

Überlegungen zur Wiedergabe formallogischer Formeln in digitalen Editionen.

Eckhart Arnold, Digital Humanities, BAdW

“Syntax matters”

Aufgabe: Addiere zwei Zahlen!

römisch:

$$\begin{array}{r} \text{MMDXL I} \\ + \quad \text{MCCCXLVII} \\ \hline \end{array}$$

arabisch:

$$\begin{array}{r} 2541 \\ + \quad 1347 \\ \hline \end{array}$$

Anlass: Planung einer digitalen Rudolf-Carnap-Edition

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

Über **Rudolf Carnap** (1891-1970):

- Philosoph und führender Vertreter des „**Wiener Kreises**“ (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 Philosophie“)
- 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:

$$q \cdot \supset . p \vee q \quad (\text{Russell-Peano})$$
$$q \supset p \vee q \quad (\text{heutige Notation})$$
$$p \cdot q \cdot \supset . r : \supset : p \cdot \supset . q \supset r \quad (\text{Russell-Peano})$$
$$[(p \& q) \supset r] \supset [p \supset (q \supset r)] \quad (\text{heutige Notation})$$

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

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?
-

Historische oder aktuelle Notation?

(Naheliegende) Antwort:

- Historische Notation für die historisch-kritische Edition (wenn Sie gedruckt erscheinen soll).
- Moderne Notation für die Studienausgabe ausgewählter Schriften.
- Digitale Edition: Beide Notationen (umschaltbar),
ggf. Bereitstellung eines Konverters (Langlebigkeit?)

Mit automatischer Konvertierung nur geringer Mehraufwand!

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

Maschinenlesbare Bereitstellung, aber wie?

1. Bereitstellung des „*digitalen mathematischen Objekts*“ ([Inhaltsform](#)) neben der [Präsentationsform!](#)

Beispiel: $f(x)$ kann bedeuten: $f \cdot x$ („f mal x“)
oder: die Funktion „ f von x “

2. Format der Bereitstellung?

- *passives Format* ([LaTeX](#), [ASCII Math](#), [MathML](#))?
- *aktives Format* ([Arbeitsform](#)), eines gängigen Beweis-Assistenz- (wie z.B. Coq: [coq.inria.fr/](#)) oder Computer-Algebra-Systems (Mathematica, Maple, SageMath) oder Prolog?

Beispiele: [www.principiarewrite.com/](#) und [us.metamath.org/mm.html](#)

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: pdfTeX, LuaTeX Online: MathJax, KaTeX)
- Insgesamt gute Werkzeuguntersützung sowohl für ein- als auch Ausgabe.
- Klartextformat mit großer Langzeitstabilität: + 30 Jahre

Nachteile:

- Nur Präsentationsform.
- Vergleichsweise kompliziert einzulesen.

Welches Datenformat?

MathML (Core?)

Vorteile:

- Erlaubt sowohl die Präsentationsform semantisch zu erfassen (aber nicht wiederzugeben!?) als auch das 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/)

