Using Formal Reference to Enhance Authority and Integrity in Online Mathematical Texts

Authors: Stuart Allen, Robert Constable, and Lori Lorigo

The amount and variety of digital information readily available to the public has become one of the defining features of the intellectual and scientific landscape. Digital information is bringing to the forefront new questions for computing and information science, e.g., how should this information be organized, searched, and evaluated. Universities, publishers, government, and other esteemed professionals bring unique and essential value to this enterprise that goes beyond their support of research – namely, intellectual authority. The imprimatur given to the information resources they own or sponsor is essential in helping individuals assess the validity of what they encounter on the Web.

Recognizing quality online is important to creators and consumers of both open-access and peer-reviewed publications, digital libraries, newsgroups, and e-learning repositories. For example, one of the important ways in which the NSDL (National Science Digital Library) seeks to distinguish itself among digital libraries is by the authority of its collections. Also, the most respected journal publishers adhere to high standards and efforts to uphold the quality of their publications.

Over the past five years we have witnessed the addition of online tools that approximately rank importance or authority of components in the respective collections. For example, the CiteSeer digital library of scientific articles allows users to sort their search rankings based on expected citations, citations, hubs, usage data, and other statistics about both the author and the document. Citebase Search is a service for the e-print repository, arXiv.org, which also approximates measures of influence including hub and authority scores. Work by Brush, Wang, Turner, & Smith (2005) describes the use of a tool connected to the Usenet newsgroup repository that can reveal implicit reputation of authors based on various statistics about their activity on Usenet. And, Elsevier reports on the most downloaded articles. These tools help to inform a novice which resources might be important to pursue, what the "hot topics" at a given time may be, and essentially who the authorities are.

These tools are invaluable, yet they reveal only an implicit level of quality, and there is no guarantee of correctness within the articles themselves. It would be inspiring if in certain collections, all of the facts were correct, all of the citations proper, all of the quotes exact, the definitions right, and the arguments logically correct. It would be inspiring if these collections' services helped readers better understand evidence and the methods of science, e-science, and related modes of computer-mediated discovery.

In the domain of mathematics, very high standards for correctness and authority are being attained through computer mediation. Indeed, for some areas of mathematics, there are digital collections that have already achieved the highest standards of correctness known, computer checked formal proofs. Moreover, in the area of formalized mathematics we see some of the most striking examples of creating new knowledge in partnership with computers. With these accomplishments a new community focused on Mathematical Knowledge Management (MKM) has recently emerged, committed to furthering capabilities for representing and accessing computer-assisted mathematics on the web.

We are interested in harnessing the levels of quality and utmost correctness that computer checked mathematics have achieved to assist users in making informed quality judgments about electronically published articles. We have defined concepts and methodology for allowing authors of expository mathematical texts to exploit this large and growing body of formal digital mathematics. In this article, we explain a new methodology for authoring mathematics articles that ensures high quality by virtue of their creation, rather than post-filtering. We begin by introducing the most central new concepts that we have developed.

Formal Reference

We say that a particular piece of text within a document, such as a mathematical formula, *formally references* those objects and data from which that piece of the document was mechanically derived. Hence, there are computer verified facts relating documents to the contents of objects they formally reference. In contrast, manual transcription of objects into a document, as is traditionally done with article citations, would not produce a formal reference.

Perhaps the simplest example of formal reference would be quoting a referenced text. A formal quotation mechanism would allow the author to stipulate what part of the referenced text to quote, and the computer would copy the quotation, thus preventing error. Not only might this save the author some trouble, which in the case of "typesetting" mathematical formulas can be quite tedious and error prone, but the reader who knows that the quotation was computer-built does not have to establish its accuracy anew by comparing the quote and the source.

Figure 1 is an excerpt from a document about the Fundamental Theorem of Arithmetic, produced using our authoring tool, exemplifying formal reference. Each formula throughout the sample text was derived automatically from a repository object, and so formally references that object. None of these formulas anywhere in the sample text was typeset explicitly by the author. Instead, the code for formulas was generated automatically instead based on references to contents of a repository of computer checked mathematics. The figure is an image of pdf text which includes links to sources for the formulas.

