Sophize Markdown and Collaboration Interface

Abhishek Chugh

Sophize Foundation
Bengaluru, India
abc@sophize.org

Abstract

Sophize is a novel mathematics library and discussion platform with a mission to help our users find and organize mathematical proofs. We present a Sophize's novel yet familiar knowledge organization scheme that is used to represent a wide variety proofs. With this knowledge organization scheme at its core, we have engineered that platform to aggregate knowledge from sources. The platform meaningfully combines knowledge from published research, encyclopedias such as PlanetMath and Wikipedia, informal computer programs and formal systems such as Metamath; and allows its users to run a federated search over them. To make this happen, two developments were necessary. First, we developed an open plugin architecture that allows proofs from computer programs to be generated as required. Second, we extended the Markdown language to represent the connections between between mathematical objects found across various sources of knowledge. In addition, we also utilized this new language to create a novel communication system built specifically to aid mathematicians in solving problems collaboratively.

1 Introduction

Background: The Sophize Platform

Sophize is a novel mathematics library and discussion platform. The author has developed the platform over the course of the last two years at `https://sophize.org`. Sophize's primary mission is to help users find existing proofs of mathematical statements, to discover new proofs, and to utilize this knowledge in their work. We combine knowledge from multiple resources and have accumulated thousands of definitions, theorems, and proofs from a wide variety of sources, including some published research, the PlanetMath encyclopedia, Wikipedia, and the Metamath formal system [Meg19].

Proofs in Sophize

Mathematical proofs are based on a variety of foundations such as ZFC, intuitionistic logic, and type theory. The arguments used in any proof are considered valid or not based on criteria that can vary. Most academic mathematics is peer-reviewed and published, but some mathematical proofs can be found in community curated sources such as Wikipedia. Proofs can also be algorithmically generated, and at the highest level of verification, they are represented and verified using a formal system.

Sophize combines such an expansive range of proofs into a dense graph of propositions and logical arguments that aggregates knowledge from several documents and other data sources. We use this graph and combine it with the set of foundations and verifications chosen by each user to create proofs tailored to their needs. This introductory video gives an overview of the platform's offerings: `https://youtu.be/Wb1JbW9Otek`.
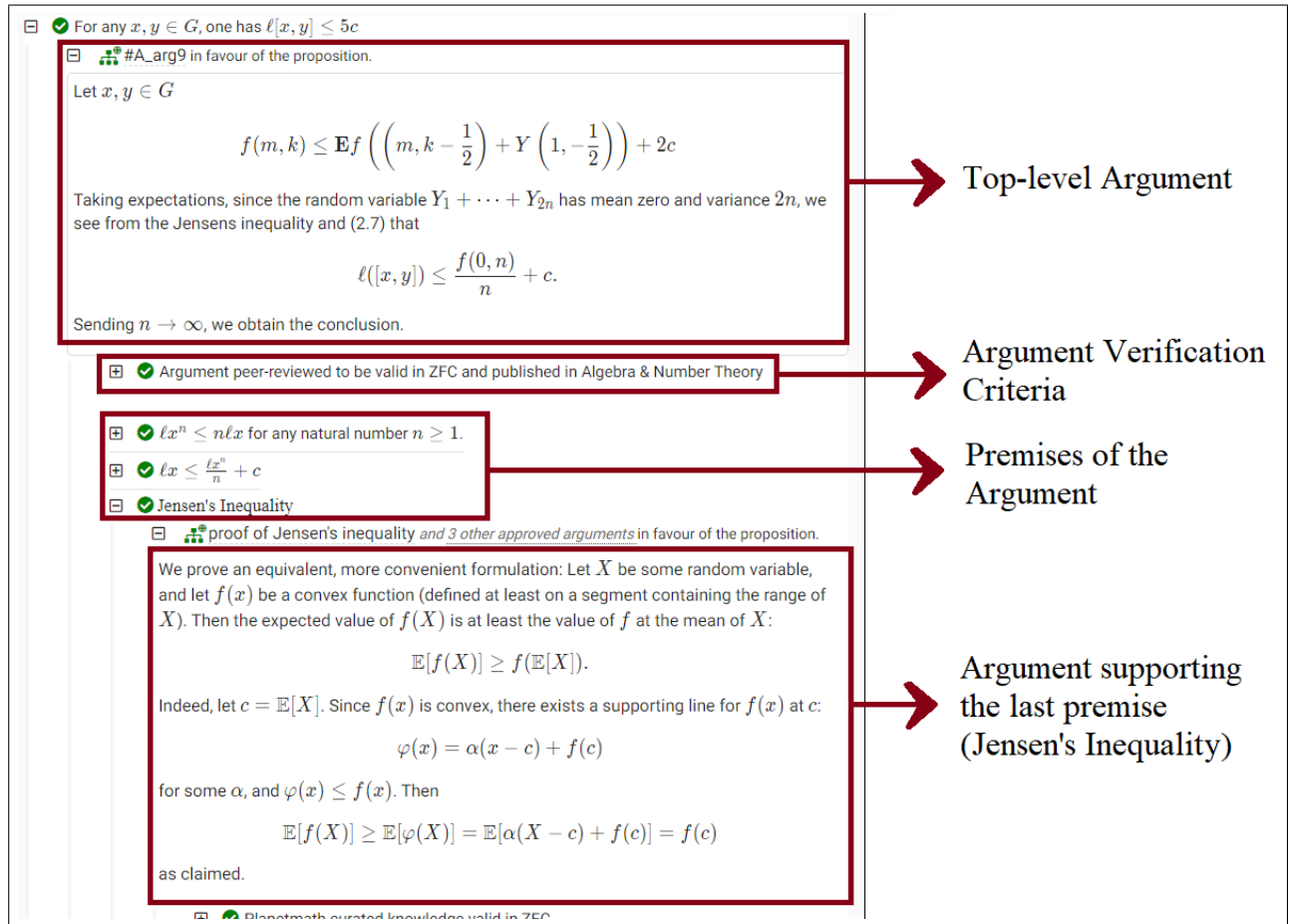

