CICM 2021

Notes by J.H.Davenport

26–30 July 2021

Contents

Chapter 1

27 July

1.1 Theresa Kruse

Coming from the NLP area. Common tecniques: pattern search, frequency count, frequency comparison, machine learning. A colleague [Schäferetal2020]. Had 882910 tokens from books and journals. Had three raters. Initially 40% had disagreements, down to 17% after discussion.

Three methods produces 5500 terms, 4000 distinct. Only 2% found by all three.

1.1.1

Q Are the two dictionaries connected?

A Yes indeed.

Q "Field"?

A Graph theory is well-behaved that way, but I have several examples. Also in German there are synonyms, e.g. for "adjacent".

Chapter 2

28 July 2021

2.1 Beautiful Formalizations in Isabelle/Naproche

By Peter Koepke (speaker, supervisor of the students), Adrian De Lon, Anton Lorenzen, Adrian Marti, Marcel Schütz and Erik Sturzenhecker.

Example 1 (Euclid) *$\#Primes = \infty$.*

"Naproche" = "Natural Language Proof Checker". Now part (Thanks Makarius) of Isabelle 2021. We have a ForTheL controlled language. This is the source code for both Naproche and LaTeX. Only text in `\begin{forthel}...` are Naproche-checked, hence allowed literate programming.

Example 2 (Cantor: `cantor.ftl.text`) *One double-column slide of readable text. Includes three definitions and four axioms (e.g. powerset of a set is a set).*

Example 3 (König's Theorem)

2.1.1

Q Struggle on longer proofs?

A We have seen e-prover struggle. We have done 50 pages of Rubin's Analysis. Would probably nede better sledgehammer.

2.2 Formalizing Axiomatic Systems for Propositional Logic in Isabelle/HOL

Asta Halkjær From, Agnes Moesgård Eschen and Jørgen Villadsen.

Various formulations: Russell–Bernays, [WR10], Łukasiewicz etc.

2.3 Cimatti: Logic at Work

Design Challenge: systems becomes more complex. E.g. railway interlocking. We want formal methods in the hands of the designer, to answer a formal version of "does the design satisfy the requirements". $M \models \phi$. Model checking looks for a trace in M that violates ϕ. Explicit model checking is great until the state space explodes. Hence symbolic verification. Initially BDD verification, then SAT-verification. This requires finite domains, hence SMT. SAT-solver passes a set ϕ to T-solver, which says "yes", or "no" ideally over $\phi' \subset \phi$. UF by Congruence Closure, LRA by simplex etc.

Hence VMT is expressing verification in the language of SMT. Bounded Model Checking is incomplete, but good for cases where there is an unsafe state. The alternative is "no error path of length $< k$". Alternative is K-induction. Abstraction is more than replacing SAT with SMT. This gives us CEGAR.

Linear has had a great deal of progress over recent years,but not nonlinear. Hence interest in SMT via incremental linearization. Apply this to NRA, NTA (transcendentals) and NIA. These are EXP-hard, of even undecidable. This is an over-approximation, so UNSAT is correct. If we get SAT, we check whether it's spurious. If genuine, we have found SAT, else add linear lemmas, such as tangent plane lemmas. Harder for NTA as some of[1] the additional lemmas may not be in terms of the base field: use Taylor series. Periodicity lemmas for sin etc.

Use MathSAT/GMP. On UNSAT, we solve more than any other (CVC4, Z3, Yices etc.). VBS is much better, though, which shows there's a great deal of complementarity.

2.3.1 Q&A

Q How do you verify the real system rather than the abstraction.

A1 (Moore) You can verify an abstraction of a microprocessor, but if you hit the microprocessor with a hammer, it will no longer function.

A2 Railway systems. We are analysing the abstractions of the circuits:need to hope this is faithful. Can check with measurements. But there is a always gap.

2.4 Inductive Benchmarks for Automated Reasoning

Márton Hajdu, Petra Hozzová, Laura Kovács, Johannes Schoisswohl and Andrei Voronkov.

