Big Proof
(Isaac Newton Institute)

Notes by JHD

10-14 July 2017

Contents

Chapter 1

10 July 2017

Introduction. This is one of the most oversubscribed workshops ever. Don't forget to credit INI and EPSRC Grant reference.

1.1 Big Proof: Hales

Lovely Cartoon. I was a 1980 Tripos student. Tribute to John Conway (first taught him Kepler: August 22, 1982) and John Thompson.

Buckminster-Fuller claimed to have a proof of Kepler in 1970s. Now known to be "nonsense". 1990s Berkeley professor (Wu-Yi Hsiang) claimed a proof, and TH had a paper of counterexamples in *Math. Intell.*, which uses this as its advertisement over "Not afiraid of Controversy". Hence worry about our making an announcement, as no-one had checked our proof.

Referee's report: divied paper into three Phases: "would preer to have Phases 2,3 checked before working on Phase 1" (theory). But this meant that no work had been done fro four years. Hence (six months) I announced the start of Flyspeck. Real aim is to eliminate referees.

> I have completed the translation of your Java to an executable Isabelle specification. A prof of correctness of this would be a proof of ....

- Don't be afraid of controversy.
- Don't be afraid of partial results.
- ABC conjecture (five years so far)
- Poincaré (three years)
- Hironaka's proof of resolution in positive characteristic (still not clear)
- HMAC controversy

- Sphere pakcing in 8D (one massive LP problem)

Problem 1 *Third part of Hilbert's 18th asks whether anisohedral tilings exist. Reinhardt constructed a 3D one in 1928. Also found five families of pentagonal tiles, but unable to prove these were complete Kershner fifty years later found three more families, and claims an unpublished proof. Gardner popularised this. James found a further in 1975 By now we have 15 families. Michael Rao has a computer-assisted proof: 5KLOC C++.*

See [Lak76, P45]: both my bibles as an undergraduate.

Problem 2 (Honeycomb) *Optimal partition of plane. First recorded in Varro (36BC), but also attributed to Pappus of ALexandria. Proved by Hales.*

Problem 3 (Kelvin, 1897) *What partition of 3D into equal volumes takes least surface area. Kelvin proposed truncated octahedra. But soap bubble experiments (Plateau) show that this can't be the right answer. Jean Taylor proved Plateau's conditions in 1970s. First description of Kelvin foam was 1990s, but one author deid next year, and proof (995 pages) not published. 1993 counter-example by Weaire–Phelan foam. But of course the conjecture hasn't been stated precisely, and the counter-example was only constructed numerically, so there's no rigorous proof of its existence.*

Now interested is a Formal Abstracts initiative. Well aware of dangers of not having proofs. Took 40 iterations to get a definition right in Kepler. But we should do it anyway. I'd like to move beyond the areas normally convered by formalization: Automorphic representation theory.

1.1.1 Clay Problems

BSD Needs analytic continuation of L-functions. Much more formal after Wiles. But needs a deept theorem to even state/

Poincaré 3-sphere and manifolds.

Hodge Needs de Rham cohomology and complex varieties.

Navier–Stokes Easy statement about PDEs.

PvNP Easy statement about Turing Machines

Riemann Hypothesis We have ζ in HOL-Light,but analytic continuation.

Yang–Mills mass gap Probably can't be stated. Quoted the "official" statement from [[]p. 6]JaffeWitten. Phrases like "predictions as described in textbooks" with no more details.

1.1.2 Q&A

Vienna YM can perhaps be made precise: there's a lot of maths underpinning the vague statements in physics testbooks.

A Doug Arnold's name keeps coming up here.

Hoare Need a theory of bubbles?

A Yes

? Should we try to formalise everything going forward, or need we look back?

A We aren't going to get everything.

Nipkow Which language and logic?

A I've used HOL-light. Not clear it's the best more generally.

Weidijk Competition between systems? But what will you do with formal abstracts.

A Open licence.

Organiser Lessons learned?

A Do it in one proof assistant. Isabelle was a useful learning step. But the import into HOL was a nightmare.

Q Many regulators have imprecise statements. Would YM work help?

A General discussion.

Q Lakatos is anti-formalisation: are you denying him?

A "Keep your conjectures bold and your refutations brutal".

Powers? Are you seeing all FAs should be based on the same underlying definitions. Is that possible?

A One would like to go that way. E.g. Calculus of Inductive Constructions?

Peter ? Hilbert can show that even stating conjectures would need hard theorems.