Figure 1: A proof displayed as a graph (tree) of arguments and propositions on the Sophize platform.

This work can also be seen as a step towards formalizing the network of information that exists in the connections of mathematical objects. The committee on planning a global library of the mathematical sciences recognized that this network is largely unexplored, and formalizing it has tremendous potential to accelerate math research [oPaGLotMS14].

Hence, we required novel knowledge organization techniques to connect mathematical entities from such a wide range of formal and informal knowledge sources. And thus, one of the main challenges towards building the Sophize platform was to create a simple way to embed mathematical entities on the web. This is the primary problem that this work addresses.

Mathematics collaborations

As a discussion platform, Sophize focuses on creating an interface that helps mathematicians discuss mathematics and collaboratively discover new proofs. The existing web platforms and technologies, while useful, are, of course, not tailored for academic mathematical discussions. It becomes very challenging to organize discussions, and participants struggle to keep up with the rapidly evolving ideas. In our current environment, where in-person collaboration opportunities have drastically reduced due to the ongoing COVID-19 pandemic, aiding mathematicians to work together online can prove to be quite beneficial.

Contribution

In this paper, we present the features of Sophize Markdown, an extension of the Markdown language. It is convenient enough to be used for casual discussions of mathematical ideas over the web. It is also powerful enough to embed mathematical entities such as definitions, theorems, and proofs from various sources, including formal systems. It thus plays an integral role in the two problems that we have mentioned above.

We then elaborate on the communication interface designed to help mathematicians discuss mathematics and collaboratively discover new proofs. The design focuses on making the collaboration more productive and enjoyable by building the right set of technical tools that aid in effective organization and summarization of existing progress.

Section 3 presents Sophize Markdown as a lightweight interface for representing math content, and Section 4 discusses its collaboration facilities. Finally, Section 5 concludes the paper.

Acknowledgements

2 Sophize's Knowledge Organization Scheme

2.1 Sophize's Data Model

Before we describe the Sophize Markdown language, it will be necessary to briefly describe how Sophize models mathematical entities. The data model is published in JSON schema [sop] and popular package managers such as MVN, npm and PyPI. We describe relevant parts of the concepts used here.

