BAdW

.dhmuc
Digital Humanities
München

Überlegungen zur Wiedergabe formallogischer Formeln in digitalen Editionen.

Eckhart Arnold, Digital Humanities, BAdW

BAYERISCHE
AKADEMIE
DER
WISSENSCHAFTEN

BAdW

Motto:

.dhmuc
Digital Humanities
München

“Syntax matters”

Aufgabe: Addiere zwei Zahlen!

römisch:

MMDXLI
\+ MCCCXLVII

arabisch:

2541
\+ 1347

BAdW

Hintergrund-Infos

.dhmuc Digital Humanities München

Anlass: **Planung einer digitalen Rudolf-Carnap-Edition**

(Christian Damböck, A.W. Carus, Stephan Hartmann, Hannes Leitgeb, Eckhart Arnold)

Rudolf Carnap (1891-1970)

- Philosoph und führender Vertreter des „**Wiener Kreis**es“ (Neupositivismus, Logischer Empirismus)
- vertritt eine Philosophie, deren Methode sich an den exakten Wissenschaften orientiert
- überwiegend **erkenntnistheoretische Ausrichtung** (Hauptwerk: „Logische Syntax der Sprache“), aber auch **religions- und ideologiekritisch** („Scheinprobleme in der Philophie“)
- Stilmerkmal: Massiver **Einsatz formaler Logik**.

Editorisch relevante Merkmale der logischen Notation Carnaps:

- Die formale Logik, deren sich Carnap bediente, ist als solche nach wie vor gebräuchlich
- Verwendete Notation (Russell-Peano) jedoch heute nicht mehr üblich. Beispiele:

`q.⊃.pvq` (Russell-Peano)

`q ⊃ pvq` (heutige Notation)

`p.q.⊃.r:⊃:p.⊃.q⊃r` (Russell-Peano)

`[(p&q)⊃r] ⊃ [p⊃(q⊃r)]` (heutige Notation)

Wesentlicher Unterschied: *Abgrenzung durch Punkte* statt der heute üblichen Gruppierung durch Klammern und Präzedenzregeln für Verknüpfungen. (plato.stanford.edu/entries/pm-notation/)

Editorische Fragen

Digital Humanities München

Wie sollte man mit logischen Formeln in einer digitalen Edition verfahren?

1. Sollte man Formeln in der historischen Notation oder einer aktuellen wiedergeben?
   Sollte man bei der digitalen Ausgabe die Wahl den Leserinnen und Lesern überlassen?

2. In welcher Form sollte man Formeln für die Erfassung durch Maschinen bereitstellen
   (im Gegensatz zur Lektüre durch Menschen)?

3. Welche Datenformate sollte man verwenden?

4. Wie kann man die Formeln am bequemsten eingeben?

1. Frage

Historische oder aktuelle Notation?

(Naheliegende) Antwort:

- Historische Notation für die historisch-kritische Edition (wenn Sie gedruckt erscheinen soll).
- Digitale Edition: Beide Notationen (umschaltbar),

  ggf. Bereitstellung eines Konverters (Langlebigkeit?)
- Auch denkbar: Gedruckte Studienausgabe ausgewählter Schriften in moderner Notation.

Mit automatischer Konvertierung nur geringer Mehraufwand!

(Demonstrator auf: github.com/jecki/logicConverter)

2. Frage

Maschinenlesbare Bereitstellung, aber wie?

- Bereitstellung des „*digitalen mathematischen Objekts*“ neben der Präsentationsform!

  Beispiel: f(x + 1) kann bedeuten: f · (x + 1) („f mal (x + 1)“)
  oder: die Funktion „f von (x + 1)“

- Format der Bereitstellung?

  - *passives Format* (LaTeX, ASCIIMath, MathML)?
  - *aktives Format*, eines gängigen Proof-Assistants (wie z.B. Coq: coq.inria.fr/ oder Lean: leanprover-community.github.io/) oder Validators (metamath.org)? Mehraufwand?

