Conferences on Intelligent Computer Mathematics 2012

Notes by J.H. Davenport — J.H.Davenport@bath.ac.uk

8–13 July 2012
(updated December 2012 to include URL to Wolfram's talk)

Contents

Chapter 1

MIR — Sunday 8 July

MIR = Mathematics of Information Retrieval.

1.1 `MIR.fi.muni.cz`: Past, Present and Future — Sojka

MIaS = Math Indexer and Searcher. based on a Lucene Core

- Format choice depends on application's purpose
- Many (e.g. CAS) have on structure.
- New canonicalisation (DML talk on Monday) influence MIR experience considerably
- Is wider unification and Content MathML indexing needed when moving towards research search? If so, this is big research area of "Math-aware NLP".
- `http://mutin.fi.muni.cz` has Dost'al indexing of formulae images.
- Ranking based on semantics profiles (MSC-based)

1.2 Searching Induced Formulae — Kohlhase

Traditional paradigm is to crawl the resources, index the search-relevant information, process user queries and rank/process the hits. Claim that this is probably not sufficient for the working mathematician, who wants to search the knowledge space generated by the literature, not just the literature itself.

Framing If we can view a as an instance of concept B, we can inherit all of B's properties. This is very common (e.g. Bourbaki). This is the "little theories" view as well. See also Kohlhase's MMT. But, of course, the graph becomes extremely complicated in practice (show part of MMT graph).

Flattening Our search engine searches over the induced space of flattening the MMT tree, and returns an URI to the flattened result. These URI's can 'explain' their genesis. The LATIN Logic Atlas has 24MB of mathematics source text, organised this way. 60,000 declarations, generated from 2,500[1].

1.3 Latest Developments of MIaS — Růžička

Large indexing phase based on a Lucene core. We use a weighting utility, with initial weight 1/(number of nodes). There are weights (costs?) associated with tokenisation and unification. Future work involves canonicalisation and unification, possibly also equivalence via CAS. Their database has an average of 380 formulae per arXiv paper.

1.4 Using LaTeXML and MathWeb in MediaWiki — Schubotz

The default behaviour in Wikipedia is the "master" view, preferably MathJax, if not then database lookup, or if an image is not found, running LaTeX. The browser sees MathJax or an image. We propose a new view where the server uses LaTeXML to produce MathML, and the browser sees this and uses either MathJax or MathML.

1.5 MIR Happening

The contestant judges were JHD and Patrick Ion (PDFI). Queries "?" indicate unknown symbols, i.e. candidates for α-conversion. WebMIaS was running off a sandpit of about 10,000 articles and 1.6M formulae.

2.1.1 `\sqrt{2} = 1 + \frac{1}{3} + \frac{1}{3\dot 4} ?- \frac{1}{3\dot 4\dot 34}`, i.e.

$$\sqrt{2} = 1 + \frac{1}{3} + \frac{1}{3\dot{4}}? - \frac{1}{3\dot{4}\dot{3}\dot{4}}. \tag{2.1.1}$$

2.1.2 `B_{p+n} = B_n + B_{n+1} \bmod p \ \text{for all}\ n=0,1,2,\dots`

$$B_{p+n} = B_n + B_{n+1} \pmod p \text{ for all } n = 0, 1, 2, \ldots. \tag{2.1.2}$$

WebMIaS just asked for `B_{p+n}` This found 451 articles, of which the first was the right one (`http://arxmliv.kwarc.info/files/1010/1010.1573/1010.1573.xhtml`, [SZ10])! This is because α-conversion was applied. JHD notes that B is in fact a literal for Bell numbers.

[1] Hence the power of factoring this way.

Schubotz tried the same (`B_{p+n}`), without α-conversion and got just the 'right' document. Both JHD and Schubotz (using MathJax) had had problems with the LaTeX coding of `\bmod`. PDFI noted that this was a common issue.

One of the audience comments that traditional IR queries have a description as well as the query. The working mathematicians may well be searching for a "formula about".

JHD also noted that B is a literal, n is a free variable (over $\mathbf{N}$), but p is actually a prime (the `\bmod p` ought to be a hint).

2.1.3 `S(g) = \frac{s(g)-s_{\text{min}}}{s_{\text{max}}-s_{\text{min}}}` i.e.

$$S(g) = \frac{s(g) - s_{\min}}{s_{\max} - s_{\min}}. \tag{2.1.5}$$

2.1.5 `a?x^2+b?y^2` i.e.

$$a?x^2 + b?y^2$$

The original formula ([MT08, p. 2]) is $ax_1^2 + bx_2^2$, but WebMIaS couldn't find it.

2.1.6 `\frac{e^2+3}42^{?l\choose 2}?n^?l` i.e.

$$\frac{e^2+3}{4}2^{\binom{?l}{2}}?n^{?l}. \tag{2.1.6}$$

WebMIaS found six hits, of which the first (correct: `http://arxmliv.kwarc.info/files/0801/0801.2554/0801.2554.xhtml`) had a score of 1.2, and rest of 0.06 or less.

2.1.7 `?P\in \sum_{i=1}^r\Z ?{P_i}`

$$?P \in \sum_{i=1}^{r} \mathbf{Z}?P_i. \tag{2.1.7}$$

WebMIaS just asked for `\sum_{i=1}^r`, and in fact the target paper[2] came up first with 0.56, with some other papers at 0.39.

2.3.1 `f_1(x_1,\ldots,x_n)<0\land f_2(x_1,\ldots,x_n)<0`, i.e.

$$f_1(x_1, \ldots, x_n) < 0 \land f_2(x_1, \ldots, x_n) < 0. \tag{2.3.1}$$

JHD had observed that this is complicated for several reasons. The text ([JPS08, p. 1]) talks about "$f_1\sigma_1 0, \ldots, f_m\sigma_m 0$", so one has to

- realise that "," is "$\land$";
- infer "$f_1(x_1, \ldots, x_n)$" from "f_1" and the earlier $f_i \in K[x_1, \ldots, x_n]$;

[2] `http://arxmliv.kwarc.info/files/0712/0712.3704/0712.3704.xhtml`

- infer "$f_1 < 0$" from "$f_1\sigma_1 0$" and the earlier $\sigma \in \{<, =, >\}^m$ (where $\sigma = (\sigma_1, \ldots, \sigma_m)$ is wholly implicit). The "$\sigma \in \{<, =, >\}^m$ " provoked some debate.

In fact, this is a remarkably hard problem, and a related question would be "what mathematically sensible queries *will* retrieve the opening paragraph of this paper?"

1.6 Mathematical Search — Michael Trott

Looking at search from the point of view of Mathematica and Wolfram Alpha. Note that it is already possible to search for facts, and get values *and* sources. What about "is every matrix similar to a Toeplitz matrix".

Idea How far does mathematical search differ?

Many challenges some shared with others.

- Structure of OCR documents
- Disambiguation
- Dummy variables
- Data — all inputs are human-verified (expensive!).

Data Real time, factual data. facts support multiple inference: 'When did X win the Nobel Prize', 'Who won the prize in2010' etc., as well as questions involving multiple facts (ratios etc).

Size e.g. 50,000 formulae for special functions.

Quote "unit conversion is a major problem".

Q It is rumoured that Alpha is hundreds of person-years: is this true?

A The whole team is indeed hundreds, and has been working for years, but the mathematical component is a small proportion: tens of person years?[3]

Q Computational Number Theory?

A Essentially what Mathematica does!

1.7 Building a POS-annotated Corpus of Scientific Papers in Mathematics — Wolska

Actually the title is "Mathematical English". POS = "parts of speech". POS is a standard starting point for any computational linguistics these days. Supervised POS tagging now gets accuracy rates in the high 90s%, at least in the news domain, and general text (the available corpora).

[3] But Patrick Ion heard "ten person years".

Data 10,000 arXMLiv documents (Bremen). Take a subset of 500 random subsets. Only used document body sentences (abstract sentences are even more bizarre).

tag set Penn Treebank (PTB) tree set. Widely-used (available tagged corpora), general models available, and fairly coarse granularity (36 tags). Needed a finer definition of "TO" (JHD thinks this was a tag, not the word).

work Two student annotators. 100 sentence double-annotated and MW-adjudicates, 2×100 annotated and MW-verified. 200 annotated by MW.

Findings The tag distribution is distinctly different from general English. Pronouns are much more common, though very few different ones — "it", "we" and "us".

Q–JHD Does PTB distinguish parts of a verb? For example indicative versus conditional.

A Yes, but only on the basis of individual words, so 3rd person present is different from the others, ditto infinitive. Not conditional.

Q "Isolated sentences"?

A Yes — the annotators were not mathematicians.

***** Apparently the phrase "open set" causes some problems for the annotators.

1.8 Annotating Mathematics Expression Definitions for Automatic Detection

Mathematical formulae and their descriptions are inseparable n a document (proof by eye movement!).

Challenge 2009–11 Track the connections between the formulae and surrounding text.

***** When is something a definition. Various texts were looked at (JHD is not quite sure he agrees with the classification). "A partition B of V is G-invariant" is a 'fragmented definition'.

1.9 Zentralblatt — Sperber

Zentralblatt (and MR) have the same ultimate goals — make mathematical knowledge searchable — as the 144-year old JFM. But much else has changed, both qualitatively and quantitatively.

1.9.1 Keyword extraction and controlled vocabulary

Zbl/Technical Information Library (Hannover). DeLiCeeMATH project. Main idea is that in fact key phrases are more important than key words. Can extract keywords/phrases against MSC classes and do intellectual filtering. Then we will combine this with MSC. Also do context analysis for formula search.

Q You compared your clustering with another: what?

A (Sojka) Hierarchical semantics clustering.

1.10 Searching and Ranking mathematical formula — Murata

Note that "integrals of sine functions should not return $\sin x \times \int \ldots$. e had data from the Wolfram Functions Site: about 8,000. The query language is MathML + extensions (e.g. "argument is" *or* "argument includes"): source is both presentation and content. Ranking is focus on length (of query and result) and clarity (ratio of matched parts with given query and frequency of appearance). See `http://webdemo.visionobjects.com/equation.html`. Currently slow because of problems of matching tree structures.

1.11 Math Information Access

In a Japanese study, 77% of researchers said that mathematics was relevant to their research. Therefore one sub-task was to extract natural language definitions[4] from formulae in a document.

[4] In questions, it came out this should be "descriptions".

Chapter 2

DML — Monday 9 July

DML= Digital Mathematical Libraries.

2.1 Wrapup of MIR — Kohlhase

Emerging consensus: two things matter:

- Mathematical Search
- Mathematical Semantics Recovery (be it from an OCRed document, born-LaTeX, or even Word/Equation).

Note from Trott's talk (section 1.6) that they throw manpower at the problem. Note the importance of unit conversions.

