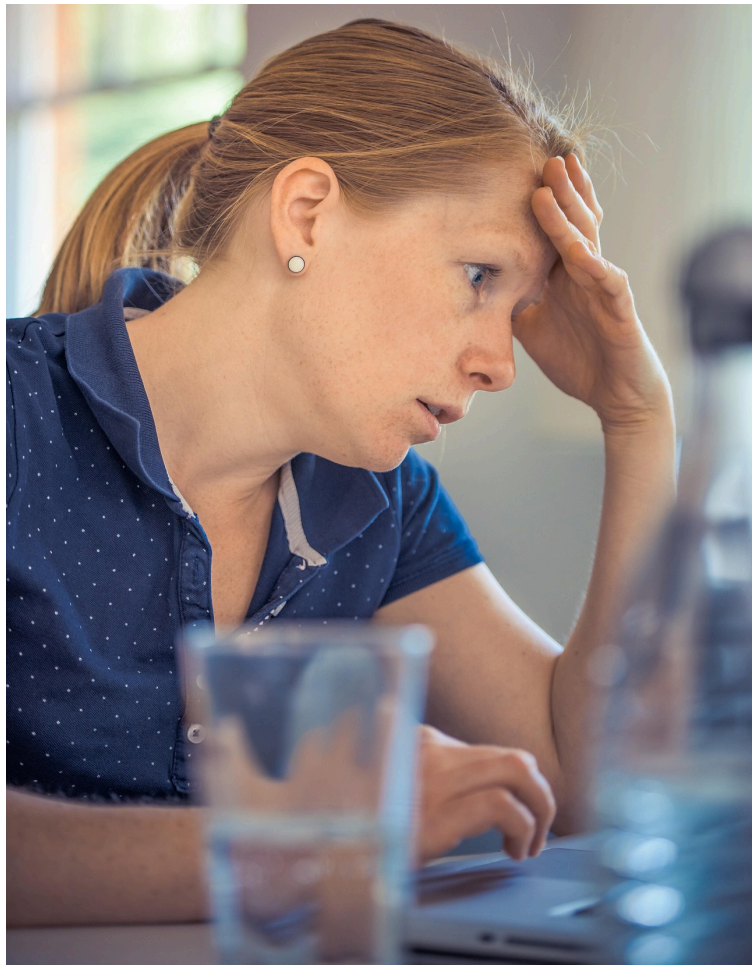
Inkrementelle Kompilierung

Auf den Spuren des Typst-Compilers

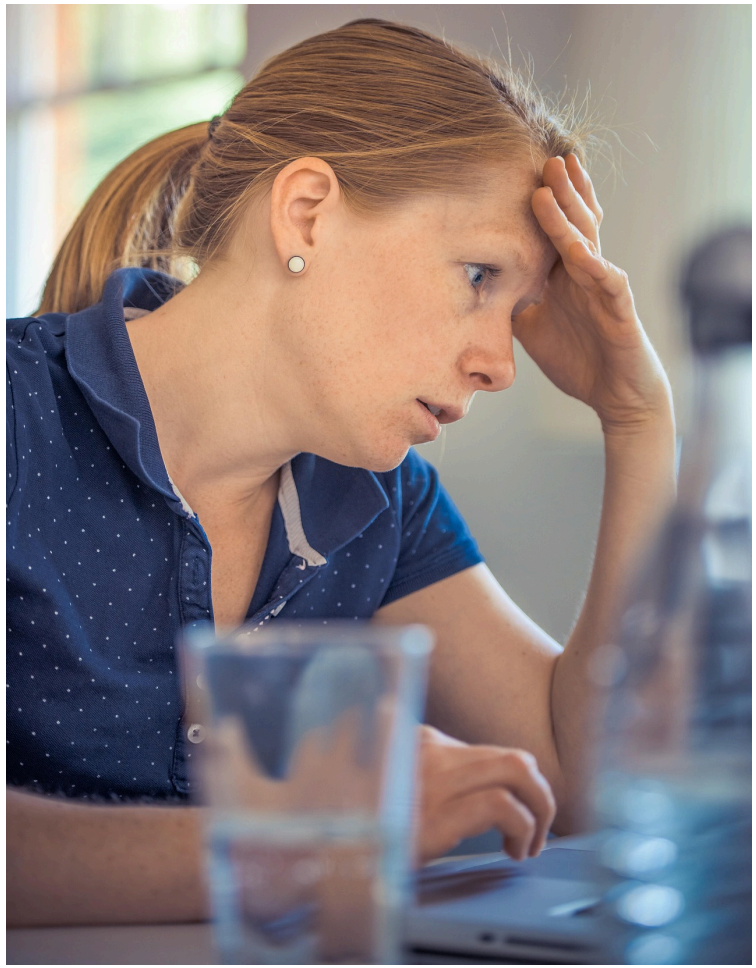
Tristan Pieper — tristan.pieper@uni-rostock.de

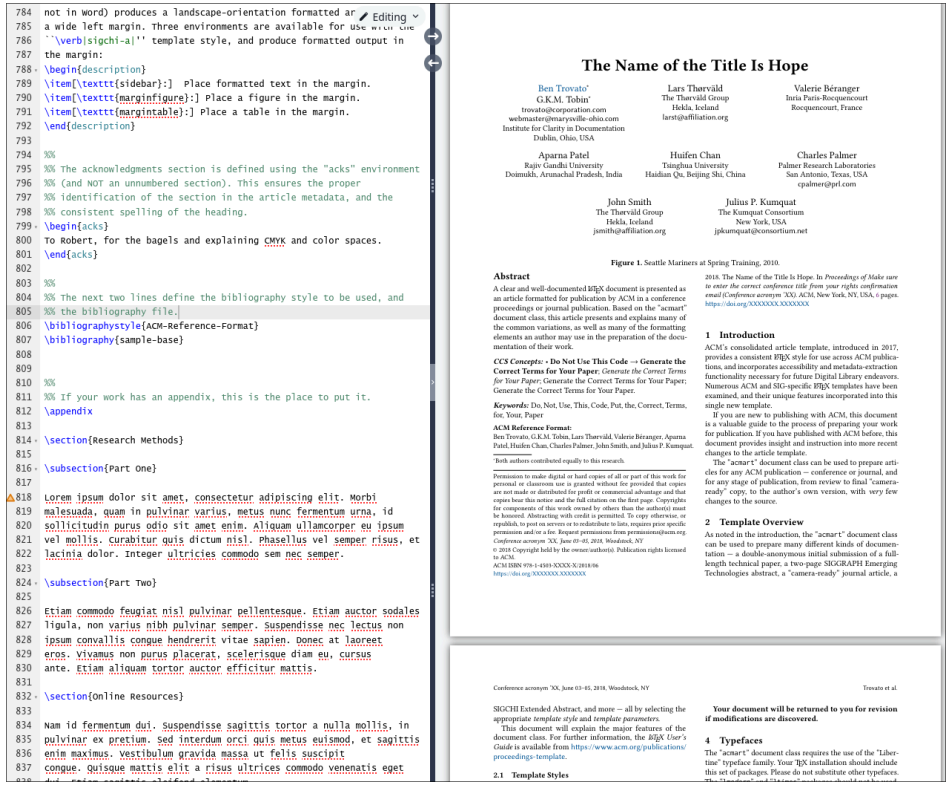
Dienstag, 16.06.2026

[1]