Umständlichkeit von MathML - Ein Beispiel: $ax^2 + bx + c$

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)
```

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

MathML = Die gute Idee, mathmatische Formeln durch ihren Strukturbau maschinenlesbar darzustellen, runiniert durch “Overengineering”?

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

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, Unicode, S-Ausdrücke, MetaMath
- **Nein:** MathML
- **Vielleicht:** MathMLCore, Coq, Tastaturnotation (falls bei der Eingabe verwendet)

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:

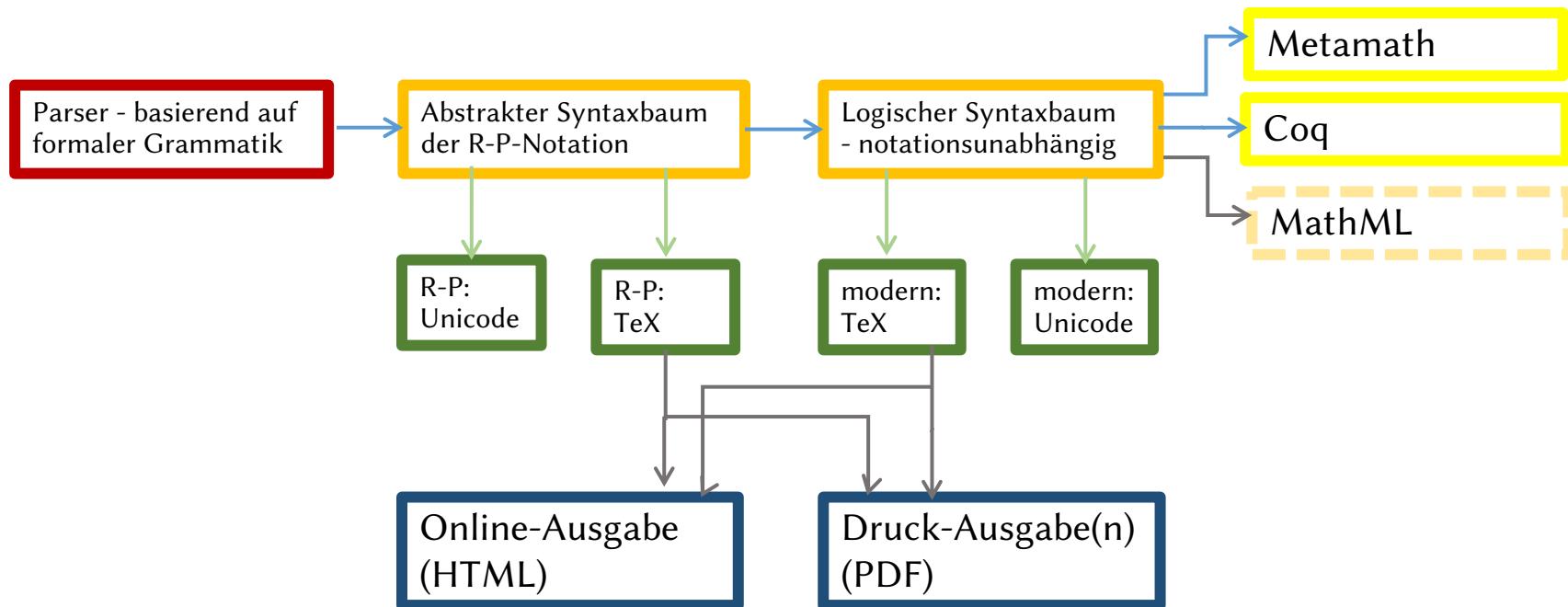
- geringste Tipp-Arbeit, **direkte Übernahme aus dem Text**, *einfachste Konvertierung*, evtl. ungewohnt
- **Werkzeuge für die Umwandlung** nach LaTeX und (Live-)Vorschau müssen erst entwickelt werden.
Im Prinzip nicht schwer, aber...

Computer-Algebra oder Beweisführungsassistenz-System:

- **Lernaufwand**, kann aber auch 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

Automatische Konvertierung von logischen Formeln in Russell-Peano-Notation:



-
1. Bei digitalen Editionen mathematischer oder formallogischer Texte besteht im Prinzip die Möglichkeit, den mathematischen und formalen Inhalt auch **aktiv bereit zu stellen** ([Arbeitsform](#)), d.h. in Form eines Quelltextes für ein Computer-Algebra- oder Beweisassistenz-System.
 2. Bei der **passiven Bereitstellung** ist zwischen der [Inhaltsform](#) und der [Präsentationsform](#) zu unterscheiden.
 3. Mit **LaTeX** gibt es einen **de-facto Standard** für mathematischen Formelsatz, aber nur für Präsentationsform.
 4. XML-basierte Formate (wie MathML) für Formeln bisher eher nicht zu empfehlen!
(Für eine maschinell leicht einzulesende Repräsentation eignet dann noch eher die Serialisierung der Strukturbäume der Formeln als S-Ausdruck.)
 5. Für die Bereitstellung in unterschiedlichen Formen und Formaten, Einsatz (teil-)**automatischer Formelkonverter** unbedingt zu empfehlen. Teilweise kann auf **vorhandene Werkzeuge** zurückgegriffen werden. Für die Präsentationsform MathJAX und KaTeX.
-