2.1.1 MIR Happening

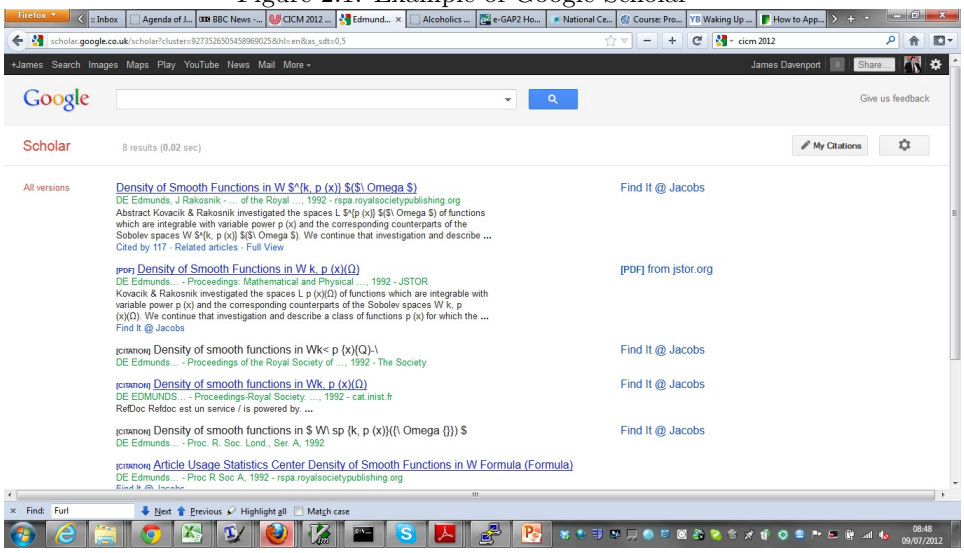
Intended to be an analogue for the CADE Competition, but we aren't ready for that yet.

2.2 JBIG2 Supported by OCR — Hatlapatka

JBIG2 is a compression standard. Shows original and compressed version: very little difference. Supports both lossless and lossy modes, in PDF since 1.4. Specialised compression is used in each region of the image, but multi-page compression is supported. Tesseract OCR.

The number of different symbols recognised for a page is several times greater than for born digital documents. Hence we wish to unify different "variants" of the symbol, which is the OCR goal. Improvement increased with number of pages, and seemed to be 10–15%. This technology is ready for integrating into DML-CZ and EuDML.

Figure 2.1: Example of Google Scholar



2.3 Normalization of Digital Mathematics Library Content — Liška

> The Google generation is beginning to demand the same functionality for mathematics as for text.

But EuDML has multiple content providers, metadata schemes etc. Google scholar : Kovacik Rakosnik returns many forms of the papers — see Figure 2.1. LaTeXML produces a certain form of MathML. Other tools (he showed MatLab) produce rather different MathML.

So far we use the UMCL Library for canonicalisation in our MIaS system. It has several problems.

- Can change the semantics (!)
- Very slow (168M formula in their corpus).

Some Presentation MathML includes elements we don't need for search, such as `mphantom`. Omit these. Also options on `mfrac` such as `thickness`. `msubsup` also provides a different encoding to `msub`/`msup`. Also drop `#x2061` (`&ApplyFunction`?).

Tool is written in , like EuDML in general.

Q Content MathML?

A No. Almost none occurs, and it has semantics anyway.

Q–MK What about the work at DesignScience? Also objects to `msubsup`–flattening. I would like an option to suppress this.

Q Also note that subscripts and superscripts in practice occur in either order.

Q–PL Do you intend to evaluate this with real mathematicians?

A We haven't really done any evaluation yet.

2.4 DynGenpar — Kofler

A Dynamic Generalised Parser for Common Mathematical Language. FMathL is intended to be a modelling and documentation language for the working mathematician. This is the parser component of FMathL.

Existing parsers were not adequate. C++ using libQtCore. Integrated into Concise, but there's a GPLv2+ stand-alone version. We need more general grammars than LR(1) or lALR(1), and parallel multiple CFGs. Need multiple parse trees for semantic disambiguation. Predictive parsing is important for user interfaces.

'Dynamic' means that rules can be added at any time which rules out precompiled tables.

Top-down parsing chokes on left-recursion: $Expr \to Expr + Term|Term$ which requires grammar transformation or work-around with precompiled tables.

Consider a directed multigraph on $\Gamma = N \cup T$ where the tokens T are the sources. The tokens T are the sources, and there is an edge from $s \in \Gamma$ to $n \in N$ iff there is a rule from a string including s to n, which is labelled by the rule name, and the number of null elements in the production.

Let $s \in \Gamma$, $z \in N$ be the target we are trying to reach. Then the *neighbourhood* $\mathcal{N}(s, z)$ is he set of edges from s to any c where c can arrive at Z. These can be computed by a (cacheable) graph walk. This ends up being a mixture of top-down and bottom-up parsing.

Our syntax tress end up looking like DAGs (packed forests!). This avoids some case distinctions, and allows for exhaustive parsing. We can also have custom parse actions. We are 5 times slower than Bison at pure parsing, but 11 times faster at grammar conversion, and doesn't require recompilation. Error correction/reporting is still pretty basic.

2.5 Writing on Clouds

Watt/Mazalov, presented by Rui Hu.

Handwriting recognition for mathematics has no fixed dictionary. We would like to keep the recognition cloud-based to share training data across several devices. This should have pooled training data and corrections (for multiple devices, but the same user).

Handwriting is a sequence of points, at device-dependent sample rates. Consider X and Y as functions of time, or arc length. Approximate via orthogonal polynomials (d of these), then X, Y is a point in $2d$ space,

and a cluster of points represent various versions of the same character. So we have a SOAP message containing coefficients of the sample, which returns a (list of) possible Unicode results. In practice a character can have multiple styles (especially if we're measuring time, i.e. direction of writing).

Evaluation The recognition error of the Nth sample is the ratio of number mis-recognised. The *Basic* strategy is adaptive: add the test sample to the training class. The *Null* strategy ignores it. Basic does much better than Null. 20 training samples for Basic seem adequate: clear cliff in the graph.

Q Can a given user mix whether you sample by time or arc-length?

A Devices return time, so can choose which.

Q–AS Different devices have different error corrections: did you use a range?

A Yes, but not sure if we tested given users on a range of different devices.

Q Speech recognition used to need large amounts of training, but now it works "out of the box". Do you envisage the same?

A Not sure, but we don't need much training.

2.6 Panel: Towards World DML: Are We on the Right Track?

JR Jiří Rákosník (DML-CZ, EuDML),

PI Patrick Ion (Mathematical Reviews),

AS Alan Sexton (University of Birmingham),

WS Wolfram Sperber (Zentralblatt Math),

JR listed stakeholders, but pointed out that mathematicians are the once most concerned, over librarians and publishers. Noted that 6th ECM in Krakow had many discussions about this. PI pointed out the "open access" movement among funders is not quite the same as WDML envisages. WS pointed out that EuDML ends as a project in Jan 2013, but becomes a consortium, open to new members. AS believes that WDML should be funded as a public good, as arXiv and INSPIRE. A commercial subscription model would be a disaster. There is, in his view, sufficient technology to make it work. A 'top down' analogy would be the UK's NHS Information System: a top-down disaster.

Q–PlanetMath This seems like a very slow way of getting anywhere.

A–AS NumDam pioneered the 'moving wall' (at least as an agreement), and EuDML has shown that it can work in the real world across a broad collection. This is the key to a smooth transition.

Q–PlanetMath But there isn't a plan!

A–PI It could have happened at the Sloan/NAS meeting, but didn't directly.

Q–MK Do we need WDML: can't EuDML just grow?

A–WS Notes that his working definition of DML has changed, from digitisation through to search etc. JR said that the consortium could grow, even now. AS remarks that we have taken what we know works, and developed it accordingly, and hence not in favour of a centralised plan.

Q Now about growing into different areas: e.g. Physics or Computer Science?

A–AS Good question. PI noted that JSTOR was a top-down plan. There is nothing to stop EuDML joining a multi-subject federation as well.

Q What is the connection with arXiv?

A PS: there's no technical problem. WS: our current system is restricted to reviewed articles. JR — as I said earlier, who knows what will happen to publishing in the next ten years.

Q Any commercial publishers?

A–JS One French one. EMS's own publishing house is in discussion, and progress *may* have been made at Krakow. Also discussions with Springer, but these are slow. There are also problems with the ownership of metadata: PI remarked that MR has problems getting metadata from AMS.

Q Why is it a good idea to have metadata but not plain text.

A–JR Added value, plus a demonstration to publishers. A further question is *who will archive*.

2.7 Challenges and Experiences in Managing Large-Scale Proofs — Daum

- 4-colour theorem: 60,000 lines of Coq.
- HOL Library in Isabelle, 66,000 lines
- Archive of Formal Proofs (AFP) 145–80,917 lines of Isabelle
- CompCert verified compiler — 100,000 lines of Coq (and 12 hours)

Rafal's observation:

Proof introspection is a significant part of proof development.

L4.verified has 25,000 human-generated lemmas (95,0000 in all). Matthias's observation:

> I must have waited weeks for Isabelle, so I am willing to sacrifice (temporarily) soundness for speed.

2.8 Dependencies in Formal Mathematics: Applications and Extraction for Coq and Mizar — Alama

Use Mizar Mathematical Library, and CoRN (Coq Repository at Nijmegen).

Definition 1 *A definition/theorem T* depends *on some definition/lemma/therem T' (or that T' is a dependency of T), if T's truth/value/... depends on the presence of T'.*

There's also

- All proofs of T use T'.
- This proof of T uses T'.
- This proof of T tries to use T'.
- This computation of T gives an error without T'

Coq This is not as easy as it seems "walk the λ-term" doesn't work. If T' is a tactic, we may never see the dependency, and these are not visible in the λ-terms.

Mizar Similarly for Mizar: *explicit* dependencies are easy. But much dependency information is missing. The Mizar environment is a (very conservative) overestimate of all items on which the article depends.

Coq tactic command structure: Ltac (domain-specific programming language) evaluation expression trees, nodes, OCAML tactics. The Coq language has various items.

Definition

Theorem

Qed ends a theorem or a definition, and saves it.

Tactic

`tmEgg` allows interleaving work in different theorems. when asked "load this theorem" only load necessary lemmas, and when changing the theorem, only invalidate those that use it.

Currently Coq has per-file dependencies, and we have added fine-grained per-item dependencies: still need to recompile the file, but only if it actually depends on a *changed* item, not an unchanged item in a changed file.

Both Massive change in amounts of dependencies depending on grain: e.g. Coq 8% probability at item basis, and 55% on a file basis.

2.9 Semantic Alliance — A. Kohlhase

Note that, for example, Excel is hard to use for difficult formulae, so why not something else (she suggested Mathematica). Stand-alone systems are great as far as they go, but insularity interrupts workflow. Invasive technology (see 2004 paper) will put one system inside another. Can have separate tools. But we wish to add an "Interaction Manager" , which is application-independent. This talks to "Application Event Handler (Alex)" which is the invasive component (one per service) and to a screen handler (Theo — currently built on Firefox). We built this for Excel and LibreOffice, and partly in GoogleDocs.

One side-effect is that the formulae display properly. Success in this depends on having an Open-API , supporting content, layout and player-API.

Q–PL Disagree strongly, e.g. Cinderella/ActiveMath.

A It would be nice if these had the same APIs.

Q–PL These are data paths, not APIs.

Q Do you think Google would be more opening the future?

A I don't know.

2.10 Extending MKM Formats at the Statement Level — Horozal

Contrasts "Mathematics" (theorem statement and "proof of theorem"), with Curry–Howard style, where the two are bound. The two are logically transformable, but look different. Extensions:

object (typically identifier) e.g. `2:=succ(succ(0))`. Defined by user.

statement (typically with keyword) as locales in Isabelle. Defined by programmer.