[1]




Editing
not in Word) produces a landscape-orientation formatted ar
a wide left margin. Three environments are available for use with the
``\verb|sigchi-a|'' template style, and produce formatted output in
the margin:
\begin{description}
\item[\texttt{sidebar}:] Place formatted text in the margin.
\item[\texttt{marginfigure}:] Place a figure in the margin.
\item[\texttt{margintable}:] Place a table in the margin.
\end{description}
%%
%% The acknowledgments section is defined using the "acks" environment
%% (and NOT an unnumbered section). This ensures the proper
%% identification of the section in the article metadata, and the
%% consistent spelling of the heading.
\begin{acks}
To Robert, for the bagels and explaining CMYK and color spaces.
\end{acks}
%%
%% The next two lines define the bibliography style to be used, and
%% the bibliography file.
\bibliographystyle{ACM-Reference-Format}
\bibliography{sample-base}
%%
%% If your work has an appendix, this is the place to put it.
\appendix
\section{Research Methods}
\subsection{Part One}
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi
malesuada, quam in pulvinar varius, metus nunc fermentum urna, id
sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum
vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et
lacinia dolor. Integer ultricies commodo sem nec semper.
\subsection{Part Two}
Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales
ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non
ipsum convallis congue hendrerit vitae sapien. Donec at laoreet
eros. Vivamus non purus placerat, scelerisque diam eu, cursus
ante. Etiam aliquam tortor auctor efficitur mattis.
\section{Online Resources}
Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in
pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis
enim maximus. Vestibulum gravida massa ut felis suscipit
congue. Quisque mattis elit a risus ultrices commodo venenatis eget
The Name of the Title Is Hope
Ben Trovato*
G.K.M. Tobin*
trovato@corporation.com
webmaster@maryville-obin.com
Institute for Clarity in Documentation
Dublin, Ohio, USA
Lars Thørväld
The Thørväld Group
Hekla, Iceland
larst@affiliation.org
Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France
Aparna Patel
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India
Huifen Chan
Tsinghua University
Haidian Qu, Beijing Shi, China
Charles Palmer
Palmer Research Laboratories
San Antonio, Texas, USA
cpalmer@prl.com
John Smith
The Thørväld Group
Hekla, Iceland
jsmith@affiliation.org
Julius P. Kumquat
The Kumquat Consortium
New York, USA
jpkumquat@consortium.net
Figure 1. Seattle Mariners at Spring Training, 2010.
Abstract
A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.
CCS Concepts: • Do Not Use This Code → Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.
Keywords: Do, Not, Use, This, Code, Put, the, Correct, Terms, for, Your, Paper
ACM Reference Format:
Ben Trovato, G.K.M. Tobin, Lars Thørväld, Valerie Béranger, Aparna Patel, Huifen Chan, Charles Palmer, John Smith, and Julius P. Kumquat.
*Both authors contributed equally to this research.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX
2018. The Name of the Title Is Hope. In Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX). ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX
1 Introduction
ACM's consolidated article template, introduced in 2017, provides a consistent LaTeX style for use across ACM publications, and incorporates accessibility and metadata-extraction functionality necessary for future Digital Library endeavors. Numerous ACM and SIG-specific LaTeX templates have been examined, and their unique features incorporated into this single new template.
If you are new to publishing with ACM, this document is a valuable guide to the process of preparing your work for publication. If you have published with ACM before, this document provides insight and instruction into more recent changes to the article template.
The "acmart" document class can be used to prepare articles for any ACM publication — conference or journal, and for any stage of publication, from review to final "camera-ready" copy, to the author's own version, with very few changes to the source.
2 Template Overview
As noted in the introduction, the "acmart" document class can be used to prepare many different kinds of documentation — a double-anonymous initial submission of a full-length technical paper, a two-page SIGGRAPH Emerging Technologies abstract, a "camera-ready" journal article, a
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY Trovato et al.
SIGCHI Extended Abstract, and more — all by selecting the appropriate template style and template parameters.
This document will explain the major features of the document class. For further information, the LaTeX User's Guide is available from https://www.acm.org/publications/proceedings-template.
2.1 Template Styles
Your document will be returned to you for revision if modifications are discovered.
4 Typefaces
The "acmart" document class requires the use of the "Libertine" typeface family. Your TEX installation should include this set of packages. Please do not substitute other typefaces.

Probleme mit LaTeX

1. 40 Jahre alte Syntax

Probleme mit LaTeX

1. 40 Jahre alte Syntax
2. kryptische Fehlermeldungen

Probleme mit LaTeX

1. 40 Jahre alte Syntax
2. kryptische Fehlermeldungen
3. langsame Compilezeit

Probleme mit LaTeX

1. 40 Jahre alte Syntax
2. kryptische Fehlermeldungen
3. langsame Compilezeit
4. uvm.

Probleme mit LaTeX

1. 40 Jahre alte Syntax
2. kryptische Fehlermeldungen

} subjektiv

3. langsame Compilezeit
4. uvm.

} objektiv messbar

Typst. Eine Lösung?[1]

```
_Hallo Welt._ Dies ist ein *test*.

#table(columns: 3,
    [a], [b], [c],
    [d], [e], [f])
```

Hallo Welt. Dies ist ein **Test**.

a	b	c
d	e	f

[1] Diese Folien sind in Typst erstellt.

Typst. Eine Lösung?

1. modernes Sprachdesign
2. verständliche Fehlermeldungen

Typst. Eine Lösung?

1. modernes Sprachdesign
2. verständliche Fehlermeldungen
3. schneller Compiler (Live-Preview)