Figure 1: Sample of Semantically Anchored Text

We have found it convenient in our development of the Fundamental Theorem of Arithmetic to use two formulations of prime factorization. Besides the notion of a general factorization in which only primes happen to have positive exponents, we can also formulate a prime factorization as an assignment of exponents to primes.

To bridge the gap between simple assignments $f \in \mathbb{N}_{\text{prime}} \rightarrow \mathbb{N}$ of exponents and multiplication over powers of consecutive integers, we introduce the notion of trivially converting an assignment to primes into an assignment to all numbers; just assign zero to the non-primes:

$$\text{prime\_mset\_complete}(f)(x) \equiv_{\text{def}} \text{ if is\_prime}(x) \rightarrow f(x) \text{ else } 0 \text{ fi} \quad \text{[Def]}$$

$$\forall f{:}(\mathbb{N}_{\text{prime}} \rightarrow \mathbb{N}).\ \text{prime\_mset\_complete}(f) \in \mathbb{N} \rightarrow \mathbb{N} \quad \text{[Thm]}$$

An assignment of natural numbers to members of some class X is sometimes considered a *multiset*, a collection where each member can "occur" any number of times. Thus, prime_mset_complete(f) converts a multiset of primes to a multiset of natural numbers generally. We can now state our formulation of when a multiset $f \in \mathbb{N}_{\text{prime}} \rightarrow \mathbb{N}$ of primes is a factorization of a number $k \in \mathbb{N}$:

$$\begin{aligned} & f \text{ is a factorization of } k \\ & \equiv_{\text{def}} (\forall x{:}\mathbb{N}_{\text{prime}}.\ k{<}x \Rightarrow f(x) = 0) \\ & \quad \& \ k = \Pi\{2..k{+}1^{-}\}^{\text{prime\_mset\_complete}(f)} \quad \text{[Def]} \end{aligned}$$

The condition $k = \Pi\{2..k{+}1^{-}\}^{\text{prime\_mset\_complete}(f)}$ simply means that one factorization of k is prime_mset_complete(f) $\in \{2..(k{+}1)^{-}\} \rightarrow \mathbb{N}$.[5] The adequacy of this formulation depends on the fact that all the prime factors of k must lie in $\{2..(k{+}1)^{-}\}$. The other condition, $\forall x{:}\mathbb{N}_{\text{prime}}.\ k{<}x \Rightarrow f(x) = 0$, constrains the set of possible factors (having positive exponents), putting an upper bound on the members of multiset f, and k is given explicitly as this bound to avoid unnecessarily introducing another variable.[6]

In the formulation "f is a factorization of k", f need only assign values to primes, but to all of them. This formulation is employed to state our FTA because it allows us to express uniqueness of prime factorization literally and simply – it's about exponents of primes:

$$\forall n{:}\{1\ldots\}.\ \exists! f{:}(\mathbb{N}_{\text{prime}} \rightarrow \mathbb{N}).\ f \text{ is a factorization of } n \quad \text{[FTA]}$$

Thus, our formulation of the Fundamental Theorem of Arithmetic is that, for every positive integer, there is exactly one assignment of exponents to primes that is its prime factorization.

[5] Note that $\forall g{:}(\mathbb{N} \rightarrow \mathbb{N}).\ g \in \{2..(k{+}1)^{-}\} \rightarrow \mathbb{N}$.

[6] An alternate formulation would have been $\exists U{:}\mathbb{N}.\ (\forall x{:}\mathbb{N}_{\text{prime}}.\ U{<}x \Rightarrow f(x) = 0)\ \&\ k = \Pi\{2..U{+}1^{-}\}^{\text{prime\_mset\_complete}(f)}$ but if any bound works for U here then k works.

How does formal reference contribute to quality of a document? Unless an authoritative text is entirely self-contained, it expresses reasoning based in part upon some other already established sources. Our special interest is in a world of authoritative texts that are not self-contained and, indeed, in which multiple authoritative documents refer to common source materials (and to each other). Reasoning expressed in a document based on sources it references normally involves adopting content from those sources, perhaps even transcribing or quoting. Consequently, the content of the referenced source matters in the referencing document. Formal reference is a computer verified bond between the contents of document and source.

