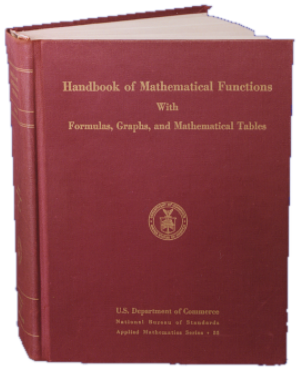
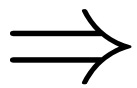
Semantic Annotation with LaTeXML

Bruce R. Miller
NIST

February 11, 2021





Digital Library of Mathematical Functions

Context & Caveats

- Application extremes:
  - DLMF: Curated source; eventual goal Content.
  - arXiv: The Wild; goal is anything we can get.
- Notational & Semantic ambiguity

$$\sin(x+y) \text{ vs. } f(x+y) \text{ vs. } a(b+c)$$

- Content usually needs both resolved;
  Accessibility often needs neither.
- Core principles:
  - Leverage grammar;
  - Minimize author typing, complexity;
    Best when natural, provides own benefits
  - Maximize ambiguity resolution.
- Reminiscent of the Defaulting discussion.
- But: Authors (& editors) choose style, not processors.

Obvious Macros

- `\cpi` → π
- `\iunit` → i;
  `eg \exp(i \iunit\cpi/3)` → $\exp(i\mathrm{i}\pi/3)$
- `\Reals` → $\mathbb{R}$
- `\abs{x}` → $|x|$
- `\floor{x}` → $\lfloor x \rfloor$
- ...

Slightly Less Obvious Macros

- `\deriv{f}{x}` → $\frac{\mathrm{d}f}{\mathrm{d}x}$
- `\ideriv{f}{x}` → $\mathrm{d}f/\mathrm{d}x$
- `\pderiv{f}{x}` → $\frac{\partial f}{\partial x}$
- `\deriv[n]{f}{x}` → $\frac{\mathrm{d}^n f}{\mathrm{d}x^n}$
- `\deriv{}{x} f` → $\frac{\mathrm{d}}{\mathrm{d}x} f$
- ...

Integrals? Choice paralysis!
Too much variety for macros (IMHO)
Need grammar support (still!)

Fancy DLMF style Macros

- `\BesselJ{\nu}` → J_ν
- `\BesselJ{\nu}@{z}` → $J_\nu(z)$
- `\BesselJ{\nu}^2@{z}` → $J^2_\nu(z)$ (power)
- `\BesselJ{\nu}''@{z}` → $J''_\nu(z)$ (derivative)

Currently Internal use;
Certain extensions & extensibility before publicizing

lxDeclare

```
\lxDeclare[role=ID,%
   tag={real variable}]{$x$}
\lxDeclare[role=FUNCTION,%
   tag={function}]{$F$}
\lxDeclare[role=OPERATOR,%
   tag={Fourier transform}]{$\mathcal{F}$}

F(x), \mathcal{F} F(x) →
```

$$F(x), \mathcal{F}F(x)$$

- What objects needs declaration?
- What attributes do they need?
- Depends on the processing model, grammar and goals!

lxDeclare Key-Values

`\lxDeclare[`*keyvals*`...]{`*pattern*`}`

- role ; Grammatical: FUNCTION, ID, OPERATOR, ...
- name, meaning, tag, description ; what it is
- scope, label ; controlling where it applies
- nowrap, replace ; more exotic

lxDeclare Patterns

```
\lxDeclare[role=ID,tag={real variable}]{$x$}
\lxDeclare[role=ID,tag={integer variable}]{$i$}
\lxDeclare[role=FUNCTION,%
   tag={function of reals}]{$f$}
\lxDeclare[role=ID,%
   tag={expansion coefficient}]{$f_?$}
```

$$f(x) = \sum_{i=0}^{\infty} f_i x^i$$

```
\lxDeclare[role=ID,tag={ecks}]{$x$}
\lxDeclare[role=ID,tag={why}]{$y$}
\lxDeclare[role=ID,tag={hatted}]{$\hat{?}$}
\lxDeclare[role=ID,tag={eckshat}]{$\hat{x}$}
```

$$x, y, \hat{x}, \hat{y}$$

[Actually '?' is \WildCard]

New Notations?

```
\lxDefMath{\baz}{\mathbf{baz}}[%
   meaning=foo,tag={Baz}]
\lxDefMath{\plus}{+}[%
   role=ADDOP,tag={Plus}]
\lxDefMath{\foo}[1]{\mathbf{foo}}[%
   meaning=foo,tag={Foo}]
```

$\baz\plus\foo{x} \to \mathrm{baz} + \mathrm{foo}$

```
\lxDefMath{\oddball}[2]{%
   \left|\left(#1\right)_{#2}\right|}[%
   tag={Oddball}]
```

$\oddball{a}{j} \to \left|(a)_j\right|$

LaTeXML's processing

- Pattern converted to XML as document is converted;
- Pattern's XML converted to XPath;
- XPath applied to XML document;
- Matched nodes annotated/adjusted;
- *Then* math parsing.

Since currently before, not interleaved with, parsing, some desirable patterns with wildcards are difficult.

Relevance to Standard?

- Parsing & Content tree may not be that relevant;
- Annotation *is*.
- Our processing model *could* be used;
- *Not* clear it's directly appropriate,
- But may inspire the right approach?