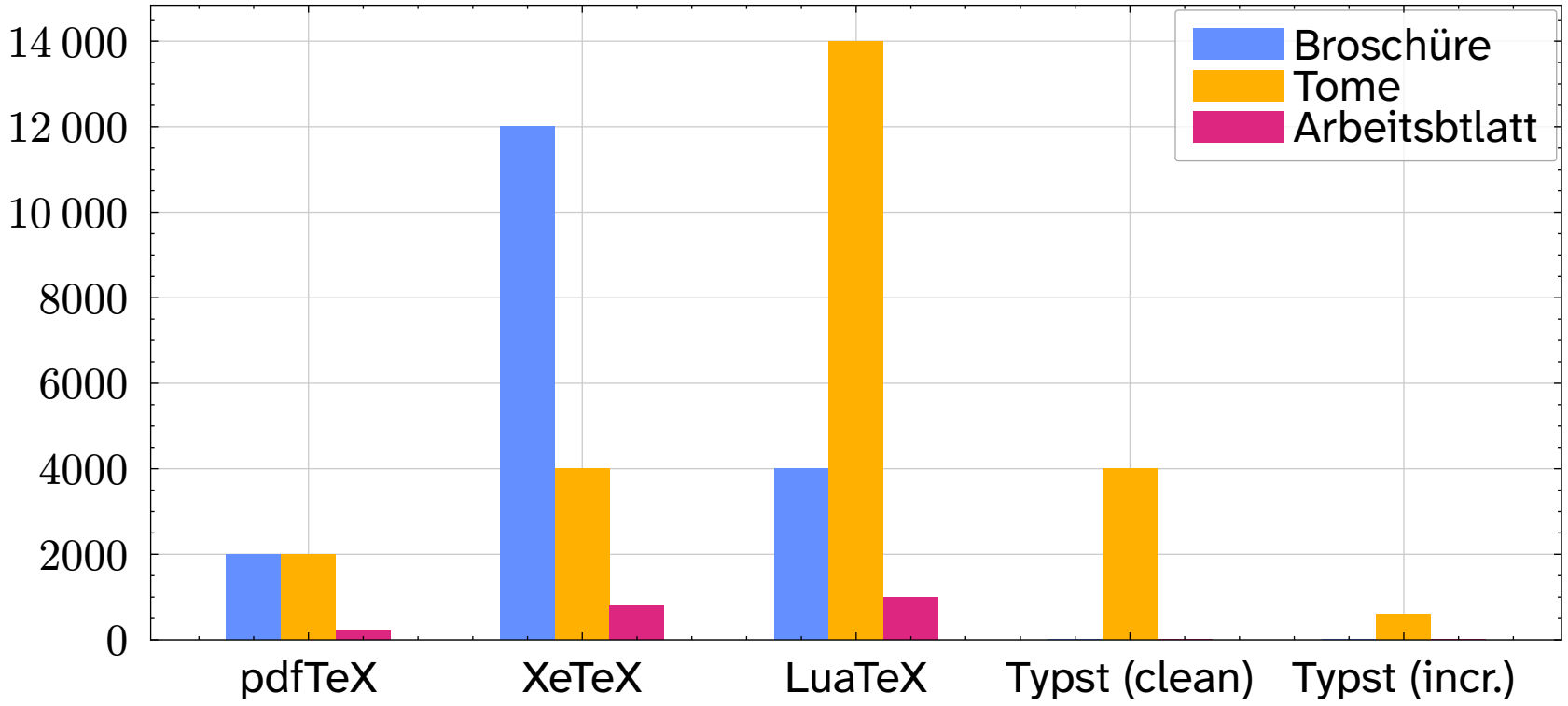


Abbildung 1: Benchmark des Typst-Compilers 2022, Angaben in ms [2, S. 70].

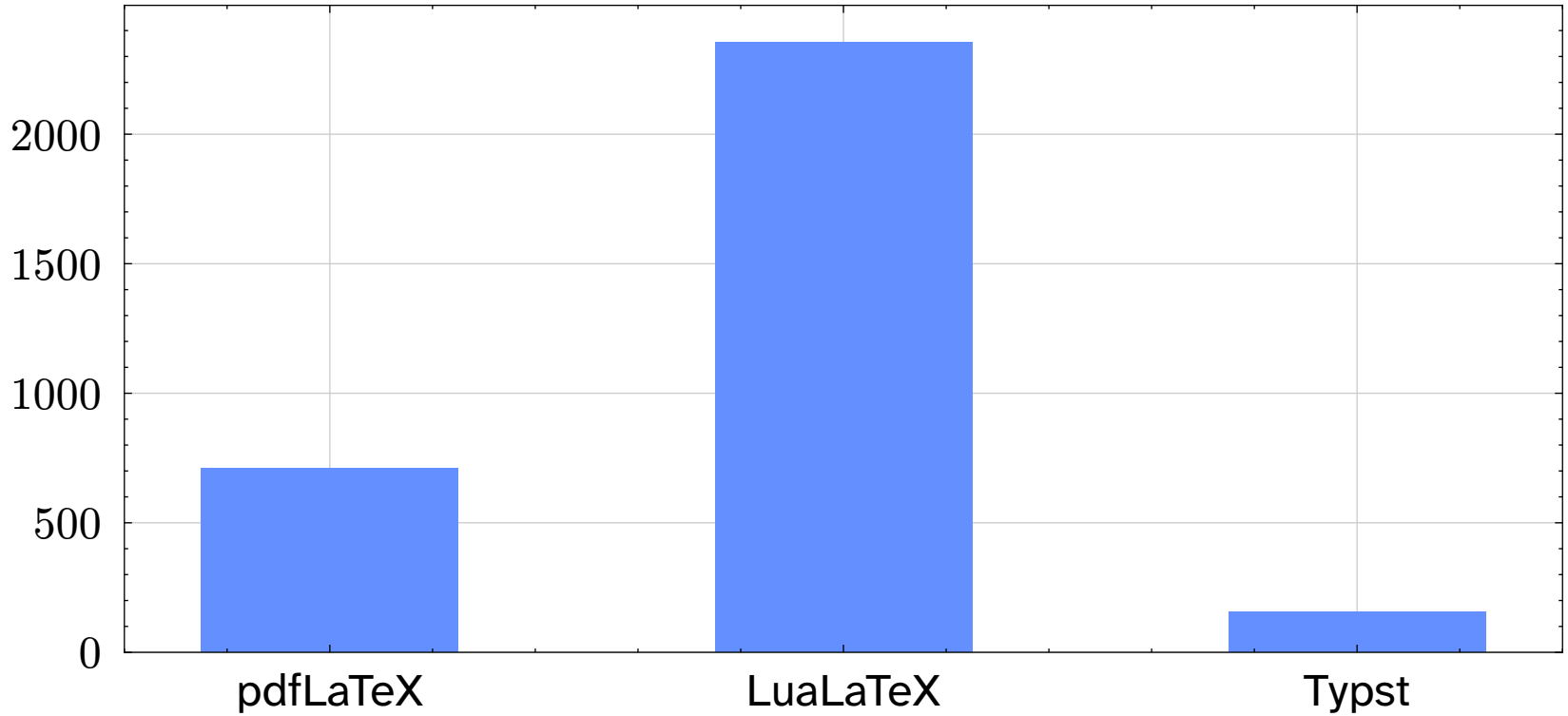


Abbildung 2: Benchmark des Typst-Compilers 2026, Angaben in ms [3].

Wie macht Typst das?

Funktionen in Typst

```
1  #let global = 3
2
3  #let increment() = {
4      global = global + 1
5  }
6
7  #increment()
```

Funktionen in Typst

```
1  #let global = 3
2
3  #let increment() = {
4      global = global + 1
5  }
6
7  #increment()
```

```
error: variables from outside
the function are read-only and
cannot be modified
   ┌─ test.typ:4:4
   │
4  │     global = global + 1
   │     ^^^^^^
```

Funktionen in Typst

Definition

Eine Funktion ist *wirkungsfrei* (ohne „side-effects“), gdw. sie den Zustand des Programmes nicht verändert.