We say that an extension principle is *unconstrained* if it is interpreted at runtime, *constrained* is there are well-formedness judgements.

OMDoc 2 has "pragmatic" and "strict" (as MathML 3).

pragmatic surface `theorem 1+1=2 (foo)`
`proof term`

pragmatic abstract

strict abstract

Strict OMDoc 2 is MMT, their foundation-independent language.

Modules

Theories

Expressions

This lets them write Mizar-style theorem definitions, and HOL-style type definitions. There is bidirectional pragmatic-to-strict translation. We have a notation parser specific to each pragmatic surface syntax. Development of which is ongoing for Twelf surface syntax, but done for sTEX.

2.11 A Streaming Digital Ink Framework for Multi-Party Collaboration — Hu

Want a portable framework[1]. We anticipate the synergy of pen-based collaboration and recognition of mathematical input, but there is no current technology. Windows provides collaboration whiteboard with no recognition, and Maple the converse. Should support mathematics and diagrams.

We have a platform-dependent layer, and invoked the JNI layer, which converts that into a platform-independent "ink provider", which talks to "Java Digital Ink" applications. We use InkML.

We have "collaboration extensions" talking to each other, using "InkChat", transmitting voice and InkML data, and doing synchronisation of the canvas. Also a Mathematical Recognition Extension (section 2.5). Also a compression extension, which supports both functional and linear approximations, useful for mathematics and diagrams respectively. Also have a round brush (pen) and teardrop brush (now adopted by InkML), as well an an (ink-free) pointer. There's an archival extension to support store/load of collaboration sessions.

Currently use Skype and GoogleTalk as communication vehicles. Common interface is very valuable for extension combinations. Can combine playback with recognition, for example. Devices, and standards, are fast-moving: it was important to be part of InkML development.

[1] Windows, Linux and MacOS, currently.

2.12 Point-and-write – Documenting Formal Mathematics by Reference — Lange

Carst Tankink did the main implementation, but he's ill.

Documentation of formalised mathematics is hardly supported. The narrative flow of the informal documentation is not the same as the logical flow of the islands of formal code. Examples.

Literate Proving required adapted workflow.

Coqdoc requires adherence to the formal workflow

Hyperlinking No juxtaposability.

Separate documentation e.g. formal snippets into a LaTeX document, which is messy (note that references are by line numbers!).

Use Wikipedia-style markup on own sandbox Wiki. Embeds by formal identifier (URI in Mizar's case). Syntax is `@{type link}`, inspired by Isar's antiquotation). Hence Wiki source code is readable, but the annotations need to be in the formal source.

For Mizar, can generate internally, for Coq, hack the HTML from CoqDoc. This is implemented in Agora (at `http://mws.cs.ru.nl/agora`). It would be nice to make the references by queries, i.e. "proof of Lemma ...". Autocompletion would also be nice (feasible via SPARQL). It would be good to survive change of identifier, as well.

Would like future CICM papers to be written in Agora.

Q–OC Used in e-Science applications.

A Not as far as I know.

2.13 Work in Progress

2.13.1 A XML-Format for Conjectures in Geometry — Quaresma

Work in Intergeo format.

2.13.2 PlanetMath/Planetary — Corneli

There's a new version at `beta.planetmath.org`. Has scanned (OCR ongoing) books, e.g. Calculus by David and Brenke. His current work is adding exercises to PlanetMath. Planetary is built on top of Drupal (apparently 7.10, Drupal 8 is the Symphony version).

2.14 DML Business Meeting

2.14.1 Introduction

PS opened the meeting. Last year, DML had agreed to join CICM. He had been appointed as Track Chair at 2011 for this meeting, and TB as a CICM Trustee from DML. DML is less formal than the others (MKM, Calculemus).

One question was the constitution of the "DML Community". It was certainly(AS, MK, PS and others) not limited to those who were members of the funded DML project.

MK noted that Calculemus and MKM had been independent bodies before they joined CICM, hence had had to have formal structures. There was no need for DML to have such a formal structure. PS said he wanted rather more formality than just a self-appointed spokesperson.

There were three papes submitted to DML formally, three to Work-in-Progress, and 10 to the MIR workshop, which was cleary DML-related. MK pointed out that Calculemus had, at one point, been down to three papers as well, and one of the points of CICM was precisely to share this sort of risk.

There is currently no formal DML mailing list, only PS's personal list. WS ponted out that EuDML will end in January 2013, hence we should use CICM as a meeting point.

OC noted that there had just been a CEUC-sponsored meeting in Washington. OC/JHD could place the future of DML discussions on the next CEIC agenda[2].

2.14.2 Actions

- MK would organise a mailing list.
- MK proposed that DML voted on the question of Track (Springer proceedings and all it implied) or Workshop. rack was carried 8–0. PS was elected as representative by acclamation.

2.15 MKM Business Meeting

2.15.1 Introduction

SA opened the meeting.

2.15.2 Track Chair Report

MW presented the track chair's report. He was definitely in favour of the rebuttal period, and the shepherding process had been used for a few papers. 19 submissions[3], of which 13 were accepted (2 after shepherding). There were

[2] Berlin 18–20 July 2012.
[3] Plus one moved to DML.

60 reviews in total. 12 external reviewers were used. The acceptance rate was slightly higher than last year, but his feeling was the standard was slightly higher. The chair was thanked.

2.15.3 Future

MW noted that the allocation of papers to sessions maybe need not follow the track organisation quite so closely. It was also noted that the division into track PCs was somewhat inflexible. There had been some "borrowing" of PC members from one speciality to another, and this had affected the percieved balance of the tracks/conference.

2.15.4 MKM 2013

This wuld be part of CICM 2013 in Bath (JHD local organiser). David Aspinall had been asked, and accepted, to be MKM Track Chair.

2.15.5 Trustees

The Board of Trustees has a permanent post of Treasurer, but since the finance has been moved to IFCOLOG, and the organisation to CICM, this was redundant as such, and the Trustees had decided to interpret his as a member-at-large, with Patrick Ion. This was applauded.

AS and JHD were finishing their terms as Trustee. DA would replace AS, and JHD's place was up for election. Nominations by the end of the CICM meeting. **But** we needed a returning officer — volunteers to SA. There were immediate nominations for Trustee of Volker Sorge and Johann Jeuring.

Chapter 3

10 July 2012

3.1 Text Mining Methods Applied to Mathematical Texts — Haralambous

There are many ways of saying the same thing.

- There is no triplet of positive non-zero integers such that the sum of the cubes of two of them is the cube of the third.
- $\neg\exists a, b, c \in \mathbf{N}^{>0} : a^3 + b^3 = c^3$
- The case $n = 3$ of Fermat's Last Theorem is true
- ....

Quotes [Zin04] as an example of difficulties "can only process two [theorems] from [[HW79]]". See also [Bau99].

Possible strategies.

1. Controlled natural languages, e.g. Mizar [RT03].
2. Use XML Markup, as in OMDoc [Koh01, Koh07].
3. Use a visual language, as in MathLang [KMW04]
4. Use statistical methods, as is parts-of-speech taggers. A syntax parser produces the most likely tree. Then pragmatics, anaphora resolution etc.

* Also, [Wat08] does classification of documents by symbol frequency analysis. Note, this solves a different problem. It is essentially "bag of words" restricted to symbols.

What happens if we move from "bag of words" to "bag of terms". Trouble is that 'term' is vaguer. Terms are used to describe symbols, or, inversely, symbols

are used to denote objects described by terms. Terms exist in documents, which are events in the real world, and have a timeline.

A simple model is topics/documents. Mor elaborately, we look at a Latent Dirichlet Distribution. An enhanced version of this is Pachino Allocation Model, which has more levels than topics/documents. Furthermore, we can look at syntax (moving beyond a simple bag).

How does "topic analysis" enrich our knowledge of the literature?

Most of the blocks of a paper (lemmas, acknowledgements etc.) are intra-documental, but theorems, definitions etc are inter-documental. Can this refine the usual digraph of citations? There are also graphs of co-authors, shared institutions etc.

There are a lot of statistical tools, which, even if approximate, may well be useful.

Q–MK Very interesting. How much have you done, and how much is you dreaming and our sharing the dream?

A There are lots of tools out there which could be used.

MK Our experience is that standard tools fall over immediately when presented with mathematics.

3.2 An essence of SSReflect — Whiteside

Based on [WAGD11]. SSReflect is a language for **big** proofs. See [Har98]. Tacticals and the "By tacticals". A "hiproof' is a proof viewed as a hierarchy. 'Hitac' is a hierarchical tactic language. This allowed recursively defined tactics. Predefined ones are all(X) (applies X to every subgoal), id, null, and rotate (the list of goals).

Scripts are made of paragraphs, and a paragraph is a non-empty list of sentences. Shows a series of translations between SSReflect and Hitac.

3.3 A Query Language for Formal Mathematical Libraries — Rabe

Note that my scope is formal mathematics, but it can be used elsewhere (presentation, narrative). Querying is a natural MKM application. The LATIN library has 4 years, c. 10 authors, c. 1000 modules and is systematically modular, but still difficult to search. Questions like

- What does s depend on
- What theories import t

are difficult to answer.

Questions, as asked by [ADL12] should be easy to answer in my MMT, but of course this is a different system. SPARQL is a classic query protocol, but

systems like Coq and Mizar have their own variants. The typical query is essentially relational, and easy to support (as long as we aren't using mathematical expressions!). The relational model is also bad at transitive closure, such as the dependency relation.

Heavyweight XML database with XQuery, SPARQL etc.

Lightweight (i.e. this talk) MMT-based query language QMT, and this raises queries about client–server protocols. Same code can be both.

Ground concepts, also types (tuples and sets, but not nested), relations (with composition, inverse etc.), propositions and queries. We therefore define a QMT signature over MMT. This makes it easy to ask "all theories which transitively include u", and can restrict to which symbols are imported etc. Has the power of DL and Zquery.

All binders are relativised by a query: $\forall x \in D$ (base type of all OM Objects is infinite, but a query is always finite). Relational symbols r and predicate symbols p are different (JHD didn't quite follow why, but suspects it's a question of level: relations might query *on* predicates). Queries from Javascript are AJAX-style: the cycle is hidden from the programmer.

This is possible to implement for other systems:

- Write an export into MMT;
- Express queries in QMT.

3.4 Abramowitz and Stegun — A Resource for Mathematical Document Analysis — Sexton

"To do MKM, we need some Mathematics!" Various sources: automatic such as arXMLiv; manual such as DLMF and Wolfram Alpha; or OCR. We've been looking at [AS64, US Government, not Dover].

So what is the process of (mathematical) OCR?

1. Image processing
2. connected components identification
3. segmentation
4. Layout analysis

* and then, in some order (the mathematical things that general OCR people don't do):

5. Formula recognition
6. Table analysis

7. Diagram analysis

It's easy to do small projects, but going beyond this requires significant infrastructure. Hence we need shared corpus/ corpora, data output for every stage, etc. We use Creative Commons Attribution 3.0 Unported Licence (CC-BY). This allows document analysis researchers to work on single stages of he pipeline without requiring a full implementation of the pipeline. Initially, we have high quality page scans, image processing results, connected component analysis, feature extraction and diagram clips. Planned: character identification, segmentation, layout plans, structure analysis, table extractions, formulas as MathML.

What else is there?

Infty I–III Large high quality manually constructed ground truth. Original page images are not freely available.

