegliche redaktionelle Bearbeitung:
<https://news.ycombinator.com/>

Hausaufgaben (bis Ende KW 43)

1. Arbeitsgruppen bilden (je 3 Personen), Projekte anlegen
`https://gitlab.imn.htwk-leipzig.de/waldmann/wki-ws22/-/blob/master/README.md`
2. Wunschthema in Ihrem eigenen Repo eintragen
(Überschrift und erster Satz von `main.tex`)
Das ist bereits der erste (bewertete) Meilenstein!
3. zur Diskussion in den Übungen: für irgendeinen der von mir geschriebenen oder darin zitierten Fachartikel:
`https://gitlab.imn.htwk-leipzig.de/autotool/all10/-/tree/master#documentation-papers-talks-theses`
untersuchen Sie die Passung von Titel, Zusammenfassung, Struktur, Einleitung, Ausblick, Beispielen.

Kurze Erläuterung: Git, Gitlab.Imn

- git (Torvalds, 2005): verteiltes System zur Quelltextverwaltung, Implementierung als freie Software
Repository = Sammlung von versionierten Texten mit Verzeichnis-Struktur und Vergangenheits-Struktur.
verteilt: es ist keine Zentralstelle nötig
- git{hu/la}b.com: zentrales Repo-Hosting. Mehrwert: Betrieb der Hardware und Zusatzdienste (Continuous Integration, Issue-Tracker (gamifiziert)). Geschäftsmodell: Auswertung der Daten, kostenpflichtige Leistungen.
(Benutzung für Lehrveranstaltungen nicht zulässig.)
- gitlab.imn: an unserer Fakultät selbst betriebene Instanz der gitlab-Software (community edition)

Wissenschaftliche Publikationen in der Informatik

Gegenstand, Überblick

- Gegenstand sind heute die Publikationen, die Sie für Ihre Arbeit benutzen (recherchieren, zitieren)
- Themen sind
 - die Arten von Publikationen (primär, sekundär, tertiär)
 - die Qualitätskontrolle für Publikationen (peer review)
 - Recherche-Methoden und -Werkzeuge
 - das Unterlaufen der Qualitätskontrolle (Spam)
- der Textsatz der Quellenangaben (mit BibTeX) wird später genauer erklärt, heute nur verwendet

Bsp: bibliographische Daten einer Publikation

- Chris Okasaki: *Red-Black Trees in a Functional Setting*, Journal of Functional Programming 9(4): 471-477 (1999)
- Autor, *Titel*, Zeitschrift, Band(Ausgabe), Seiten, Jahr
- Publikation ist durch diese Angaben eindeutig bestimmt und kann damit wiedergefunden werden
- durch Zitate
 - belegen Sie, daß Sie den Stand der Technik kennen,
 - und sparen Zeit und Platz (Sie müssen das Fahrrad nicht neu erfinden, sondern beschreiben Ihre Eigenleistung)

Bsp (Fortsetzung) Forschungsdatenbanken

- bibliographische Daten der genannten Publikation sind u.a. hinterlegt in den Datenbanken
 - der Hochschulbibliothek
(<https://bibliothek.htwk-leipzig.de/>,
Suche im Katalog)
 - des Verlages, über Digital Object Identifier (DOI)
<https://doi.org/10.1017/S0956796899003494>
 - DBLP (ursprünglich *data base for logic programming*,
Univ. Trier, inzwischen für sehr viele Bereiche der
Informatik, Leibniz-Zentrum für Informatik,
[https://dblp.org/rec/journals/jfp/
Okasaki99.html](https://dblp.org/rec/journals/jfp/Okasaki99.html)

Bsp (Fortsetzung) Technik der Quellenangabe

- vom Eintrag bei DBLP: export record ... BibTeX

```
https://dblp.org/rec/journals/jfp/Okasaki99.html?
view=bibtex
```

```
@article{DBLP:journals/jfp/Okasaki99,
  author    = {Chris Okasaki},
  title     = {Red-Black Trees in a Functional Sett.
```

- in Datei `main.bib` kopieren. In `main.tex` so:

```
Okasaki~\cite{DBLP:journals/jfp/Okasaki99} beschreib
\bibliographystyle{plain} \bibliography{main}
```

- kompilieren: `bibtex main; pdflatex main,`

evtl. mehrfach wegen Vorwärtsverweisen

- bis Ende KW 45: drei solche Literaturangaben einfügen

Arten von Publikationen (Erfindungsabstand)

- Nach dem Abstand von der eigentlichen Erfindung unterscheiden wir:
 - primäre Quelle: enthält eine (Welt)Neuheit, ist das Resultat einer Forschungstätigkeit.
Formen: Workshop-Beitrag, Konferenz-Beitrag, Zeitschriften-Beitrag (deren Unterschiede: später)
 - sekundäre Quelle: ist Zusammenfassung, Erläuterung, Bewertung von Primärquellen.
Formen: Monographie, Lehrbuch
 - tertiäre Quelle: enthält Verweise auf Sekundärquellen.
Bsp: Artikel in Lexikon, Wikipedia.
- Sie zitieren Primär- und Sekundärquellen, keine tertiären!

Bewertung von Publikationen

- (Idealfall?) objektive Bewertung: Publikation enthält Aussage mit Beweis, Leser rechnet den Beweis nach.
- aber: Beweis fehlt of (zu wenig Platz), könnte vom Leser gar nicht überprüft werden (nur von Experten),
- deswegen *peer review*. Leser muß dann einschätzen, ob diese überhaupt und korrekt stattgefunden hat.
- muß dazu die Reputation der Herausgeber, der Programmkomitee-Mitglieder einschätzen
- wenn er die nicht kennt: muß sich auf Einschätzung von ihm bekannten Experten verlassen.
- \Rightarrow bei Recherche für studentische Arbeiten: beginnen Sie immer bei der in der Lehrveranst. angegebenen Literatur!

Begutachtung (peer review)

- die Qualität einer Publikation wird durch Begutachtung (durch *peers*: andere Wissenschaftler des gleichen Fachgebietes) hergestellt und gesichert
- für jede Einreichung ca. 3 unabhängige Gutachten,
- Gutachter schätzt die Arbeit ein (korrekt?, neu?, lesbar?), stellvertretend für den zukünftigen Leser
- Resultat eines Gutachtens ist eine Empfehlung: ablehnen, überarbeiten und neu einreichen, überarbeiten und akzeptieren, sofort akzeptieren.
- Entscheidung trifft der jeweils Verantwortliche (Programmchef der Konferenz, Hrsg. der Zeitschrift)

Erscheinungsform und Qualitätskontrolle

- im Idealfall wird in der Informatik jedes Resultat nacheinander (!) in diesen Formen publiziert:
 - (keine Begutachtung durch andere) *technischer Bericht*
 - (oberflächliche Begutachtung) *Workshop-Beitrag*
 - (genaue Begutachtung, Platz- und Zeit-Schranke) *Konferenz-Beitrag*
 - (sehr genaue Begutachtung, keine Platz- und Zeit-Schranken) *Zeitschriften-Beitrag*
- bemühen Sie sich, jeweils die hochwertigste Form des Resultates zu zitieren.
- einige Quellen sind hinter Bezahlschranken, mit evtl. alternativen Versionen z.B. auf Webseiten des Autors

Bewertung von Büchern

- Fachbücher, Lehrbücher werden *lektoriert* (durch den Verlag), aber meist nicht wie Fachartikel begutachtet
- denn das Buch faßt eigene und fremde (in jedem Fall bereits referierte) Publikationen zusammen.
- für studentische Arbeiten: beginnen Sie mit Büchern, die der Dozent empfiehlt (Literatur in Modulbeschreibung)
- für (Ihnen) unbekannte Bücher:
 - nehmen Sie Stichproben des Inhaltes und der Form, z.B.: wird im Buch selbst ordentlich zitiert?
 - schätzen Sie Kompetenz und Reputation des Autors z.B.: referierte Publikationen, akademische Position

Das Erschließen weiterer Quellen

- ausgehend von einer seriösen Quelle (z.B. Literaturempfehlung aus Vorlesung):
- die in dieser Quelle angegebene Literatur (d.h., frühere)
- Artikel, die die gegebene Quelle zitieren (d.h., spätere)
- an gleicher Stelle erschienene Artikel (z.B. andere Beiträge der gleichen Konferenz/Zeitschrift, auch frühere und spätere Jahrgänge)
- andere Artikel des gleichen Autors, von Personen aus der gleichen Arbeitsgruppe

Bsp. für Quellenarten und -Verwendung

- Thema: eine Eigenschaft einer Programmiersprache.
- Primärquelle: der Sprach-Standard.
- Sekundärquellen: eine Konferenz über diese Sprache oder dieses Paradigma, veranstaltet von einer etablierten Fachgesellschaft (Bsp.: GI, ACM)
- unwissenschaftlich: Vermutungen, die keine peer review durchlaufen haben
- eine solche unwiss. Quelle könnte ein interessantes Beispiel enthalten—das kann man zitieren, muß es dann aber selbst analysieren, mit Bezug auf wissenschaftliche Quellen.