A We'll do what we can do.

Q Is there intrest informalising experimental mathematics?

A Difficult, but a good point. Ideas?

1.2 UniMath: Present and Future: Voevodsky

UniMath is three things.

- univalent foudnation of mathematics
- A subset of the type-in-type mode of the language of COQ
- A github Library.

"Nobody knows what is the foundation of mathematics".
By a foundation we understand on object with three necessary components.

1. A formal deduction system such as FOL or Martin-Löf or ...
2. An interpretation of the formal primitives as mathematical obejcts of actions
3. A collection of fundamental constructons that, using the interpretation above, defines formal counterparts for the main mathematical concepts such as set, function or natural number.

List of type constructors selected from Coq: includes dependent product and disjoint sum. Exampl eUniMath code defining categories from precategories. Definitions `has_homsets` etc. Importance of readability: code is written to be read, as well as used: more like a paper (with improvements suggested etc.) than a program.

If we do not use the resizing rule (see later) then I believ ethat we know how to interpret any construction of UniMath as follows.

- Types are Kan simplicial sets, elements of types are points of these sets, elements of the identity types are paths between the corresponding points, the universe of the base of the universal Kan fibration.

* The element of belief comes from the Initiality Conjecture that we hope to prove.

The third component is "univalent type theory". The key concept is a *contractible type*.

$$\texttt{iscontr}(T) := \sum_{t0:T} \prod_{t1:T} \texttt{paths}t1t0.$$

This is a Kan simplicial set which is non-empty iff ....

$$\texttt{isofhlevel}0T := \texttt{iscontr}(T).$$

$$\texttt{isofhlevel}(Sn)T := \prod_{t0,t1 \in T} \texttt{isofhlevel}n(\texttt{paths}t0t1)$$

.

Level 1 are propositions, level 2 are sets, level n are $(n-2)$-types.

$$\texttt{homot} fg := \prod_{t:T} \texttt{paths}(ft)(gt)$$

A fiber, or h-fiber of a function $f : T_1 \to T_2$ over $t_2 : T_2$ is the type

$$\texttt{hfiber} f t_2 := \sum_{t_1:T_1} \texttt{paths}(ft_1)t_2.$$

Definition 1 (Naïve equivalence) *Quadruple* $(f_1, f_2, h_1.h_2)$ ...

Definition 2 (Equivalence) *A pair* (f, w) *where* $f : T_1 \to T_2$ *and* w *has*

$$\texttt{isweq} f := \prod_{t_2:T_2} \texttt{iscontr}(\texttt{hfiber} f t_2)$$

Write $\texttt{weq} T_1 T_2$.

This participates in the Univalence Axiom. If we tried to use naïve equivalnce we would get inconsistency, so this is quite subtle. Related to hlevel concept.

This goes back to 2010: we've edited since for readability, but not semantics.

1.2.1 UniMath Library

Grows in two ways.

Directed Someone comes to us, and we give them a needed project.

> **Problem 4** *Let HSET be the category of sets, Given S, to construct an equavalence between categories $HSET/S$ and*
>
> Unimath has three objects corresponding to small category: precategory, category and univalent_category. The first is what you get obviously (objects, morphisms, composition functions, three axioms). For category we need the types of morphisms to be sets. For third, we also need the type of isomorphisms between two objects to be equivalent to the type of paths between the two objects.
>
> Types form a precatgeory, and sets a category. If we add the univalence axiom then we can proe the category of sets is univalent.
>
> **Problem 5** *Let C be a precategory,a nd P a presheaf of sets on C. Let $\int P$ be the category of elements of P. To construct an equivalence between the precategories $\int P$ and $(PreShvC)/P$.*
>
> There's a solution in `elems_slice_equiv.v`.

Problem 6 *A split object extension strucure on a category C is given by*

1. *A function $Ty \to HSET$.*
2. *For every $\Gamma : C$ and $A : Ty(\Gamma)$ a pair*
   (a) *an object $\Gamma.A$ of C*
   (b) *A projection morphism $\pi_A : \Gamma.A \to \Gamma$*

For $F : \Gamma' \ldots .$

Then construct an equivalence between the types of functorialterms structures and split q-morphisms ....

Not true for sets: intrinsically uses level-3 objects.

