Extend your Math with Semantics and Services

Dennis Müller[1] and Michael Kohlhase[1]

Computer Science, FAU Erlangen Nürnberg, Germany

This paper uses sTeX. An interactive HTML version is available at `http://mathhub.info?a=Papers/26-ICMS-Semantics&d=paper&l=en`

There is a great difference between reading a mathematical article or textbook and discussing with a colleague before a blackboard. Both have their respective uses: the former presents the material in highly polished, carefully proofread, and maximally generalized way, the latter allows challenges, answers questions, and is tailored towards the respective interlocutors. But in the age of interactive, adaptive media, we do not have to choose and can combine aspects of both by embedding semantic annotations into scientific (mathematical) documents and using them to create **active documents**, in which semantics-informed interactive services help the reader explore the underlying meaning rather than only the presentation of the text itself.

Over the last decade, we have experimented with, and invested heavily into, realizing an ecosystem for active documents. Recently this culminated in a complete toolchain for semantically augmented documents (see [Mü26]) consisting of 1. sTeX [sTeX], a LaTeX package for semantic annotations, as a document source format, 2. FTML, a collection (and specification) of dedicated HTML attributes for semantic annotations (producable from sTeX) based on OMDoc [Koh06], 3. the FLAMS (**Fl**exiformal **A**nnotation **M**anagement **S**ystem [Fla]) system for managing FTML documents, and 4. a dedicated FTML viewer in the form of a JavaScript/WebAssembly script that provides in-situ document services by (optionally) connecting to a (remote) FLAMS system for "global context".

This paper shows how adding semantics in sTeX directly translates into FTML viewer services by developing an real-life example step by step.

1 Step 1 – No Markup

We will use the following as a running example:

`https://gl.mathhub.info/Papers/26-ICMS-Semantics/-/blob/main/source/fragments/nomarkup.en.tex?ref_type=heads`

Theorem 1. Let $Q, E_1, \ldots, E_{n_E}, U_1, \ldots, U_{n_U}$ be random variables and $D := \mathrm{dom}(U_1) \times \ldots \times \mathrm{dom}(U_{n_U})$. Then:

$$\mathbf{P}(Q \mid E_1 = e_1, \ldots, E_{n_E} = e_{n_E}) \\ = \alpha \left(\sum_{\langle u_1, \ldots, u_{n_U} \rangle \in D} \mathbf{P}(Q, E_1 = e_1, \ldots, E_{n_E} = e_{n_E}, U_1 = u_1, \ldots, U_{n_U} = u_{n_U}) \right).$$

We call Q the **query variable**, $E_1, \ldots, E_{n_E}$ the **evidence**, and $U_1, \ldots, U_{n_U}$ the **unknown** (or **hidden**) **variables**, and computing a conditional probability distribution this way **enumeration**.

This particular theorem is taken directly from our course *Artificial Intelligence II* [Ai2] and adapted from [RN21]. Briefly, the theorem states that the conditional probability *distribution* for a (finite domain) random variable Q, represented as a column vector of probabilities, given evidence ($E_i = e_i$ for similar random variables E_i with observed values e_i) can be computed by first summing up the probabilities of all possible conjunctive outcomes over all remaining *unknown* random variables U_i (component-wise for every value in the domain of Q), and subsequently *normalizing* the result by scaling it back to a probability distribution; i.e. dividing by the sum of all entries so the resulting vector has sum $= 1$; where $\alpha(\cdot)$ represents that normalization function (this, as throughout the course, implicitly assumes we know the *full joint distribution* of all random variables involved).

As is probably obvious, this is the kind of mathematical result that, for a second semester course, requires a decent familiarity with all the notations and concepts involved, and hence regularly causes confusion among students, even though mathematically, not much is happening here that does not straightforwardly follow from basic definitions (i.e. of conditional probabilities) and facts (i.e. marginalization, product rule).

Naturally, as dilligent educators, we want to help our students with understanding this result as much as possible.

2 Step 2 – Symbol references

Using sTeX and FTML, we can make this particular theorem significantly more accessible. As a first step, we can take the technical vernacular in our statement and annotate it with existing symbols; in our case, those are *"random variables"* and *"conditional probability"*.

We first add the `stex` package via `\usepackage{stex}`, and then (after the `\begin{document}`) import an existing module that gives us the symbols we need via `\usemodule{mod?probability-distribution}`.[1] This module also re-exports its own dependencies, so we get *random variable* "for free" as well.