UW-III 25 pages of mathematical content, no longer available.

Ashida *et al.* Not available

[BSSS11] We intend to release this, but it's only 10 pages.

One-column and two-column format totally mixed-up. This is *very* challenging. PreTEX—itself interesting. The small fonts (5 pt for the prime numbers!) are quite OCR-challenging. Plots — quotes [AS64, Figure 20.13]. So what do we have.

lossless PDF 55MB at 300dpi,binarised, deskewed, with small connected components removed.

JBIG2 of above 13MB.

All pages 300/600dpi TIFF g4 54/111MB.

All 267289 connected component images. 600dpi monochrome RGBA TOFF 392MB.

Geometric features for all connected components 255MB.

...

Complete Gray scale 19GB

`http://www.cs.bham.ac.uk/~aps/research/projects/as`.

Q–JHD Fantastic, but runs the risk of distorting the field, since everyone now will work on [AS64]. It would be great to have some other text (even small by comparison) done through the *same* process.

A Yes — happy to do more if I have *totally copyright-free* material.

***** General debate on what was MKM, and/or DML.

3.5 Cost-Effective Integration of MKM Semantic Services into Editing Environments — Jucovschi

Calim: Service + Editor Integration = Better Service. So how do we do this.

1. Editor extensions: generally available. But very editor-specific.
2. Editor generation frameworks. CFG of language gives an editor in 30 minutes. Example is xText. Note that *requires* a CFG (in practice, certainly).
3. Real-Time Documentation Synchronisation (me!).

His system allows contributor-control of reversion/merging, where 'contributor' might be a service. The API allows multiple entities to change the document in real-time. Allows entities to exchange private messages, such as auto-complete.

Direct editor–service integration breaks separation of concerns. A document-centred approach pays off in the long run. `https://github.com/jucovischi/SharesJSServices`. His four services work, but probably not very stable yet.

3.6 Proof, message and certification — Asperti

The set $\mathcal{V}$ of true arithmetical formulas is a productive set. The formal system $\mathcal{T}$ may provide a (forcibly incomplete) r.e. approximation of $\mathcal{V}$ (usually expressed as an existential projection of a recursive set of proof-statement pairs). A formal proof for a given formal system $\mathcal{T}$ is a ....

"Ever since Euclid, mathematical proofs have served a dual purpose: certifying that a statement is true, and explaining why it is true. Now, these two epistemological functions may be divorced. In the future, the computer assistant may take care of the certification and leave the mathematician to look for an explanation that humans can understand" [Mackenzie2010].

Q–PL Note that the teaching of 'proof' in schools varies greatly by country.

Q–JJ Computer algebra systems also do this very differently from humans.

3.7 Understanding the Learners' Actions when using Mathematics Learning Tools — Libbrecht

Risks of interactive exercises: cognitive load (complaint about answer not being accepted). Shows an applet for formulating induction properly (based on Vaxima/ OpenMath). SMALA — Salience ....

Smala has a statistical log, aimed at the teacher's learning.

Q The system does not keep track of scores?

A Again only statistically.

3.8 Towards Understanding Triangle Construction Problems

Presented by Marić. Geometry constructions have been with us for 2,5000 years. How can computers help? Claims we need four components.

Analysis finding properties that enable a construction.

Construction A concrete construction procedure — essentially a program: technically for straight edge (ruler with no marking) and collapsing compass (can't transfer distances).

Proof that the construction meets the properties.

Discussion Uniqueness etc.

After removing trivial symmetries, there are 139 Wernick's triangle construction problems. 25 contain redundant information (A, B and midpoint of AB) or have a locus restriction; some are unsolvable etc.28 are "unknown". The authors claim there are 72 solution rules used. A Prolog program could solve almost all known to be soluble Wernick's problems, with maximal depth 9, and in at most one second, so this is easy.

The real complexity issue is the number of rules. Narrowed down to 28 lemmas, 18 construction primitives and 11 definitions. Could expand to another corpus, but only two more lemmas were needed, hence we can assume this will stabilise. Future work involves production of formal (Isabelle, probably) proofs to match the constructions. Might also go beyond triangle constructions.

3.9 Teasers for Systems & Projects

3.9.1 Tentative Experiments with Ellipsis in Mizar — Kornilowicz

Added flexary conjunction and disjunction to Mizar. This let us replace some lemmas, and shortened some articles, e.g. by 25%.

3.9.2 Algorithmic Structuring and Compression of Proofs — Hetzl

These proofs are very hard to understand, without help.

3.9.3 A web interface for Matitia — Ricciotti

Give markup a more semantic nature.

3.9.4 Open geometry Textbook — Chen

A case study of Knowledge Acquisition via Collective Intelligence.

3.9.5 Reimplementing the MSC as a Linked Open Dataset — Ion

Previously a large TEX file only understood by authors. `http://msc2010.org/mscwork`. It's deployed in PlanetMath and elsewhere

3.9.6 Planetary Project: Towards eMath 3.0 — Kohlhase

3.9.7 MaxTract — Sorge

Converting PDF into LATEX, MathML and Text. It now has a web interface.

3.10 Distributed Ontology Language — Lange

There's an upcoming ISO standard in this area, apparently feature complete, and with some tool support. Use DOL for expressing the relationships between first-order logical settings (LATIN?).

Chapter 4

UITP — July 11

"UITP" = User Interfaces to Theorem Provers.

4.1 Theorema 2.0 — Windsteiger

You've probably all seen Theorema 1.0. The redesign was prompted by experience of non-developer users.

Base Mathematica wasn't a bad choice, so we built on Mathematica 8 for this.

1.0 In Theorema 1.0, `Definition` was a Mathematica command, with Mathematica evaluation rules for its arguments etc. Similarly `Lemma`, `Prove` etc. `Prove` had a choice of prover etc., so users in general were faced with a variety of little-understood options.

2.0 Get a graphic interface, and "commander". There are 'activities' (e.g. 'prove a theorem'), composed of various 'actions'. The model is that of the "installation wizard", guiding you through the steps. Hence we select proof goals, knowledge bases, and proof methods. Such a method (new in 2.0) a combination of inference rules and a proof strategy.

Output Having hit 'prove' one gets a visual representation of the proof tree, as well as the old-style "proof notebook". The tree is clickable, which may be more useful than navigating the old-style notebook.

N.B. As in 1.0, the user is responsible for the knowledge base, so the user can place things into the knowledge base that have not been proven. hence circular proofs are still possible.

Q–CL What is really needed is help in knowing what to put in the knowledge base.

A Yes, but how?

Q–VS What percentage is old code.

A 100% new code, but old ideas/concepts are recycled when appropriate.

4.2 ProofTool: GUI for the GAPT Framework

GAPT = General Architecture for Proof Theory. `http://code/google/com/p/gapt`. Includes a theorem prover, but is really a proof transformer, not a prover. GAPT has typed λ-calculus, first and higher order logics, formula schemata. The calculi are LK (Gentzen), LKS (Gentzen with Schemata) and Resolution.

Chapter 5

OpenMath — July 11

5.1 Mathematical Computations for Linked Data Applications — Wenzel

From the Frauenhofer Institute for Production Engineering. Data from many sources needs integration and their calculations. Use RDF. for data integration, and OpenMath. One use case is calculation of performance indicators.

- Query RDF data from OpenMath: done via a CD. The RDF model is based on triples. `http://www.openmath.org/cd/contrib/cd/rdf.xhtml` uses a subset of Manchester syntax. Has a native OpenMath encoding for literals.
- Embed formulae in RDF: an OpenMath ontology `http://numerateweb.org/vocab/math`. This is a verbatim mapping of the OpenMath XML. This can be queried with SPARQL. In practice use a Popcorn representation. The OpenMath RDF can then be used in a DL reasoner.
- Mathematical rules and reasoning. We have `Cnstraint` which related rdf:property to OM:Object.

  ```
  @e:bmi = @e:mass/@e:height^2
  ```

  Then can apply this as a λ-expression over a group of people.

  Our reasoning architecture has a common triple store, OWLIM Lite, for rules and data. Computer Algebra System 'Symja' is extended with RDF interface,

Need to scale the inference for large data sets and many formulae. Need a Popcorn-based Web REPL. Issues are the complexity of reasoning, especially in the resolution of cyclic dependencies. This might also leverage OWL2 meta-modelling.

5.2 The Gf Mathematical Grammar Library — Saludes

Came out of multilingual WebALT project, but in 2003 Gf was not mature enough. Gf is a functional language capable of representing natural grammars as well as formal ones. The Mathematical Grammar Library has 15 languages so far `svn://molto-project.eu/mhl`. over the ground layer, we have OpenMath. The CDs are viewed as abstract models, such as `arith1` Then `arith1l` will be the concrete version, and `arith1eng` the English-specific part. Above this layer, we have 'operations', such as simple exercises, commands (Computer, Assign, Assert, Approximate, BeginBlock, EndBlock) and word problems ("apples and oranges").

Gf has levels 'abstract' and 'concrete'. ValNum, ValSet etc are MathObj, which is a Noun Phrase. ValFun is a Noun Phrase with extra information. Prop is a clause with polarity, but FullProp is a sentence. In Gf we can only make questions from clauses, not sentences.

In English, we combine the adjective "absolute" with the noun "value" to get a common nun (CN). This and "of" also gives a CN. "the" then gets a noun phrase NP. Similar in German, but need to specify noun plural as well, and gender, for 'value' = 'Wert'.

The real problems, at least in English, is with functions: "f at 3" versus "sin of 3" versus "the derivative of sin at 3".

"compute the integral of the function mapping x to the square of x from minus infinity to infinity". Need various transfer rules: 'square' needs expanding, "minus infinity" needs to be the right symbol, interval needs specifying etc. This is not easy in Gf..

Working in 3 languages. A new language needs: Resource Grammar support (should already be in Gf); needs to fill the lexicon, and a review cycle. A new module (= CD) needs an abstract module $M, entries in the lexicon for \mathrm{M}, concrete modules for \mathrm{M}$, with input form language and maths exerts, and a review cycle.

5.3 OpenMath Business

JHD's note: as at 15 July 2012, these are unchecked minutes.

5.3.1 Introduction

MK opened the meeting. He reminded us of the formal agenda.

1. Election of the Chair of the Meeting — MK elected.

2. Election of the Meeting Secretary and Minute Checkers. JHD was elected as Meeting Secretary. JWK and CL were elected as checkers.

3. Annual Report. Last open meeting 2010 in Paris (none in 2011, partly because of delay in starting). Formal Meeting in Bremen in 2011 to pre-

serve legalities. There had been no financial transactions. The following non-financial contributions were noted.

- The webserver has been donated by DFKI.
- MS pays for the domain.
- TUE runs the mailing lists.

There is, following Paris, an OpenMath Infrastructure Team and mailing list (`infrastructure@openmath.org`).

4. It was noted that the list on the web page needs updating. Of today's speakers, Saludes should be a member: **elected**. It was observed that Robert Miner had unfortunately passed away.

5. The Balance sheet was adopted, and the Committee discharged.

6. Committee. It was noted Dewar and Gaëtano had not been at the meeting for the next few years. Dewar had contributed electronically, and NAG collectively were major contributors to OpenMath. The same could not be said of Gaëtano, as INRIA were no longer contributing.