Independent A pull request from some-one we don't know. Tomi Pannila (triangulated categories), Anthony Bordg (modules over rings)

1.2.2 Future

New participants define the directions, so I can only talk about a subset. One is syntax and semantics of dependent types. This is going well. In one branch it needs a 2-category library. We

1. a 2-precatgrory is ...
2. or 2-precatgeory is a type with a famoly of precategories of morphisms where the compositions are functions.

Currently have a start along the second.

Another issue (Bernays lecture 2014 ETH) is to formalise a proof of Milnor's conjecture on Galois cohomology. Doing this classically is not very interesting, but constructive is much more interesting. But the proof uses Merkurjev/Suslin ..., and ultimately AC.

Personally most interesting in the modern theory of geometry and topology But no-one knows how much of this can be made constructive.

Thanks to Coq team.

Pitts What's your view on constructivism: why does it matter.

A Taught to care by constructivists whom I respect. I see classical as a subset of constructivist. Non-constructive existence proofs (especially if provably not constructible) leave an empty taste.

? We have a paper on the equivalence between slices and .... Hales had a question on formalising sheaves: doe sthis help?

A We have presheaves. So merely (!) a matter of formalising Grothendieck topologies.

Hoffman (Munich) Which theorems are worth formalising non-constructively?

A ?Don't know.

Q Last week you said you wanted proofs written by hand.

A Certified complex computations would be good. But I want the logic of the proof to be written by hand.

1.3 Proof Assistants: From Symbolic Logic to Real Mathematics?: Paulson

Recitation of examples.

Q McCune/EQP wasn't very special.

A It had a very specific aim, though.

Note that 4CT was checked in Coq

1.3.1 History

Boyer–Moore (1971): ancient, but lives on in ACL2 [BM79]. HOL(1988) but a descendant of LCF (1978). Lego (1991), Automath (1968), Mizar(1973) were designed to do mathematics rather than to verify computer code.

Harrison 2000 Verifying floating point algorithms. Needs both a mathematicaltheory and a theory of IEEE in HOL.

Hurd2003 formalised measure and probability to verify probabilistic algorithms in HOL.

Boldo (2013) verified a numerical analysis program fro solving a wave equation "covering all aspects from partial differential equations to actiol numerical results" in Coq.

***** And in maths.

Fleuriot 1998 A formalisation of geometry and nonstandaind analysis to check infinitesimal proofs in Newton's *Principia*.

Avigad/Harrison Prime Number Theorem in Isabelle/HOL Light.

Gonthier it et al. Odd-order theorem.

Paulson Gödel etc. Inspired by question at INI some years ago.

See page 118 of Jech's *Axiom of Choice*. Also [[]p. 201]Rota Indiscete Thoughts] OlgaTT works fro three years to fix errors in Hilbert's papers before the collected works could be published.

1.3.2 Choices

Types/sets Note the extent to which Russell's letter was a shock. De Bruijn believed that set theory was wrong. Type theory can be polymorphic (HOL) or predicate subtypes (PVS). Or depednent types (Automath, Coq, Agda). If you want sets, there's Mizar and Isabelle/ZF. Noe that type class polymorphism (as in Isabelle) allowed you to axiomatically define groups etc., and once I prove V is a Euclidean space, I inherit all the properties. Less flexible than dependent types, but useful.

Definedness "What is 1/0".

- Don't case, e.g. HOL/Isabelle. Makes higher concepts hard.
- Dependent types require you to prove $y \neq 0$ before x/y is statable.
- Free logic: a formalism in which `defined`$[x/y]$ is statable.

Search versus automation Some automation is vital. Matthias showed that Bourbaki's 1, fully expanded, need $> 10^{12}$ signs.

- Decision procedures. Fast, predictable, powerful, limited scope.
- Heuristic methods. Ad hoc changes invalidate prior proofs.

Syntax Mathematical notation is elegant but ambigupus, and machine notations are hideous. Shows a HOL-Light code, where $\mathbf{R}$ is distinguished from $\mathbf{R}^1$ etc. Example of same as a sructured proof. Claims structure is necessary for maintenance, re-use and eventually translation into more advanced systems. Furthermore, legibility breeds confidence.

1.3.3 Are proof systems ready fro mathematics?

Soundness, at least relaive to a small kernel. Automation is getting a lot better. Scalability is very good (these days).

Q I am not easy in the easy stuff — that's why I'm here.

A Agreed.

Q Programmers have to write readable code, or not be employed.

A Yes.

Q What will you do to Isabelle other that grow the libraries?

A Hire mathematicians and built theories. Structured proofs.

Q Compare type classes?

