9-1-2003

System Description: Analytica 2

Edmund M. Clarke
Carnegie Mellon University, emc@cs.cmu.edu

Michael Kohlhase
Carnegie Mellon University

Joel Ouaknine
Carnegie Mellon University

Klaus Sutner
Carnegie Mellon University, klaus.sutner@cmu.edu

System Description: Analytica 2

Edmund Clarke, Michael Kohlhase, Joël Ouaknine, Klaus Sutner

Carnegie Mellon University
{emc|kohlhase|ouaknine|sutner}@cs.cmu.edu

Abstract. The ANALYTICA system is a theorem proving system for 19^{th} century mathematics written on top of the *Mathematica* computer algebra system. It was developed in the early 1990's by X. Zhao and E. Clarke and has since been dormant. We describe recent work to resurrect the theorem prover and port it to newer versions of *Mathematica*. The new system ANALYTICA 2 can still prove the same theorems, but has been significantly cleaned up. The code has been restructured and documented, the declarative knowledge has been separated from a logical kernel, and the system is being made available as a MATHWEB service.

1 Introduction

The ANALYTICA system [CZ92,BCZ98] is a theorem proving system for 19^{th} century mathematics.

It has been able to prove theorems from elementary calculus and number theory, including a proof of the Bernstein approximation theorem and the theorems and examples in the second chapter in Ramanujan's Collected Work [Ber85,CZ92]. The system was developed in the early 1990's by Xudong Zhao and Edmund Clarke and has since been dormant.

ANALYTICA is written on top of the *Mathematica* computer algebra system [Wol02], a large commercial computer algebra system that offers a highly developed document-centered front-end that facilitates communication with the kernel and that allows for the development of multi-modal electronic documents, so-called notebooks, that can contain code, text, graphics, and data. Notebooks can render mathematical formulae in near-typeset quality. Moreover, Notebooks are symbolic structures that can be manipulated by the *Mathematica* kernel like any other symbolic expression in the system. They can also be exported in LaTeX and MATHML format. We suggest that this computational environment naturally supports the design and implementation of fairly complicated software systems using symbolic computation. A description of a similar effort in the area of computational automata theory can be found in [Sut02].

2 Porting the Code Base to *Mathematica* Version 5

ANALYTICA was originally written for *Mathematica* version 1.2, which lacked many of the features of current versions of the product. In particular, no graphical front-end was available and all communication to the kernel was handled by a text-based interface similar to a command shell. In our work on the ANALYTICA prover we make substantial use of four new capabilities of *Mathematica*: the

notebook front-end We have used the front-end in the documentation of the code base, and as user interface: As formula output in the *Mathematica* frontend approaches that of TEX and notebooks supports a powerful folding operation, ANALYTICA's original LATEX output routines for proofs are now obsolete and were deleted from ANALYTICA.

external system interface JLINK is used for interfacing to knowledge exchange formats like OMDOC (see Section 3).

native XML **processing capabilities** are used heavily in communication with XML based services.

added symbolic computation capabilities in the *Mathematica* kernel: For instance, support for symbolic summation and trigonometric simplification has dramatically improved in *Mathematica* since version 1.2. Nonetheless, we have retained existing ANALYTICA modules for these areas as plug-ins, loadable on demand. These implementations are transparent to the theorem prover and can thus be used to document proofs and computations that would be opaque if carried out by *Mathematica*'s built-in version.

The first step was to convert the formerly 50 plus source files into two large notebooks, one each for the prover and knowledge base parts (see Section 3). Code for the prover is represented using a special SOURCE style sheet that tightly integrates the actual *Mathematica* code with accompanying documentation, examples and test code. From the SOURCE style notebook one can automatically generate files that augment the *Mathematica* help browser and provide online help for the ANALYTICA system. From the same source document one can also extract pure code files that can be bundled together with the online documentation into an add-on package (see [CKOS03] for details). Installation of this package is very straightforward, and requires no more than to copy the package files to the appropriate place in the *Mathematica* file structure.

3 Separating Mathematical Knowledge from Code

There are two kinds of code in ANALYTICA: the program code and mathematical knowledge used in proof search. To separate causes and make ANALYTICA easier to port to other mathematical domains, these are separated in ANALYTICA2. Originally, the mathematical knowledge used in ANALYTICA was represented as the following *Mathematica* code.

```
(* Rules for simplifying expressions involving the  absolute value function. *)
UnProtect[Abs];
Abs[a_ b_] := Abs[a] Abs[b];
Abs[a_^n_] := Abs[a]^n;
Protect[Abs];

(*  Local rule  used in simplification. *)
AbsRule = {Abs[a_] :> If[TrueQ[WeakSimplify[a >= 0]], a,
                If[TrueQ[WeakSimplify[a <= 0]], -a, Abs[Factor1[a]]]]};
```

The first block specifies some rewriting rules for the *Mathematica* symbol `Abs` that are subsequently used by *Mathematica*'s built-in simplifier. The second code block specifies a rewrite rule used in a special simplification engine in

ANALYTICA. The correctness of the ANALYTICA system depends on a couple of hundreds of such rules.