So, assessing the quality of a document requires assessing the quality of the sources it references, assessing the quality of the reasoning in the document itself, and assessing the accuracy of transcription from the referenced sources into the document. For the purposes of this article, we take the authority of the computer checked formal mathematical source for granted and focus on practical exploitation of computer verification and generation of content transcription.

The remaining quality assessment problem, for the text of the document itself, is largely beyond our technical scope, and we acknowledge the importance of editorial assessment; but even this human process would receive some mechanical assistance for ascertaining the accuracy and coherency of all the transcriptions. For example, one could easily tell whether two different theorem statements represented in a document, or in multiple documents, apparently about greatest-common-divisor, say, were accurately reproduced and actually used the same definition for greatest-common-divisor (of which there are various related conceptions in current use).

Also beyond our present scope is how to formally reference, i.e. verify transcription from, other kinds of source material, such as primary data that served as a basis for scientific papers. But that is a natural path for generalizing this methodology for establishing authoritative texts.

Semantically Anchored Texts

If part of a document that formally references some data also provides a computationally usable path to that data, we say it is *semantically anchored*. Anchoring provides various benefits beyond unanchored formal reference. First, the anchors may lead to a world of data which is highly elaborated independently of the documents that reference it. Second, those formally referenced structures can be analyzed by computer and provide the basis for a computer checkable criterion for common reference (e.g., two documents sharing the same formal definition of greatest-common-divisor). Third, new documents can be created that intentionally refer to the same objects and concepts used in already existing documents by following the anchors to the formal referents (Allen & Constable, 2005). Fourth, the computer verified facts relating documents to the contents of objects they formally reference can be scrutinized and independently reverified.

Mathematical and program text is unique because significant parts of it have a well understood formal semantics, expressible in logical theories that are implemented by computer systems. When elements of expository text, such as definitions and theorems, are formally linked to their implemented counterparts, we call the texts semantically anchored. We have built tools that allow nonspecialists to easily semantically anchor texts and lessons.

Those places in a document that formally reference other objects or texts are semantically anchored to those referents when there is a computationally effective path from those places in the document to the sources from which they were derived. Four of the formulas automatically formatted in Figure 1, are semantically anchored with links variously labeled "Def", "Thm" or "FTA". If we had submitted the actual document instead of an image snapshot of it for the purposes of our present publication, and you were reading this article online, you could follow the links to Web presentations of the formal mathematics these anchored formulas were derived from, where you would find more detail and various derived information. In this example, the four links, Def, Thm, Def and FTA would lead you to the appropriate math.

Making precise this notion of semantic anchoring was what led us to the more basic concept of formal reference above.

Replete Documents

We have adopted a conception of a *replete document* that comprises much more than the ordinary text, or even hypertext sources, someone reads. A document in this extended sense is a structured entity that is computationally bound to the material it formally references. It includes authored "sources" for the document with embedded "commands" for generating formal references and anchors, but also includes auxiliary tables built implicitly by the authoring tools during the authoring process that are used to complete automatic generation of referencing text, and to accomplish Web-publication (principally for calculating the right Web addresses to link to). Multiple editions of readable (hyper)texts are automatically generated from these same sources, sometimes making obsolete earlier editions and sometimes coexisting as alternative presentations. Replete documents also include various data derived from the sources.

Under our conception of a replete document, the texts a person normally reads (via pdf in our prototype) constitute just one aspect of the document as a whole. Let us call this the surface of the document. Surface pages are derived from data relating the author's text intimately to other contents in the repository which it formally references. Any replete documents which are themselves included in the repository will inherit the repository's generic services, including inclusion in Web-published editions, and get the benefits of the repository's policies for preserving content and integrity.

The replete document comprises a structured collection of source texts together with various tables used to computationally determine how to identify and present the contents of formally referenced texts, which in our prototype means producing Latex code, typically for mathematical expressions contained in referenced texts.