A I don't know a great deal.

Q–MK If we want to help mathematicians, they are good at proofs. Not so good at search, and de-duplication. Is verification barking up the wrong tree?

A I do proof tools.

Q–UHM There are lots of varieties of mathematicians.

1.4 Mathematical Knowledge at Scale: Watt

Early vision was digitise, OCR for searching, and links in references. The "digitise" component is fading (but old papers are still important, unlike many other subjects). Quotes 2006 WDML charge from IMU GA. Tondeur (NSF) was promoting this. Quotes Dennis and Ginsparg. Claimed 80K items/year. Very important point is that most of the world does not have access to most published mathematics.

- What to include
- How to include it
- Copyright
- Interoperability
- Sustainability

We've practically reached the Jurassic boundary when "Not online means doesn't exist".

ID's 2014 e-mail. Beyond document capture there is actually knowledge capture. Then there's a knowledge base. Paraphrase of Gowers' wishlist. "Is it known" (aim yes, to build on); "Is it known" (aim no, to publish); etc. Then further goals.

What is the literature: formal, informal, monographs, specialised collections like DLMF, OEIS etc.

Q What do you need?

Tao E-mail and MathSciNet.

IMKT main areas.

- Ontologies
- Formalized Mathematics
- Document Analysis
- Machine Learning (human effort just won't cut it).

Quotes [CDJW00].

1.4.1 Seed projects

- Formal Abstracts project
- Fornal Harmony, e.g. special functions.
- Document analysis: n-gram analysis of documents.

We will need lots of independent projects. What does capturing knowledge mean? Differing degrees of formalism. "A little inaccuracy saves a world of explanation". De Bruijn factor: FW thougt it was 4, and probably still is, as it's the point

1.4.2 Documents

- Metadata
- Subject classification and eywords
- References
- and the article itself.

Take the point of view of the compiler, which converts source code into a semantic object. Work on document segmentation (Sexton/Sorge). Equation recognition (Suzuki). Quotes [MW12], and the fact that "is (1) an equation" is undecidable in some languages.

Note the area of Ontology Engineering. Many different meanings for "semantic capture". Note that capturing the actual literature includes the ability to capture wrong statements (see Hilbert above). Good Venn diagram. Formailsed Math $\subset$ FAbstract $\subset$ Published, but the intersection with "True" isn't clear (except that Formailsed Math $\subset$ True).

Instead of busily importing libraries, can we link them? Need translation theorems.

We will never do all this by hand. Should we ever get to full automation? Approximation goes a long way. There are levels of formality. But hopefully we have usability in the less formal direction. Can we get anywhere with Mathematical Object Identifiers?

1.4.3 Concluions

> To extract and mechanise the world's mathematical knowledge.

Various more detailed ones.

Q-organiser Big challenges?

A Waterloo in 1980s: tried to digitlise OED, which seemed a vast project at the time. But caused OpenText, now the biggest Canada software house, Boston Marathon explosions analysis etc.

Q–FW Copyright

A Springer participated. 80% on board. Moving wall debates. Copyright covers the form not the content.

JHD Most subjects have much smaller time horizons than we have, which is to our advantage.

A We will need to work with publishers.

Q Applied Mathematics in adjacent areas?

A Good question.

Q:Harding How will deal with issues of style? Can we produce better texts this way?

A Good question - change the language, or the mathematics, or a combinations.

Q Differences and similarities with QED project.

A We have to be prepared to get things wrong in the process.

Chapter 2

11 July 2017

Workshop Group Photograph 11 a.m. tomorrow.

2.1 Impredicative Encodings in HoTT: Awodey (CMU)

Or: Toward a Realizability ∞-Topos. Introducer: one of the first people to recognise the connections HoTT/Martin-Löf. We are are really talking about "Big Types".

2.1.1 HoTT Introduction

Homotopy theory is the theory of space up to continuus transformations: "coffee cup = doughnut". Hence we can use intensional Martin-Löf to prove results in higher mathematics, e.g. homotopy group of sphere. Type Theory is great,but when we add new axioms, we have to be careful. Types, Terms, and Dependent Types $x : A \vdash B(x)$. $\sum_{x:A} B(x)$ and $\prod_{x:A} B(x)$. Them a logic, e.g. rules of inference.