These rules are now collected in a notebook using as special Knowledge Representation style that captures the information implicit in the original code fragments. We have used a variant of the `nb2omdoc` transformer [Sut03] to transform KNOWLEDGE style notebooks into the OMDOC format (Open Mathematical DOCuments [Koh03]), an XML-based format for representing mathematical knowledge in the large. OMDOC can be used as a basis for communicating with other mathematical software systems and in particular, the MBASE mathematical knowledge base [KF01], which acts as an external knowledge repository for ANALYTICA 2 (see section 4).

In the transformation we have made explicit and thus documented the mathematical knowledge used in ANALYTICA. In the case of our example above, this is given by the 4 theorems:

#	formalization	the absolute value function ...
1	$\forall a, b.\lvert a \cdot b\rvert = \lvert a\rvert \cdot \lvert b\rvert$	commutes with multiplication
2	$\forall a, n.\lvert a^n\rvert = \lvert a\rvert^n$	commutes with exponentiation
3	$\forall a.a \geq 0 \Rightarrow \lvert a\rvert = a$	is the identity on $\mathbb{R}^+$
4	$\forall a.a \leq 0 \Rightarrow \lvert a\rvert = -a$	is the negative identity on $\mathbb{R}^-$

In the generated OMDOC representation, these theorems are represented in a special `assertion` element that combines the formalization in OPENMATH [CCAMC02] representation with the natural vernacular. Note that the *Mathematica* code fragments contain other information than the logical theorems, mostly of heuristic or computational nature, like the direction of the equation in the simplification rules. Therefore, the OMDOC representation also embeds the original *Mathematica* code. As *Mathematica* has a native XML (and thus OMDOC) parser, ANALYTICA can directly read OMDOC documents.

The main problem in the transformation to OMDOC is that the ANALYTICA logic is based on and uses many of the unique term representation features of the *Mathematica* language, which are geared for programming with mathematical objects, but whose logical foundations are insufficiently explored ([Mar03,Kut03] are recent exceptions).

For instance, functions in *Mathematica* are polyadic (they can have variable arities). To make this palatable to the user and programmer, *Mathematica* employs sequence variables in pattern matching. Consider for instance the following fragment from the definition for continuous functions.

```
Continuous[f_[a__], x_, x0_] :=
    Apply[ and, Map[ Function[z, Continuous[z, x, x0]], List[a]]] /; ContFunction[f];
```

The function `Continuous` takes three arguments, an expression e, a (bound) variable x, and a point x_0; it is true, if e is continuous at x_0 when viewed as a function in x. The interesting part is that the expression e is of the form

$f(a_1, \ldots, a_n)$, where the variable `a` is a sequence variable that stands for the sequence $a_1, \ldots, a_n$[1].

Our OMDoc transformation currently treats sequence variables like arbitrary variables, and represents this as

$$\forall f, a, x, x_0 . \mathbb{C}(f(a), x, x_0) \Leftrightarrow \mathbb{C}^0(f) \wedge apply(\wedge, map(\lambda z \mathbb{C}(z, x, x_0))), list(a)$$

where we use $\mathbb{C}$ for `Continuous` and $\mathbb{C}^0$ for `ContFunction`. Of course, this is not a standard logical representation, and to communicate with other mathematical software systems we will need to translate this into more standard representations. One approach we are experimenting with at the moment is to encode sequence variables into higher-order logic with Currying, e.g. for communication with the TPS theorem prover for higher-order logic [ABI+96]. For the example above, an equivalent representation in such a logic would be the two formulae $\forall F, A, x, x_0 . \mathbb{C}((FA), x, x_0) \Leftrightarrow \mathbb{C}(F, x, x_0) \wedge \mathbb{C}(A, x, x_0)$ and $\forall F, x, x_0 . \mathbb{C}(F, x, x_0) \Leftarrow \mathbb{C}^0(F)$. A proof of $\mathbb{C}(x^2 + x, x, 1)$ would involve a derivation of $\mathbb{C}(+, x, 1)$ from $\mathbb{C}^0(+)$ and of $\mathbb{C}(x^2, x, x_0)$ from $\mathbb{C}(x, x, x_0)$. From that we get $\mathbb{C}(+x^2, x, x_0)$, and finally $\mathbb{C}(x^2 + x, x, 1)$; this computation is in fact what the ANALYTICA simplifier executes internally, justified by the piece of knowledge above.