The `\sn`-macro (short for `\symname`) is used to reference an existing symbols, the variant `\sns` trivially pluralizes the concept's name by appending an `-s` at the end. Hence, we replace `random variables` and `conditional probability distribution` by `\sns{random variable}` and `\sn{conditional probability distribution}`.

[1] Such modules can be found in multiple ways, e.g. using the FLAMS extension for VSCodium/VSCode. Here, we have made an effort to keep everything needed for this paper self-contained, so we use dedicated modules with minimal dependencies.

In the pdf, this does not make much of a difference, except for coloring the two terms differently (this behaviour can be configured to the author's liking). In the FTML, however, we get our first added-value services: Users can hover over the highlighted terms and get a popup with a definition of the hovered term. They can also *click* on the words and quickly see all available definitions and examples in mutliple languages, if available (see Figure 1).

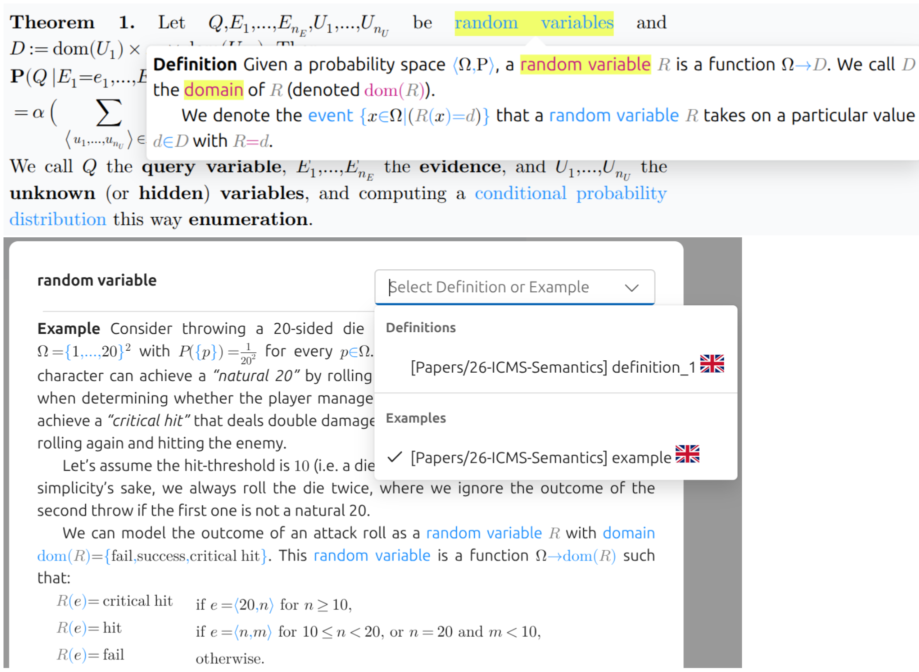

Fig. 1: Definitions on Hover and Examples on Click

Alternatively, we can mark up `random variables` with the more precise symbol for *finite domain random variable* (short:"finrandvar"), which shows the following more specific definition on hover.

https://gl.mathhub.info/Papers/26-ICMS-Semantics/-/blob/main/source/mod/finite-domain-random-variable.en.tex?ref_type=heads

If a random variable R has a *finite* domain, i.e. $\mathrm{dom}(R) = \{d_1, \ldots, d_n\}$, we can represent R as the vector of the probabilities of the d_i; i.e. we can identify R with $\langle P(R = d_1), \ldots, P(R = d_n)\rangle$

For that, we can use the `\sr`-macro (short for `\symref`) that allows us to provide a precise text to be annotated by doing `\sr{finrandvar}{random variables}`

Readers can now quickly recap the definitions of technical terms and refamiliarize themselves with the materials without having to leave the document, (to a large extent) obviating the need for *recap*-sections almost entirely.

3 Step 3 – Semantic Macros

One major source of confusion among AI2 students is the equivalent usages of *actual* probabilities of events $\mathrm{P}(R = e)$ and discrete distributions over (finite) random variables $\mathbf{P}(R)$. This is quite understandable, once conditions and conjunctions with actual events are mixed in freely in equations such as ours. The conventions that arithmetic operations on distributions are always treated component-wise allows us to write results on distributions exactly as those for (actual) probabilities, and hence much more succinctly, compactly and intuitively obvious, but at the cost of superficial ambiguity. This is not uncommon in mathematics, that notations deliberately hide (assumed to be "obvious") complexity as to not distract from the actual matter at hand, where non-experts would *initially* benefit more from cumbersome but explicit details.