Curry–Howard correspondence. $A + B$ corresponds to $A \vee B$ etc. $\sum$ corresponds to (constructive) $\exists$. Add a primitive identitytype $x, y : A$ has $\mathrm{Id}_A(x, y)$, which corresponds logically to $x = y$. $p, q : \mathrm{Id}_A(a, b)$ is two proofs of $a = b$. Then what s $\alpha, \beta : \mathrm{Id}_{\mathrm{Id}_A}(p, q)$ are two proofs that this proofs are equivalent etc. Until 10 years ago, this was "junk". Now we thing of $p : \mathrm{Id}_A(a, b)$ corresponds to a path from a to b. Then α, β are homotopies between paths.

Logic then *forces* dependent types to be fibrations etc. In topology, the points and paths in any space are a groupoid. The laws of identity (RST) are identity, inverse and composition in the groupoid. However, the groupoid equations only hold "up to homotopy". Hence the fundamental ∞-groupoid of a type. [Lumsdaine, Garner–van den Berg]. Goes back to a conjecture of Grothendieck.

Then the universe of types is stratified by the level at which the fundamental ∞-groupoid becomes trivial (if it ever does). A is contractible $\sum_{x:A} \prod_{y:X} \mathrm{Id}_A(x,y)$ a is essentially a point, and so on (as in section 1.2). This revives "Propositions as Types".

An Impredicative system such as Girard's F one can form new types by quantifying $\prod_X$ over all types X. This encodes many things, e.g.

$$\mathbf{N} := \prod_X (X \to X) \to (X \to X).$$

Hence a justification for the rules of inductive types, and these new objects are constructed n the system (therefore relatively consistent) rather than being added (and possibly adding inconsistency). Originally we had CC (Coquand, Huet). This justifies the Inductive in CIC. The orignal encodings of System F did not yield the usual elimination rules. Technically, no η ruke. Using HoTT we can sharpen up these encodings and get the right elimination property.

We have the universe U rule.

$$\frac{A\ \mathtt{Type} \quad x : A \vdash B(x) : U}{\prod_{x:A} B(x) : U}$$

whereas the usual rule requires $A : U$. Girard's F:

$$A + B = \prod_X (A \to X) \to ((B \to X) \to X)$$

(roughly speaking, we need maps from both A and B into X). This really does satisfy the encoding of $A \vee B$, where we restrct to $X : Prop$ in VV's stratification.

If A and B are sets, with $X : Set$ and $Set := \sum_{X:U} Set(X)$, the same construction only has a weak elimination property. In HoTT, dependent elimination is equivalent to η by a recent result. Ww have an embedding/retraction.

$$\begin{array}{ccc} A & \to_r & \prod_{X:Set}(A \to X) \to X \\ & =\searrow & \downarrow_e \\ & & A \end{array} .$$

We use identity types to "sharpen up" the encodings.

$$S^1 = \prod_X \prod_{x:X} (x = x) \to X$$

was suggested earlier, but this had the same weakness. The universal mapping property of the circle is $(S^1 \to X) \simeq \sum_{x:X}(x = x)$. Then the main lemma applies. We have higher coherence conditions.

This was fun, but is it safe? Is it consistent to have a universe that is impredicate and univalent? In a topos, the subobject classifier Ω is a universe of propositions that are both. But what a proof-relevant universe U i.e. not a poset. Models of System F and CoC can be made using realizability: the

category $\mathcal{A}sm$ of assemblies has an internal category $\mathcal{M}$ of modest sets that is complete and not a partial order [Hyland].

In order to get amodel with arbitrary n-types, we generalise from groupoids to ∞-groupoinds. inside a realisability model $\mathcal{A}sm$ of impredicativity. These are $\mathcal{A}sm$-valued Kan complexes, i.e. certain presheaves with values in $\mathcal{A}sm$. Our presheaves are cubical rather than simplicial, since [Coquand] this makes for better Kan complexes. The realizability ∞-topos RT_∞ is a QMC based on cubical presheaves in $\mathcal{A}sm$. Our present gioalis to show that the complete internal subcategrory $\mathbf{M}$ consisting of the **modest Kan complexes** provides a univalent universe.

Q Linear-time equivalence? But what about introduction rules?

A Don't know the first. But the introduction rules follow.

***** Doubts.

Q What properties do you need of $\mathcal{A}sm$.

A Locally cartesian-closed. We don't assume co-complete, so some arguments aren't available.

Q Apparent paradox.

A You can't construct a power set (?that way).

2.2 Logic in univalent type theory: Escardó

I want totalk about generally doing mathematics. Aunivalent type theory s a mathematicallanguage for expression definitions etc. that is invariant under isomorphism. $P(X) \wedge X \simeq Y \to P(Y)$.