   Though we have no assets, we **do** need a Treasurer (Currently Christine Müller). Kohlhase, Dewar, Caprotti, Müller, and Seppälä were re-elected. JHD proposed, PDFI seconded the election of CL as Member-At-Large. This was carried.

7. OpenMath 3. There has been no progress, basically due to MathML 3 overload. It was not proposed to resume this work, but rather to proceed less ambitiously via "Technical Notes" etc.: JHD noted that we might make more progress by being less ambitions, and this was generally agreed. It was noted that MathML 3 is currently in "maintenance mode". The W3C working group Charter runs out in March 2013. There are no plans for MathML 4.

   CL pointed out that we need a route for doing these "Technical Notes". JHD agreed: noting that the list was non-empty ('endianness bug', MathML 3 as an encoding, "DefMP" sprang to mind).

8. CD Management. MK reminded the meeting that the process of making a CD into "official" status was via the CD Editor, JHD, and suggestions should be mailed to him. He would organise reviewers, and circulate a proposal to the mailing list.

9. PL noted that his name is on the bottom of all pages, and wishes that it were `infrastructure@openmath.org`. This was agreed. Note also the site `http://www.openmath.org/infrastructure/`. He also noted that DFKI would not necessarily continue hosting, and wanted a proposed replacement. Bremen was willing to take this over, and give PL an account. JWK would check whether TUE could take over the website.

This concluded the formal business meeting. Next meeting at CICM 2013 in Bath, whether or not there was a formal OpenMath workshop.

5.3.2 Workshop

CL asked about the future of the workshop. 2009 was large, and 2010 was small. 2011 was a failure of timely organisation. 2012 was organised in time, and yet only had two submissions, albeit of high quality. What should be done.

It was noted that people could submit OM-related papers to the CICM tracks, and this was the route to formal publication.

It was suggested that the title could be broader. There was no really competing venue. If we broadened out too much, we might run into, say, "Semantic Web" territory. The "Semantic Web" conferences were actually very expensive. The general feeling was that we should continue more-or-less as we are, possibly opening up the title, since CDs were relevant to people using Content MathML as their encoding. This did not stop us organising more "evangelical" events elsewhere.

It was noted that in 2009 several people had given short "CD talks" — the format still existed, but hadn't been taken up since.

Chapter 6

Doctoral Programme — July 11

6.1 Pen-Based Collaboration — Hui

Western; advisor Windsteiger.

Pen input is common, as is collaboration,, but no technology does both. Pen-based is very good for mathematics 3× faster and 2× less erroneous.

Portability i.e. platform-independence. Team members have different technology, and may use different devices at different times. Data portability is provided by InkML.

Multimodal Pen+voice. Hence "InkChat".

Q–OC Is it possible to correlate calligraphic style with anything?

A Don't really have the data.

Q–VS There's a French group doing something similar, see iJCAR

A Not aware.

Q–DJW Do you propose to assess your tool as in quotes?

A Yes, undergraduates.

Q–PS You train on characters, but people write complete formulae, and styles change depending on context.

A Have tried this, but don't really have the data.

6.2 Knowledge Management in Computer-Aided Design — Iacob

From Jacobs :-). There is the object and its documentation. Assemblies have parts and constraints. parts have geometric constructs, transforms and constraints. Change management is a major problem.

The FormalSAFE project at DFKI was a precursor of Formal CAD. A domain-specific language for engineering calculations. A geometry library and formalisation in HasCASL. Export from SolidWorks into HasCASL, and interaction with Isabelle/HOL. MathLang in a bridge between natural language like LaTeX and formal mathematics, as an alternative to OMDoc. Mizar. Open Geometry Textbook: OMDoc at object level with relations at statement level. Interaction with theorem provers and visualisation software. MMT elaborates the formal part of OMDoc. Infrastructure for presenting and querying. Interchange format with Twelf, and Mizar under development.

Looking at the Semantic Alliance Framework (see section 2.9).

Example 1 *Let C be the set of points in the plane defined by*

$$\{(xy) \in \mathbf{R}^2 | \text{the harmonic mean of } x \text{ and } y \text{ lies in } [-1,1]\}.$$

Because of the inclusion of text, this can only be handled by OMDoc as Presentation MathML.

Q Where are you getting the source form? You showed hand-written documents.

A I am assuming engineers who know LaTeX (general disbelief!).

6.3 Representing Declarative Languages and their Translations — Horozal

Jacobs — adviser Kohlhase. There is the expressivity/simplicity trade-off.

Categorical frameworks: Burstall, Goguen etc. Generalise concepts but know constraints on concrete theories.

Declarative frameworks, such as LF. Tool support for theories, such as type-checking. No abstract representation of theories,

How might we give both abstract and concrete presentations of theories in one framework? Working within the LATIN framework. In particular, use MMT and LF.

$$\begin{array}{cc} L^{Mod} & \\ \uparrow L^{mod} & \\ L^{Syn} & \hookrightarrow \\ \uparrow & \cdots \\ \cdots & \end{array}$$

FL Theories have n-ary function symbols f, n-ary predicate symbols p and first -order axioms. I make the shape of declarations explicit in the framework.

A mapping $\mu : L \to L'$ must map the declaration patterns as well as the symbols. If Σ_i is a theory extending L, then functor(μ, Σ_i) is a theory extending L', and $\sigma : \Sigma_2 \to \Sigma_1$ gives rise to functor(μ, σ) mapping functor(μ, Σ_2) to functor(μ, Σ_1). This allows to add an ellipsis constructor to existing systems.

Patterns for Mizar theories, though complicated, can in fact be written in this framework.

6.4 Content-based Formula Search — Schubotz

Jacobs — Kohlhase? Use cases:

- Patent Search: US Patent Office has the MathML, and this is useful since people use different terms;
- Enterprise search/corporate knowledge management;
- Researcher hinting (consider Wolfram Alpha).

Example: Jensen's inequality. Note that uses $\langle \cdots \rangle$ to mean "mean", but this isn't obvious, and not really encodable in MathML. Hence my requirements:

- semantics instead of presentation;
- text as well as formulae;
- context (hence meaning of symbols).

Consider the see-saw equation $M_1 a = M_2 b$, where we should recognise (form the diagram?) that M_1 and a are properties of the same object.

Q–PL Maybe you need better use cases, in the HCI style. I know you're a user, but you never do a good tool for yourself.

A Possibly.

6.5 Semantic Understanding of Mathematica Formulae — Almomen

This is important for searching, or further processing, and translating to speech, re-flowing etc. Context can be

local Do we interpret $f(a+b)$ as `&InvisibleTimes;` or `&FunctionApplication;`?

document

domain e.g. group theory. JHD: tends to change the interpretation of π.

Need to understand the formula: conversion to Content MathML, OpenMath or LaTeXML. I can't find any tool to convert to OpenMath. LaTeXML only discovered this week.

For context, "... if there exists σ with $w\sigma = t$, then $\nu_k \mapsto s \in \sigma$.

Currently evaluating Maple, Mathematica and SnuggleTeX (tool from Edinburgh). For $f(a+b)$, SnuggleTeX finds multiplication, but Maple (i.e. Maple's LaTeX parser) finds `&FunctionApplication;`. With a space after f, Maple finds multiplication. Need to build some ground truth to compare the tools to. Aim to use sentence pattern work with Wolska (see section 1.7).

Q–PL WebEQ had a web service (ex Robert Minor), which he has access to. Aso Wiris is an OpenMath-generating tool.

A–JHD Look at [Wat08], which seems to find context from symbol occurrences.

Q–WW What documents?

A PDFs (from others) which I extract.

6.6 Real Geometry and Connectness — Wilson

EPSRC Project, joint with UWO and Maplesoft. Define quantifier elimination on Tarski formulae. Has just seen a paper at Western which took 15 minutes to derive a major published paper. Note [Col75], and the general complexity [DH88]. Note new U.W.O. algorithm [CMMXY09].

Example 2 *$ax^2 + bx + c$ given 9(?) cells over complex and 27 over real.*

Laziness [CDM+10] seems to allow a drop in complexity, at the cost of ignoring lower-dimensional components. Can we do the same for CAD? Not clear, since knowing all the full-dimensional cells implies, in some sense, the others. For many applications, e.g. robotics, only the full-dimensional cells matter, since these are the only feasible ones.

- Can we get `LazyRealTriangularize` to interact with `MakeSemiAlgebraic`?
- Can we prune the tree before `CylindricalDecompose`?
- Also, paper at Calculemus [WBD12].
- Note repository at [Wil12]. Looking at the number of cells, which is strongly correlated with time in practice, but also an invariant.

6.7 Stephen Wolfram

A recording of this talk is available at `http://www.cicm-conference.org/2012/slides/StephenWolfram.mov`.

> I still think symbolic programming is "ahead of its time", even after 24 years of Mathematica (more if you include the prehistory of SMP).

Originally we implemented known algorithms, but the fraction of algorithms which are original in Mathematics is increasing, and now high. I spend a lot of time ensuring upwards compatibility. Automation is very important — leave as little as possible to the human: e.g. "solve an ODE" should automate the choice of method, and much of our effort goes into these "meta-algorithms". People don't use the over-ride controls we *do* build in. This includes things like choosing the number of points on a graph, the scale, and for diagrams the general layout.

As a example, he typed "picture of a beaver" ("frog" didn't work!), which retrieved one via Wolfram Alpha, and applied commands such as 'edge detection' on it.

An important area is "implicit representations", such as `Root`. Recently added transcendental roots. Solutions of recurrence equations will also generate these, and even "difference root" objects, presumably for non-soluble cases.

"Mathematica is a totally precise system, but Wolfram Alpha is very different". Will offer, for example, plots, special values, etc. "Enable Interactivity" button uses Wolfram's CDF to provide sliders etc. Not limited to Mathematics, tried "weather in Bremen", and then asked for a ten-year average etc. Typed in an apparent DNA sequence, and got matches in the human genome.

Wolfram Alpha is now 15M lines of Mathematica. Showed "p and q or not r", which gave truth table, alternate forms, Venn diagram, circuit etc.

The paradigm is computational. Very good for Applied Maths, but not so suitable for Bourbaki. How do we support this? The answer is more Alpha-like than Mathematics-like. Claims there's 3,000,000 theorems in the (curated) literature. What can we do with these? To what extent is Alpha a universal Turing machine?

For example, we can ask "what is the simplest axiom system for Boolean algebra", with an automated proof that it's correct (he tried to make it understandable, but failed). We haven't put much effort into making these understandable, since there's no demand for it. We can generate a vast number of theorems — which ones are interesting? The last theorem in Euclid, 5 Platonic solids, has the deepest dependency graph (32).

For example, if we regard "named" theorems as interesting, we can do the following for Boolean algebra. Enumerate theorems lexicographically, and ask "which cannot be proved from the previous ones", and these are the named ones (except De Morgan's Laws, which *can* be proved from the previous, but the proof is long). Hence we *may* have a usable definition of "interesting".

Claims that the typical mode of developing mathematics has been

> Here's a neat idea (e.g. unique factorisation). What's the largest class of things for which it's true?

This makes certain statements about mathematics self-fulfilling.

30% of students doing any kind of mathematics use Wolfram Alpha at least once a week, and expect saturation in a couple of years. This gives us some interesting data about what queries actually get asked.

Q Make the knowledge open source?

A Some already is: see `functions.wolfram.com`, but the amount of usage of these facts has grown exponentially since its inclusion in Wolfram Alpha. Hence I believe you need the searching/question answering framework.