Funktionen in Typst

Definition

Eine Funktion ist *wirkungsfrei* (ohne „side-effects“), gdw. sie den Zustand des Programmes nicht verändert.

Bsp.: alle mathematischen Funktionen sind wirkungsfrei

Inkrementelle Kompilierung

- jede Funktion in Typst ist $f : \mathrm{Value}^n \to \mathrm{Value}$

Inkrementelle Kompilierung

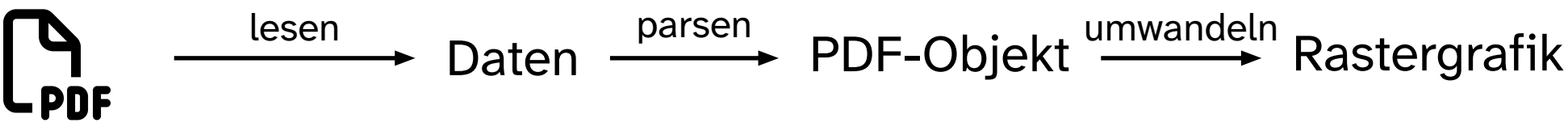
- jede Funktion in Typst ist $f : \mathrm{Value}^n \to \mathrm{Value}$
- Cache konstruieren:

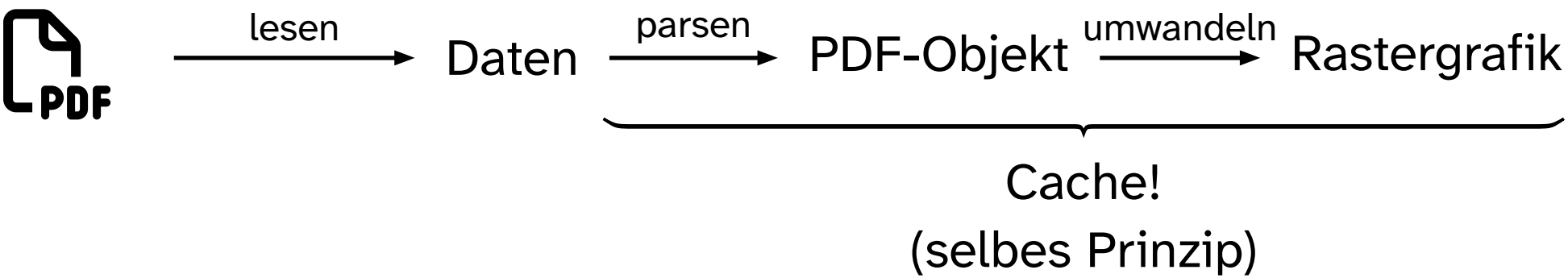
$\mathrm{hash}(\langle f, v_1, v_2, \ldots \rangle)$	v_{ret}
`1011172c6...`	[Hallo]
`bad8c0dc4...`	4
`12fccd699...`	table(columns:(auto,auto),chil...

Inkrementelle Kompilierung

In main.typ:

```
#image("bookcover.pdf")
#include "chapters/chap01.typ"
#include "chapters/chap02.typ"
...
```


PDF
lesen
Daten
parsen
PDF-Objekt
umwandeln
Rastergrafik


PDF
lesen
Daten
parsen
PDF-Objekt
umwandeln
Rastergrafik
Cache!
(selbes Prinzip)

```
1  impl PdfDocument {
2
3      pub fn new(data: Bytes) → PdfDocument {
4          ...
```

```rust
impl PdfDocument {
    #[comemo::memoize]
    pub fn new(data: Bytes) -> PdfDocument {
        ...
```

```rust

fn eval(sourceCode: &Source) → Value {
    ...
}
```

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source) -> Value {
    ...
}
```

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source) -> Value {
    ...
}
```

```typst
#image("bookcover.pdf")
#include "chapters/chap01.typ"
#include "chapters/chap02.typ"
...
```

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source) -> Value {
    ...
}
```

```typst
#image("bookcover.pdf")
#include "chapters/chap01.typ"
#include "chapters/chap02.typ"
...
```

⟹ im Cache nicht alle Abhängigkeiten berücksichtigt

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: &Fs) → Value { ... }
```

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: &Fs) → Value { ... }
```

↑ `fs.load(path)`

path	source
/chapters/chap01.typ	The palace still shook occasionally as ...
/chapters/chap02.typ	The Wheel of Time turns, and Ages ...
...	...

```
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: Tracked<&Fs>) → Value { ... }
```

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: Tracked<&Fs>) → Value { ... }
```

fs.load(path)

path	$\mathrm{hash}(v_{\mathrm{ret}})$
/chapters/chap01.typ	03f200a3b8bb…
/chapters/chap02.typ	2084ff93ce00…
…	…

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: Tracked<&Fs>) → Value { ... }
```

fs.load(path)

path	$\mathrm{hash}(v_{\mathrm{ret}})$
/chapters/chap01.typ	03f200a3b8bb…
/chapters/chap02.typ	2084ff93ce00…
…	…

⟹ nur relevantes Verhalten von `fs` wird getracked

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: Tracked<&Fs>) → Value { ... }
```

fs.load(path)

path	$\mathrm{hash}(v_{\mathrm{ret}})$
/chapters/chap01.typ	03f200a3b8bb…
/chapters/chap02.typ	2084ff93ce00…
…	…