Martin-Löf Type Theory (MLTT) has the property that we can't exhibit any property of $\mathbf{N}$ that $\mathbf{N} \times \mathbf{N}$ does not. Note that this is not true is set theory as such.

1. Spartan MLTT+univalence+resizing is UniMath
2. More generous MLTT + univalence + higher-inductive types is HTT book
3. Cubical Type Theory
    - has univalence as an axiom
    - has some tigher-inductive types
    - extends a spartan MLTT
    - has intrinsic computational content

I will deliberately not commit myself to any particular univalent type theory here. I wil formulate the univalence axiom afte we have developed enough material.

Spartan MLTT: $(0, 1, 2, \mathbf{N}$, cartesian product, function types, etc.

$\mathrm{Id}_{\mathbf{N}}(0,0) = 1$, $\mathrm{Id}_{\mathbf{N}}(0, n+1) = 0$, $\mathrm{Id}_{\mathbf{N}}(m+1, 0) = 0$, $\mathrm{Id}_{\mathbf{N}}(m+1, n+1) = \mathrm{Id}_{\mathbf{N}}(m, n)$. In this case $\mathrm{Id}_X(x, y)$ is a subsingleton, but in geneal there may be more than one way to identify two objects.

We assume a large type U. This has many uses: define type families as functions $X \to U$, also type of groups etc. People say "univalence means equivalent types are equa", but this doesn't really make sense.

Curry–Howard Propositions are types, and proofs are elements of types.

Univalent Propositions are subsingleton types (as in topos logic), and proofs are elements of types.

Both are constructive by default. We can consistently postulate excluded model and choice if we wish. $A \to B$ transforms a proof of A into a proof of B etc. A prof of $\exists x : XA(x)$ is a witness $x \in X$ and a proof $A(x)$. Hence "there are infintely primes" is a function $n \to p : p > n \wedge p$ prime.

Suppose for the moment that two functiosn that two functions whihc are pointwise equal are equal (this will be a consequence of univalence. A type is a proposition if any two lements are equal (subsingleton) $funext \to isProp(isProp(X))$

Theorem 1 (MLTT) $\prod(y : X), isSIngleton(\sum(x : X), x = u)$.

But the type IsIsomorphism need nt be propoistion please of higher-dimensionaltypes. But isEquivalence is (VV).

2.2.1 Univalnece for universe

1. $C \simeq Y \equiv \sum(FLX \to Y), isEquivalnce(F)$.
2. $UA \equiv \prod(XLU), isSIngleton(\sum(Y : U), X \simeq Y)$ - not the original formulation by VV (this needs less baggage).
3. Theorem of MLTT (VV) UA is a proposition
4. Meta theorem (VV) UA is consistent with MLTT (simplicial setmodel)
5. Theorem of MLTT+UA $P(x) \wedge X \simeq Y$ implies $P(Y)$
6. Theorem of sparten MLTT woth two universis: the univalence axiom formulates with crdude isomorphism rather than equivalence is false.

Univalence is an extensionality suitable for type theory rather than set theory.

Suppose we define Image in the usual way. Then $image(f) \simeq X$ which is not we expect. We don't expect the image of the unique function $2 \to 1$ to be 2!

$||X||$ is the truncation of a proposition.

$$\underbrace{||X||}_{\text{small type}} \Leftrightarrow \underbrace{\prod(P : U), isProp(P) \to (X \to P) \to P}_{\text{large type}}. \tag{2.1}$$

The image fo a function $f : X \to Y$ is defined with $|| \dots ||$ and now works.

Theorem 2 (Impredicateive characterisation) *We get the usual intuitionistic higher-order logic, which reduces everyting to implication $\to$ and universal quantififtcaion $\prod$ with a set of equavalnes small/large like (2.1).*

We can have a univalent excluded middle, but restricted to propositions only. There is a mth that propoitional truction erases information. It doesn't: the witness is there all the time.

$$||\sum(n : \mathbf{N})f(n) = 0|| \to \sum(n : \mathbf{N}, f(n) = 0$$

Plays a game between "root" and "minimal root".

Correct formulation of unique existence. Instead of usual statement, use $isSingleton(\sum(x : X), A(x))$. A unique $x : X$ with $A(x)$ is not enough.

Choice just holds in Curry–Howard. If $R : X \times Y \to U$ is a relation .... However, univalent excluded model implies unique choice.

