Formal Mathematics on Display:
A Wiki for Flyspeck

Carst Tankink\textsuperscript{1}, Cezary Kaliszyk\textsuperscript{2}, Josef Urban\textsuperscript{1}, and Herman Geuvers\textsuperscript{1,3}

\textsuperscript{1} ICIS, Radboud Universiteit Nijmegen, Netherlands
\textsuperscript{2} Institut für Informatik, Universität Innsbruck, Austria
\textsuperscript{3} Technical University Eindhoven, Netherlands

Abstract. The Agora system is a prototype “Wiki for Formal Mathematics”, with an aim to support developing and documenting large formalizations of mathematics in a proof assistant. The functions implemented in Agora include in-browser editing, strong AI/ATP proof advice, verification, and HTML rendering. The HTML rendering contains hyperlinks and provides on-demand explanation of the proof state for each proof step. In the present paper we show the prototype Flyspeck Wiki as an instance of Agora for HOL Light formalizations. The wiki can be used for formalizations of mathematics and for writing informal wiki pages about mathematics. Such informal pages may contain islands of formal text, which is used here for providing an initial cross-linking between Hales’s informal Flyspeck book, and the formal Flyspeck development.

The Agora platform intends to address distributed wiki-style collaboration on large formalization projects, in particular both the aspect of immediate editing, verification and rendering of formal code, and the aspect of gradual and mutual refactoring and correspondence of the initial informal text and its formalization. Here, we highlight these features within the Flyspeck Wiki.

1 Introduction

The formal development of large parts of mathematics is gradually becoming mainstream. In various proof assistants, large repositories of formal proof have been created, e.g. in Mizar\textsuperscript{4}, Coq\textsuperscript{5}, Isabelle\textsuperscript{6} and HOL Light\textsuperscript{7}. This has led to fully formalized proofs of some impressive results, for example the odd order theorem in Coq\textsuperscript{8}, the proof of the 4 color theorem in Coq\textsuperscript{9} and a significant portion of the proof of the Kepler conjecture\textsuperscript{10} in HOL Light.

Even though these results are impressive, it is still quite hard to get a considerable speed-up in the formalization process. If we look at Wikipedia, we observe that due to its distributed nature everyone can and wants to contribute, thus generating a gigantic increase of volume. If we look at the large formalization projects, we see that they are very hierarchically structured, even if they make use of systems like Coq, that very well support a cooperative distributed way of working, supported by a version control system. An important reason is that the \textit{precise} definitions \textit{do} matter in a computer formalised mathematical theory:
some definitions work better than others and the structure of the library impacts
the way you work with it.

There are other reasons why formalization is progressing at a much slower
rate than, e.g. Wikipedia. One important reason is that it is very hard to get
access to a library of formalised mathematics and to reuse it: specific features
and notational choices matter a lot and the library consists of such an enormous
amount of detailed formal code that it is hard to understand the purpose and use
of its ingredients. A formal repository consists of computer code (in the proof
assistant’s scripting language), and has the same challenges as a programming
source code regarding understanding, modularity and documentation. Also, if
you want to make a contribution to a library of formalized mathematics, it
really has to be all completely verified until the final proof step. And finally,
giving formal proofs in a proof assistant is very laborious, requiring a significant
amount of training and experience to do effectively.

To remedy this situation we have been developing the Agora platform: wiki
technology that supports the development of large coherent repositories of for-
malised mathematics. We illustrate our work by focusing on the case of a wiki
for the Flyspeck project, but the aims of Agora are wider. In short we want to
provide proof assistant users with the tools to

1. Document and display their developments for others to be read and studied,
2. Cooperate on formalizations,
3. Speed up the proving by giving them special proof support via AI/ATP
tools.

All this is integrated in one web-based framework, which aims at being a “Wiki
for Formal Mathematics”. In the present paper we highlight and advocate our
framework by showing the prototype Flyspeck Wiki. We first elaborate on the
three points mentioned above and indicate how we support these in Agora.

**Documenting formal proofs.** An important challenge is the communication of
large formalizations to the various different communities interested in such for-
malizations: PA users that want to cooperate or want to build further on the de-
velopment, interested readers who want to understand the precise choices made
in the formalization and mathematicians who want to convince themselves that
it is really the proper theorem that has been proven. All these communities have
their own views on a formalization and the process of creating formalization, giv-
ing a diverse input that benefits the field. Nonetheless, communicating a formal
proof is hard, just as hard as communicating a computer program.

Agora provides a wiki based approach: Formal proofs are basically program
code in a high-level programming language, which needs to be documented to be
understandable and maintainable. A proof development of mathematics is spe-
cial, because there typically is documentation in the form of a mathematical text
(a book or an article) that describes the mathematics informally. This is what
we call the informal mathematics as opposed to the formal mathematics which
is the mathematics as it lives inside a proof assistant. For software verification