Mnay inductive system now, such as Zeno, CVC4, improvements to Vampire. $>$ 3500 benchmarks, some handcrafted. SMT-LIB and many other formats.

[1]Many basic lemmas, e.g. $\exp(x) > 0$ are well-posed.

Also include inductive datatypes, Nat, List(Nat). Tree(Nat). Examples such as associativity of multiplication. Also examples with existential quantifiers.

We had 120 integer benchmarks based on programming examples.

Chapter 3

July 29

3.1 Automated Generation of Exam Sheets for Automated Deduction

By: Petra Hozzová, Laura Kovács and Jakob Rath (TU Vienna). Setting is Covid-19 conversion to online examinations, so need to give students different problems of similar difficulty.

Inplemented in Haskell, inspired by QuickCheck [CH00].

Example 4 (SAT) *Constraints during generation: #connectives, each of* $\leftrightarrow$, $\rightarrow$, $\neg$ *appears at least once; no trivialities like* $p \wedge \neg p$ *etc., which meant we needed a backtracking generator*

Example 5 (First order ATP) *Same prinicples. Fix a signature, three random literals, and then a substitution.*

Example 6 (Random variation of templates) *Used for SMT. Example shown required arithmetic, arrays and UIF. Random generation might too easily provide trivia, so we provide a template and tweak the integer constants, so the reasoning task is the same. Same technique for ground superposition, where the template goes through larger clauses which are nevertheless smaller w.r.t.* $\succ$.

250–300 lines of Haskell/Python.

Students performance similar to previous years, and the end-of-year survey had good satisfaction.

3.1.1 Future Directions

1. Encode constraints via SMT, rather than backtackig explicitly
2. More problems.
3. More support for grading.

3.1.2 Q&A

Q–MK Using these in exercises, as well as exam?

A Not yet.

Q Liaison with other universities.

A–LK Some liaison with Manchester. But this was an emergency solution, and we need to do more.

Q–PL Fairness?

A Manual currently.

Q–FR Will you continue this way?

A We would need a larger pool of problems.

A–LK Out of our hands. Also, these questions are only part: also need hard mathematical problems.

3.2 10 Years Later: The Mathematics Subject Classification and Linked Open Data

Susanne Arndt, Patrick Ion, Mila Runnwerth, Moritz Schubotz and Olaf Teschke. See [AIR+21].

Note that MSC is 30 years old. But there's no real home for MSC, and MSC2000 was almost lost — see a Spanish website. MSC 2020 has 63 top level, 1037 second level and 5000+ 3rd level. New this year is a meta-class on research data and software.

There was a SKOSified version of MSC 2010. Was SKOS Core + `mscvocab` extension, but this made Protégé very unhappy. There were a lot of lessons learned for SKOS version of MSC2020.

Note that the Unif classification tool requires a SKOS model. Flattened all the maths (was mixed MathML/LaTeX) to MathML. Added a licence, but still in search of a proper home.

3.2.1 Q&A

Q–JHD Permanent home: how about IMU.

A Hadn't tried this — thanks.

Q The licence is NC.

A Yes, the SKOS version has the same as the base MSC2020. Admitted a problem.

3.3 WebMIaS on Docker: Deploying Math-Aware Search in a Single Line of Code

Dávid Lupták, Vít Novotný, Michal Štefánik and Petr Sojka.

Conventional searchers can't handle $E = mc^2$ etc. LaTeX-search in Springer,

Chapter 4

30 July: OpenMath

4.1 JHD

See [Dav21] and slides at `https://staff.bath.ac.uk/masjhd/Slides/JHDatOpenmath2021.pdf`.

4.2 Ken Wenzel: OpenMath-RDF

We often need mathematical formulae for engineering use cases. Linked Data and Semantic Web. Four basic rules:

1. URIs
2. http URIs
3. Provide useful information using Open Standards
4. Always provide information. Vocabulary can be extended with ontologies like RDF Schema, OWL etc.