2.2.2 Conclusions

1. $\sum$ is used to express structure/data, the type of groups
2. Truncation $\sum$ is used to express existence
3. at the moment it seems to be an art to decide when to truncate, i..e. structure/data or propositions.
4. The main point is that our mathematical langauge allows this distinctions

2.2.3 Q&A

Q Are you proposing this terminology? To separate these logics.

A I just needed it for my talk.

2.3 HoTT in Lean: Floris van Doorn

2.3.1 HoTT

I HoTT, types are interpreted as spaces. This leads to a new programmme: *synthetic homotopy theory.* To study typesin type theory as spaces in homotopy theory. Thismethod for homotopy teoory is more general; constructive and

easier to formally verify. A type has points a, b, paths p, q, paths between paths $(r : p = q)$.

We make equality precise by path induction. If $C : \prod(x : A), a = x \to Type$, to prove/construct an element of $\prod(x : A).\prod(p : a = x), C(x, p)$ ot suffices to construct an element of $C(\ldots)$. Example of composition: both formula and three lines of lean.

Given A, we can n-trunctate $||A||_n$ which is the best approximation of A which is n-truncated. Given a opinted type (A, a_0) we have a loop space $\Omega A = (a_0 = a_o, relf_{a_0})$. This is not quite a group as it has higher sructure, but the truncation solves this. $\pi_1(A) := ||\Omega A||_1$. This is much simpler than a traditional definition via continupus functions. Contrast [Bohua Zhan ITP 2017]: 13KLoC theory 3500 KLoC on ML, 5 person-months. Traditionally, all work with homotopy group has to prove invariance under homotopy, but again this is trivial.

In TT there are inductive types, e.g. **N**. In homotopy theory, we can build cell complexes. S^1 is base + loop: base=base. Hence functions from S^1 define $f(base)$ and prove that f respects loop.

Hence the suspension in homotopy theory, unfortunately denoted ΣA. Add North, South and merid:$A \to$ (North=South).

2.3.2 Lean

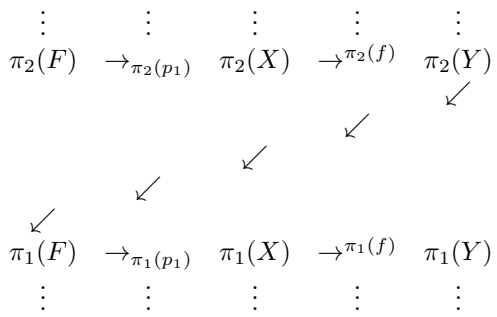
Various proof assistants support HoTT. Lean by Leonardo de Moura. Apache 2.0 licence. There is a standard and HoTT instantiation (broken in latest instantiation, apparently). Small kernel (Hierachy of universes indexed by **N**, universe polymorphism, dependent products, inductive types as in Dybjer: 1500–2000 LoC depending on kernel) and powerful elaborator. Allows proof terms or tactics. Emacs mode with on-the-fly checking. Kernel doesn't have termination checker, fixpoint operations, pattern matching, etc., but the elaborator seems to compile ML-lookig pattern matching into the kernel.

For HoTT we add univalence axiom, and the HIgher Inductive Types quotients and truncations.

```
HIT quotient (A:Type) (R : A -> A -> Type): Type :=
| i:A -> quotient A R
| ...
```

HoTT library is 33KLoC, and 11KLoC of work-in-progress. [LumsdaineShulman2017] that not all HITs can be reduced to quotients. However, many can be, which works for us. We can do homotopy pushout, $\pi_k(S^n)$ for $k \leq n$ and $\pi_3(S^2)$.

In particular long exact sequence of homotopy groups.



Hence spectral sequences.

Automation is not much used in Lean-HoTT. Type-class inference is used. We could perhaps do more. There are isuses.

- HoTT is *proof-relevant*, i.e. is matters which proof you use.
- When using dependent types,computation matters a lot.

But also there are many cases where it doesn't depend on the proof.

2.3.3 Conclusions

HoTT is a convenient language for homotopy theory. More general, and the language notions are primitive. Path induction is a novel means of reasoning (and more?). Constuctive, but not anti-classical. Formal verification does work in practice. Lean is a good proof assistant for this: significant results done.

2.3.4 Q&A

Q Synthetic versus non-synthetic?

A Here you cannot take a point out of a space (not homotopy invariant)

Q Quotient looks really like a pre-quotient. Can one put in an equality on the quotient?