Interessenkonflikte

- der eigentliche Zweck der Publikation (Dokumentation des wissenschaftlichen Fortschritts)
tritt in Konflikt mit anderen Interessen:
- die Politik (als Geldgeber öffentlicher Forschung) verwendet Anzahl von Publikationen zur „Leistungs“messung und Geld-Zuteilung
⇒ wer da mitspielt, muß auch Mikro- und Garnicht-Fortschritte zu Publikationen aufbauschen
- Verlage verdienen an jeder Publikation
- seriöse Verlage achten mglw. trotzdem auf ihren Ruf, das eröffnet eine Markt für betrügerische Verlage und Konferenzen (Spamference)

SEO-Spam

- search engine optimisation (SEO): Webseitengestaltung mit dem Ziel der Irreführung von (vermuteten) Bewertungsalgorithmen von Suchmaschinen
Ziel: höhere Einnahmen aus Anzeigenplatz-Vermietung
- Zusammenkratzen (scraping) von Publikationen, Angeboten (evtl. nur teilweise) unter eigener Domain
- (angebliche) Herausgabe der versteckten Teile nach Preisgabe weiterer Daten (Account anlegen), aus denen genauere Personenprofile konstruiert werden
- Suchmaschinen liefern dann überwiegend solche *Spam*-Resultate. Wenn der Suchmaschinenbetreiber gleichzeitig der Werbemakler ist, bleibt das auch so.

The Cost of Knowledge

- Tim Gowers et al., <http://thecostofknowledge.com/> 2012. (Zugriff am 3. 11. 2022).
- die traditionellen Aufgaben der Wissenschaftsverlage:
Organisation der Begutachtung, Textsatz, Druck/Versand.
- sind weitgehend weggefallen oder automatisiert:
der Autor setzt den Text selbst, gedruckt wird gar nichts.
- die verbleibende Arbeit bezahlt der Staat:
1. die Arbeitszeit des Autors, 2. der Gutachter
- und 3. kauft dem Verlag dann noch die Publikation ab

Open Access

- frei verfügbare peer-reviewed Publikationen: z.B.
 - Logical Methods in Computer Science (LMCS)
`https://lmcs.episciences.org/`
 - Leibniz Intl. Proceedings in Informatics (LIPICS) `https://drops.dagstuhl.de/opus/institut_lipics.php`
- freie Dokument-Speicher (ohne Begutachtung) z.B.
 - Computing Research Repository (CoRR)
`https://arxiv.org/archive/cs`
- wer zahlt die Betriebskosten? Bsp. LIPICS: 60 EUR pro Artikel (bis 20 Seiten), bezahlt Konferenz-Veranstalter, wird auf Teilnahmegebühren umgelegt – da zahlt eine staatliche Stelle (Univ. des Autors) and eine andere

... und Open Access

- um Kosten vom Leser auf den Autor umzulegen
(in beiden Fällen zahlt schließlich der Staat)
- z.B. Springer LNCS: 30 EUR (plus MWSt.) pro Seite
(üblich bei Konferenzen: 15 Seiten pro Artikel)

Quelle (gelesen am 3. 11. 2022)

<https://www.springer.com/gp/computer-science/lncs/open-access-publishing-in-computer-proceedings>

- z.B. Elsevier Theoretical Computer Science:
article processing charge (APC) 2.280 EUR.

Quelle (gelesen 3. 11. 2022)

<https://www.elsevier.com/open-access>

Weiteres zum Recherchieren und Zitieren

- dazu gibt es viele Hinweisen (z.B. von Bibliotheken): das ist nützlich, aber überlegen Sie bei und vor Anwendung:
- Technik der Suche (in Katalogen):
Qualitätskontrolle ist dort oft nicht erklärt.
- Verwaltung und Benutzung (Formatierung) des eigenen Literaturverzeichnisses: viele Anweisungen beziehen sich auf unzureichende Software.

Benutzen Sie (LaTeX und) BibTeX, verwenden Sie Einträge aus DBLP als Beispiel, ändern Sie diese nicht.

Werkzeuge zur Recherche, Zitatverwaltung

- z.B. Citavi: ist nicht fachspezifisch (Informatik), ist m.E. unnötig (verwende stattdessen DBLP und Textfiles)
ist unfreie Software, mit Lock-Angeboten zum Einsammeln personenbezogener Daten.
- freie Software mit ähnlicher Funktion ist z.B. Zotero.
Vorsicht - Versuch des Geldverdienens/Datensammelns mit Zotero Storage (standardmäßig abgeschaltet)
- Suchmaschinen: überlegen Sie, wem Sie Ihre Daten zur unbegrenzten (Zweck, Zeit) Nutzung überlassen
- Hinweise der Hochschulbibliothek (Stand November 2022) <https://bibliothek.htwk-leipzig.de/kurse-und-beratung/online-tutorials/online-recherchekurs-barrierefrei/>

Zusammenfassung, Empfehlungen

- Qualität und Eignung einer Quelle, einer Quellensammlung (Suchmaschine) müssen Sie selbst einschätzen (das ist fachspezifisch)
- die wesentlichen Quellen-Angaben sind (für Bücher) Autor, Titel, Verlags-Ort, Jahr
- andere Publikations-Formen: den Ort sinngemäß anpassen (z.B. \Rightarrow Name der Konferenz, der Zeitschrift)
- reine Online-Quellen: Ort \Rightarrow URL (vollständig), *mit Zugriffsdatum*.

Vermeiden Sie das! Qualität ist unsicher, Inhalt ist instabil
mglw. hilft (dem Leser) `https://web.archive.org/`,
Internet Archive Wayback Machine

Aufgaben (bis Ende KW45), Organisation

1. recherchieren Sie für Ihr Thema.

Legen Sie einen Literatur-Verantwortlichen fest. (in `main.tex` eintragen, damit ich den Namen sehe)

Zitieren Sie wenigstens ein Buch, einen Zeitschriften-Artikel, einen Konferenz- oder Workshopbeitrag.

in `main.bib` eintragen, in `main.tex` verwenden, siehe **Beispiel** `https:`

`//gitlab.imn.htwk-leipzig.de/waldmann/wki-ws22`

Übernehmen Sie dabei wenigstens einen Datensatz aus DBLP, schreiben Sie wenigstens einen solchen Datensatz selbst.

Sie können diese Literaturliste später noch ändern.