A **resource** is an abstract concept inherited by all other top-level concepts like terms, propositions, and arguments. Each resource has a URI and contains fields such as search tags and citations.

A **URI** consists of two parts: a namespace-like identifier called its dataset-id, which indicates the data source, and a resource-id that specifies the resource type and its unique name in the data source. The dataset-id may be omitted if it can be inferred from the surrounding context.

For example, the Pythagorean theorem (a **P**roposition) represented in the Metamath project may have the URI *metamath/P_pythagorean* and the definition of cone (a **T**erm) extracted from Wikipedia may have the URI *wiki/T_cone*. When used inside another resource in the 'wiki' dataset, cone's definition can be referred to simply as *T_cone*.

A **term** is a clearly defined entity that can be used to make up a valid proposition. It can be a mathematical object, operator, symbol, data structure, algorithm, or even a person. 'Meaningless' primitives in formal theories are also categorized as terms.

A **proposition** is a grammatically valid statement that can be either true or false. Axioms, theorems, conjectures, hypotheses, lemmas, corollaries, and converses are all classified as propositions.

An **argument** is a set of propositions called premises along with a concluding proposition that is claimed to follow from the premises. In addition, most arguments include supporting text that explains how the conclusion follows from the premises. A proof is seen as a directed graph of arguments and propositions.

2.2 Polymath Projects

In order to better understand the nature of online collaborations, we studied the Polymath projects [pol], which are massively collaborative online mathematical projects. Conveniently, the project leaders publically analyzed these projects and suggested technical improvements suitable for such collaborations. In fact, a large portion of the interface is designed specifically to overcome the issues noted by Timothy Gowers [Gow09] and Terence Tao [Tao09].

3 Sophize Markdown

Markdown is a lightweight markup language for creating formatted text using a plain-text editor. Markdown is very widely adopted on the web, and it makes it quite simple to add lists, headers, bold or italic fonts, images,

and more. One can choose from several slightly varying Markdown specifications. We start with the widely supported CommonMark specification and create extensions for the features that we need. The extensions are implemented using the 'markdown-it' [mar] JavaScript parser. The parser for Sophize Markdown produces an abstract syntax tree (AST). A separate renderer module utilizes the AST tree to create the appropriate HTML or Single Page Application (SPA) libraries. Many of the features of Sophize Markdown are demonstrated at: `https://youtu.be/5UYOpQwcjCk`

3.1 Embedding structured data

With Sophize Markdown we can easily embed resources such as terms, propositions, and arguments. Links can be added using the resource's URI, and the following formats are supported:

- #URI[| OPTIONS]

  Examples: #wiki/T_cone, #planetmath/P_covering_lemma|NO_LINK|LC

- #(URI, 'Custom Text'[| OPTIONS])

  Examples: #(wiki/T_cone, 'cones'), #(P_green_tao_theorem, 'Green–Tao Theorem'|NAV_LINK)

There are multiple resource link options to embed a resource in the Markdown. These are required to make it convenient for the content creators to provide a rich experience to their readers. Some of the commonly used options are summarized below.

3.1.1 Link Text Options

NAME (default): This option fetches the resource and sets the link text to the name of the resource. For terms, the link text is set to the phrase of the term.

LOWER_CASE (LC) or UPPER_CASE (UC): These options become quite useful when adding a link in the beginning or middle of a sentence where adding the name in its default case would be grammatically incorrect.

Custom text: A link can also explicitly specify the link text by enclosing it in single quotes.

The following example shows the above options in use. Note that we assume that the dataset-id of URIs is available from the context and thus not specified in the URIs.

```
#P_cosine_law|UC is a generalization of the #P_pythagorean for all kinds of
#(T_triangle, 'triangles').
```

The above code is rendered as:

> Law of cosines is a generalization of the Pythagorean theorem for all kinds of triangles.

3.1.2 Link Type Options

OVERLAY_LINK (default): By default, clicking on links opens up a modal dialog with the resource's summary.

