Sophize Markdown and Collaboration Interface

Abhishek Chugh

Sophize Foundation
Bengaluru, India
abc@sophize.org

Abstract

We extend the markdown language to represent the connections of mathematical objects that exist across various sources of knowledge. Using the new language, we demonstrate an interactive interface that helps its users effortlessly explore mathematics content on the web. We also utilize this new language to create a novel communication system built specifically to aid mathematicians in solving problems collaboratively.

1 Introduction

Sophize is a novel Mathematics library and discussion platform. Our primary mission is to help our users find existing and discover new proofs of mathematical statements and utilize this knowledge for their work. All mathematical proofs can be seen as directed acyclic graphs of logical arguments that span across a large number of documents and databases. These proofs arise from a variety of foundations such as ZFC, intuitionistic logic and type theory. The arguments used in any proof are considered valid based on varying criteria - most academic mathematics is peer-reviewed, 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. Thus, aggregating proofs from such a wide variety of sources that utilize a wide variety of foundations and verification criteria is a challenging problem that requires novel knowledge organization techniques and user interfaces needed for such an organization. This article focuses on these user interfaces that help to navigate and understand existing proofs and discover new ones. Our 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 this network is largely unexplored, and formalizing it has tremendous potential to accelerate math research[oPaGLotMS14]. This introductory video gives an overview of how structured knowledge can help: `https://youtu.be/Wb1JbW9Otek`.

In this paper, we first discuss the details of Sophize Markdown, an extension of Markdown language. It is convenient enough to be used for a casual discussion of mathematical ideas over the web. It is also powerful enough to embed mathematical entities such as definitions, theorems, and proofs from a wide variety of sources, including formal languages. Then, we elaborate on the communication interface designed to help mathematicians discuss mathematics and collaboratively discover new proofs. The design focuses on making it easy for the participants to understand the current progress even when many collaborators have posted a large number of comments. We studied the nature of collaboration on Polymath projects - massively collaborative online mathematical projects to develop our interface design. Conveniently, the leaders have 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 Prof Timothy Gowers [gow09] and Prof Terence Tao[wha09]. We

believe that these features will not only aid large-scale collaborations like Polymath but will also be useful for private collaborations and workshops like AiM.

2   Preliminaries

Before we describe the Sophize markdown language, it would be necessary to briefly describe how we model 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.

Resource

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 dataset-id, which indicates the data source, and a resource-id that indicates what kind of resource it is along with its unique name in the data source. The dataset-id may be omitted if it can be inferred from the surrounding context.

For example, Pythagoras's theorem (**P**roposition) represented in the Metamath project may have the uri *metamath/$\boldsymbol{P}$_pythagoras* and the definition of cone (**T**erm) extracted from wikipedia may have the uri *wiki/$\boldsymbol{T}$_cone*. When used inside another resource in the 'wiki' dataset, cone's definition can be referred to simply by *T_cone*.

Term

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, entity, data structure, algorithm, or even a person. 'Meaningless' primitives in formal theories are also categorized as terms.

Proposition

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

Argument

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 have a supporting text that explains how the conclusion follows from the premises. A proof is seen as a directed graph of arguments and propositions.

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/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 that are needed for our requirements. The extensions are implemented using the 'markdown-it' [MI] JavaScript parser. The parser for Sophize markdown produces an AST tree. A separate renderer module utilizes the AST tree to create the appropriate HTML or Single Page Application (SPA) libraries. Many of the features described here are demonstrated at: `https://youtu.be/5UYOpQwcjCk`

3.1   Parser Extension: Tex Math

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.)

3.2 Parser Extension: Resource Links

This extension allows easily adding links to resources such as terms, propositions, and arguments. Links can be added using the URI, and the following formats are supported:

- #URI[| OPTIONS]

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

- #(URI, ['LINK_TEXT'][| OPTIONS])

  Examples: #(wiki/T_cone, 'cones'), #(P_greens_lemma, EXPAND)

3.2.1 Options

The different resource link options provide different ways to embed a resource in the markdown. These are required to make it convenient for the content creators to provide a rich experience for their readers. The following are most commonly used:

NAME (default)

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.

EXPAND

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, the premises, conclusion, and the argument text is added in place.

OVERLAY_LINK (default)

In the case of an overlay link, clicking on the link opens up a modal dialog to show a resource summary instead of navigating to another URL.

NAV_LINK

Clicking the link causes the browser to navigate to the web page indicated by the resource's URI.

LOWER_CASE or LC

Lower cases the link text. This becomes quite useful when adding a link in the middle of a sentence where adding the name in its default case would be grammatically incorrect.

SHOW_TVI (default)

For propositions only. Adds a truth value icon indicating whether or not there is a proof available for the proposition. Clicking on the icon brings up the proof as a graph of arguments.

HIDE_TVI

The truth value icon is hidden.

3.3 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 delightful interface, where the user types in the native language, and the final output automatically allows users to deeply explore all concepts than make up the input statement. Currently, this is demonstrated in the Sophize Markdown with the Metamath language.

4 Sophize Collaboration Interface

Sophize collaboration interface is designed to allow multiple researchers to simultaneously discuss multiple approaches to solve the problem of interest. In 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 dissuades even the experts and the highly motivated. This work is primarily influenced by recommendations made by Professor Timothy Gowers and Professor Terence Tao after organizing a few polymath projects. 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`.

Requirements

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

Some of the other technical requirements were discussed by some project participants and summarized by Prof. Tao on his blog[wha09] (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 Prof Gowers, to make sense of the numerous ideas discussed, they need to be organized in a natural hierarchical way. The 'proof discovery tree', as Gowers calls it [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 would quickly inform 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 would be difficult to follow. To overcome this problem, we allow project moderators to create summaries of discussions that have taken place so far. For example, they can club together multiple comment chains that are dead ends so that each reader doesn't waste their time. Or to save time for future readers, 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 different projects can be references by providing the URI (#URI/COMMENT_NUMBER).

4.4   Moderation Tools

Sophize gives moderators 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 choose to see them if they wish to.

Sources

Sophize markdown parser and its Angular renderer source code is available with 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'.

Acknowledgements

...

References

[gow09]   Can polymath be scaled up?, Mar 2009.

[MI]   Markdown-It. markdown-it/markdown-it.

[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.

[Sop]   Sophize. Sophize/datamodel-json.

[wha09]   Imo 2009 q6 mini-polymath project: impressions, reflections, analysis, Jul 2009.