2. KW 45 keine Übung (Sie haben nur 0.5 SWS Übung lt. Modulbeschreibung)

aber Konsultationen möglich (zu den im Studienplan angegebenen Zeiten und Räumen) — bitte vorher anmelden per Mail

3. lesen Sie

J. Waldmann: *Die wissenschaftliche Publikation*, Vortrag im Rahmen der Kompetenzschule für Doktoranden, HTWK Leipzig, 20. 10. 2011, <https://www.imn.htwk-leipzig.de/~waldmann/talk/11/pub/>

Dokumentstruktur, logisches Mark-Up

How to Write a Great Research Paper

- Simon Peyton Jones: *How to Write a Great Research Paper*, ICFP PLMW 2015

<https://www.cis.upenn.edu/~sweirich/icfp-plmw15/>

1. nicht warten, sondern schreiben
2. über eine Idee—genau eine
3. erzähle eine Geschichte (baue die passende Struktur)
4. zeige sofort und konkret, was wesentlich/neu ist
5. sprich über verwandte Arbeiten: später
6. denke an die Leser, erst der Plan (mit Beispielen), danach Details
7. höre auf die (Test-)Leser (Gutachter)

Die passende Dokument-Struktur

- Simon Peyton Jones: *How to Write a Great Research Paper*, ICFP PLMW 2015

empfohlene Struktur — wahrscheinliche Anzahl der Leser
(bei einer wissenschaftlichen Publikation)

- Titel (1000 Leser)
- Zusammenfassung (4 Sätze) (100 Leser)
- Einleitung (1 Seite) (100 Leser)
- Aufgabenstellung (1 Seite) (10 Leser)
- Idee der Lösung (2 Seiten) (10 Leser)
- Details der Lösung (5 Seiten) (3 Leser)
- andere Arbeiten (2 Seiten) (10 Leser)
- Zusammenfassung, Ausblick (0.5 Seiten)

Struktur durch logisches Markup realisieren

- ```
\documentclass[12pt,a4paper]{article}
\usepackage{hyperref}\usepackage{german}
\author{A. Uthor} \title{Alles Quatsch}
\begin{document} \maketitle
\begin{abstract} ... \end{abstract}
\section{Einleitung}\label{sec:ein}
... siehe Abschnitt \ref{sec:exp}.
\section{Beispiel}\label{sec:bsp}
\section{Experimente}\label{sec:exp}
\end{document}
```
- Struktur durch Blöcke (abstract, document),  
Gliederungsbefehle (section), Verweise (label/ref)
- Numerierung und Formatierung (layout) automatisch
- das ist *logisches* Markup (Gegensatz: visuelles)

# Warum logisches Markup?

- weil man für visuelles Markup (z.B. manuelles Einstellen der etwas größeren und dickeren Schrift für Kapitel-Überschriften) keine Zeit und keine Fachkenntnisse hat
- weil logisch markierte/strukturierte Texte leichter zu schreiben sind (Konzentration auf Inhalt, nicht Form) und besser zu bearbeiten sind (z.B. Vertauschen von Abschnitten - bei Erhaltung aller Verweise)
- aus logischem Markup erzeugte visuelle Struktur ist einheitlich, damit leichter zu lesen
- ein weiteres wichtiges Argument für  $\text{T}_{\text{E}}\text{X}$  ist der hervorragende Formelsatz (mehr dazu nächste VL)

# Warum kein WYSIWIG?

- WYSIWYG (what you see is what you get — Editieren in der formatierten (gerenderten) Darstellung)
- ein  $\text{\LaTeX}$ -Dokument schreibt man wie Quelltext (mit einem Text-Editor), das Formatieren (Kompilieren, Rendering) ist separater Vorgang
- im  $\text{\LaTeX}$ -Quelltext kann man Unterprogramme (definieren und) aufrufen, z.B. für häufige Notationen
- der Quelltext enthält Struktur-Merkmale, die in der Formatierung unsichtbar sind:  
für den Schreiber wichtig, aber nicht für den Leser  
z.B.: Verteilung auf mehrere Dateien, Namen für Labels, Kommentare (!), Zeilenumbrüche (!)

# Zeilen-Umbrüche

- Änderungs-Behandlung der Quelltextverwaltung (`git diff`) ist zeilenorientiert
- Umbrüche im Quelltext bestimmt der Autor, nach logischen Aspekten (neuer Gedanke  $\Rightarrow$  neue Zeile)
- Zeilen- (und Seiten-)umbrüche im gerenderten Dokument werden automatisch bestimmt  
( $\text{T}_{\text{E}}\text{X}$  löst Optimierungsaufgabe: den Platz gut füllen und möglichst wenige Wörter trennen)
- kleine Quelltext-Änderung (zusätzliches Wort am Absatz-Anfang) kann viele Umbrüche ändern
- aus dem gleichen Grund: niemals paragraph-fill (Absätze füllen) im Editor, auch keine überlangen Zeilen

# Technische Details zu $\text{\LaTeX}$ (grundsätzlich)

- Folgen Sie Empfehlungen (vom Autor (Lamport), von mir, von anderen)—die Sie verstehen. Vermeiden Sie Kunsthandwerk. Einfache Texte kann man einfach schreiben. Kompliziertes Design brauchen wir nicht.
- jede  $\text{\LaTeX}$ -Installation enthält Beispiele und Hinweise in

```
/usr/share/texlive/texmf-dist/tex/latex/base/small2e
/usr/share/texlive/texmf-dist/tex/latex/base/sample
```

Wörter werden durch Leerzeichen getrennt, Absätze durch Leerzeilen. Zusätzliche Leerzeichen und -Zeilen ändern die Ausgabe nicht.

# Technische Details (Schriften, Formatierung)

- Keine Kunsthandwerk! Benutzen Sie voreingestellte Schriften für Text und Überschriften, Maße für Ränder.

```
\documentclass[12pt,a4paper]{article}
```

- Hervorhebung durch `\emph{vor}` (logisches Markup)
- Listen (itemize: Punkte, enumerate: Numerierung)

```
\begin{itemize}\item Foo ... \item Bar\end{itemize}
```

- Code in der Zeile: `\verb|List<Foo> bar|`, in Absatz:

```
\begin{verbatim}
List<Foo> bar;
\end{verbatim}
```

- Kommentare im Quelltext: von `%` bis Zeilen-Ende

# Hausaufgaben

1. die Kriterien aus *How to write a great . . .* überprüfen.

- für einen von Ihnen recherchierten Artikel
- oder für einen Artikel von S. Peyton-Jones.

2. bis Ende KW47

- Struktur deutlich machen durch `\section`-Überschriften
- Einleitung schreiben (die erste `\section`). Soll bereits ein konkretes Beispiel enthalten. Von dort auf andere Abschnitte verweisen (label/ref).

# Überblick (restliche) Meilensteine

1. Ende KW 43: Projekt und Repo anlegen
2. Ende KW 45: Literatur (Bibtex)
3. Ende KW 47: Abschnitt-Überschriften, Einleitung
4. Ende KW49: je ein Beispiel für Themen aus VL KW 48 (Formelsatz, Tabelle, Grafik)
5. bis Ende KW 53: Dokumente abgeben
6. bis Ende KW 55: Gutachten schreiben
7. bis Ende KW 57: Dokumente überarbeiten

# Text- und Formelsatz

## T<sub>E</sub>X: Beispiel

D.E. Knuth: The Art of Computer Programming, Bd. 3, Addison-Wesley  
1998

### \*5.3.2. Minimum-Comparison Merging

Let us now consider a related question: What is the best way to merge an ordered set of  $m$  elements with an ordered set of  $n$ ? Denoting the elements to be merged by

$$A_1 < A_2 < \cdots < A_m \quad \text{and} \quad B_1 < B_2 < \cdots < B_n, \quad (1)$$

we shall assume as in Section 5.3.1 that the  $m + n$  elements are distinct. The  $A$ 's may appear among the  $B$ 's in  $\binom{m+n}{m}$  ways, so the arguments we have used for the sorting problem tell us immediately that at least

$$\left\lceil \lg \binom{m+n}{m} \right\rceil \quad (2)$$

comparisons are required. If we set  $m = \alpha n$  and let  $n \rightarrow \infty$ , while  $\alpha$  is fixed, Stirling's approximation tells us that

$$\lg \binom{\alpha n + n}{\alpha n} = n((1 + \alpha) \lg(1 + \alpha) - \alpha \lg \alpha) - \frac{1}{2} \lg n + O(1). \quad (3)$$

Untersch. Text/Formel; Zentrierung, Numerierung; . . .