NAV_LINK: We can create a hyperlink to the resource's URL using the using NAV_LINK option.

NO_LINK: NO_LINK creates a non-clickable element whose hover text is the URI of the resource.

For example, the following Markdown code has all three link types:

```
#T_matrix_multiplication|NO_LINK is a #(T_commutative_property,
'non-commutative'|NAV_LINK) #T_binary_operation.
```

The above code is rendered as:

> Matrix multiplication is a non-commutative binary operation.

The first element shows '#wiki/T_matrix_multiplication' on mouse-hover. The second link navigates the page to a different URL. Clicking on the third link pops up a modal dialog box like so:

Figure 2: Modal dialog box for the term binary operation. These boxes can also have content that uses all three types of links.

3.1.3 Resource Expansion

Instead of creating a link, this option expands the resource in place. For terms and propositions, the definition and the statement are added in place, respectively. For arguments, its premises, conclusion, and the argument text is added.

The following is a sample article where we re-use concepts and theorems already extracted from multiple sources (say, Wikipedia and the Oxford dictionary):

```
# Conic section

A conic section (or simply conic) is a curve obtained as the intersection of the
#(wiki/T_conical_surface, 'surface') of a #wiki/T_cone with a #wiki/T_plane. There are
three types of conic sections.

## Ellipse
#oxford/T_ellipse|EXPAND #wiki/P_ellipse_area|EXPAND

## Parabola
#oxford/T_parabola|EXPAND The area of a parabola is unbounded.
...
```

The above code is rendered as:

> ## Conic Section
>
> A conic section (or simply conic) is a curve obtained as the intersection of the surface of a cone with a plane. There are three types of conic sections.
>
> ### Ellipse
>
> An ellipse is a regular oval shape, traced by a point moving in a plane so that the sum of its distances from two other points (the foci) is constant, or resulting when a cone is cut by an oblique plane which does not intersect the base. The area $A_{ellipse}$ enclosed by an ellipse is
>
> $$A_{ellipse} = \pi a b$$
>
> where a and b are the lengths of the semi-major and semi-minor axes, respectively.

Parabola

A parabola is a symmetrical open plane curve formed by the intersection of a cone with a plane parallel to its side. The area of a parabola is unbounded.

...

3.1.4 Status Indicators

For propositions, a Truth Value Icon indicating whether or not there is a proof is added next to the link. Clicking on the icon brings up the proof graph as shown in Figure 1. Similarly, an icon is added next to an argument which indicates whether the argument is valid or not. These options can be turned on or off depending on the input and the context. Further details require an in-depth understanding of the Sophize knowledge organization scheme which is not our focus here.

3.2 Formal Language Support

In formal languages, definitions of terms, statements of propositions, and supporting argument text may be parseable by an external parser. We can convert the externally parsed output into Markdown in such a case, where each term is automatically linked to the appropriate resource. This provides a convenient interface, where the user types in the native language, and the final output automatically allows users to explore all concepts that make up the input statement in depth. Currently, this is demonstrated in the use of Sophize Markdown with the Metamath language.
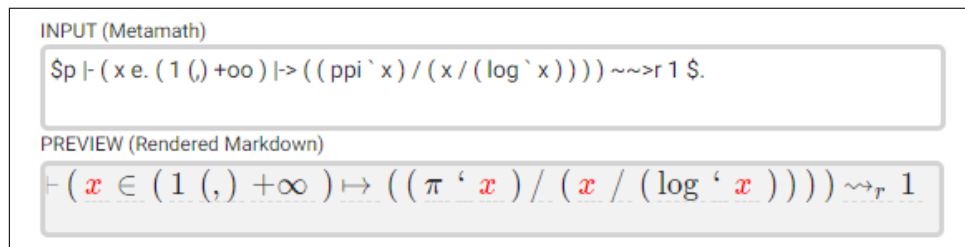

Figure 3: The prime number theorem written in Metamath language and rendered as Markdown in real time.