Then we can run queries with SPARQL etc. We could annotate with provenance information. Could link outside for well-defined constants etc. OpenMath might be useful for constraint checking: see screenshot. Conversely we could use standard RDF tools and databases to edit OpenMath objects and CDs.

But this means we need an OWL ontology for OpenMath objects. Literals rely on XML schems datatypes, which causes problems with hexadecimal encodings. His XML-RDF mapping fits on one slide. Examples of SPARQL queries.

Demonstration with POPCORN-LD (linked Data). `https://github.org/numerateweb`. Also pointed out the RDF CD.

4.2.1

Q Are hex literals the only obstacle?

A Yes, and there are solutions via subclassing the literals.

MK Literal subclassing sounds like to route.

Q Embedded RDF?

A Yes, compatible.

4.3 Ken Wenzel: Pattern Matching for Mathematical Expressins with OM

MathWebSearch and SEWELIS (etc.) use pattern variables. MathWebSearch is MathML+proprietary extensions. SEWELIS introduced SPARQL+, with mathematical constructors, and traversal operators. Out CD has nine symbols: `all_of` etc. Note the `bind` construct analogous to `mq:generic` in MWS.

Example 7 *Find integrals containing quadratic functions at any nesting level. Uses* `calculis1:int`

4.3.1 Q&A

Q–MK Adopt this in OM?

A Yes, can you help?

MK Tom Weising can.

4.4 Discussion

Encodings TW said that it would be nice to have a list of these and systems supported.

MK The systems page on `openmath.org` is out-of-date. But a volunteer is needed (speaker sort-of did).

MK What about KW's idea of this as an encoding.

JHD Popcorn showed that a new encoding can be valuable.

MK Why doesn't KW convert his paper into a specification, and then reviewed (+TW, LH) and action it.

Q Who looks at pull requests for CDs?

A–MK JHD

JHD I'm not seeing them! Apparently there are three at `https://github.com/OpenMath/CDs/pulls`.

Chapter 5

30 July: MathUI

5.1 Story-based content presentation in 𝕄ATh: Michael Junk

Homo Sapiens is a story-telling animal. Recipes are a kind of story. But "take two large eggs", doesn't say "take two large eggs, arbitrary but fixed". Why do we?

5.1.1 Q&A

Q–MK I know you teach this, but have you evaluated this?

A Not formally.

5.2 John Schihada

We wanted to disentangle Knowledge Management from Games Design. MMT is the MKM system we choose. So the knowledge is in modular theories.

In the scroll formuation, we have added dynamic descriptions. Example from the tree-measuring game. Used JSON as exchange medium.

5.3 PaiDavenport

See [PD21] and slides at `https://staff.bath.ac.uk/masjhd/Slides/MathUI21Slides.pdf`.

5.3.1 QA

Q–PL Why so slow?

A Generating the images from LaTeX.

Q–MK You need arrow keys

A Yes

Q–AK Educationally, would be good to have overview of the methods first. Also "side-by-side" isn't really that.

A Good points, There is an "instruction" button.

Q–PL Could make a web page that aligns the two.

MK Bottle framework: `https://bottlepy.org/docs/dev/`.

A Thanks

Bibliography

[AIR+21] S. Arndt, P. Ion, M. Runnwerth, M. Schubotz, and O. Teschke. 10 Years Later: The Mathematics Subject Classification and Linked Open Data. https://arxiv.org/abs/2107.13877, 2021.

[CH00] K. Claessen and J. Hughes. QuickCheck: a lightweight tool for random testing of Haskell programs. In *Proceedings of the fifth ACM SIGPLAN international conference on Functional programming*, pages 268–279, 2000.

[Dav21] J.H. Davenport. Digital Collections of Examples in Mathematical Sciences. https://arxiv.org/abs/2107.12908, 2021.

[PD21] A. Pai and J.H. Davenport. ATLAS: Interactive and Educational Linear Algebra System Containing Non-Standard Methods. https://arxiv.org/abs/2107.13942, 2021.

[WR10] A.N. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1910.