Q–PL Do you see Wolfram Alpha communities?

A There are facilities to share Wolfram Alpha results via Facebook, and we see a lot of this. Textbooks are starting to include Wolfram Alpha. One thing that scares teachers is the "show steps" button on, say, integrate. Note that these are *not* the internal steps in the integrator, but are reconstructed to be human-comprehensible. There are apps for various courses. There are also 8,000 demonstrations based on CDF technology.

Q–APS Are you seeing a future in which research mathematicians becomes mere reviewers of these results?

A Compare Poincaré's response to the Hilbert programme. People make progress by automating the lower steps. Humans will still be asking "what should be computed". I *do* expect "experimental mathematics" to grow,and probably the lines to become more blurred.

Q–PDFI Will the nature of proof change?

A To back up a claim of "truth", the computer will be accepted, but the rôle of proof in 'understanding' will continue. A proof in code is more "flexible" than scratchings on a piece of paper. We do get asked questions by education ministries about mathematics curricula — what would a mathematics curriculum for the 21st century be? The questions we might ask in the 21st century are different, and the tools we have to answer them are different.

Q Apparently there are negotiations between Wolfram Research and the Ministry of Education in Estonia, but my friend couldn't find anything out from them. What's happening?

A I don't know personally, even if others in my organisation (probably Wolfram Europe) might. For general developments, see "computer based math education" summit conference in London in November (1–2: "math $\neq$ calculating") — `computerbasedmath.org`.

Q–MK Computational mathematics is mostly 19th-century as well.

A How any people should learn Bourbaki-style mathematics in middle school? probably few. "Set theory in middle school", which was the 1960s 'new mathematics' was not a success. How do we routinely automate abstract mathematics? We have had a small project in continued fractions.

Coda I'm happy for others to work on these ideas, we can't do it on our own, and I'd be happy to work with others as well, and to provide what technology we can. It is a fairly complex project (!).

Chapter 7

Calculemus — 12 July

7.1 A Perspection on Reflection — McBride

Note Wolfram's comment on the shifting balance of labour between humans and computers.

Claims that reflecting "types as data" lets one write a generic equality.

Example 3 (after Gauss) $2\sum_{i=1}^{n} i = n(n+1)$. *How can we prove this?*

- Proof by induction
- Test for $n \in \{0, 1, 2\}$

The second is in fact sufficient: has a data type poly2 of polynomials with degree, and claims that summation increases the degree. Hence

Theorem 1 (Theorem of sufficient tests) *Define* $p \approx_n q$ *meaning "equal at* $0, \ldots, n-1$*". If* p, q *have degrees* $\leq n$ *and* $p \approx_{n+1} q$*, then really* $p = q$*.*

JHD notes subsequently[1] that, to apply this here, we also need to know that δ^{-1} (provided we fix the "constant of summation", e.g. $\delta^{-1}(f)(0) = 0$) is a *total* function from polynomials to polynomials. Forward difference $\delta : \delta(f)(n) = f(n+1) - f(n)$ is the inverse of summation, and if f is a polyomial of degree k, $\delta(f)$ is a polynomial and has degree *precisely* $k-1$, i.e. $\delta : \text{poly2}(k) \to \text{poly2}(k-1)$. Since $\delta(\frac{1}{k}x^k) = x^{k-1} + g$ with $g \in \text{poly2}(k-2)$, induction on k shows that every element of $\text{poly2}(k-1)$ is an image, i.e. δ is surjective.

Alternatively, it is easy to see that δ is linear and the constant polynomials are precisely the kernel of δ, and, thus, since δ goes from a vector space of dimension k to one of dimension $k-1$ and the kernel has dimension 1, δ has got to be surjective.

In particular $\delta^{-1}(i)$ is a polynomial of degree 2, and Theorem 1 then tells us which.

[1] Thanks are due to Prof. Recio for raising this question, and providing the alternative argument.

7.1.1 A Difficulty

Suppose we have eval:$\mathbf{N} \to (\mathbf{N} \to \mathbf{N})$ so that .... Then (*sic*) evil: $n \mapsto 1 + (eval(n))(n)$ has the property that it can't be any $eval(n)$: essentially Russell's paradox[2].

7.1.2 Reinventing type theory as rationalised LISP

$TM :=$	$0 \mid sTM$	Testably different
$\mid$	TM, TM	cons cells
$\mid$	λ terms	first class objects
$\mid$	...	

We can then make tagged objects by having lists whose first element is $s^n(0)$

Types are a human way of making sense of things. Hence we get a typecheck bidirectionality. This construct has included all the basic type building we need. A type includes a first-class description of itself. Shows a candidate type of descriptions,and its interpretation in `Set`. There are two temptations: `Set` = `Desc Zero` and `Desc` I = μ don't quite coincide. Still "set of all sets" problem, hence we need a cumulative hierarchy: $\texttt{Set}^0 : \texttt{Set}^1 : \texttt{Set}^2 \ldots$; also cumulativity $\texttt{Set}^0 \subset \texttt{Set}^1 \ldots$, and polymorphism.

So far, we've only encoded data structures and types. Mutually define `Context`:`Set` and `Env`:`Context` $\to$ `Set`.

7.1.3 Summary

Europe Type theorists define inductive datatypes then notice that some have the substitution structure of syntax

America logical frameworks people define syntax but suppress inductive structure to avoid representing 'exotic' non-terms.

Me A datatype is a ...

Q—JAC What do you bring most to the party?

A People write tactics in ML to control Isabelle, which has to do type-checking: we should be able to write this *in* Isabelle. The challenge is always "how do you take a problem and regard it as data".

[2]Subsequently, Prof. McCusker informs me that this is Rice's Theorem [Ric53].

7.2 Verifying an Algorithm for Computing Discrete Vector Fields for Image Processing — Heras

We relate Algebraic Topology to Digital Images: simplicial complexes and chain complexes. $C_0 = \mathbf{Z}_2[vertices]$, $C_1 = \mathbf{Z}_2[edges]$, $C_2 = \mathbf{Z}_2[triangles]$. Why?

e are analysing biomedical images: efficiency and reliability are the goals. The bottleneck is computing the chain complexes. We would like to formalise in Coq/SSReflect a process for reducing the chain complex while preserving the homology.

A chain complex is a pair of sequences $C_* = (C_q, d_q)_{q \in \mathbf{Z}}$ where

- For every q, C_q is a $\mathbf{Z}$-module; ...

Concept of a reduction of a chain complex [RS10]. Can define a 'discrete vector field' on a chain complex. It's a collection of pairs. An admissible v.f. is one in which all paths from a cell have length bounded by a function of the cell. A critical cell is one which appears in no (admissible?) vector field.

Theorem 2 *There is a canonical reduction to a chain complex generated by the critical cells.*

Can represent admissible vector fields via matrices. Has a nice piece of Coq - four lines, but it's non-deterministic. Ned an effective method which takes a matrix M and produces an admissible vector field. We implemented in Haskell (similar-enough to Coq), used QuickCheck to test, then verified (??) with CoqSSReflect.

For a 100×300 matrix, Coq takes 12 seconds. Reduces to 5×30, which takes milliseconds, and the reduction is quick also. In real life, Coq can't cope, but the reduced images work in 25 seconds.

7.3 CDCL-Based Abstract State Transition System for Coherent Logic — Nikolic

Coherent logic is a fragment of FOL with conjunction $\Rightarrow$ disjunction of existentials. There are no function symbols of positive arity. No negation First used by Skolem. Semi-decidable. Many theorems can be formulated in CL, e.g. Euclidean Geometry, It is expressible, allows direct, readable and machine-verifiable proofs. Simple natural deduction reasoning based on forward ground reasoning. A conjecture is proved directly (no refutation, Skolemisation or clausal form), and existential quantifiers are eliminated by introducing witnesses.

- Euclid (Kordić etc.)
- many others CL provers, but generally not very efficient.

Show a (very readable) proof from ArgoCL.

On the other hand, there are CDCL provers, such as SAT and SMT solvers, which are pretty mature. Support for quantifiers depends on the theory solvers (and most only allow quantifier-free formulae). Readable proofs are often challenging (and not even considered).

We would like to combine the two! Krstić and Giel's system has certain inference rules (eight).

State = (constants, formulae: axioms+lemmas, model, C_1, C_2, l). An accepting final state is a lemma which implies the conjecture. e slights extend CL to allow universal quantifiers on the left-hand side. The rules are more complex here, partly because they are FOL not propositional, and have rules like "explain left $\exists$", "explain right $\exists$": in SAT, "explain" is essentially resolution. This is sound, and complete with an additional rule for iterative deepening. Allows first-order reasoning (rather than just ground reasoning).

7.4 Theory Presentation Combinators — Carette

Title in programme was "Theory expression combinators" — apparently the same thing. Want "efficiency", but rather efficiency of development, and for the users, not computer efficiency. We aren't (unlike section 7.1) trying to type the syntax.

Monoids, commutative monoids, additive monoids, combinations of these. "one idea — one line".

Shows the "algebra zoo", and a lage number of arrows. The middle layer is very largely combinations. We now have a decent library of theories, and an expander (users want to see flat theories), and a (mostly) complete export of the expanded versions to MMT/OpenMath and Matita (two don't: not Matita's fault) and this does type checking for us. Working on export to metaOcaml and Template Haskell.

But what does this mean? Intuitively, we are working in some category of signatures. Extend = embedding, renaming = renaming and combine = pushout. Goguen. Burstall etc. But this doesn't work well enough.

```
T1:= Theory { n: Integer}
T2 := Theory { n: Natural }
T3:= Combine T1, T2 over Empty
```

This either gives you gensym-like names or fully qualified names, both of which are user-unfriendly. We need a semantics of our syntax qua syntax. We've been worrying about the objects, but category theory tells us that the arrows are what matter. Hence focus on the intensional content.

The algebraic people look at the category of Theory Presentations. We've learnt that this is like the theory of context + substitutions, but all the arrows go "the wrong way": pullbacks, rather than pushouts etc. Note that Cartmell mods out by the names: we don't do that. A morphism $\Gamma \to \Delta$ are assignments $[y_0 \to t_0, \ldots]$. We use the category of *nominal assignments*, $\mathbf{B}$, whose only

morphisms have labels as terms. If every label occurs at most once, we have a *general extension*.

Theorem 3 *The functor* $\mathrm{cod} : \mathbf{E} \to \mathbf{B}$ *is a* fibration.

Hence the MathScheme Theory Presentation Language. `combine` used to have `over`, which was the base of the pushout. No longer needed, since we naturally combine A_1 and A_2 over their natural intersection, except that the user specifies enough renamings to avoid confusion.

In the algebra zoo, there are sub-diagrams which look similar. Therefore functorial semantics should lead to diagram-level constructions, We know how to define this, but need to port the library to the new semantics. All this structure in mathematics should be leveraged to make the builder's life easier. We have learnt to follow the *right* mathematics, rather what we intuit should be right.

7.5 Formalizing Frankl's Conjecture: FC-families — Marić

Example: 4-colour, first unverified computer calculations [AH76], then formalised in Coq [Gon08].

Conjecture 1 (Frankl) *If a family of sets in union-closed, then an element has to occur in at least half of the sets.*

Still open. Proved for at most 36 sets, possibly 40, and $|U| \leq 12$. Note that "brute force" fails: $|U| = 12$ implies considering $2^{2^{12}} = 2^{4096}$ cases. Very complex Java programs. How can this proof be trusted? New versions of the program generate proof traces that can be independently verified.