3.3 LaTeX Support

While multiple applications extend Markdown to support TeX, there is no standardized syntax specification. However, Pandoc is the most widely used tool for converting LaTeX to Markdown, and its specification is well documented and tested. Thus, we use their specification for our extension [pan]:

> Anything between two $ characters will be treated as TeX math. The opening $ must have a non-space character immediately to its right, while the closing $ must have a non-space character immediately to its left, and must not be followed immediately by a digit. Thus, \$20,000 and \$30,000 won't parse as math. If for some reason you need to enclose text in literal $ characters, backslash-escape them and they won't be treated as math delimiters.

> For display math, use $$ delimiters. (In this case, the delimiters may be separated from the formula by whitespace. However, there can be no blank lines between the opening and closing $$ delimiters.)

The following example summarizes the specification:

```
Einstein's most famous equation is $E=mc^2$ but it is his handwritten theory of
happiness is what fetched $1.3 million recently. While it may not fetch many \$s, I
like the field equations more: $$G_{\mu \nu }+\Lambda g_{\mu \nu }=\kappa T_{\mu \nu }$$
```

The above Markdown code is rendered as:

Einstein's most famous equation is $E = mc^2$ but it is his handwritten theory of happiness is what fetched \$1.3 million recently. While it may not fetch many \$s, I like the field equations more:

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = \kappa T_{\mu\nu}$$

4 Sophize Collaboration

Sophize's communication interface is designed to allow multiple researchers to collaboratively solve a problem. In such open online collaborations, a lot of different ideas and approaches get introduced rapidly. Going through all these comments and making sense of the rapidly evolving ideas becomes a daunting task that can dissuade even the experts and the highly motivated. We believe that we have created a novel interface that can significantly mitigate this problem. The work is primarily influenced by recommendations made by Timothy Gowers and Terence Tao after organizing several Polymath projects [pol]. We believe that the interface can aid many collaboration types - with few or many participants. We demonstrate the various features of this interface by incorporating actual data from a Polymath project at: `https://youtu.be/d3gaalJ7UQM`.

Figure 4: Some elements of the Sophize Collaboration Interface.

Requirements

Helping users make sense of the conversation and allowing them to get up to speed with existing progress as quickly as possible is perhaps the most important requirement for managing online collaborations. To solve this problem, we need to organize the ideas in an intuitive way and to summarize the progress. A reader should be able to quickly grasp the current state, and project moderators need a way to manage the direction of various threads to avoid the big picture from being obscured.

Other technical requirements were discussed by some project participants and summarized by Tao on his blog [Tao09] (lightly edited):

- LaTeX support

- Group moderation

- Comment editing and preview

- Comment numbering and/or threading

- Wiki-like features, i.e., group-editable, publicly-viewable documents with version control

- Some way to view all recent comments or developments in a project

- Easy registration process

- Easy commenting process (i.e., no technical knowledge required)

- Permanent URLs for posts and comments (for link-back purposes)

We note that Sophize provides all the above features except that it doesn't yet allow users to look at the previous versions of documents. We summarize innovative features of the platform below:

4.1 Hierarchical Page Organization

As noted by Gowers, to make sense of the numerous ideas discussed, they need to be organized in a natural hierarchical way. The 'proof discovery tree', as described by Gowers [Gow09], would have a precise approach to the main problem, and each of the subproblems that come up would have its own trees. The leaves of the tree would thus be sub-problems that get solved without division into sub-problems.

Sophize brings the 'Proof-discovery tree' to life by organizing the project using neatly organized wiki-like pages. Pages have a parent-child relationship allowing them to be organized like a tree. Discussion comments can be posted on any page, and moderators can rearrange them the way they think is best for the project. Moderators are also encouraged to add an overview on each page that quickly informs readers of the significant ideas that they should be familiar with before they start contributing.

4.2 Comment Summaries

Even after the hierarchical division of comments, some pages can still have a large number of comments, which make them difficult to follow. To overcome this problem, we allow project moderators to create summaries of discussions that have taken place so far. Comment summaries get embedded in the comment tree as a parent for comments they summarise. For example, they can club together multiple comment chains that are dead ends so that each reader doesn't waste time on them. Or they could summarize a sub-proof that was arrived at after dozens of comments.

