0.1 Induction Challenge OMDoc Manager (ICOM)

Project Home	`http://www.cs.nott.ac.uk/~lad/research/challenges/challenge_manager.html`
Authors	Thomas D. Attfield, Monica C. Duarte, Lin Li, Ho-Ying Mak, Adam M. Neal, Lewis M. Toft, Zixuan Wang, Louise A. Dennis School of Computer Science and Information Technology, University of Nottingham

We describe work in progress to create a system for organising and presenting a set of challenge problems collected by the Induction Theorem Proving community. These challenge problems come from a number of sources and are presented in different logics using different presentation conventions.

The intention is to provide a system which will allow these problems to be stored in a unified format and will support the collection, browsing and extraction of the problems.

OMDoc is an obvious choice for representing such problems and the system is able to take advantage of much existing work on the manipulation of XML documents.

0.1.1 The Induction Challenge Problems

Inductive Theorem proving is a small field. The main theorem provers within this field are NQTHM [BM79] (now re-engineered as ACL2 [KM96]), INKA [AHMS99], the series [BvHHS90, RSG98] and RRL [KZ95]. TWELF [PS99] also looks at the automation of inductive proof in the context of logical frameworks. Within the field it is hard to assess claims for the superiority of any given system since there is naturally a tendency to report "successes" – difficult or challenging problems automatically proved. There is also a desire within the community to develop a store of shared knowledge about the challenges that face the automation of proof by mathematical induction.

TPTP (Thousands of Problems for Theorem Proverss) [SS98] is a library of test problems for first-order ATP systems. They provide the ATP community with a comprehensive library complete with unambiguous names and references. All the problems are stated in a standardised formulation of first-order logic and are widely used to benchmark first-order systems. They are also used as the test set for the CASC competition [Sut01] which compares such systems. One of the benefits of the TPTP library to the ATP community is the existence of a common set of problems by which comparisons can be made.

It is not practical for inductive theorem provers to follow the pattern of the TPTP library. Various attempts have been made to build a similar corpus of problems requiring inductive reasoning. The most mature of these was based on the Boyer-Moore [BM79] corpus[1]. This corpus was unpopular partly because there was repetition within the problem set and partly because many problems depended on a few particular function definitions. But the major objection was that induction theorem provers use a number of different logics, some of which are typed and some of which are not, which made it difficult to agree on a standard format. The use of other logics also raised translation issues and a fully automated process for converting the theorems, even into an agreed typed language was never produced.

A group of researchers within the community[2] agreed that instead of a large set of benchmarks in a standard logic they would each put forward a number of "Challenge Problems". These should present interesting challenges to the automation of inductive proof or illustrate important features which an inductive prover should be able to handle. A set of these problems would be collected which would remain sufficiently small that an individual could represent them within their own theorem proving system as they saw fit[3]. These challenge problems are currently described in a high-level way and written up in an ad hoc fashion. The descriptions contain both mathematical notation and commentary. They are difficult to read, navigate or use in any particular system.

OMDoc seems ideally suited as a format for representing these challenge problems: it can represent both text and formulae; it is not tied to any particular logic and it supports the extraction of data into a number of different formats. As an added benefit its hyper-text features would potentially allow definitions to be stored separately and shared between problems. Individual theorem provers can then concentrate on translations between OpenMath content dictionaries and their own logics and individuals submitting problems can specify the appropriate content dictionary for the problem.

0.1.2 System Description

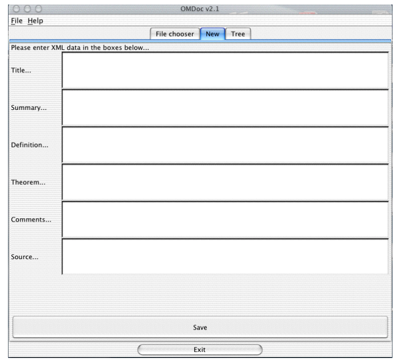
The Induction Challenge OMDoc Manager (ICOM) is designed to be a system which will ease the submission and extraction process for the problems. Our intention is to provide a submission interface that will create a simple OMDoc markup for the problems which can subsequently be edited by a user and to provide browsing and extraction capabilities.

Each challenge problem description contains six distinct sections (e.g. Summary, Definitions, Comments). Currently a user who wishes to enter a problem into our system is presented with the form shown on the right with a field for each section.

[1] This has become known as the Dmac corpus after David McAllester who translated a fragment of the NQTHM corpus into a simpler language.