# $\text{T}_{\text{E}}\text{X}$ und $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ : Geschichte

- Donald E. Knuth ist der Autor der Softwaresysteme
  - $\text{T}_{\text{E}}\text{X}$ : zum Satz (automatische Anordnung von geschachtelten Grafik-Blöcken auf Druckseiten, Buchstabe  $\rightarrow$  Wort  $\rightarrow$  Zeile  $\rightarrow$  Absatz  $\rightarrow$  Seite)
  - Metafont: zum Schrift-Design (Beschreibung Zeichen(sätzen) durch parametrische kubische Splines)
  - literate programming: aus *einer* Quelle werden erzeugt: Quelltext *und* Beschreibung (Kommentar)

Ziel ist: Computer-Textsatz „wie vom Fachmann“ (manuell, mit Blei-Lettern)
- $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  (Autor: Leslie Lamport) ist eine Bibliothek von  $\text{T}_{\text{E}}\text{X}$ -Unterprogrammen für logisches Markup

# Donald E. Knuth

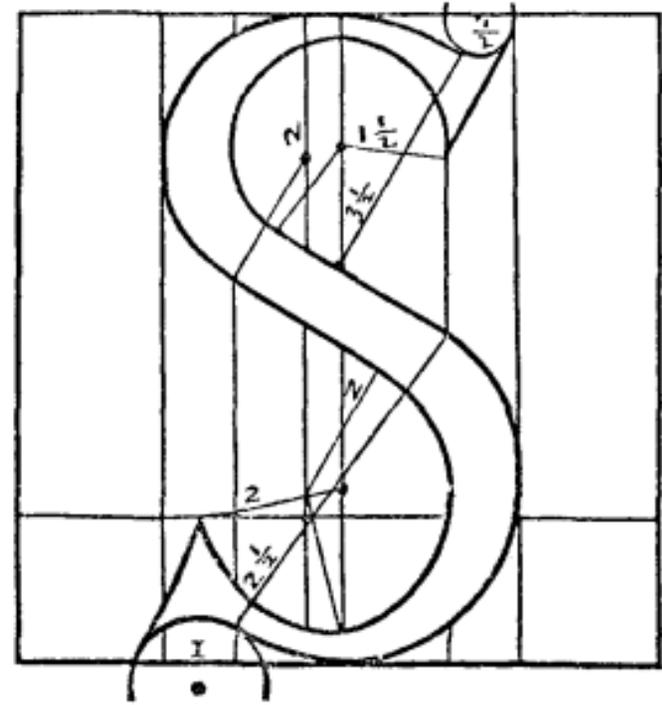
- Professor für Informatik, Stanford Univ.  
`https://www-cs-faculty.stanford.edu/~uno/`
- The Art Of Computer Programming (1968, ...)  
(... Band 3: Sortieren und Suchen, Band 4A1 (2015)  
Kombinatorische Algorithmen)
- T<sub>E</sub>X, Metafont, Literate Programming (1983, ...)
- Attribut-Grammatiken (1968) (syntaxgesteuerte  
Semantik, Grundlage f. Definition von  
Programmiersprachen, Compilerbau)
- Anwendung der Landau-Notation ( $O(f)$ , aus Analysis)  
und Erweiterung ( $\Omega, \Theta$ ) für asymptotische Komplexität

# The Letter S

SEVERAL YEARS AGO when I began to look at the problem of designing suitable alphabets for use with modern printing equipment, I found that 25 of the letters were comparatively easy to deal with. The other letter was 'S'. For three days and nights I had a terrible time trying to understand how a proper 'S' could really be defined. The solution I finally came up with turned out to involve some interesting mathematics, and I believe that students of calculus and analytic geometry may enjoy looking into the question as I did. The purpose of this paper is to explain what I now consider to be the 'right' mathematics underlying printed S's, and also to give an example of the **META-FONT** language I have recently been developing. (A complete description of **META-FONT**, which is a computer system and language intended to aid in the design of letter shapes, appears in [3, part 3].

Before getting into a technical discussion, I should probably mention why I started worrying about such things in the first place. The central reason is that today's printing technology is essentially based on discrete mathematics and computer science, not on properties of metals or of movable type. The task of making a plate for a printed page is now essentially that of constructing a gigantic matrix of 0's and 1's,

.i., con quello tondo quale ha lo suo puncto de mezo fora del quadro, longe da la inferiore linea del quadro puncto mezo. Poi largo lo circino puncti .2., ponendo una puncta dove finisti la inferiore parte del .S. qual fu facta a drita linea, cioè longe da la linea del spacio da parte drita puncti .2., e altri



puncti .4. da la [linea] inferiore del quadro. L'altra puncta longe da quella del spacio da parte sinistra puncti .2. descenderai in tondo verso man drita tanto che oionoi sopra la me-

D.E. Knuth: *The Letter S*, Mathematical Intelligencer 2(1980), Nachdruck in: Digital Typography, Stanford 1999.

# Layout mit T<sub>E</sub>X

concerned, the letter ‘A’ is a box ‘□’ and the letter ‘p’ is a box ‘□’. So the given instruction causes T<sub>E</sub>X to form the hbox



representing ‘A line of type.’ The hboxes for individual lines of type are eventually joined together by putting them into a “vbox” (vertical box). For example, you can say

```
\vbox{\hbox{Two lines}\hbox{of type.}}
```

and T<sub>E</sub>X will convert this into



i.e., Two lines  
of type.

The principal difference between T<sub>E</sub>X’s method and the old way is that metal types are generally cast so that each character has the same height and depth; this makes it easy to line them up by hand. T<sub>E</sub>X’s types have variable height and depth, because the computer has no trouble lining characters up by their baselines, and because the extra information about height and depth helps in the positioning of accents and mathematical symbols.

D.E. Knuth, The TeX Book, Addison-Wesley, 1996

# TeX als literate Program

- D.E. Knuth, TeX-The Program, Addison-Wesley, 1988

1056. As an introduction to these routines, let's consider one of the simplest cases: What happens when '\hrule' occurs in vertical mode, or '\vrule' in horizontal mode or math mode? The code in *main\_control* is short, since the *scan\_rule\_spec* routine already does most of what is required; thus, there is no need for a special action procedure.

Note that baselineskip calculations are disabled after a rule in vertical mode, by setting *prev\_depth* ← *ignore\_depth*.

```
<Cases of main_control that build boxes and lists 1056> ≡
vmode + hrule, hmode + vrule, mmode + vrule: begin tail_append(scan_rule_spec);
 if abs(mode) = vmode then prev_depth ← ignore_depth
 else if abs(mode) = hmode then space_factor ← 1000;
 end;
```

See also sections 1057, 1063, 1067, 1073, 1090, 1092, 1094, 1097, 1102, 1104, 1109, 1112, 1116, 1122, 1126, 1130.

- `texk/web2c/tex.web` Zeile 20515 ff.

Note that baselineskip calculations are disabled after a rule in vertical mode, by setting `|prev_depth:=pdf_ignored_dimen|`.

```
@<Cases of |main_control| that build...@>=
vmode+hrule,hmode+vrule,mmode+vrule: begin tail_append(scan_rule_spec);
 if abs(mode)=vmode then prev_depth:=pdf_ignored_dimen
 else if abs(mode)=hmode then space_factor:=1000;
 end;
```

<https://tug.org/svn/texlive/trunk/Build/source/>

# Notationen für Mengenlehre und Logik

- in wissenschaftlichen Arbeiten müssen Begriffe und Aussagen präzise formuliert und notiert sein, in der Informatik benutzt man dazu die Sprache der Mathematik
- **Mengen:**  $3 \notin \{x \mid x \in \mathbb{N} \wedge x > 7\}$ ,  
`$3 \notin \{x \mid x \in \mathbb{N} \wedge x > 7\}$`  
 $(A \cap B) \times (C \cup \emptyset) \subseteq D$ ,  
`$(A \cap B) \times (C \cup \emptyset) \subseteq D$`
- **Aussagenlogik:**  $\neg(p \vee q) \wedge r \rightarrow \bar{s}$ ,  
`$(\neg(p \vee q) \wedge r) \rightarrow \overline{s}$`
- **Prädikatenlogik:**  $(\neg \forall x : P(x)) \iff (\exists x : \neg P(x))$ ,  
`$(\neg \forall x : P(x)) \iff (\exists x : \neg P(x))$`
- das sind trotzdem nur Symbolfolgen,  $\text{\LaTeX}$  prüft weder Syntax (Klammern) noch Semantik (Typ, Wert).