Creating Semantically Anchored Documents

The production of anchored documents depends on the medium intended for them, the editing tools and source-text formats the author prefers to work with, and the kind of access the author has to the underlying repositories of formal mathematics. The important processes are (1) the repository of mathematics, (2) its Web-publication service, especially the generation of formally referencing subtexts, and (3) tool-assisted editing by authors while interacting with the repository, the Web, and a conventional text editor.

Repositories of Formal Content

Mathematical content formalized for computer processes such as verification, represents content, or internal abstract structure, completely and unambiguously. These repositories represent vast, richly meaningful complexity accessible to mechanical elaboration. One can find dependencies between meaningful formal texts such as proofs, definitions and programs.

How one makes this content conveniently readable for humans is a distinct, though important, matter. This distinction is nicely made for mathml which distinguishes content markup from presentation markup of formulas. It is possible to present the same content in different ways, with different notations. Notation might change to reflect improvements developed over time, or the same content might be presented to distinct readerships having different notational needs or traditions.

While it might be possible to carefully maintain such mathematical content repositories as subcollections of Web pages, that is not a reasonable requirement at this time. How to organize formal mathematical repositories and how to represent inter-object reference is a repository design issue rather than a document publication issue; in particular, the repository we used exploits an "abstract" name space, so we cannot assume a global name space for objects (Allen, 2004). Our methodology is to use the Web as an access method and publication medium for repository content. Our prototype configuration uses the Cornell Formal Digital Library (FDL) (Allen et. al., 2002) as a repository. We used this repository because it has plenty of content, we believe it's a good example of a repository design for formal mathematics, and we have experience using it. We expect the increasing widespread use and availability of other such repositories for formally developed content. HELM, C-CoRN, and Mizar are others, for example.

Web Publication Services

Our main purpose for this effort has been to find ways to make it practical for authors to create documents integrated formally with texts they reference and to make their creations widely accessible to readers. We have assumed the Web will be the principal medium for readers to access authored documents and the formal materials they refer to. While we certainly want readers to directly access the repository containing a document should they desire it, we will not require such access to make the document readable. Note that part of projecting repository objects onto the Web is to provide a path back to those objects in the repository.

Our prototype methods include a Web-publication[1] service that creates Web pages representing repository objects, including authored documents of the repository. Our prototype uses Latex and pdf as the basic data formats because Latex is the current standard for documents with special mathematical notations and from Latex one can

[1] We use the term "publication" here with some reservation; we use it to mean making texts available to the public, without necessarily alluding to the other obligations that may be associated with "publication."

produce pdf pages that link to other Web pages. The authored documents can also be put in standard data formats suitable for publication in paper journals.

Other kinds of texts in the FDL repository such as mathematical definitions and theorems or programs, which expository documents can reference, are independently projected onto the Web using methods considered appropriate for them by the publication service providers.

A further element of the Web-publication service for expository documents is the publication of auxiliary data derived from the authored document such as a digest of what materials it formally references. A structured representation of the mathematical expressions used in the document would facilitate search for the document based upon structure and content of the mathematical expressions as opposed to their appearance as character configurations.

When a new document is projected onto the Web, one must decide which Web addresses to link to. The Web Publication Basis is a “consistent” collection of Web projection identifiers described above. We assume that initially the author will want to link to the same pages he or she is reading during document preparation. When the document is eventually regenerated as part of a new edition, a new document will be created by the publication service that has a newer Web-publication basis.

Editing Example

In our methodology, authors do not cut and paste math into documents. Instead, they indicate sources for math text to be inserted by a computer. Our prototype configuration supports Latex as the source language for the author to edit. Our tools for creating documents are most fully integrated for users of the Emacs editor, although the various components could be used with some author overhead with any string-text editor. The core element for generating text for formal references in a document is a process for converting abstract mathematical structures in the repository to Latex code; we call it the Dynamic Math Formatter (DMF).

Commands embedded in author sources can be used to refer to repository objects, and to parts of the content of such objects, and can describe how to arrange those contents in the document when formatting them. For example, one could specify a whole definition, the left hand side of a definition, a whole theorem statement, or a component of a complex statement.

Here we present examples of some of those commands and the formulas automatically aid out by them. These are from the document of which figure 1, on page 3, is an excerpt.