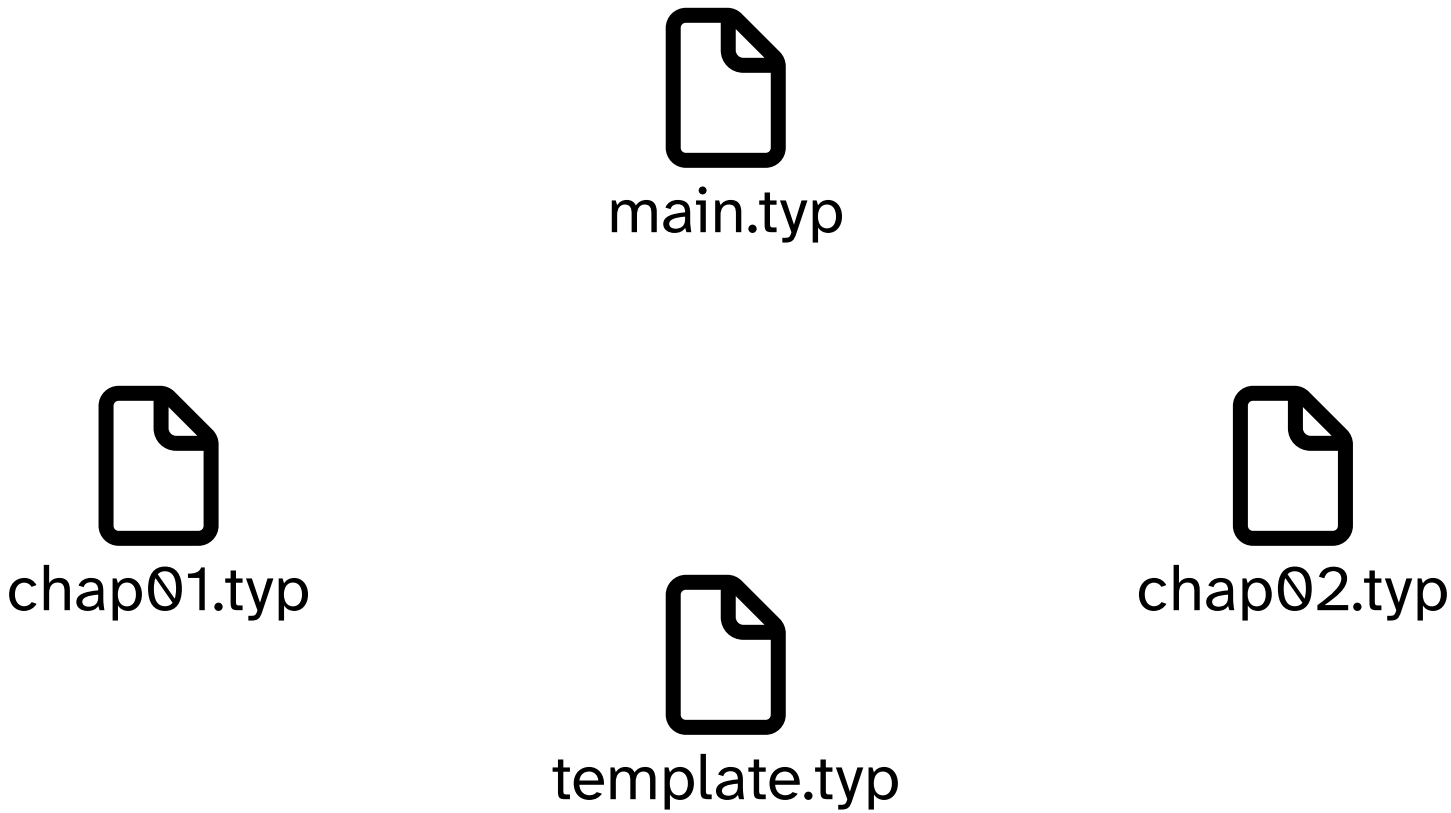
⟹ nur relevantes Verhalten von `fs` wird getracked

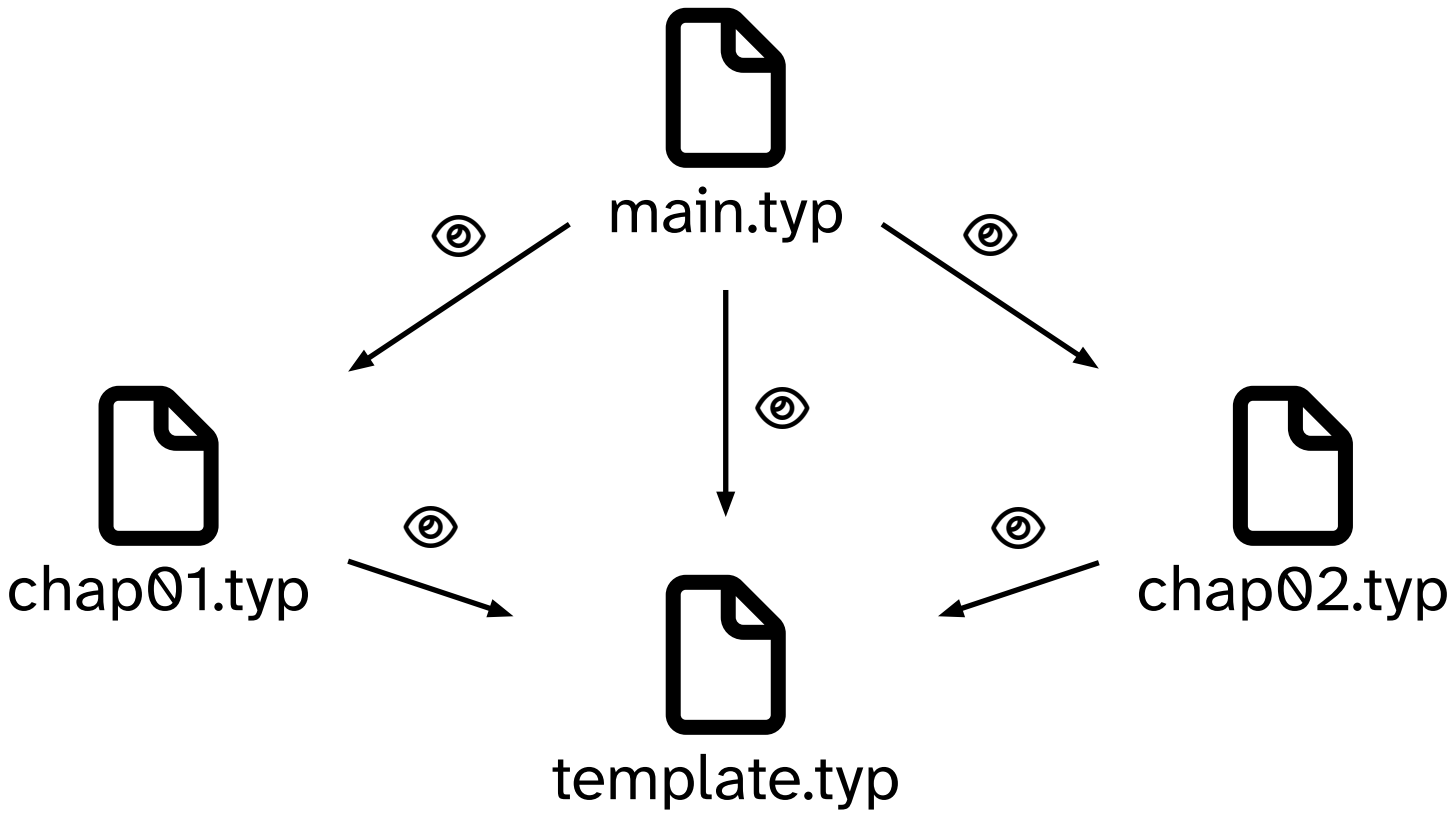
⟹ getrackte Argumente bilden „Constraints“ für Cache-Eintrag

```rust
#[comemo::memoize]
fn eval(sourceCode: &Source, fs: Tracked<&Fs>) → Value { ... }
```

- Caching dann wie folgt:

$\langle v_1, v_2, \ldots, v_n \rangle$	Constraints	v_{ret}
…	<table><tr><th>path</th><th>$\text{hash}(V_{\text{ret}})$</th></tr><tr><td>/chapters/chap01.typ</td><td>03f200a3b8bb…</td></tr><tr><td>/chapters/chap02.typ</td><td>2084ff93ce00…</td></tr></table>	…


main.typ
chap01.typ
template.typ
chap02.typ


main.typ
chap01.typ
template.typ
chap02.typ

Inkrementelle Kompilierung

- Wirkungsfreiheit beschreibt unabhängigen Code
- Typst nutzt Wirkungsfreiheit um Funktionen zu cachen
- Constraints ermöglichen effiziente Validierung des Caches

(Diese Folien wurde in 2,5 s und mit Caching in 150 ms kompiliert.)