# Lokale Struktur-Elemente (Blöcke)

- logisches Markup für Definition, Beispiel, Aussage, ...  
(vgl. `begin/end document/abstract/verbatim`)
- (empfohlen) <https://www.ctan.org/pkg/amsthm>

```
\usepackage{amsthm}
```

```
\theoremstyle{definition}
```

```
\newtheorem{defn}{Definition}
```

```
\newtheorem{satz}[defn]{Satz} % Numerierung wie defn
```

```
\theoremstyle{remark}
```

```
\newtheorem{bsp}[defn]{Beispiel}
```

- `\begin{defn}\label{defn:kreuz}`

Das Kreuzprodukt  $A \times B$  der Mengen  $A, B$  ist die Menge  $\{(a,b) \mid a \in A \wedge b \in B\}$ .

```
\end{defn}
```

- Mit der Bezeichnung aus Definition `\ref{defn:kreuz}`

# Text und Formeln

- $\text{\LaTeX}$  (und  $\text{\TeX}$ , aber wir trennen das nicht) unterscheidet
  - Text-Modus: Abschnitte, links und rechts justiert, automatischer Zeilenumbruch mit Silbentrennung, Schrift: Romanisch (Druckschrift)
  - Mathematik-Modus: Schrift: Italisch (Schreibschrift)
    - \* inline (im Fließtext) „If we set  $m = \alpha n$  and let“  
If we set `$m = \alpha n$` and let
    - \* display (eigener Absatz, zentriert, Zeilenumbruch manuell, optional Numerierung)  
merged by `\[A_1 < A_2 < \dots.\]` we
- Mathe-Operatoren funktionieren *nur* im Mathe-Modus.
- ... beim Versuch der automatischen Fehlerbehebung wird Modus umgeschaltet, das ergibt Folgefehler

# Zweidimensionale Formel-Struktur

- Indizes: unten:  $A_1: A\_1$ , oben:  $B^2: B^{\wedge} 2$ ,  
 $\binom{n+1}{2} = \sum_{k=1}^n k$ , `\binom{n+1}{2} = \sum_{k=1}^n k`
- Gruppierung durch Klammern  $\{ \quad \}$ ,  $M_{i,j}: M_{\{i, j\}}$
- Bruchstrich:  $\frac{x \cdot y}{x+y}$ : `\frac{x \cdot y}{x+y}`
- gestapelte Operatoren:  $p \xrightarrow{\epsilon} q$   
`p \stackrel{\epsilon}{\longrightarrow} q`
- Matrizen:  $\begin{pmatrix} 1 & 20 \\ 300 & 4 \end{pmatrix}$   
`\begin{pmatrix} 1 & 20 \\ 300 & 4 \end{pmatrix}`  
nächste Spalte: `&`, nächste Zeile: `\\`

# Arrays und Tabellen

- im Mathe-Modus:

|    |            |      |
|----|------------|------|
| 1  | $\sqrt{x}$ | 3333 |
| 44 |            | 6    |

```
\begin{array}{r|cc}1 & \sqrt{x} & 3333 \\44 & & 6 \\ \hline\end{array}
```

- im Text-M.:

| Modus | Silbentrennung                                   |
|-------|--------------------------------------------------|
| Mathe | nein ( $\sin(x)$ hat keine Silben)               |
| Text  | ja, wenn passendes Trennverzeichnis benutzt wird |

```
\begin{tabular}{l||p{0.5\textwidth}}
 Modus & Silbentrennung \\ \hline
 Mathe & nein ($\sin(x)$ hat keine Silben)
 Text & ja, wenn passendes Trennverzeichnis
\end{tabular}
```

- Spalten-Definitionen (für Ausrichtung d. Elem.): `l,c,r,p`

# Mehrzeilige und numerierte Gleichungen

- Ausrichtung wie nach `\begin{array}{rcl}`:

$$0 \cdot y = 0 \tag{1}$$

$$(1 + x) \cdot y = y + x \cdot y \tag{2}$$

Wir verwenden Gleichung (1).

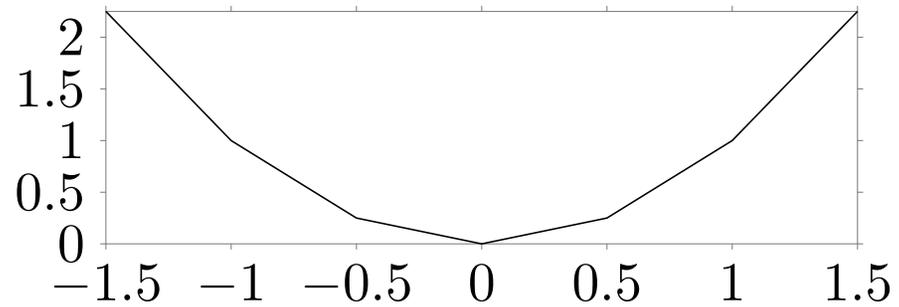
```
\begin{eqnarray}
 0 \cdot y &=& 0 \label{eq:0y} \\
 (1+x) \cdot y &=& y + x \cdot y
\end{eqnarray}
```

Wir verwenden Gleichung `\eqref{eq:0y}`.

- ohne Numerierung: `eqnarray*`

# Grafiken: Zeichen-Technik (Beispiel)

- programmatische Beschreibung von (Vektor)grafiken  
Till Tantau et al.: *The TikZ and PGF Packages, Manual for Version 3.1.9a-2022*, <https://tikz.dev/>,  
gesehen 1. 12. 2022



- Bsp. aus Abschnitt 80.2:

```
\usepackage{pgf} \usepackage{tikz}
\usetikzlibrary {datavisualization}
\begin{tikzpicture}\datavisualization[scientific axes
 x axis={length=10 cm}, visualize as line]
 data { x, y
 -1.5, 2.25
```

# Externe Grafiken

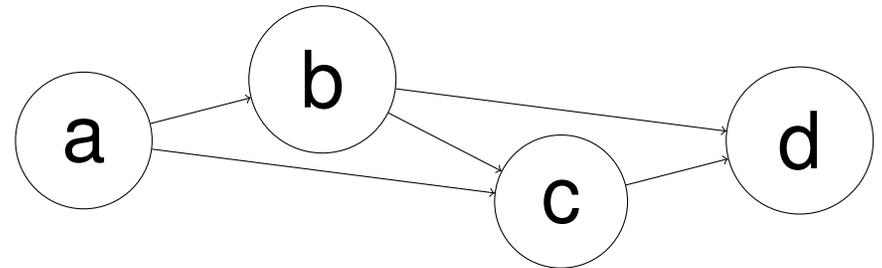
- Bsp: aus Datei `bild.png`

```
\centerline{
 \includegraphics[width=0.75\textwidth]{bild}
}
```

- OK: für Grafiken mit primären Daten (z.B. Foto eines Versuchsaufbaus, Spektrogramm eines Signals)
- meist nicht OK: falls Grafik nur entsteht, weil man das Zeichnen mit  $\text{\LaTeX}$  vermeiden wollte:  
die Schrift (und erst recht Formel-Schrift) innerhalb der Grafik paßt dann oft nicht zum Text  
ist evtl. schlecht lesbar (gerastert)

# Externe Grafik-Werkzeuge (Beispiel)

- ```
digraph g {a -> b -> d;  
a -> c -> d; b -> c;}
```



Emden R. Gansner, Stephen C. North: *An open graph visualization system and its applications to software engineering*, *Softw. Pract. Exp.* 30(11): 1203-1233 (2000), <https://graphviz.org/>

- ```
dot2tex -ftikz --codeonly automat.dot > automat.tex
```

Kjell M. Fauske: *dot2tex - A Graphviz to LaTeX converter*, <https://dot2tex.readthedocs.io/>, gesehen 1. Dez. 22

- ```
\begin{tikzpicture}[every node/.style = {draw,circle}]  
\input{automat.tex} \end{tikzpicture}
```

Grafiken: allgemein

- Grafiken können nützlich sein, aber auch unzweckmäßig oder irreführend. Visualisierung ist ein eigenes Forschungs- und Lehrgebiet. Eine klassische Quelle ist Edward Tufte: *The Visual Display of Quantitative Information*, Cheshire, 2001. vgl. https://www.edwardtufte.com/tufte/books_vdqi ... detailed analysis of how to display data for precise, effective, quick analysis.
- klare Beziehung zwischen Grafik und Daten herstellen:
 - durch Beschriftung in der Grafik (z.B. Achsen),
 - und durch Erläuterung im Text (z.B. Meß-Methoden)