The command \DMFtyping{eval_factorization_wf}{f} stipulates that the typing assumed for variable f in the named theorem $(\forall a,b{:}\mathbb{Z},\ f{:}(\{a..b^{-}\}\rightarrow \mathbb{Z}).\ \Pi\{a..b^{-}\}^{f} \in \mathbb{Z})$ should be formatted thus: $f \in \{a..b^{-}\}\rightarrow \mathbb{Z}$.

Use of \DMFob[Def]{prime_factorization_of} in the author source indicates that the content of the object named prime factorization of should be laid out as

$$f \text{ is a factorization of } k$$
$$\equiv_{\text{def}} \ (\forall x{:}\mathbb{N}_{\text{prime}}.\ k{<}x \Rightarrow f(x) = 0)\ \&\ k = \Pi\{2..k{+}1^{-}\}^{\text{prime\_mset\_complete}(f)}$$

and juxtaposed to a link (in pdf) labeled “Def ” to the object itself.
Similarly, \DMFob[eval_factorization]{eval_factorization} is replaced by formatting code for the contents of the named object (along with a link so labeled) that lays out as:

$$\Pi\{a..b^{-}\}^{f} \ \equiv_{\text{def}} \ \Pi\ i{:}\{a..b^{-}\}.\ i^{f(i)}$$

The author can insert the left hand side $(\Pi\{a..b^{-}\}^{f})$ of this definition into the document with the command \DMFlhs{eval_factorization}.

It is even possible for the author to assemble new expressions from elements of other DMF commands to describe a new expression. For example,

\DMFitermul{i}{\DMFvar{e}{\DMFvar{i}}}{\DMFvar{a}}{\DMFvar{b}}

is formatted to produce

$$\Pi\ i{:}\{a..b^{-}\}.\ e(i)$$

even though this expression does not already occur exactly in the repository (it will once this document is added). This expression is deemed to refer to the definitions of each operator occurring in the expression, and is a faithful representation of it.

Normally, the DMF command will be considerably simpler than the Latex code inserted in the intermediate file by preprocessing, and it prevents transcription errors, and so provides a useful editing service to the author. But the more profound aspect of this functionality is that it is possible for readers to ascertain that such transcriptions from referenced texts were performed automatically, and neither error nor deception by the author has occurred at these points in the document. Variations of the document can be automatically generated by the publication service making very plain where these certified transcriptions have occurred.

The DMF renders pure structured data of the FDL as printable text; it is principally applied to mathematical expressions, but is actually more general than that, and is also used for formatting program text and sometimes for ordinary English text. The DMF is designed to be used in an environment where abstractly structured objects are stored independently of how they are to be rendered for the eye. One then supplies a collection of display forms which specify how various forms of structured data are to be displayed. The DMF then applies these display forms to the abstractly structured data to produce characters or symbols or instructions in another rendering format such as Latex or html. This is analogous to converting a pure content mathml expression to presentation mathml.

Given our design choices for the repository, Web, and DMF, here is the scenario we implemented. The author writes a Latex document using an Emacs text editor that has been enhanced to access a repository, the FDL. The author edits in tandem with a Web browser which is used to view the repository contents as well as the output document, in pdf, as it is developed. In the Latex source text being prepared, at those places where the author wants the pdf text to formally reference a mathematical object, the author can insert a link to the object and, if desired, further indications as to how the referencing text is to be generated from the referenced data. As often as desired, the author may have a fresh copy of the pdf output text generated for viewing. The pdf output text will contain the automatically generated text for formally referenced objects and links to the Web addresses representing the formally referenced objects.