Was ist mit (La)TEX?

„Don tried very hard not to make TeX a programming language. Unfortunately, he didn’t succeed.“[1]

— Michael Pass, Schüler von Donald Knuth, dem Erfinder von TeX

[1] Norman Ramsey: Stackoverflow. https://stackoverflow.com/questions/1931238/if-tex-is-a-programming-language-how-could-i-start-programming-in-tex, letzter Zugriff: 1.6.2025, 11:39 Uhr.

Pure Functions/Macros möglich?

```tex
...
\def\A{A}
"\A"
...
```

Pure Functions/Macros möglich?

```tex
...
\def\A{A}
"\A"
...
```

„A“

Pure Functions/Macros möglich?

```tex
...
\def\A{A}
\def\B{\def\A{C}}
"\A \B \A"
...
```

Pure Functions/Macros möglich?

```
...
\def\A{A}
\def\B{\def\A{C}}
"\A \B \A"
...
```

„AC“

Pure Functions/Macros möglich?

```tex
...
\def\A{A}
\def\B{\def\A{C}}
"\A \B \A"
...
```

„AC“

$\Longrightarrow$ `\B` nicht wirkungsfrei

Inkrementelle Kompilierung

- Neukompilierung nur dessen, was sich ändert
- Rest mittels constrained Memoization speichern
- durch Wirkungsfreiheit des Typst-Codes und im Compilerdesign besonders effektiv

Typst:

Diese Folien:

Bibliografie

[1] Pexels, „Adult, Annoyed, Blur image. Free for use.“. Zugegriffen: 3. Juni 2026. [Online]. Verfügbar unter: https://pixabay.com/photos/adult-annoyed-blur-burnout-1850268/

[2] M. Haug, „Fast Typesetting with Incremental Compilation“, Masterarbeit, Berlin, 2022. Zugegriffen: 12. Mai 2026. [Online]. Verfügbar unter: https://web.archive.org/web/20221022111033/https://www.user.tu-berlin.de/mhaug/fast-typesetting-incremental-compilation.pdf

[3] Speedata, „Speedata/typesetting-benchmark: Compare typesetting speed of SP, typst, weasyprint, FOP, latex and LuaLaTeX“. Zugegriffen: 31. Mai 2026. [Online]. Verfügbar unter: https://github.com/speedata/typesetting-benchmark

[4] L. Mädje, „Typst. A Programmable Markup Language for Typesetting“, Masterarbeit, Berlin, 2022. Zugegriffen: 12. Mai 2026. [Online]. Verfügbar unter: https://laurmaedje.github.io/programmable-markup-language-for-typesetting.pdf

[5] L. Mädje, „Frozen State“. Zugegriffen: 12. Mai 2026. [Online]. Verfügbar unter: https://laurmaedje.github.io/posts/frozen-state/

[6] L. Mädje, „Types and Context“. Zugegriffen: 12. Mai 2026. [Online]. Verfügbar unter: https://laurmaedje.github.io/posts/types-and-context/

[7] L. Mädje, „What If LaTeX Had Instant Preview?“. Zugegriffen: 12. Mai 2026. [Online]. Verfügbar unter: https://laurmaedje.github.io/posts/comemo/

[8] L. Mädje, „TeX and Typst: Layout Models“. Zugegriffen: 12. Mai 2026. [Online]. Verfügbar unter: https://laurmaedje.github.io/posts/layout-models/

[9] D. Salomon, *The Advanced TEXbook*. New York, NY: Springer New York, 1995.

[10] L. Mädje, „Designing for incrementality“. Zugegriffen: 31. Mai 2026. [Online]. Verfügbar unter: https://www.youtube.com/live/6LTwtyuj69M

Weitere Typst-Beispiele

Theorem 1.1. $f(n) = 7n^4 - 4n^3 + \lceil n^e + \pi \rceil$

Es gilt $f \in \Theta(n^4)$.

Beweis:

1. $\Theta(n^4) := O(n^4) \cap \Omega(n^4)$, d. h. $f \in \Theta(n^4) \leftrightarrow (f \in O(n^4) \wedge f \in \Omega(n^4))$.
2. $f \in O(n^4)$?
   a) Ja, denn:

$$|f(n)| = \left|a_0 + a_1 n + a_2 n^2 + \ldots + a_m n^m\right| \leq \left(\sum_{i=0}^{m} \frac{|a_i|}{n^{m-i}}\right) n^m \leq \left(\sum_{i=0}^{m} |a_i|\right) n^m$$

$$c = \left(\sum_{i=0}^{m} |a_i|\right)$$

   b) Mit gewähltem c gilt, dass f nicht schneller wächst als n^4.
3. $f \in \Omega(n^4)$? D.h. $f \in \Omega(n^4) \leftrightarrow g(n) = n^4 \in O(f)$. Also $g \in O(f)$? Siehe dazu Schritt 2: c kann in Bezug auf g ebenfalls gewählt so werden, dass $g \in O(f)$.
4. Daher $f \in O(n^4) \wedge f \in \Omega(n^4)$, also $f \in \Theta(n^4)$.

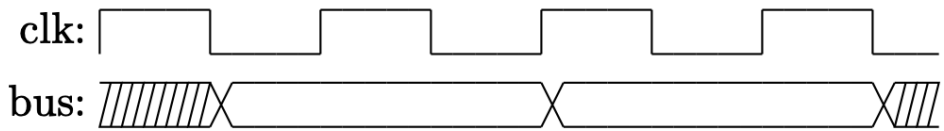
□

$$A^{\mathsf{T}}, \boldsymbol{\nabla} \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}, \quad \Lambda^{\mu}{}_{\nu} = \begin{pmatrix} 1 & \\ & \mathbb{R} \end{pmatrix}, \quad f(x,y)\,\mathrm{d}x\,\mathrm{d}y, \quad \mathrm{d}^3\boldsymbol{x}\,\mathrm{d}y, \quad \Delta^2 x \wedge \Delta^2 y, \quad \frac{\mathrm{D}\varphi}{\mathrm{D}t} = \frac{\partial \varphi}{\partial t} + \boldsymbol{u}\boldsymbol{\nabla}\varphi$$

$$H(f) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix}, \quad \boldsymbol{v}^{\boldsymbol{a}} = \sum_{i=1}^{n} \alpha_i \hat{\boldsymbol{u}}^{\boldsymbol{i}}, \quad \left\{ (x,y) \,\middle|\, \frac{\partial^3 f}{\partial x^2 \partial y} + \frac{\partial^3 f}{\partial x \partial y^2} < \varepsilon \right\}$$

$$-\frac{1}{c^2}\frac{\partial^2}{\partial t^2}\psi + \nabla^2\psi = \frac{m^2c^2}{\hbar^2}\psi, \quad \left|n^{(1)}\right\rangle = \sum_{k \notin D} \frac{\left\langle k^{(0)} \middle| V \middle| n^{(0)} \right\rangle}{E_n^{(0)} - E_k^{(0)}} \left|k^{(0)}\right\rangle, \quad \int_V \mathrm{d}V \left(\frac{\partial \mathcal{L}}{\partial \varphi} - \partial_\mu \left(\frac{\partial \mathcal{L}}{\partial(\partial_\mu \varphi)} \right) \right) = 0$$

$$\mathrm{d}^2 s = -\left(1 - \frac{2GM}{r}\right) \mathrm{d}^2 t + \left(1 - \frac{2GM}{r}\right)^{-1} \mathrm{d}^2 r + r^2\,\mathrm{d}^2\Omega$$