Grafiken anordnen und mit Text verbinden

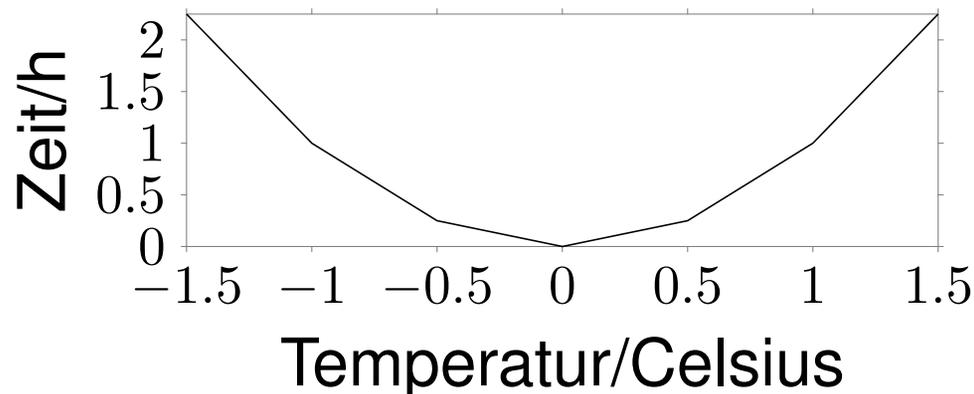


Abbildung 1: Laufzeit als Funktion der Außentemperatur

- ```
\begin{figure} \centering \begin{tikzpicture} \datavisualization[x axis={label={Temperatur...}} \end{tikzpicture} \caption{Laufzeit als ...} \label{fig:lauf} \end{figure}
```
- *gleitendes* (floating) Objekt, automatische Platzierung vermeidet Seitenumbrüche. Ebenso für `table`

# Bezeichner (Wörter) in Formeln

- math. Bezeichner müssen *immer* in Mathe-Modus:  
falsch:  $x$  und  $y$  ( $x$  und  $y$ ); r.:  $\$x\$$  und  $\$y\$$  ( $x$  und  $y$ )  
Begründung: konsistente Erscheinung (Schrift-Art)
- traditionell gibt es in der Mathematik keine mehrbuchstabigen Bezeichner, Begründung: damit spart man in Produkten  $xyz$  die Punkte ( $x \cdot y \cdot z$ )
- Ausnahmen: Namen von Funktionen (sin, log, exp, ...)   
richtig:  $\sin(x) = 0$ ,  $\$\sin(x) = 0\$$   
falsch:  $\sin(x) = 0$ ,  $\$sin(x) = 0\$$   
Begründung (siehe oben): *sin* bedeutet  $s \cdot i \cdot n$ .
- eigene Funktionsnamen:  $\text{foo}(x)$ :  $\$\text{textrm}{foo}(x)\$$

# Benutzerdefinierte Bausteine

- häufig vorkommende Teilausdrücke

```
$\textsf{foo}(x) = 1/\textsf{foo}(y)$
```

sollten heraus-faktoriert werden:

```
\newcommand{\foo}{\textsf{foo}}
```

```
$_foo(x) = 1/_foo(y)$
```

- alle diese Definitionen am Text-Anfang sammeln
- übliche mathematische Zeichen: gibt es schon, Bsp.  $\mathbb{R}$

```
\usepackage{amssymb}\newcommand{\RR}{\mathbb{R}}
```

- Suche nach Zeichen: [https:](https://detexify.kirelabs.org/classify.html)

```
//detexify.kirelabs.org/classify.html
```

# Meilenstein Textsatz (Ende KW 49)

- wenigstens eine `newtheorem`-Umgebung (Bsp: Beispiel) definieren, benutzen, markieren (`label`) und darauf verweisen (`ref`).
- wenigstens eine formale Spezifikation für wenigstens einen der Begriffe aus Ihrem Text schreiben, dabei Mathe-Modus benutzen.
- wenigstens eine Tabelle (Umgebung `tabular`) schreiben, in Umgebung `table` einbetten, mit Legende (`caption`), Marke (`label`) und Verweis (`ref`) von außen.
- wenigstens eine Grafik (`tikzpicture` oder `includegraphics`) in Umgebung `figure` einbinden, mit Legende, Marke und Verweis. Wenn nicht eigene Grafik, dann mit Quellenangabe.





# Zwischenstand: Dokument-Entwürfe

## Plan

- KW 50 (heute): Kommentare und Diskussion zu Textstruktur und -Gestaltung, an Beispielen
- bis Di. KW 54: Dokumente abgeben: Quellen (tex, bib, kein bbl u.ä.) im Repo (master!), Artefakt (pdf) in CI
- KW 54: VL zu Peer Review,  
bis Di. KW 56: Gutachten schreiben
- KW 56: VL zum Präsentieren,  
bis So. KW 57: Dokumente überarbeiten und abgeben
- KW 61 (noch nicht amtlich): Prüfung (Präsentationen)

# Dokument-Struktur allgemein

- soll aussehen wie die zitierten Papiere (von 10 Seiten Umfang): Titel, Abstrakt, Text, Literatur direkt nacheinander, ohne künstlichen Platzverbrauch.
- *keine* separate Titelseite, Verzeichnisse (Inhalt, Abbildung, Tabelle) - außer Literatur, die muß sein.
- Abschnitte (`section`) verwenden,  
Unterabschnitte (`subsection`) kaum (keinesfalls in Einleitung, Schluß)
- bei Bachelorarbeit ist das alles anders—die ist auch deutlich länger.

# Einzelkritik (Reihenfolge ist zufällig))

1. Wörter mit Bindestrich werden nicht automatisch getrennt. `Trenn\ -Po\ -si\ -tio\ -nen` explizit angeben oder Trennung vermeiden durch Umformulieren
2. für jede Abbildung (Bsp. Cookie-Banner auf einer Webseite) Quelle angeben
3. In dieser Arbeit soll ein Überblick über ... verschafft und Schwachstellen beleuchtet werden: grammatisch falsch (Schwachstellen: Plural *sollen* ), holprig, unpersönlich.  $\Rightarrow$  Wir geben ... und beleuchten .....
4. Literatur: [Union, 2011] Union, E. (2011) General Data Protection Law.  $\Rightarrow$  Autor? Ort? (Genaue Dokument-Bezeichnung.)

5. keine Fußnoten! nicht für Quellen, denn die stehen in der Bibliographie, nicht für Aussagen: wenn diese wichtig sind, dann sollen sie im Text stehen, wenn unwichtig, dann nirgendwo.

6. Markup für Auszeichnung des definierten Begriffs: in eckigen Klammern, als optionales Argument für die `definition`-Umgebung

```
\newtheorem{definition}{Definition}
```

```
...
```

```
\begin{definition}[Foo]\label{def:Foo} Ein Foo ist
```

7. richtige Anordnung von Literatur-Verweisen:

```
... ist quadratisch~\cite{foobar}.
```

- vor dem Satzschlußpunkt (nicht danach),

- vom vorausgehenden Wort durch ~ (getrennt  
Dieses ~ ist stabiles Leerzeichen (unbreakable space),  
dort wird nicht umgebrochen.

### 8. Bindestriche, Kommas, Wortwahl!

EBP und ESP sind CPU Register welche ...

⇒ EBP und ESP sind CPU-Register, die ...

### 9. unterschiedliche Länge und Bedeutung von Strichen im Text-Modus:

(a) kurz (-): Trenn-Strich Trenn-Strich

(b) mittel (–): Bereiche, Seiten 5–15 5--15

(c) lang (—): Gedankenstrich, *ohne* umgebende  
Leerzeichen benutzen! Besser—ganz vermeiden.

im Mathe-Modus nur – als Minus ,  $x - y$ ,  $\$x-y\$$

0. Anführungszeichen: falsch (englisch): `\`ba'` "bar"

deutsch richtig: `\glqq foo\grqq „foo“`

evtl. besser `\emph{foo}` *foo*

1. die if-Klausel betreten: ist unklar. den Zweig betreten?  $\Rightarrow$   
Problem vermeiden: Anweisung im Quelltext markieren,  
im Text die Marke benutzen

2. `\context: Person\`

$\Rightarrow$  mehrbuchstabile Bezeichner im Mathemodus müssen  
in `\textrm{context}` ...

Zeilenumbrüche in Formel:  $\Rightarrow$  (eqn)array verwenden

3. laeuft haeufig ohne:  $\Rightarrow$  Umlaute verwenden!

4. Zum Beispiel könnte eine formale Spezifikation des

...-Algorithmus so aussehen,

Kann man folgende Formel verwenden, um ... zu berechnen:

⇒ könnte, aber tut es dann doch nicht? Nicht um die Sache herumreden! Der ...-Algorithmus erfüllt die Spezifikation ....

5. table, figure: caption immer *unter* das Bild, nicht über.