Related Work

As stated much work is being done to infer authority from online documents after they are produced, whether based on citation, usage, or other metrics. Our formal reference mechanism is new in that it allows for a strict notion of correctness at the time of creation of mathematics articles. The importance of tools that facilitate expression of and access to mathematics is witnessed by several significant accomplishments in the mathematical editing and publishing domains. Mathml has gained popularity as a medium for disseminating mathematical content and presentation on the Web. Authoring mathml documents typically involves three steps: (1) authoring of the article without math content using a familiar tool, (2) writing mathml objects using separate Mathml equation editors, and (3) integrating or embedding the mathml objects into specific locations of the original article. Mathml editing capabilities exist in Design Science's MathType editor and Integre's mathml equation editor. We have experimented with Integre's software, and hope it can provide an additional outlet for our work in the future. Because we were targeting a very widespread, browser independent solution that has at the same time an allowance for semantic anchoring, we did not find current off-the-shelf WYSIWYG tools best suited. Hindrances included limited access to the mathematical source objects with generic tools after the embedding.

In the domain of formal mathematics, OMDoc, for Open Mathematical Documents, (Kohlhase, 2000) specifies a format specifically for formal mathematics, or mathematical proof. OMDoc extends xml to provide structured content-markup for formal mathematical objects, including allowance for informal text. While OMDoc may provide an additional representation format for users of our tools in the future, we wished to work in language already familiar to many mathematicians and scientific researchers. It should be noted that use of the Latex format does not require that authors use standard text editors; TeXmacs, for example, is a WYSIWYG editor based upon TeX.

Conclusion

Several methods for inferring quality of online information exist today. We have presented one new element of ensuring correctness in mathematical texts at the time of document preparation. The semantically anchored texts that we produce contain formally-grounded explanations; by this we mean an explanation that ultimately can be reduced to a readable machine checked proof in a formal logic. They serve as exceptionally thorough models of fundamental elements of education, namely, explanation and evidence. Whether we are talking about an English essay, a chemistry experiment, a legal argument or a mathematical demonstration, students are taught to answer the question “How do you know?” They are taught to give evidence and to say what statements follow from others. This ability to provide evidence and evaluate arguments is critical to a liberal arts education or an engineering one.

Authors may also want to deposit documents into the FDL repository itself in order for such documents to be automatically subject to ongoing repository services such as cross referencing and production of new “editions” exploiting improvements in notation and automatic metadata generation. To do so, authors will need write-permission to the FDL, but the functionality needed for writing documents to the FDL is fairly narrow since it doesn’t have to support the creation of formal mathematics generally such as proof development, so simple protocols for authors should suffice.

Furthermore, as we described, our prototype tools support Latex in order to serve authors who want versions of their documents suitable for standard mathematical journal publication. But we believe there are more prospective authors who do not use Latex and are more interested in writing Web documents for other audiences, such as teachers preparing documents for a class or speakers preparing pages for presentation at seminars. Making document source formats and tools that support html and produce mathml instead of Latex/pdf would be valuable in these cases. Furthermore, the methodology we have developed for the FDL would be immediately useful once adapted to any of the large collections of formalized mathematics developed by internationally established groups of specialists (Kaufman & Moore, 1997; Cornes et. al., 1995; Paulson, 1994; Gordan & Melham, 1993; Harrison, 1996; Allen et. al. 2005; Rudnicki, 1992; Hickey et. al., 2003; Benzmuller et. al., 1997, Sorge, 2001; Shankar, et. al. 1999; Schurmann, et. al., n.d.; Wolfram, 1988; Geddes, Czapor, & Labahn, 1992), increasingly being made available through the Web.

Acknowledgements

The authors thank Richard Eaton, lead programmer of the FDL for his valuable input. This work was supported by the National Science Foundation, NSDL Grant # 0333526 entitled “Enabling Large-Scale Coherency among Mathematical Texts in the NSDL”.

References

Allen, S., & Constable, R. (2005). Enabling large scale coherency among mathematical texts. Submitted to *Journal of Digital Information*, sponsored by British Computer Society and Oxford University Press.

Allen, S. (2004). Abstract identifiers, intertextual reference and a computational basis for record keeping. *First Monday*, 9(2), 2004. Available online at http://www.firstmonday.org/issues/issue9/allen/index.html.

Allen, S., Bickford, M., Constable, R., Eaton, R., Kreitz, C., & Lorigo, L. (2002). FDL: A prototype formal digital library. PostScript document on website, May 2002. Available online at http://www.nuprl.org/html/FDLProject/02cucs-fdl.html.