4 A MATHWEB Interface for ANALYTICA

For the communication with the MBASE system, we have equipped ANALYTICA with an XMLRPC interface, see [Com]. This allows ANALYTICA to store the OMDoc-encoded knowledge externally and request the fragments needed for the proofs of the respective theorems. The XMLRPC interface is built on *Mathematica*'s JLINK facility. and makes the XML representations of the protocol documents available to *Mathematica*, whose native XML facilities are used to convert them into ANALYTICA's internal representations.

MBASE is part of the MATHWEB [ZK02] service infrastructure, which connects a wide-range of reasoning systems (*mathematical services*), such as automated theorem provers, (semi-)automated proof assistants, computer algebra systems, model generators, constraint solvers, human interaction units, and automated concept formation systems, by a common *mathematical software bus*. Reasoning systems integrated in the MATHWEB can therefore new services to the pool of services, and can in turn use all services offered by other systems.

The next step in this development will to offer ANALYTICA as a MATHWEB service, making it possible to send problems in OMDoc form and receive OMDoc-encoded proofs in return. The main problem here lies in the *Mathematica*/ANALYTICA logic as we have seen above. We plan to augment the proof output of ANALYTICA to point to the justifying theorems to make ANALYTICA proofs independent of the ANALYTICA prover itself: ANALYTICA does not currently produce proof objects; rather, a trace of the proof search is output as

[1] The postfix __ after the variable name marks `a` as a sequence variable for *Mathematica*. A single underscore marks a normal variable, and a triple one a possibly empty sequence variable

a side-effect. Eventually, we plan to supply proofs from first principles for all the knowledge used in the prover, so that ANALYTICA proofs are grounded in axiomatics, as they should be for a theorem prover for mathematics.

5 Conclusion

We have described a recent effort to port the code base of the ANALYTICA theorem prover to the newest version of the *Mathematica* language and to restructure it, so that can be extended to new mathematical areas. The knowledge part of ANALYTICA is translated to the OMDOC framework for mathematical knowledge representation. This general setup seems ideal for a knowledge-rich deduction component like the ANALYTICA theorem prover, and for the combination of computer algebra methods with proof engines.

References

[ABI+96] P. Andrews, M. Bishop, S. Issar, D. Nesmith, F. Pfenning, and Hongwei Xi. TPS: A theorem-proving system for classical type theory. *J. of Automated Reasoning*, 16:321–353, 1996.

[BCZ98] A. Bauer, E. Clarke, and X. Zhao. Analytica — an Experiment in Combining Theorem Proving and Symbolic Computation. *J. of Automated Reasoning*, 21(3):295–325, 1998.

[Ber85] B. C. Berndt. *Ramanujan's Notebooks, Part I*, pp. 25–43. Springer, 1985.

[CCAMC02] O. Caprotti, D. P. Carlisle, and eds. A. M. Cohen. The Open Math standard, version 1.1b. Technical report, The Open Math Society, `http://www.nag.co.uk/projects/OpenMath/omstd/`, 2002.

[CKOS03] E. Clarke, M. Kohlhase, J. Ouaknine, and K. Sutner. Resurrecting the ANALYTICA theorem prover. In *First QPQ Workshop on Deductive Software Components*, CADE-19, Miami, USA, 2003.

[Com] XML Remote Procedure Call Specification. `http://www.xmlrpc.com/`.

[CZ92] E. Clarke and X. Zhao. Combining symbolic computation and theorem proving: Some problems of Ramanujan. In D. Kapur, editor, *CADE-11*, volume 607 of *LNCS*, pages 66–78, 1992. Springer Verlag.

[KF01] M. Kohlhase and Andreas Franke. MBase: Representing knowledge and context for the integration of mathematical software systems. *Journal of Symbolic Computation*, 32(4):365–402, 2001.

[Koh03] M. Kohlhase. OMDOC an open markup format for mathematical documents (version 1.2). Technical report, Computer Science, Carnegie Mellon U., 2003. forthcoming.

[Kut03] T. Kutsia. Matching in flat theories. In UNIF2003 [UNI03].

[Mar03] M. Marin. Functional programming with sequence variables. In UNIF2003 [UNI03].

[Sut02] K. Sutner. `automata`, a hybrid system for computational automata theory. In J.-M. Champarnaud and D. Maurel, editors, *CIAA 2002*, pages 217–222, Tours, France, 2002.

[Sut03] Klaus Sutner. Converting mathematica notebooks to OMDOC. to appear in [Koh03], 2003.

[UNI03] 17^{th} *Workshop on Unification*, Valencia, Spain, 2003.

[Wol02] S. Wolfram. *The Mathematica Book*. Cambridge Univ. Press, 2002.

[ZK02] J. Zimmer and M. Kohlhase. System Description: The Mathweb Software Bus for Distributed Mathematical Reasoning. In A. Voronkov, editor, *CADE-18*, number 2392 in LNAI, pages 139–143. Springer, 2002.