    Beispiele: www.principiarewrite.com/ und us.metamath.org/mm.html

Exkurs: Proof-Assistants

Digital Humanities München

Proof-Assistants: **Coq:** coq.inria.fr/ **Lean:** lean-lang.org/ und leanprover-community.github.io/,

- Nicht nur Überprüfung von Beweisen, sondern auch Unterstützung beim Entwickeln von Beweisen
- Reiche Programmiersprachen, mit einigem Komfort. (Aber: Mehr Fehlerquellen durch Größe?)
- Lean zugänglicher und einfacher. Mehr Anfängerliteratur – taucht auch schon in Mathe-Vorlesungsskripten auf: loeh.app.uni-regensburg.de/teaching/prooflab_ws2122/lecture_notes.pdf

Proof-Validators:

- **metamath.org** (Ansatz beschrieben im „MetaMath-Book“ us.metamath.org/downloads/metamath.pdf)
- Versteht sich in erster Linie als vollständige (im Sinne unverkürzter Beweisführung) Mathematik-Datenbank mit bewusst minimalen (ca. 500 Code-Zeilen) Theorem-Validator.
- Philosophie: Ein Beweis ist die Transformation von (inhaltslosen) Zeichenketten nach Regeln.
- Kritik: Zeichkettenbasierter Ansatz (im Gegensatz zu Syntax-Bäumen) schließt syntaktische Ambiguitäten nicht aus. (Vgl.: https://github.com/digama0/mm0)

3. Frage

Welches Datenformat für die Datenbereitstellung?

LaTeX

Vorteile:

- Weit-verbreitet, de-facto Standard
- Ergonomisch: Gut durchdachte Syntax. Relativ leicht zu lernen und zu lesen.
- Menschenlesbare Darstellung im Druck und im Netz mit verfügbaren Werkzeugen ohne weitere Konvertierung sofort möglich! (Druck: pdfLaTeX, LuaLaTeX Online: MathJaX, KaTeX)
- Insgesamt gute Werkzeuguntersützung sowohl für ein- als auch Ausgabe.
- Klartextformat mit großer Langzeitstabilität: + 30 Jahre

Nachteile:

- Nur Präsentationsform. (Aber: Mit zusätzlichen Annotationen auch Inhaltsform möglich!)
- Vergleichsweise kompliziert einzulesen. (u.a. wg. Eigensyntax von Ergänzungen / Pakteten)

3. Frage

Welches Datenformat?

MathML (Core?)

Vorteile:

- Erlaubt sowohl die Präsentationsform semantisch zu erfassen (aber nicht wiederzugeben!?) als auch die Inhaltform (digitales mathematisches Objekt)
- leicht von einer Maschine einzulesen (aber deswegen noch nicht leicht zu verarbeiten!?)

Nachteile:

- Umständlich und praxisfern, taugt daher auch nicht zur Eingabe von Formeln
- kaum Werkzeugunterstützung (Teilmengen werden durch einzelne Browser unterstützt)
- „It is a huge and standalone specification."
- „It does not contain any detailed rendering rules."
- „It is not driven by browser-implementation."
- „It lacks automated testing." (die letzten vier von: w3c.github.io/mathml-core/)

3. Frage

$ax^2 + bx + c$

MathML:

```
<math>
    <apply>
        <plus/>
        <apply>
            <times/>
            <ci>a</ci>
            <apply>
                <power/>
                <ci>x</ci>
                <cn>2</cn>
            </apply>
        </apply>
        <apply>
            <times/>
            <ci>b</ci>
            <ci>x</ci>
        </apply>
        <ci>c</ci>
    </apply>
</math>
```

S-Ausdruck:

```
(plus
  (times a
(power x 2))
  (times b x)
 c)
```

LaTeX:

```
ax^2 + bx +c
```

Wurden die Fehler von TEI-XML und XHTML wiederholt? Werden sie mit MathML-Code korrigiert?

MathML = Die gute Idee, mathmatische Formeln durch ihren Strukturbaum maschinenlesbar darzustellen, runiniert durch "Overengineering"?

Beispiel aus en.wikipedia.org: t1p.de/7vay3

3. Frage

ASCII- bzw. Unicode-basierte Formate

- Logische Formeln lassen sich in der Regel in Unicode darstellen
- ASCIIMath: http://asciimath.org/ Keine deutlichen Vorteile gegenüber LaTeX
- Tastatur-Notation, die die Formeln mit den auf der Tastatur verfügbaren Zeichen abbildet.