Universität Rostock, Institut für Informatik
Einführung in die Theoretische Informatik
Prof. Dr. Jens M. Schmidt

Übung 9
Sommersemester 2025, 13.06.2025
Tristan Pieper, Jan Marten Noack, Björn Christer Weiss

Übung 9

Aufgabe 1 — Diagonalsprache der RAM

(i) Seien für diese Teilaufgabe $\ulcorner\alpha\urcorner$ objektsprachliche Ausdrücke mit $\alpha \in \Sigma^+$. Sei $w \circ v = wv$, dementsprechend $\ulcorner a\urcorner \circ \ulcorner b\urcorner = \ulcorner ab\urcorner$.

Sei für alle RAMs R mit der Befehlsmenge B eine Gödelnummer definiert als $G(R) = \langle R\rangle$, $\langle R\rangle \in \{\ulcorner 0\urcorner, \ulcorner 1\urcorner\}^*$.

Dafür sei $\mathrm{bin} : \mathbb{N} \to \{\ulcorner 0\urcorner, \ulcorner 1\urcorner\}^+$ die Binärdarstellung einer Zahl $x \in \mathbb{N}$. Zusätzlich sei $\mathrm{double} : \Sigma^+ \to \Sigma^+$ diejenige Funktion, die jedes Symbol eines Wortes verdoppelt:

$$\mathrm{double}(x) = \begin{cases} x \circ x & \text{wenn } x \in \Sigma \\ \mathrm{double}(v) \circ \mathrm{double}(w) & \text{wenn } x = vw \text{ und } v, w \in \Sigma^+ \end{cases}$$

Sei $G_B : B \to \{\ulcorner 0\urcorner, \ulcorner 1\urcorner\}^*$ wie folgt definiert:

$$G_B(x) = \begin{cases} \ulcorner 0\urcorner & \text{wenn } x = \ulcorner \mathrm{LOAD}\urcorner \\ \ulcorner 1\urcorner & \text{wenn } x = \ulcorner \mathrm{STORE}\urcorner \\ \ulcorner 10\urcorner & \text{wenn } x = \ulcorner \mathrm{ADD}\urcorner \\ \ulcorner 11\urcorner & \text{wenn } x = \ulcorner \mathrm{SUB}\urcorner \\ \dots & \\ \mathrm{bin}(n) & \text{wenn } x = B_n,\ n < |B|,\ n \in \mathbb{N} \end{cases}$$

ALLER ANFANG IST… KLAGERISCH

KÜK

[Klager-Überlebens-Koffer für das Lehramtsstudium der Philosophie an der Universität Rostock]

Der KÜK umfasst grundlegende, methodische und inhaltliche Überlebenstipps für das Bestehen der Klausur „Einführung in die Philosophiedidaktik", die Seminarvor- und Nachbereitung und die Prügungsvorbereitung am Institut für Philosophie der Universität Rostock. Die 17-teilige Grundausrüstung richtet sich an alle Lehramtsstudent*innen der Philosophie und soll den Start in das Studium erleichtern. Der KÜK versteht sich prozessorientiert und erlaubt individuelle Ergänzungen. Die zusammengefassten Hinweise sind jedoch für das Lehramtsstudium der Philosophie als wichtig anzusehen. Diese Sammlung ersetzt nicht das Nachfragen und Nachdenken; wenn Inhalte, Formalia oder Methoden unklar erscheinen, sollten Dozierenden und Mitstudierenden um Rat gefragt werden.

Verantwortung und Betreuung: Gruppe „Philo lernen" © 2023-2024

Autor: Tristan Pieper, Version: 2024-12

Planung

SEQUENZPLANUNG

Eine Sequenzplanung (auch „Einheitenplanung") spezifiziert, wie viele Unterrichtsstunden einer gegebenen Länge sich mit welcher Thematik auseinandergesetzt wird. Es werden Richtziele genannt, die über die geplante Sequenz behandelt werden sollen. Die Thematik der Sequenz sollte – genauso wie jede Unterrichtsstunde – angemessen motiviert werden. Eine Sequenz „Gottesbegriffe" könnte etwa wie folgt aussehen:

Universität Rostock, Institut für Philosophie
Schulpraktische Übung zum Philosophieren mit Kindern
[Dozent*in]
[Schule], [Klasse]

Sequenzplanung
[Namen der Lehrpersonen]
[Datum]

Sequenzplanung

Thema: Gottesbegriffe
Richtziel: Die SuS setzen sich kritisch mit der Frage nach Gott und verschiedenen Gottesbegriffen auseinander.
Anzahl der Unterrichtsstunden: 6
Dauer einer Unterrichtsstunde: 45min

	Datum	Lehrperson(en)	Thema	Bemerkung	
1	Mi., 17.4.2024	C. Muster	Einführung: Gottesbegriffe	Stunde hat nur 20 min., Mindmap zu „Gott"	Gottesbegriffe
2	Mi., 24.4.2024	A. Joe	Warum ist man auf der Suche nach Gott?		
3	Mi., 8.5.2024	B. Schmidt	Gotteswahrnehmung und Gottesbeweise – (Wie) Nimmt man Gott wahr?		
4	Mi., 15.5.2024	C. Muster	Gottesbegriffe – Was versteht man unter „Gott"?		
5	Mi., 22.5.2024	A. Joe	Gottesbilder – Kann man Gott darstellen?		
6	Mi., 29.5.2024	B. Schmidt	Gottesbegriffe	Test	

Standardmäßig wird in der Spalte „Thema" das Stundenthema genannt, in „Bemerkung" die dazugehörige philosophische Leitfrage, die bearbeitet werden soll, wie auch konkrete Hinweise für die Stunde. In diesem Beispiel wurde beides vereint: Das Thema ist eine philosophische Leitfrage, in den Bemerkungen wurden hier nur organisatorische Umstände aufgezählt.

KURZENTWURF

Für ein Beispiel der Inhalte eines Kurzentwurfes siehe S. 29. Wichtig ist, dass Sie für jede Aufgabe der fertig geplanten Stunde einen Erwartungshorizont entwerfen und nötige Anlagen anhängen.[171]

Definition 2.13 *(lebendige Transition).* Eine Transition t ist lebendig bei m, gdw. $\forall m' \in R_N(m) : \exists m'' \in R_N(m') :$ aktiviert(t). Also dass für alle $m' \in R_N(m)$ eine erreichbare Markierung m'' existiert, in der t aktiviert ist.