[RN21], in a minimal deference to this dichotomy, therefore use a bold $\mathbf{P}$ for (conditional) distributions, and a normal P for concrete probabilities – unfortunately, we learned the hard way that in beamer mode, the two are (if at all) barely distinguishable. We'd rather make the difference more noticable by using a blackboard $\mathbb{P}$.

If in our original soure we actually wrote `\mathbf{P}(Q \mid E_1=e_1,...` everywhere, we would now have to go through the arduous process of manually finding and replacing all relevant occurences of `\mathbf` by `\mathbb`. Good practice in LaTeX is therefore to offload such *notational conventions* into macro definitions and preambles, so that subsequent changes can be made easily in one place and be propagated everywhere; e.g.

`\newcommand\probdist[2]{\mathbf{P}\left(#1\mid#2\right)}`.

But of course, we would also like to have those notations be highlighted, hoverable, and clickable, making it even easier to clearly distinguish between distributions and concrete probabilities by inspecting on-hover definitions.

Naturally, sTeX allows us to associate symbols with semantic macros we can use in math mode to do exactly that. Semantic macros can also be associated with arbitrarily many notations so we can switch conveniently. For example, we can replace `\mathrm{dom}(U_1)` by `\dom{U_1}`, the first `\mathbf{P}(Q \mid E_1=e_1)` by `\CondDist{Q}{\rveq{E_1}{e_1},...`, etc. (we refer to the linked source file below for all the details).

We can provide a new notation for `CondDist` (for conditional probability distributions) using `\notation{CondDist}[bb]{..` instead of the `\newcommand` above, and invoke it directly using either `\CondDist[bb]{Q}{E_1=...}`, or set it as the new default notation everywhere using `\setnotation{CondDist}{bb}`. Now not only do we get our new blackboard notation everywhere, all the symbols are also highlighted, hoverable and clickable.

https://gl.mathhub.info/Papers/26-ICMS-Semantics/-/blob/main/source/fragments/macros.en.tex?ref_type=heads

Theorem 1. Let $Q, E_1, \ldots, E_{n_E}, U_1, \ldots, U_{n_U}$ be random variables and $D := \mathrm{dom}(U_1) \times \ldots \times \mathrm{dom}(U_{n_U})$. Then:

$$\mathbb{P}(Q|E_1 = e_1, \ldots, E_{n_E} = e_{n_E})$$
$$= \alpha\Big(\sum_{\langle u_1,\ldots,u_{n_U}\rangle \in D} \mathbb{P}(Q, E_1 = e_1, \ldots, E_{n_E} = e_{n_E}, U_1 = u_1, \ldots U_{n_U} = u_{n_U})\Big).$$

We call Q the **query variable**, $E_1, \ldots, E_{n_E}$ the **evidence**, and $U_1, \ldots, U_{n_U}$ the **unknown** (or **hidden**) **variables**, and computing a conditional probability distribution this way **enumeration**.

Additionally, on clicking on a symbol in the FTML, users (e.g. our students) can switch notations *dynamically* based on their own presonal preferences – those who are confused by us diverging from the notations employed in the text book [RN21] can choose to change all usages of the symbol to use the notation they are familiar with (see Figure 2).

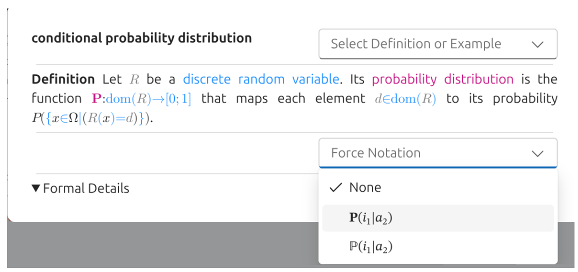

Fig. 2: Notation Switching on Click

4 Step 4 – Modules and Symdecls

Our theorem introduces a few new concepts, namely *"query variable"*, *"evidence"*, *"hidden variable"* and *"enumeration"*. Now is the time to make them symbols, so we can have all our document viewer services be available for them as well.

Good practice dictates that we should put our theorem into a separate `.tex` file for easier reuse. We then wrap the entire body of the document in the `smodule` environment and give it a name; say `inference-by-enumeration`:

```
\begin{document}
  \begin{smodule}{inference-by-enumeration}
    ...
  \end{smodule}
\end{document}
```

For details on (functional) symbol declarations with semantic macros and arguments, we refer to the sTeX documentation [Ste]. Here we focus on the simplest case that suffices for our example, namely that all we need is *pure vernacular*.