6. ... werden durch das sogenannte ... *erschaffen*. ⇒ schwülstige Sprache ersetzen, besser: hergestellt, angelegt, erzeugt.

7. Energiekonsum ⇒ Energieverbrauch. Da man Energie nicht verbrauchen kann, nur umwandeln (hier: in Wärme), mglw. besser: E-Aufwand?

8. Mathe-Modus: passende Klammern-Größe  $\left(\frac{x}{y} - 1\right) \cdot z$   
`\left(\frac{x}{y}-1\right)\cdot z`

9. Die Behebung der Sicherheitslücke ist eine Performance-kritische Abwägung.  $\Rightarrow$  eine Behebung ist eine Abwägung? Auf Deutsch sagen, was gemeint ist. Wenn die Lücke geschlossen wird, sinkt die Leistung?

10. Die Menschheit sammelt und speichert seit tausenden von Jahren ...  $\Rightarrow$  kürzer! Möglichst früh im Text mit möglichst konkretem Beispiel schreiben, worum es tatsächlich geht.

11. ein Absatz über mehr als eine halbe Seite  $\Rightarrow$  aufteilen in mehrere Absätze.

2. Abgabe immer im master-Branch! andere Zweige gern benutzen, aber nur intern.
3. immer prüfen, ob `make` auf eigenem Rechner *und in CI* funktioniert.

```
make: *** No rule to make target 'main.bib', needed by 'main'.
Cleaning up project directory and file based variables
ERROR: Job failed: exit code 2
```

```
! LaTeX Error: File `google_quantum_nature_cover_ar
```

# Begutachtung (Peer Review)

## Begriff, Ziele, Quellen

- Begutachtung durch Fachleute (peers) *sichert* und *verbessert* Qualität von Publikationen
- in LV, bei BA, MA: Begutachtung durch Dozent, Prüfer.  
zur Übung (in dieser LV): durch andere Studenten
- im Berufsleben: Sie schreiben technische Dokumente für Kollegen (Vorgesetzte, Anwender)—und lesen diese
- Ian Parberry: *A Guide for New Referees in Theoretical Computer Science*, Information and Computation 112(1) 1994, *A Form for Referees in Theoretical Computer Science*, SIGACT News 25(4), 1994.

<https://ianparberry.com/pubs/>

# Ziele (für wissenschaftliche Arbeiten)

- (nach Parberry) Gutachter soll (im Auftrag des Herausgebers) diese Qualitätsmerkmale überprüfen:
  - 1. Korrektheit: für Resultate und Beweise
  - Neuheitswert: neue Ergebnisse, neue Methode, oder neuartige Kombination von bekannten Methoden
  - Bedeutung (für das Fachgebiet), Interesse (anderer Fachleute), Zeitnähe
  - Kürze (effizienter Ausdruck), Verständlichkeit (für Nicht-Spezialisten)
  - Eleganz (der Methoden-Benutzung), Lesbarkeit, Stil
- es nützt der Fach- und Nachwelt, wenn Arbeiten guter Qualität publiziert werden (und schlechte nicht)
- Autor kann Merkmale verbessern, Hrsg. kann ablehnen,

# Empfehlungen für Abschlußarbeiten

- Klaus Hänßgen: *Hinweise für die Erarbeitung von schriftlichen Arbeiten* <https://www.imn.htwk-leipzig.de/~haenssge/lehre/hinweise-s-arb-v3.htm>, HTWK Leipzig, 2007 (gelesen: 12. Jan. 2023)  
besonders: strukturelle Fragen und häufig vorkommende Fehler
- Johannes Waldmann: *Diplomarbeiten richtig vorbereiten, ausführen und verteidigen*, <https://www.imn.htwk-leipzig.de/~waldmann/edu/diplom/faq/>, HTWK Leipzig, 2020 (gelesen: 12. Jan. 2023)  
besonders: Gestaltung der Arbeit, Textbausteine für Gutachten

# Ziele (für technische Dokumentation allgemein)

- nicht nur Ihre BA/MA (das ist das direkte Ziel unserer LV), sondern auch vorher und nachher:

Entwurfsdokumente (global—für ein gesamtes Softwareprojekt, lokal—für Komponenten),  
Schnittstellenbeschreibungen, Anwenderdokumentation,  
Anwendungs-Programmierer-Dokumentation (API),  
Hilfe-Texte (, Quelltexte)

- die weitgehend übereinstimmenden Ziele sind: *Klarheit, Kürze, Klang* (so der Titel eines Buches von Peter Rechenberg: , Hanser 2006 <https://www.hanser-elibrary.com/doi/10.3139/9783446409576.002> )

# Ziele beim Schreiben, beim Begutachten

- (warum reden wir jetzt noch über die Ziele, wenn die Dokumente schon geschrieben sind?)
- beim Schreiben der Gutachten (bis Mitte KW 56)  
sollen Sie sehen, wie andere Autoren vorgehen,  
und bewerten, wie gut diese die Ziele erreichen
- die Gutachten über Ihr eigenes Dokument  
geben eine reale Einschätzung (versteht der Leser, was  
Sie sagen)  
und sind nützlich zur Verbesserung (bis Ende KW 57)

# Taxonomie von Gutachten (Parberry)

- Themenbereich verfehlt
- Resultate schon bekannt (an anderer Stelle publiziert)
- die betrachtete Frage ist zu einfach
- die Ergebnisse sind nicht interessant genug
- enthält schwere (irreparable) Fehler
- langweilig (kann komprimiert werden)
- langsam (hoher technischer Aufwand mit wenig Ergebnis)
- schlecht geschrieben (unfertig, unklar)
- gut geschrieben (kann mit geringen Änderungen akzeptiert werden)
- *The Perfect Paper*

# Struktur und Form eines Gutachtens

- 1. Gesamt-Eindruck und Empfehlung des Gutachters  
(kurze Inhaltsangabe in Worten des Gutachters!)
  - 2. Kritik an Inhalt und Präsentation
  - 3. Änderungsvorschläge
  - 4. Tipp- (und Formatierungs-)Fehler
- 
- Gutachten als Text (kein Latex). Gliederung durch Leerzeilen, evtl. Markdown (#) zur Auszeichnung von Überschriften
  - für jede Aussage im Gutachten: konkreter Textbezug:
    - Verweis auf Gliederung (Abschnitt 2, Abbildung 3)
    - Seiten-Nr., Zeilen-Nr. (negative Nr.: von unten)

# Ein Beispiel-Gutachten

- für Geser, Hofbauer, Waldmann: *Sparse Tiling Through Overlap Closures for Termination of String Rewriting*, FSCD 2019 `https://drops.dagstuhl.de/opus/volltexte/2019/10773/`
- **drei Gutachten** (`review-*.text`) in `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox/-/tree/master/paper/tiling/fscd19/paper`
- **Bsp: einige Zeilen aus Review 3**  
Overall evaluation: 2 (accept),  
Overall evaluation: The main idea is to ...  
Pros: ... , Cons: ...  
Proposition 6.5: mixed: should be defined, ...

# Zuordnung von Gutachtern bei Konferenzen

- gegeben sind:
  - Menge  $G$  von verfügbaren Gutachtern
  - Menge  $P$  von eingereichten Papieren,
  - Konflikt-Relation  $K \subseteq G \times P$ :  $(g, p) \in K$ , falls  $g$  im Interessenkonflikt mit Autoren von  $p$  steht,
  - Präferenz-Funktion  $m : G \times P \rightarrow \{\text{nein, vielleicht, ja}\}$   
 $m(g, p) =$  wie sehr  $g$  ein Gutachten für  $p$  schreiben will
- gesucht ist: Relation  $S \in G \times P$  („schreibt Gutachten für“),
  - die Konflikte vermeidet ( $K \cap S = \emptyset$ ),
  - Arbeit gleich verteilt (z.B.,  $\forall g : |S(p)| \leq 6$ )
  - je 3 Gutachten garantiert ( $\forall p : |S^-(g)| \geq 3$ )
  - (gewichtete) Summe der nein und vielleicht minimiert

# Zuordnung von Gutachtern, Erläuterungen

- Interessenkonflikt z.B., falls: Gutachter ist Autor, ist verwandt mit Autor, arbeitet am gleichen Institut  
Konflikte werden vom Gutachter selbst deklariert
- Zahlen (Bsp: FSCD 2023) 30 Personen im PC (Programmkomitee), 60 Einreichungen (geschätzt), also 180 Gutachten benötigt, also jeder 6 Gutachten. Davon ca. 4 durch Sub-Gutachter.
- Lösungsverfahren der Optimierungs-Aufgabe sind Teil von Konferenz-Management-Systemen, z. B.  
<https://easychair.org/> (oft benutzt, aber nicht open source), <https://github.com/kohler/hotcrp/> (OS, selbst-hosting möglich)

# Der Nutzen mehrerer Gutachten

- zwei Gutachter bemängeln am gleichen Dokument je 10 Fehler.