The key idea is weighted sums of occurrences of elements in sets.

Lemma 1 *F is Frankl iff there is a weight function ...*

Definition 2 *F_c is an FC (Frankl-Complete) family if for all union-closed families F containing F_c*

They had a (7,4,3) FC-Family which was part of the proof of the 12-F proof.

Q Have you tried relating back to the Java?

A Based in the Java, but 200 lines of Java became 10 lines of Isabelle. But I haven't been directly back.

Q Did you find bugs in the Java?

A No!

7.6 Teaser Talks

Also a repeat of section 4.1

7.6.1 New Developments in Parsing Mizar — Alama

It used to be impossible to parse Mizar: now I can, and have an `http` interface.

7.6.2 Isabelle/jEdit — A prover IDE within the PIDE framework —- Wenzel

had a slide of the Tower of Babel when man was still optimistic — this is one of these!"

7.6.3 On Formal Specification of Maple Programs — Taimor Khan

Particular attention to typing in this untyped (at least statically) system. Tested on difference/differential package.

Chapter 8

13 July

8.1 Increasingly correct scientific programming — Ionescu

Speaker from PIK, the Potsdam Institute for Climate Impact Research. "The rôle of the Institute is to slam the hard facts on the table" — Director.

Notes the Wikipedia article, points out that it deals with objects which are continuous rather than discrete.

Definition 3 *An allocation is* **Pareto efficient** *if there is no feasible allocation that dominates it strictly everywhere.*

The usual utility is Cobb–Douglas $u(x_1, x_2) = x_1^{\alpha_1} x_2^{1-\alpha_2}$ etc. Normally take logs to make linear. If the have prices, then we have Walrasian equilibria, and

Theorem 4 (First welfare theorem) *Walrasian equilibria are Pareto efficient.*

- Where do prices come from?
- Which equilibria are selected?

The Gintis model [Gin06, Gin07] is used at PIK in the Lagom project. We (and Gothenburg, see [EM09, section 3.1.1]) have had problems reproducing this. Both groups discovered a serious bug in the original implementation of the model.

$$\frac{\sum_j p_{ij} x_{ij}}{\sum_j p_{ij} o_j}$$

was implemented as

$$\frac{\sum_j p_{ij} x_{ij}}{\sum_j p_{ij} x_{ij}} \qquad (= 1)$$

which accounted for the very rapid convergence Lagom had observed.

There were other problems with this, and the "specifications" were not actually sufficient.

GEM-E3 is an applied general equilibrium model. Used in several DG of the EU. The general specification of a household is

$$\max U(q(t)) = \int_{t=o}^{\infty} e^{-deltat} u(q()) dt$$

however, this isn't in the code, but the result of several steps of mathematics. Hence the speaker feels that there is an enormous gap between the mathematical specification and the code.

[Martin-Löf1984]: "The correctness of a program is proved at the same time as the program is synthesised". We formulated different equilibria in Agda and Idris. We could write Walrasian equilibria in Idris. The good news is that we can do this, but the bad news is that we can express the hypotheses, and the rules, but not the connection between them. Economic theory is mostly non-constructive. Most modellers are not numerical analysts, and want to use external routines. There is no usable numerical library for the constructive reals.

However, the fact that we can't do it all doesn't mean we should give up. Having specifications is better than not. Having specifications that can be partially machine-checked is better than ones that can't. Having classical proofs of correctness is better than no proof of correctness.

For example, we now have a specification for maximising utility over a finite set. Tail-recursive implementation, but problems with dependent types — not good for economists. par How do we specify that the outputs of a program $X \to Y$ have to be in relation R with inputs? Two styles.

Nordström $f : (x : X) \to (y : Y * *R(x, y))$

Thompson $(f : X \to Y * *(x : X) \ldots)$

As another example, a major model ReMIND-R has the intertemporal social welfare equation. The discretisation of this can easily have an off-by-one error. Showed various code examples which don't have this problem. Shows an Agda-like (Agda/ Idris?) model for the deterministic case, and then shows the general stochastic case, when `Float` becomes some kind of distribution, and that's about the only change.

Note that current practice is to use an external optimiser and trust the results, and at least his code has a formal specification of that fact. Can also pass the (closed form of the) function to interval arithmetic functions, and have a *classical* proof of correctness.

Future fork: need to specify more common external routines, such as interpolation. A technical problem is that the notation for dependent types leads to very long names, hence we need a `where` clause.

Q "Constructive Type Theory" is about the hardest there is — why not use something easier?

A We do have executable specification, which matters.

Q–JC Another approach is to transform the model mathematically to the point where one can generate code from it.

A I agree that there is such code in Mathematica, but economists don't use Mathematica, but rather a mixture of mathematics, data, code etc. The country-wide correlation between PC members and submissions was weak (possibly negative).

8.2 Speeding-up Cylindrical Algebraic Decomposition by Gröbner Bases — Davenport

JHD presented [WBD12]. See slides at `http://staff.bath.ac.uk/masjhd/Slides/CICM2012.pdf`.

Q What happens if you compute the real radical of the Gröbner base?

A We haven't tried that, or even computing the radical as such. Good question.

8.3 Towards Formal Specification and Verification of Maple Programs — Taimoor Khan

Noted that Gauss [Mon93], an attempt to put strong typing in Maple, is now "Maple Domains".

We are attempting to specify and verify untyped computer algebra. This could be Mathematica or Maple: our specific case is Dönch's DifferenceDifferential package in Maple. Our method is static analysis: type consistency and preconditioning.

We actually use MiniMaple, which has all the syntax, but not all the operators. Global variables in Maple are always untyped, and may be reassigned at will. Local variables can only be specialised.

Note that run-time type tests are used to affect control flow. He showed some MiniMaple with

8.4 Calculemus Business Meeting

1. JHD ended up as scribe.
2. GdR reported as Programme Chair. There was a deadline extension of one week, 8 submissions, a short (< 1 week) rebuttal period, 6 acceptances (one after shepherding).
3. The Calculemus Track Chair for 2013 will be Wlfgang Windsteiger.
4. Trustees. RR is the CICM delegate. His term on the Trustees expires, but he will stay on the rustees mailing list to keep communications open.

5. There are many retiring trustees, so we wil extend Paul Jackson for one year to improve rotation. We still

6. JC has been Calculemus Secretary for ten years. GdR has been appointed secretary, and JC will arrange a handover.

7. MW/GdR nominated JHD for Trustee. WMF/MW nominated AA. GdR/MW nominated JvdH, but it had to be confirmed that he was willing.

8. It was noted that the number of submisisons was low. Many people thought this was due to a lack of *targeted* advertising by people known to the target community, e.g. MW to the Isabelle community. RR reported that Paris 2010 had also sent personal invitations, which many people thought was excellent, if labour-intensive.

9. JHD will advertise Calculemus 2013 at ISSAC 2012, which is 21–24 July. In 2013, ISSAC is before CICM.

8.5 Reasoning on Schemata of Formulae — Peltier

We have parameterised formulae in some base language. These are ubiquitous in mathematics and computer science. Essentially they are the base language plus induction. The first idea would be to encode them into expressive languages (logics with inductive definitions, etc), but we don't have efficient proof procedures here, and have lost all the properties (decidability etc.) of the base language.

Therefore we intend to extend the basle logic $\mathcal{L}$ instead. We have an algebraic structure $\mathcal{A}$ (e.g. $\mathbf{N}$). We add two new sets of symbols. A set of parameters (of a sort in $\mathcal{A}$) and a set of indexed formulae. Each defined symbol ϕ is mapped to a set of rewrite rules of the form $\phi_{f(i_1,\ldots,i_k)} \to F$. We want each ϕ_t to have a unique normal form, and if t is ground, we have a ground term.

If $\mathcal{L}$ is decidable, we have semi-decidability by enumeration. But we'd rather reduce to a finite set of base formulae. We use a tableaux-based procedure (proof trees). Hence $\frac{\phi_t}{F_\sigma}$ if $\pi_s \to F$ and .... The goal is to decompose into conjunctions of base formulae and ϕ_a (or $\neg\phi_a$) where a is a parameter. Quantifiers are left to $\mathcal{L}$.

This procedure is sound, in the sense that if all branches can be closed then the root formula is unsatisfiable. Complete for model generation: if the root formula is satisfiable, then there is a branch such that .... However, almost never terminates. We can help termination via a loop detection rule. Has a good one for monadic signatures, since the number of parameters does not increase, but there are problems with non-monadic ones.

Theorem 5 *If F is irreducible and a is maximal, then $a = f(b_1, ldots, b_n) \wedge F$ is sat-equivalent to F.*

Note that n must be instantiated during the proof search. Hence our loop detection rule , instead of detecting $\psi, \ldots \phi$ where ϕ is equal to ψ up to renaming, we need to have $\psi \subset \phi$ up to renaming.

This applies to many algebraic structures (trees, DAGs etc.), but needs a depth-first strategy to ensure termination in the case of multiple parameters and/or non-monadic signatures.

8.6 Real Algebraic Strategies for MetiTarski Proofs – Passmore

MetiTarski [AP10] is a theorem prover for real valued special functions. We know the polynomial theory is decidable — what about the extended. But since the first order theory of the rationals is not decidable, we can't just add sin, so we don't always have decidability. Hence we employ "systematic heuristics".

Examples: $0 \leq x \leq 1.46 \times 10^{-6} \Rightarrow$ inequality; $0 \leq x \wedge 0 \leq y \Rightarrow$ inequality.

The purely existential RCF is singly-exponential (in theory) unlike the full theory, which is doubly-exponential [DH88].

- Transcendental functions are replaced by rational function upper and lower bounds

! Building up good families of transcendental function bounds takes a lot of work, just assumed as given in this talk.

- Eventually end up with purely polynomial (algebraic) inequality subproblems — these can be handled by RCF

- All done systematically through extensions to a superposition (modern resolution) calculus (and prover) — Metis. Clauses are disjunctions of literals. Trying to contradict the negation of the goal. each step combined two clauses and yields new clauses which are simplified.

New Algebraic literal deletion (our main focus); algebraic redundancy checking (subsumption); formula normalisation and simplification; modified Knuth–Bendix ordering; case splitting etc.

MetiTarski keeps a data structure of all ground algebraic clauses: an algebraic context. Any literal inconsistent with the context can be deleted. MetiTarski can generate tens of thousands of calls to RCF decision. Note that number of variables never grows beyond the input. These RCF decisions only contribute when they *refute*: The best practical RCF (they have used QEPCAD-B [Bro03] and Mathematica [Str08]) are still CAD-based.

8.6.1 Motivating Hypotheses

- By studying the RCF problems, we can generate specialised solvers

- This leads to better RCF for our purpose.
- Model sharing: past models for SAT exRCF subproblems allow us to satisfy (hence not study) subsequent ones
- Polynomial factorisation generally doesn't happen (for our problems), but this is a waste of time, and should be disabled where possible (Mathematica's PCAD can't!).