4.3 Sophize Markdown benefits

Project collaborators get all the benefits of Sophize Markdown, such as easy linking of existing math content, LaTeX support, and live preview of pages and comments being written or edited.

Comments in Sophize are numbered serially (starting with 1 for each collaboration), each comment can be referenced with the hashtag notation (e.g., '#3' for the 3rd comment). Comments in an external project can be referenced by providing the project's URI (#URI/COMMENT_NUMBER).

Also, we make it extremely easy to extract terms, propositions and proofs from comments. These resources not only help in better understanding of the project, it also helps build Sophize's library.

4.4 Reducing noise

Sophize gives project moderators the usual tools like spam removal and fine-grained access control to manage participants. We also add a new feature where certain comments can be marked hidden. Often, threads of dozens of comments can be just noise because they were based on confused concepts or mistaken assumptions. Such comments should not be marked as spam. By marking such threads as 'hidden', moderators can ensure that these are not shown by default. But any user can view them when required.

5   Conclusion and Future Work

We have presented Sophize Markdown, a lightweight language for representing mathematical knowledge on the web. Using this language, we can easily embed mathematical entities like definitions, theorems, and proofs, making it very convenient to find and browse relevant content. Sophize Markdown plays a central role in organizing proofs from various data sources on the Sophize platform. While there are other ways to represent semantic knowledge, such as content MathML and OpenMath, their scope is somewhat limited to specifying the meaning of the mathematical formula. The sTeX system [Koh08] is perhaps most similar to Sophize Markdown but it needs creation of LaTeX documents and use of LaTeXML systems. This makes its use in real-time web workflows impractical.

We utilized the Sophize Markdown language to create an innovative mathematics communication interface. It is specially designed to aid online mathematics collaborations and can help actualize their vast latent potential. We believe that it can not only aid large-scale collaborations like the Polymath projects but will also be useful for private collaborations and workshops like those organized by AiM.

In the future, we want to make a substantial portion of math literature embedded with structured data available at a much greater scale. Thus, we will improve Sophize Markdown's support for LaTeX features like equation numbering, tables, and references. In addition, we are developing an online tool to easily extract content like definitions, theorems, proofs from math literature by automating some of the tedious tasks. This process will be aided by the recent supervised learning techniques [GM19] that classify scientific statements in literature.

Sources

The Sophize Markdown parser and its Angular renderer source code are available under an MIT license at `https://github.com/Sophize/sophize-md-parser` and `https://github.com/Sophize/ngx-sophize-md-renderer`, respectively. The NPM packages for these projects are named 'sophize-md-parser' and 'ngx-sophize-md-renderer'.

References

[GM19]   Deyan Ginev and Bruce R. Miller. Scientific statement classification over arxiv.org, 2019.

[Gow09]   Timothy Gowers. Can polymath be scaled up? `https://gowers.wordpress.com/2009/03/24/can-polymath-be-scaled-up/`, Mar 2009.

[Koh08]   Michael Kohlhase. Using latex as a semantic markup format. *Mathematics in Computer Science*, 2(2):279–304, Dec 2008.

[mar]   markdown-it parser. `https://github.com/markdown-it/markdown-it`.

[Meg19]   Norman D. Megill. *Metamath: A Computer Language for Mathematical Proofs*. Lulu Press, Morrisville, North Carolina, 2019. `http://us.metamath.org/downloads/metamath.pdf`.

[oPaGLotMS14] Committee on Planning a Global Library of the Mathematical Sciences. Developing a 21st century global library for mathematics research, 2014.

[pan]   Pandoc user's guide. `https://pandoc.org/MANUAL.html#math`.

[pol]   Polymath blog. `https://polymathprojects.org/`.

[sop]   Sophize datamodel. `https://github.com/Sophize/datamodel-json`.

[Tao09]   Terence Tao. Imo 2009 q6 mini-polymath project: impressions, reflections, analysis. `http://tiny.cc/polymath-reflections`, Jul 2009.