Zwei der Fehler werden von beiden Gutachtern genannt.

Wieviele Fehler enthält das Dokument tatsächlich?

- Ansatz: Das Dokument enthält  $x$  Fehler. (unbekannt)

Die Wahrscheinlichkeit, daß ein Gutachter einen Fehler bemerkt, ist  $p$ . (unbekannt)

Die Wsk, daß ein Fehler durch beide bemerkt wird, ist  $p^2$

(wenn die Gutachten unabhängig sind und die Fehler alle gleich schwer zu finden. Aber trotzdem . . .)

# Bemerkungen zur Begutachtung in der LV

- für 20 Papiere: 60 Gutachten bestellt.  
Kriterien bei Zuteilung (Gutachter  $G$  für Papier  $P$ ):
  - Expertise (Gruppe von  $G$  hat ähnliches Thema wie  $P$ )
  - Streuung (für ein  $P$  nicht zwei  $G$  aus gleicher Gruppe)
- 48 Gutachten erhalten, zugestellt am 26. Jan.
- nach Stichproben: die Gutachten sind besser (prägnanter, genauer) als die Papiere (es ist leichter, eine fremde Arbeit zu kritisieren als die eigene)
- zahlreiche Formfehler (Dateiformat, Titel nicht genannt)
- Papiere überarbeiten bis Ende KW 57, dann Bewertung (und Publikation, vor der Präsentation)







# How To Give A Great Talk

## Ziel, Quelle

- Der Vortrag (auf einer Konferenz) wirbt dafür, das Paper zu lesen. Er ersetzt das Lesen aber nicht.  
Der Vortrag soll die wesentliche Aussage des Papers *nennen* und *durch Beispiele verdeutlichen*.
  - 20 Prozent: Motivation
  - 80 Prozent: wesentliche Idee/Aussage — *illustriert*  
besser wenige Aussagen gründlich, als viele oberflächlich
- Simon Peyton Jones, John Launchbury, John Hughes,  
SIGPLAN Notices 28(11), Nov 1993.

<https://www.microsoft.com/en-us/research/academic-program/give-great-research-talk/>

# Beispiele, Beispiele, Beispiele

- durch Beispiele:
  - das Thema zeigen  
(welche Aufgabe soll gelöst werden)
  - den Lösungsplan zeigen  
(mit welcher Methode wird die Aufgabe gelöst)
  - die Realisierung des Plans zeigen  
(die Implementierung der Methode)
  - die Grenzen des Verfahrens zeigen  
(was wurde erreicht, was wurde nicht erreicht)
- (JW) diese Beispiele müssen natürlich bereits im Papier stehen, werden für den Vortrag nur extrahiert  
auch deswegen: Folien mit Latex schreiben

# Weglassen: die sinnlose erste und letzte Folie

- die Gliederung!  
(ist am Anfang unverständlich und später im Weg)
- verwandte Arbeiten  
(müssen im Paper zitiert werden, auf der Folie kurz genannt, aber deren Präsentation verdeckt die wesentliche/eigene Idee)
- technische Details (denn die stehen im Paper)
- (JW) die letzte „Danke/Fragen?“-Folie  
das soll man sagen, aber nicht schreiben!  
die letzte Folie bleibt zu Beginn der Diskussion stehen,  
enthält wesentliche Aussagen, konkrete Fragen

# Richtig präsentieren

- die Folien (auswendig) kennen
- die ersten Sätze (aber nur diese) auswendig lernen
- das Publikum ansehen (nicht den Rechner oder die Leinwand)
- auf Fragen achten, diese einladen und beantworten
- pünktlich beginnen, pünktlich schließen (zur Not Sprung zur sinnvollen letzten Folie)
- (JW) der Versammlungsleiter (*session chair*, oft ein Mitglied des Programm-Komitees) erteilt pünktlich das Rede-Recht und entzieht es

# Folien richtig gestalten

- optisch: max. 10 Zeilen, kein sinnloser leerer Platz
- zeitlich (meine Erfahrung): Konferenz: 2 min pro Folie (10 F/20 min)

NB: Vorlesung: 5 min pro Folie (16 F/90 min) (wg. Rechnungen an der Tafel, im Computer, Interaktion)

- wir benutzen Latex mit `documentclass{beamer}`

**Vorlage:** `https://gitlab.imn.htwk-leipzig.de/waldmann/wki-ws22/-/tree/master/slides`

damit können Elemente des Papiers übernommen (kopiert) werden (Grafiken, wichtige Definitionen und Aussagen—nicht: erklärender Text)

# Beispiele Folien (Latex)

- Dennis Renz, Sibylle Schwarz, J. W.: *Check Your (Students') Proofs - With Holes*, WFLP 2020
  - **Papier:** <https://arxiv.org/abs/2009.01326> ,
  - **Folien, mit Latex-Quellen** <https://www.imn.htwk-leipzig.de/~waldmann/talk/20/wflp/>,  
<https://gitlab.imn.htwk-leipzig.de/waldmann/cyp/-/tree/paper-wflp20/doc/wflp20/talk>
- Alfons Geser, Dieter Hofbauer, J.W.: *Approximating Relative Match-Bounds*, WST 2022
  - <https://www.imn.htwk-leipzig.de/~waldmann/talk/22/wst-approx/>, <https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox/-/tree/master/paper/epsilon/wst22/talk>

# Weitere Hinweise zum Präsentieren

- S. Schwarz, HTWK Leipzig: *Verteidigung von Master-, Bachelor- und Projektarbeiten*

<https://www.imn.htwk-leipzig.de/~schwarz/lehre/arbeiten/vortragsrichtlinien.html>  
(gelesen 31. 1. 2023)

- J. Waldmann, HTWK Leipzig: *Diplomarbeiten richtig vorbereiten, ausführen und verteidigen,*

<https://www.imn.htwk-leipzig.de/~waldmann/edu/diplom/faq/> (gelesen 31. 1. 2023)

# Organisatorisches (Vorträge)

- jede Gruppe (3 Personen) schreibt *einen* gemeinsamen Foliensatz mit klarem Bezug auf das Papier
- einreichen (wie Papier) durch Commit in *gitlab.imn*, kurz (1 Tag) vor Präsentation (Folien werden von *einem* Rechner aus präsentiert, um Umbau zu vermeiden)
- jede Präsentation wird aufgeteilt (je Person 5 min, d.h., maximal 3 Folien)
- Diskussion (insgesamt 5 min): jede Person erhält Fragen zum *gesamten* Vortrag

# Organisatorisches (insgesamt)

- Papiere bis Ende KW 57 überarbeiten  
(ansonsten gilt die Bewertung aus den Gutachten)
- Publikation der Papiere, vor den Vorträgen
  - als PDF, nur an TN der LV verteilt
  - Autoren anonymisiert?
  - mit Gutachten?
- Folien bis Ende KW 60 (Präsentationen Anfang KW 61)  
Format: Latex, Fragen dazu: Übungen KW 57  
Einreichung: gitlab.imn (wie Papier)  
Befristung und Bewertung: durch Koll. Hodgson

# Zusammenfassung, Ausblick

- das Fach Wissenschaftskommunikation in der Informatik:  
*wichtig, nützlich, interessant, aufwendig*  
das gilt für Student und Dozent gleichermaßen
- brauchen wir WKI in Zukunft (= wenn chatgpt die Welt regiert)? — Ja, umso mehr müssen wir: Quellen bewerten, Eigenes präsentieren.
- Auswertung der Umfragen: <https://www.imn.htwk-leipzig.de/~waldmann/edu/ws22/wki/umf/>
- ich wünsche uns spannende Vorträge (KW 61)
- wir sehen uns schon sehr bald wieder: SS 23
  - Fortgeschrittene Programmierung
  - Automaten und formale Sprachen