`max-sin-2`: $\forall xin(-8,5)\max(\sin(x),\sin(x+4),\cos(x)) > 0$. 600 lines, with 62 RCF instances used, but 2776 RCF's were generated. 2221 of these *are* satisfiable (hence useless), but took over 70% of the time. Max. total degree 24, average 3.53, max. coefficient bits 102, average 21.03. 37 rational models satisfy 2172 of the 2221 (the rest require a unique model). Have a prioritised data structure of most successful past problems

Can use QEPCAD-B, Mathematics (must faster for 3–4 variables in Mathematica 8) and Z3 (de Doura, Microsoft Research). Z3 plus above proves many more results within 30 seconds. This finds 120 theorems substantially faster than any other, whereas other methods are 30 or less.

8.6.2 Conclusions

- Expensive decision procedures shouldn't be seen as black boxes
- Hence they shouldn't be written as such

Q–DJW Do you know which models are the most useful?

A Not really. We always try to lift over full-dimensional cells where possible.

Q–JAC Which came first, the problem or the solution?

A Our general plan (grand) is to find specialised RCF decision procedures.

Q–JHD Yes, the tuning of factorisers is biased to cases that do factorise. See [Dav87]

8.7 MathWebSearch 0.5 Scaling an Open Formula Search Engine — Kohlhase

Also talking about efficiency. Note that Mizar knows what the universal variables are, we can instantiate these during theorem search.

1. Crawl the web for formulae (in 2005 we found 13 Content MathML in three months)
2. Represent them as first-order terms
3. index then for instantiation

4. Find a query language — hard.

Then came LaTeXML. Can use this for partially-remembered formulae. Also to find applicable theorems — showed an instance finding Hölder's Theorem. Note that both databases and ATP use term indexing in general, but for ATP the index is relatively large. trees in corpora written by humans, aren't deep (maximum size 50) hence time is essentially constant. In Mizar we see a great deal of sharing, but the arXiv seems essentially linear, and would need 200GB to index, but this should be RAM! Machines this large aren't common, hence we need a distributed solution. Top-level hashing isn't enough, as the tree is very unbalanced.

Keep the tree in 2^{31}-byte chunks, and when it's full, search for a fringe that leaves $\approx 60\%$ and split here. Currently load balancing is pretty dumb, and fault tolerance is negligeable.

Can build a LaTeXML editor which sends off the formula being typed every 100ms.

8.8 A System for Axiomatic Programming — Dos Reis

"Practical type checkers from mainstream programming languages are among the most successful and widely used theorem provers today". These are mostly Hindley–Milner [Mil78]: usually phrased as a system with 5 rules (there are 4-rule descriptions). Unintended consequences: emphasis on bottom-up views and initial algebras. Gazilions of explicit (type) parameters.

8.8.1 An Alternative View

Think of program behaviour as a Geometric Object. Parametric Equations or Implicit Equations? In his case, implicit functions. This work is based on the provable correct code of C++[1]. The type systems in continuously extensible with user-defined axions and properties; implicit relationships between entities using equations.

Shows fairly revolting piece of GCC internals as example of what *not* to do. There are many parameters, but *not* independent. The top-down proposal would be "how do I use this thing", rather than "what is this thing".

This gives us a final coalgebra point of view. Example (three-line definition): `concept HomogeneousFunction`. Specialises to `concept Operation`, thence `concept BinaryOperation`. Need congruence-based type checking, and Skolemisation.

Would like to integrate with existing proof assistants (but without turning off real programmers). Wants a better understanding of axioms and runtime assertions (programming by contract).

[1] Apparently most of the Higgs boson work was in C++.

Q To what extent is this related to our obsession with witnesses?

A I like constructivism, but it may go too far.

Q This reminds me of Coq's 'canonical structure'. But I'm not convinced by the implementation.

A–JC But Gonthier is the only person who can drive it!

8.9 Management of Change in Declarative Languages — Iancu

Example is LATIN atlas [CHK+11]: highly modular, but stull difficult to keep an overview. "Which declarations does a symbol depend on?" etc. Uses MMT: theories contain constant declarations. Our change language has add and delete for constants, update components and renames constants. Notes that a change can have multiple non-trivial representations. How do these changes affect the theory graph?

Change Detection $\mathcal{G}-\mathcal{G}'$ gives add/delete/updates, then refine to use rename where applicable. This leads to the concept of impact propagation: renames are done, and others wrapped with OpenMath `error` terms. Rename is good for users (simplicity) but adds effort. Therefore the 'rename' refinement mentioned above should become user-definable, rather than part of the kernel of change management.

Q–JAC How does this relate to the world of version control?

A Haven't looked at that yet. Not immediately relevant.

8.10 A Combinator Language for Theorem Discovery — Scott

Had a messy graph of dependencies, and too much combinatorial reasoning. Wanted to express things in a data flow language. Used "Monad +Monoid".

Represent the search for theorems as an infinite stream of collections in a commutative monoid αm. Showed a prettified version of the OCAML. This leads to a variety of tree-merging operations. First cut was very slow, since he wasn't using idioms, for which the (OCAML) compiler has much better type inference — different in Haskell (but not, JHD understood, better).

He has an ML library for the idioms and transformers, and a language for guiding discovery. Needed to do substantial work to make ML's lazy list library truly lazy — concatenation was eager (!). Doesn't have efficient subsumption (and termination of subsumed discoverers).

Q–JC Have you used `PA-Monad`?

A That's OCAML 4, not 5.

JC Should port easily — ask a developer, e.g. me!

Bibliography

[ADL12] D Aspinall, E. Denney, and C. Lüth. Querying Proofs. In *Proceedings LPAR*, 2012.

[AH76] K.I. Appel and W. Haken. Every Planar Map is Four-Colorable. *Bull. A.M.S.*, 82:711–712, 1976.

[AP10] B. Akbarpour and L.C. Paulson. MetiTarski: An Automatic Theorem Prover for Real-Valued Special Functions. *J. Automated Reasoning*, 44:175–205, 2010.

[AS64] M. Abramowitz and I. Stegun. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing. *US Government Printing Office*, 1964.

[Bau99] J. Baur. Syntax und Semantik mathematischer Texte. Master's thesis, (Diploma) Diplomarbeit Uiversität Saarbrücken, 1999.

[Bro03] C.W. Brown. QEPCAD B: a program for computing with semi-algebraic sets using CADs. *ACM SIGSAM Bulletin 4*, 37:97–108, 2003.

[BSSS11] J.B. Baker, A.P. Sexton, V. Sorge, and M. Suzuki. Comparing Approaches to Mathematical Document Analysis from PDF. In *Proceedings 2011 International Conference on Document Analysis and Recognition*, pages 463–467, 2011.

[CDM+10] C. Chen, J.H. Davenport, J.P. May, M. Moreno Maza, B. Xia, and R. Xiao. Triangular Decomposition of Semi-algebraic Systems. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 187–194, 2010.

[CHK+11] M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, and F. Rabe. The LATIN Logic Atlas. In J.H.Davenport *et al.*, editor, *Proceedings CICM 2011*, pages 289–291, 2011.

[CMMXY09] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing Cylindrical Algebraic Decomposition via Triangular Decomposition. In J. May, editor, *Proceedings ISSAC 2009*, pages 95–102, 2009.

[Col75] G.E. Collins. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

[Dav87] J.H. Davenport. Looking at a set of equations. Technical Report 87-06, 1987.

[DH88] J.H. Davenport and J. Heintz. Real Quantifier Elimination is Doubly Exponential. *J. Symbolic Comp.*, 5:29–35, 1988.

[EM09] P. Evensen and M. Märdin. An Extensible and Scalable Agent-Based Simulation of Barter Economics. Master's thesis, Gothenburg University, 2009.

[Gin06] H. Gintis. The Emergence of a Price System from Decentralized Bilateral Exchange. *The B.E. Journal of Theoretical Economics (Contributions) pp*, 6, 2006.

[Gin07] H. Gintis. The Dynamics of General Equilibrium. *Economic Journal*, 117:1280–1309, 2007.

[Gon08] G. Gonthier. Formal Proof — The Four-Color Theorem. *Notices A.M.S.*, 55:1382–1393, 2008.

[Har98] J. Harrison. Proof style. In E. Giménez and C. Paulin-Mohring, editors, *Proceedings Types for Proofs and Programs*, pages 154–172, 1998.

[HW79] G.H. Hardy and E.M. Wright. An Introduction to the Theory of Numbers (5th. ed.). *Clarendon Press*, 1979.

[JPS08] G. Jeronimo, D. Perrucci, and J. Sabia. On sign conditions over real multivariate polynomials. `http://arxiv.org/abs/0801.0586`, 2008.

[KMW04] F. Kamareddine, M. Maarek, and J.B. Wells. MathLang: Experience-driven Development of a New Mathematical Language. *Electronic Notes in Theoretical Computer Science*, 93:138–160, 2004.

[Koh01] M. Kohlhase. OMDOC: Towards an Internet Standard for the Administration, Distribution, and Teaching of Mathematical Knowledge. In J.A. Campbell and E. Roanes-Lozano, editors, *Proceedings Artificial Intelligence and Symbolic Computation*, pages 32–52, 2001.

[Koh07] M. Kohlhase. OMDoc — An Open Markup Format for Mathematical Documents. *Springer Lecture Notes in Artificial Intellligence 4180*, 2007.

[Mil78] R. Milner. A Theory of Type Polymorphism in Programming. *J. Comp. System Sci.*, 17:348–375, 1978.

[Mon93] M.B. Monagan. Gauss: A Parameterized Domain of Computation System with Support for Signature Functions. In A. Miola, editor, *Proceedings DISCO '93*, pages 81–94, 1993.

[MT08] B. Mourrain and Ph. Trébuchet. Stable normal forms for polynomial system solving. `http://arxiv.org/abs/0812.0067`, 2008.

[Ric53] H.G. Rice. Classes of Recursively Enumerable Sets and Their Decision Problems. *Trans. Amer. Math. Soc.*, 74:358–366, 1953.

[RS10] A. Romero and F. Sergeraert. Discrete Vector Fields and Fundamental Algebraic Topology. `http://arxiv.org/abs/1005.5865`, 2010.

[RT03] P. Rudnicki and A. Trybulec. On the Integrity of a Repository of Formalised Mathematics. In *Proceedings Mathematical Knowledge Management 2003*, pages 162–174, 2003.

[Str08] A.W. Strzeboński. Real root isolation for exp-log functions. In *Proceedings ISSAC 2008*, pages 303–314, 2008.

[SZ10] Z.-W. Sun and D. Zagier. On a curious property of Bell numbers. `http://arxiv.org/abs/1008.1573`, 2010.

[WAGD11] I. Whiteside, D. Aspinall, G. Grov, and L. Dixon. Towards Formal Proof Script Refactoring. In J.H.Davenport *et al.*, editor, *Proceedings CICM 2011*, pages 260–275, 2011.

[Wat08] S.M. Watt. Mathematical Document Classification via Symbol Frequency Analysis. In S. Autexier *et al.*, editor, *Proceedings AISC/Calculemus/MKM 2008*, pages 29–40, 2008.

[WBD12] D.J. Wilson, R.J. Bradford, and J.H. Davenport. Speeding up Cylindrical Algebraic Decomposition by Gröbner Bases. In J. Deuring *et al.*, editor, *Proceedings CICM 2012*, pages 279–293, 2012.

[Wil12] D.J. Wilson. Polynomial System Example Bank. `http://opus.bath.ac.uk/29503`, 2012.

[Zin04] C.W. Zinn. *Understanding Informal Mathematical Discourse.* PhD thesis, University of Erlangen-Nürnberg, 2004.