Ein gangbarer Weg, aber vllt. eher für die Eingabe und ggf. die Suche

Ergebnis (Bereitstellungsformate):

- Ja: LaTeX, S-Ausdrücke, MetaMath
- Nein: MathML (ältere Versionen)
- Vielleicht: MathML Core, Coq, Unicode/Tastaturnotation (falls bei der Eingabe verwendet)

4. Frage

.dhmuc Digital Humanities München

Eingabetechnologien

LaTeX:

- Bearbeiter können sofort loslegen, kein Entwicklungsaufwand für Druck und Online-Vorschau
- ggf. anspruchsvolle Entwicklung eines Parsers für die Umwandlung in maschinenlesbare Formen
- Inhaltsform könnte zusätzliche Annotation erfordern (bei logischen Formeln unwahrscheinlich)

Tastatur-Notation:

- geringste Tipp-Arbeit, direkte Übernahme aus dem Text, *einfachste Konvertierung*, evtl. ungewohnt
- Werkzeuge für die Umwandlung nach LaTeX und (Live-)Vorschau müssen erst entwickelt werden. Im Prinzip nicht schwer, aber…

Computer-Algebrasystem:

- Kann zugleich als aktives Bereitstellungsformat dienen
- Sinnvoll, wenn LaTeX-Export möglich
- Hinsichtlich Vorschau etc. ebenfalls etwas umständlicher als LaTeX

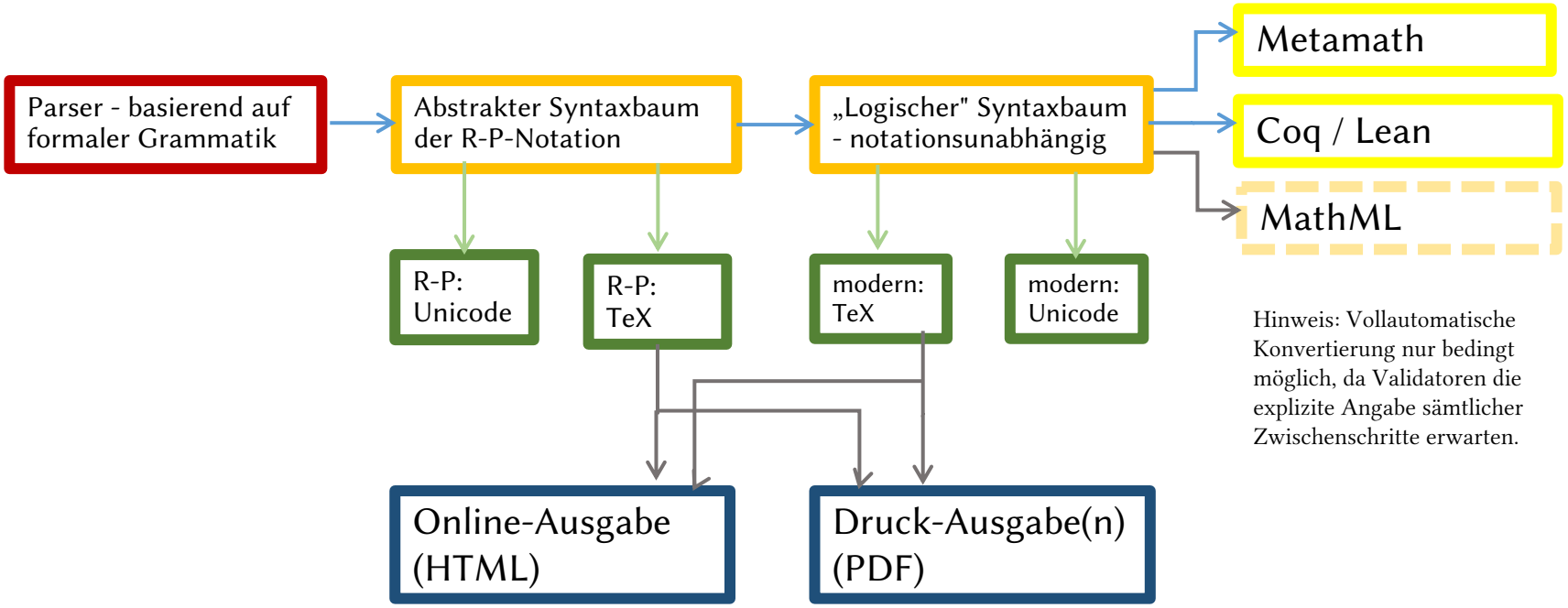
eher nicht: Formeleditor

- für logische Formeln nicht notwendig
- langsame Eingabe

Exkurs

Digital Humanities München

Automatische Konvertierung von logischen Formeln in Russell-Peano-Notation (R-P):



Hinweis: Vollautomatische Konvertierung nur bedingt möglich, da Validatoren die explizite Angabe sämtlicher Zwischenschritte erwarten.