To do so, all we need is to declare them as follows:

```
\symdecl*{query variable}
\symdecl*{evidence}
\symdecl*{hidden variable}
\symdecl*{enumeration}
```

And we can subsequently mark up the last paragraph thusly:

```
We call $Q$ the \sn{query variable}, $E_1,\ellipses,E_{n_E}$ the
\sn{evidence}, and $U_1,\ellipses,U_{n_U}$ the
\sr{hidden variable}{unknown} (or \sr{hidden variable}{hidden})
\sr{hidden variable}{variables}, and computing a \sn{conditional
probability distribution} this way \sn{enumeration}.
```

We can also now import this module using `\usemodule{<path>?inference-by-enumeration}` elsewhere to reuse our new symbols.

5 Step 4 – Definitions and Theorems

So far so good, but of course, without definitions, examples, notations, etc., there is nothing to see on hover.

Let's change that by marking up our theorem as an actual theorem, and the last paragraph as a definition. That way, the FTML viewer knows to show us this paragraph whenever a reader hovers over one of our new symbols.

For the former, we wrap our entire paragraph in an `sassertion` environment (which is used for *assertions*: theorems, lemmata, axioms, etc.). We want to *style* it as a theorem, so:

```
\begin{sassertion}[style=theorem]
    Let $Q,E_1,\ldots,E_{n_E},...$
\end{sassertion}
```

The analogous environments for *definitions* and *examples* are (naturally) `sdefinition` and `sexample`. Such *semantic paragraphs* can be nested, so we could as well wrap the entire theorem in an `sdefinition` environment, or selectively, just the last paragraphs, depending on preference. All three environments also have an *inline*-analogon as a macro though, namely `\inlineass`, `\inlinedef`, and `\inlineex`.

For a definition, we can also specify what symbols the definition is *for* – and if we want it to be shown on hover, we need to do so. We can either do that via an optional argument `for={...}`, or by marking the *definienda* in the text explicitly – this also changes the highlighting and deactivates *hover*,but keeps on-click functionality (so we can look at *other* definitions and examples). Hence:

```
\inlinedef{
  We call $Q$ the \definame{query variable}, $E_1,\ellipses,
  E_{n_E}$ the \definame{evidence}, and $U_1,\ellipses,U_{n_U}$
  the \definiendum{hidden variable}{unknown} (or
  \definiendum{hidden variable}{hidden}) \definiendum{hidden
  variable}{variables}, and computing a \sn{conditional
  probability distribution} this way \definame{enumeration}.
}
```

6 Step 5 – Variables

Last but not least, as with symbols we can also mark up *variables*. This, again, helps to resolve some confusion – when hovering over a *bound* variable, the FTML viewer will also highlight the *bindeing operator*, making it clear where the particular variable is quantified over. In our case, hovering over any of the u_i will thus highlight the $\sum$, to show us that this is the value being summed over (rather than some other variable).

For the details, we again refer to the sTeX documentation [Ste]. Suffice it to say, that variables can be declared like symbols are, except for using `\vardef` instead of `\symdecl`, and that they always need a notation.

Since variables don't have (global, explicit, natural language) definitions, FTML viewer will instead show their *types* on hover. These types can be annotated via an optional argument for `\vardef`. For example:

```
\vardef{nQ}[name=Q,type=\PosNat]{n}
\vardef{vQ}[name=Q,type=\finrandvar{\nQ}]{Q}
```

this labels the variable n as a non-zero natural number, and Q as a finite random variable with a domain of cardinality n, and give us the semantic macros `\nQ` and `\vQ`. If any formal expression is fully annotated with well-typed symbol only, FLAMS will also check the formal correctness of the expression and allow us to do type inference on subexpressions.

https://gl.mathhub.info/Papers/26-ICMS-Semantics/-/blob/main/source/fragments/variables.en.tex?ref_type=heads

Theorem 1. Let $Q, E_1, \ldots, E_{n_E}, U_1, \ldots, U_{n_U}$ be random variables and $D := \mathrm{dom}(U_1) \times \ldots \times \mathrm{dom}(U_{n_U})$. Then:

$$\mathbb{P}(Q|E_1 = e_1, \ldots, E_{n_E} = e_{n_E})$$
$$= \alpha(\sum_{\langle u_1,\ldots,u_{n_U} \rangle \in D} \mathbb{P}(Q, E_1 = e_1, \ldots, E_{n_E} = e_{n_E}, U_1 = u_1, \ldots, U_{n_U} = u_{n_U})).$$

We call Q the **query variable**, $E_1, \ldots, E_{n_E}$ the **evidence**, and $U_1, \ldots, U_{n_U}$ the **unknown** (or **hidden**) **variables**, and computing a conditional probability distribution this way **enumeration**.