Benzmuller, C., Cheikhrouhou, L., Fehrer, D., Fiedler, A., Huang, X., Kerber, M., Kohlhase, M., Konrad, K., Melis, E., Meier, A., Schaarschmidt, W., Siekmann, J., & Sorge, V. (1997). Ωmega: Towards a mathematical assistant. In William McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction*, volume 1249 of Lecture Notes in Artificial Intelligence. Springer, July 1997.

Bertot, J., Bertot, Y., Coscoy, Y., Goguen, H., & Montagnac, F. (1997). User Guide to the CTCOQ Proof Environment. INRIA, Sophia Antipolis, February 1997. System Revision 1.22; Documentation Revision 1.31.

Brush, A., Wang, X., Turner, T., & Smith, M. (2005). "Assessing differential usage of usenet social accounting meta-data", In *Proceedings of CHI 2005*, ACM Press (2005), 889-898.

Cornes, C., el Courant, J., Filliatre, J., Huet, G., Manoury, P., Paulin-Mohring, C., Munoz, C., Murthy, C., Parent, C., Saıbi, A., & Werner, B. (1995). The Coq Proof Assistant reference manual. Technical report, INRIA, 1995.

Geddes, K. O., Czapor, S. R., & Labahn, G. (1992) Algorithms for Computer Algebra. Kluwer, Boston, 1992.

Harrison, J. (1996). HOLLight: A tutorial introduction. In *Formal Methods in Computer Aided Design (FMCAD'96),* volume 1166 of Lecture Notes in Computer Science, pages 265–269. Springer, 1996.

Hickey, J., Nogin, A., Constable, R. L., Aydemir, B. E., Barzilay, E., Bryukhov, Y., Eaton, R., Granicz, A., Kopylov, A., Kreitz, C., Krupski, V. N., Lorigo, L., Schmitt, S., Witty, C., & Yu, X. (2003). MetaPRL — A modular logical environment. In David Basin and Burkhart Wolff, editors, *Proceedings of the 16th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2003),* volume 2758 of Lecture Notes in Computer Science, pages 287–303. Springer-Verlag, 2003.

Kaufmann, M., & Moore, J. (1997). An industrial strength theorem prover for a logic based on Common Lisp. *IEEE Transactions on Software Engineering,* 23(4):203–213, April 1997.

Kohlhase, M. (2000). OMDoc: Towards an Internet Standard for the Administration, Distribution and Teaching of mathematical Knowledge. In *Proceedings of Artificial Intelligence and Symbolic Computation*, Springer, Lecture Notes in Artificial Intelligence, 2000.

Kohlhase, M. (2000). OMDoc: An Infrastructure for OpenMath Content Dictionary Information. *Bulletin of the ACM Special Interest Group for Algorithmic Mathematics,* SIGSAM, 2000.

Owre, S., Rushby, J. M., & Shankar, N. (1992). PVS: A prototype verification system. In Deepak Kapur, editor, *Proceedings of the 11th International Conference on Automated Deduction*, volume 607 of Lecture Notes in Artificial Intelligence, pages 748–752, Springer-Verlag, 1992.

Rudnicki, P. (1992). An Overview of the Mizar Project, *Proceedings of the 1992 Workshop on Types for Proofs and Programs,* Chalmers University of Technology, Bastad, 1992.

Schurmann, Pfenning, & Shankar. Logosphere. A Formal Digital Library. Logosphere homepage: http://www.logosphere.org.

Shankar, N., Owre, S., Rushby, J. M., & Stringer-Calvert, D. W. J. (1999). PVS Prover Guide. Computer Science Laboratory, SRI International, Menlo Park, CA, September 1999.

Sorge, V. (2001). Ω-ANTS: A Blackboard Architecture for the Integration of Reasoning Techniques into Proof Planning. PhD thesis, Saarland University, Saarbruken, Germany, November 2001.

Trybulec, A. (1984). On a system of computer-aided instruction of logic. Bulletin of the Section of Logic PAS, 12, Nr.4, 1984.

Wolfram, S. (1988). Mathematica: A System for Doing Mathematics by Computer. Addison Wesley, 1988.