Definition 2.14 *(k-beschränkte Stelle).* Eine Stelle heißt k-beschränkt bei m, gdw. in allen $m' \in R_N(m)$ gilt, dass $m'(s) \leq k$.

Definition 2.15 *(sichere Stelle).* Eine Stelle s heißt sicher, gdw. sie 1-beschränkt ist.

Definition 2.16 *(Homestate).* Eine Markierung m wird Homestate eines Petrinetzes $\langle S, T, F, W, m_0 \rangle$ genannt, gdw. $\forall m' \in R_N(m_0) : m' \to m$. Also wenn m von allen erreichbaren Markierungen erreichbar ist.

2.3. NETZEIGENSCHAFTEN

Für ein Netz $N = \langle S, T, F, W, m_0 \rangle$ gelten die folgenden Eigenschaften.

Definition 2.17 *(Verklemmungsfreiheit).* Ein Netz N ist verklemmungsfrei, gdw. in jedem $m \in R_N(m_0)$ mindestens eine Transition aktiviert ist.

Definition 2.18 *(lebendiges Netz).* N heißt lebendig, gdw. alle Transitionen in N lebendig sind.

Theorem 2.1 Wenn N lebendig ist mit $|T| \geq 1$, dann ist N verklemmungsfrei.

Beweis. Angenommen N ist lebendig und es gibt mindestens eine Transition t. Wäre N nicht verklemmungsfrei, dann gäbe es eine Markierung m, in der keine Transition aktiviert ist, auch nicht t. Von dort aus ist keine Markierung m' erreichbar, in der t aktiviert ist. Damit wäre t nicht lebendig. Das ist ein Widerspruch. ↯ Also ist das Theorem wahr. ■

Definition 2.19 *(Reversibilität).* N ist reversibel, gdw. $\forall m \in R_N(m_0) : m \to^* m_0$.

Theorem 2.2 Wenn N reversibel ist, dann ist jede Transition bei m_0 entweder tot oder lebendig.

Beweis. Wenn t in m_0 nicht tot ist, dann gibt es ein m' in dem t aktiviert ist, mit $m_0 \to^* m'$. Da N aber reversibel ist, gilt $\forall m \in R_N(m) : m \to^* m_0$. Also auch $\forall m \in R_N(m) : m \to m'$. Das ist gerade die Definition von Lebendigkeit. Wenn also eine Transition t nicht tot ist und N reversibel, dann ist t lebendig. Da eine Transition nicht sowohl lebendig als auch tot sein kann, ist jede Transition in einem reversiblen Netz entweder tot oder lebendig. ■

Theorem 2.3 N ist reversibel, gdw. m_0 ein Homestate ist.

Beweis. N ist reversibel. m_0 ist aus jeder Markierung $R_N(m_0)$ erreichbar. Das ist die Definition eines Homestates. Die Rückrichtung funktioniert analog. ■

Theorem 2.4 N ist reversibel, gdw. alle erreichbaren Zustände Homestates sind.

Beweis. Für alle Zustände gilt $m \to^* m_0 \to^* m'$ für $m, m' \in R_N(m_0)$. Damit ist jeder Zustand aus jedem erreichbar und damit ein jeder ein Homestate. ■

3. ZUSTANDSRAUMANALYSE

3.1. ERREICHBARKEITSGRAPH

Definition 3.1 *(stark Zusammenhängende Komponente (SZK)).* Eine SZK ist ein gerichteter Teilgraph $G_{SZK} \subseteq G$, sodass es für jede Knoten $v, v'' \in G_{SZK}$ einen Weg zwischen diesen gibt. Es gilt die Äquivalenzrelation $v \sim_{SZK} v'$.

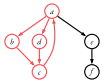


Abbildung 3.1: Die roten Knoten bilden eine stark zusammenhängende Komponente.

Definition 3.2 *(terminale SZK (TSZK)).* Eine terminale SZK ist eine SZK, wobei aus keinem $v \in V_{SZK}$ ein Weg zu einem anderen $v' \notin V_{SZK}$ führt.

Abbildung 3.2: Die beiden roten Teilgraphen sind jeweils voneinander unabhängige TSZK, e ist eine einzelne SZK, die jedoch nicht terminal ist, da sie in die anderen beiden überführen kann. Sobald man aber in einer der roten TSZK ist, kann man diese nicht mehr verlassen.

Im Folgenden ist der Graph, den wir analysieren wollen, der *Erreichbarkeitsgraph*, d.i. der Graph an Markierungen.

Definition 3.3 *(Erreichbarkeitsgraph).* Der Erreichbarkeitsgraph eines Petrinetzes N ist ein beschrifteter Graph $\langle V, E \rangle$, wobei $V = R_N(m_0)$ und $(m, t, m') \in E \longleftrightarrow m \to' m'$.

Abbildung 3: Ein einfaches Petrinetz mit Erreichbarkeitsgraph.

Man erkennt im Erreichbarkeitsgraphen, dass dieser eine SZK bildet. Da die Marke immer im Kreis wandert und sich das Petrinetz nicht stark verändert, sieht auch der Erreichbarkeitsgraph entsprechend aus.

Tiefensuche zur SZK-Bestimmung:

- kleinste dfs-Zahl ist Wurzel der SZK (Was heißt das?)

Zur Entdeckung von SZK und TSZK kann man die Tiefensuche verwenden. Aus dem Tiefensuchbaum können wir die einzelnen Klassifizieren und mit bestimmten Zusammenhängen SZK und TSZK identifizieren. Zuerst zu den SZK. Je nach Anwendung des Tiefensuchalgorithmus, kann ein unterschiedliches Ergebnis rauskommen. Daher kann es für jedes Beispiel hier unterschiedliche Tiefensuchbäume geben. Das macht uns die Eigenschaften, die wir suchen wollen, jedoch nicht kaputt.

Definition 3.4 *(Kantenarten im Tiefensuchbaum).* In einem gerichteten Graphen G ist die Kante $e = \langle v, v' \rangle \in E_G$ mit Tiefensuchbaum T, ...

- eine Baumkante, falls $e \in E_T$.
- eine Vorwärtskante, falls es einen Weg zwischen v und v' in T gibt, aber e keine Baumkante ist.
- eine Rückwärtskante, falls es einen Weg zwischen v' und v in T gibt.
- eine Querkante, gdw. es sonst zu keiner Kantenart gehört.

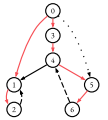


Abbildung 3.4: Die roten Kanten sind die des Suchbaumes T. Gepunktete Kanten sind Vorwärtskanten, gestrichelte sind Rückwärtskanten und die durchgezogene schwarze Kante ist eine Querkante.

Theorem 3.1 *(Kanteneigenschaften im Tiefensuchbaum).* Ist $\langle v, v' \rangle \in E_T$, $\forall x \in V_T$: x.dfs die Tiefensuchnummer und handelt es sich dabei um eine ...

- Querkante, dann ist v.dfs $> v'$.dfs.