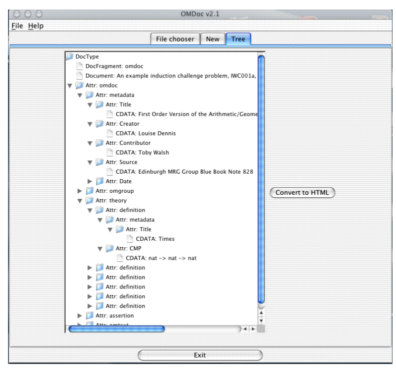
[2] At the 2000 CADE Workshop on the Automation of Proof by Mathematical Induction.

[3] The current set can be found at `http://www.cs.nott.ac.uk/~lad/research/challenges`.



Each section, once entered by a user, is placed in a `CMP` tag. These tagged fragments are wrapped in standard OMDoc headers and footers to produce a valid OMDoc. This completed document is then written to disk and stored. We are currently working on a simple parser to translate equations into `OMOBJ` structures which a user will then be able to edit (for instance to specify the appropriate content dictionaries). We hope this will be easier than adding all the OpenMath tags by hand.

An existing document can be displayed as a tree and from this tree the document can be directly manipulated. This tree display also allows the user to see the structure of the document more clearly. It is also possible to extract an HTML view of the contents of the document so it can be displayed in a web browser and read by a human.



Our implementation language is JAVA and we use its JAXP DOM API. DOM [DOM] is a W3C standard which uses a tree-based model (storing data in hierarchies of nodes). This means that once an OMDoc has been created or opened all the document's data is in memory and so data can be accessed rapidly. DOM also enables simple modification of documents by adding or deleting nodes. Although SAX (an alternative model) achieves better performance and less memory overhead than DOM, it is easier to traverse and modify XML documents using a DOM tree structure. Since we anticipate that users may wish to modify the initial OMDoc produced by our system we adopted the DOM model instead.

0.1.3 Further Work

ICOM is still in the early stages of development. Currently our most pressing aim is to provide improved support for entering equations. Once this is in place we hope to add searching facilities and provide better mechanisms for links to be created between challenge problems. We would also like to experiment with the automatic extraction of problems into a theorem prover via an MBase [KF00] and a MathWeb [FHJ$^+$99].

References

AHMS99. S. Autexier, D. Hutter, H. Mantel, and A. Schairer. System description: INKA 5.0 - a logical voyager. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1732 of *Lecture Notes in Artificial Intelligence*, Trento, 1999. Springer.

BM79. R. S. Boyer and J S. Moore. *A Computational Logic.* ACM monograph series. Academic Press, New York, 1979.

BvHHS90. A. Bundy, F. van Harmelen, C. Horn, and A. Smaill. The Oyster-Clam system. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, pages 647–648. Springer-Verlag, 1990. Lecture Notes in Artificial Intelligence No. 449. Also available from Edinburgh as DAI Research Paper 507.

DOM. Document object model DOM. web page at `http://www.w3.org/DOM/`.

FHJ+99. A. Franke, S. Hess, C. Jung, M. Kohlhase, and V. Sorge. Agent-Oriented Integration of Distributed Mathematical Services. *Journal of Universal Computer Science*, 5(3):156–187, March 1999. Special issue on Integration of Deduction System.

KF00. M. Kohlhase and A. Franke. MBase: Representing knowledge and context for the integration of mathematical software systems. *Journal of Symbolic Computation*, 2000.

KM96. M. Kaufmann and J S. Moore. ACL2: An industrial strength version of Nqthm. In *Compass'96: Eleventh Annual Conference on Computer Assurance*, page 23, Gaithersburg, Maryland, 1996. National Institute of Standards and Technology.

KZ95. D. Kapur and H. Zhang. An overview of rewrite rule laboratory (RRL). *J. Computer and Mathematics with Applications*, 29(2):91–114, 1995.

PS99. F. Pfenning and C. Schürmann. System description: Twelf — A metalogical framework for deductive systems. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, pages 202–206, Trento, Italy, 1999. Springer-Verlag LNAI 1632.

RSG98. J.D.C. Richardson, A. Smaill, and I. Green. System description: Proof planning in higher-order logic with lambda-clam. In C. Kirchner and H. Kirchner, editors, *Conference on Automated Deduction (CADE'98)*, volume 1421 of *Lecture Notes in Computer Science*, pages 129–133. Springer-Verlag, 1998.

SS98. G. Sutcliffe and C. Suttner. The TPTP problem library: CNF release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.

Sut01. G. Sutcliffe. The CADE-17 ATP system competition. *Journal of Automated Reasoning*, 27(3):227–250, 2001.