We can also provide more detailed information in types by additionally annotating the "formal type" with a comment (see Figure 3), which allows readers to better distinguish the query, hidden variables, and evidence in our expressions.

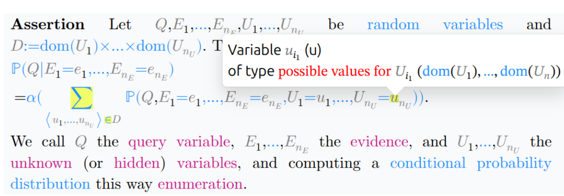

Fig. 3: Hovering Over a Variable Shows the Type and Highlights the Binder

```
\vardef{vQ}[name=Q,type=\commented{\finrandvar{\nQ}}{
    \text{the query variable}
}]{Q}
```

Type inference and symbolic computation is our latest feature, which is still in its early stages and under active experimentation and development, but it should be noted that especially in situations such as ours ("is this a probability or a probability *distribution*? The usual multiplication, scalar multiplication or *component-wise* multiplication?"), this is extremely informative for readers.

When a user selects any subexpression, a popup will show us a button in the corner, which when pressed will send a request to FL∀M∫, which will do exhaustive computation of anything that can be computed (e.g. arithmetic expressions on concrete values, beta-reduction), attempt to infer the type of the expression, and present the results to a user; e.g. when selecting $E_1 = e_1$, we see the result in Figure 4.

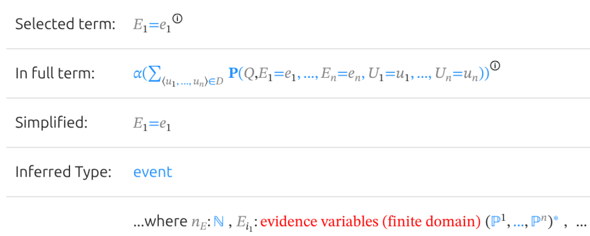

Fig. 4: Type Inference, Exhaustive Computation, and Relevant Context for a Selected Subexpression

7 Conclusion

We have presented the sTeX ecosystem for flexiformal – semantically annotated documents presented interactively in an active documents player, exhibiting the contribution of a growing set of annotations to the reading/understanding experience.

We have been using it as the technical basis of the ALeA system (Adaptive Learning Assistant [ALeA]), which has been used for three years on up to 1000 students per semester at FAU Erlangen-Nürnberg. ALeA orthognally extends the flexiformal sTeX ecosystem by learner modeling and thus learner-adaptive content selection/generation.

We believe sTeX software ecosystem is mature enough for use in AI-supported learning/teaching and generally for scientific publication (which is essentially a learning/teaching exercise not tied to a formal educational setting). We invite all our colleagues to give sTeX a spin and join the sTeX community.

Acknowledgments The work reported in this article was conducted as part of the VoLL-KI project (see `https://voll-ki.de`) funded by the German Research/Education Ministry under grant 16DHBKI089 and the ongoing FAUstairs project (see `https://faustairs.fau.de`) funded by the Stiftung Innovation in der Hochschullehre under grant StIL:1001-3096.

References

[Ai2] *AI-2 Course Home — ALeA*. URL: `https://alea.education/FAU/ai-2/SS26` (visited on 04/08/2026).

[ALeA] *ALeA: Adaptive Learning Assistant*. URL: `http://alea.education` (visited on 07/16/2024).

[Fla] FLAMS *Documentation*. URL: `http://kwarc.github.io/FLAMS/` (visited on 05/09/2025).

[Koh06] M. Kohlhase. *OMDoc – An open markup format for mathematical documents [Version 1.2]*. LNAI 4180. Springer Verlag, Aug. 2006. URL: `http://omdoc.org/pubs/omdoc1.2.pdf`.

[Mü26] D. Müller. "The Flexiformal Manifesto Made Manifest". In: *The 2026 International Conference on Breakthroughs in Artificial Intelligence (BAI 26)*. Ed. by K. Daimi and A. Alsadoon. LNNS. to appear May 16th, 2026. Springer, 2026. URL: `https://mathhub.info/?a=Papers/26-BAI-FLAMS&rp=paper.tex`.

[RN21] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 4th. Pearson, 2021.

[Ste] *The sTeX Documentation*. URL: `https://mathhub.info/?a=sTeX/Documentation&rp=documentation.tex` (visited on 04/08/2026).

[sTeX] *sTeX: A semantic Extension of TeX/LaTeX*. URL: `https://github.com/sLaTeX/sTeX` (visited on 05/11/2020).