A Yes: someone (name missed) has done this.

Q Time?

A 2–3 minutes to parse the whole library, but some aspects took lomg to run: hard to find out where it's spending time.

2.4 Small Proof: Dan Licata

I like to make sure everyone's equally confused: shows x86 assembler. Talks about high-level languages and DSLs, and also how we wouldn't have thought of parallelism until it was useful. Hence analogy that x86 or ARM might be ZFC or type theory. The HLLs might be Agda/Gallina/Isar. The DSLs would correspond to MLTT+UA+HITs, or Modal Type Theory. We can take various perspectives, and regard HoTTT as a HLL to be compiled into ZFC, or we can treat it as a foundation.

Then the "big proof" is that your compiler/model of DSL: if we have that right, then the application should be small. But this requires cleverness, specialists. There may be limits (Gödel, de Bruijn) limits on any particular language, and practical limits on DSLs.

2.4.1 HoTT'13 (MLTT+UA+HITs

DSL for homotopy types that should compile to set theory; simplicial sets model *and* other setting for homotoy theory.

Theorem 3 (Blakers–Massey) *Z has functions f, g to X and Y. The pushout is $W : -X +_Z Y$ with maps v_1, v_2 from X, Y. Then there's a pullback to $X \times_Z Y$. Then there's a statement on multiplicities (map from Z to $X \times_Z Y$ has multplicity $mult(f) + mult(g)$).*

We make a pushout .... $f : Z \to X$ induces an isomorphism between homotopy groups,

$$\begin{array}{cc} t2c : T \to S^1 \times S^1 & c2t : S^1 \to (S^1 \to T) \\ t2c\ a = (base, base) & c2t\ base = \{base \mapsto a; loop_x \mapsto q_x\} \\ t2c\ p_y = (loop_y, base) & \\ t2c\ qx = (base, loop_x) & c2t\ loop = \{base \mapsto p_y; loop_x \mapsto f_{x,y}\} \\ t2c\ f_{x,y} = (loop_y, loop_x) & \end{array}$$

These maps are inverses

Theorem 4 (Brouwer Fixed Point) *Any continuous map $f : D^2 \to D^2$ has a fixed point.*

Let f be a counterexample. For any $z \in D^2$ .... This theorem is really about topological spaces and continuous functions, not about homotopy types.

Externally, we are seeing that type theory $\to$ HTT $\to$ topology. Shulman had real-cohesive HoTT. Note that the homotopy circle has two objects, base and loop, and therefore is topologically discrete. New concept $\int$ (pronounced "shape". Related ∞-groupoinds with topologyon each level to straight ∞-groupids. This idea of "cohesion" has multiple uses:

- Types as smooth manifolds [Schreiber; Wellen]
- Types as parameterized spectra [Finster,Joyal]

- Directed type theory [Riehl, Shulman]

There's a framework (adjoint logic). Build a "mode theory" which can be used as a parameter

2.4.2 Q&A

FW I was told that "if you're a constructivist, topological spaces are not the right point of view". Shpuldn't you be "point-free"?

A If you prove different versions. you

Q Status of the "Big Proof" compiling these?

A This hasn't been fully worked out and/or verified. I'd personally like to see that this was useful before putting great effort into the Big Proof.

2.5 Schemas and Semantics for Higher Inductive Types: Peter LeFanu Lumsdaine

Apologies for not saying schemata. Higher Inductive Types says that the type is generated by elements and *equalities/paths*. Introduced 2011 andused 2012–13 IAS special year, Crucial for synthetic HoTT. Some payoffs for set-levelmathematics in type theory. The semantics seem reasonable: many important models of type theory have some HTTs. Would like to write $S^1 := |base : S^1|loop : base \mapsto base$ and it's the last part we can't write in, say, Coq.

Less trivially, a pushout type. $X, Z : Type.\ Y : X \to Z \to Type\ X + Z :=$
$|inl(x) : \quad X + Z;$
$|inr(z) : \quad X + Z \quad .$
$|h\{x\}\{z\}(y : Y\ x\ z) : \quad inl\ x = inr\ y$

The problem is that these are done *ad hoc*, and there's no general theory. We'd like a flexible general scheme of declarations, as in inductive types. Pratcical for programming. Should include all examples in HoTT book. Reasonable semantic account. Interpreted in important models.

On the way to that. How about:

- A few specific HITs, c.f. $\sum, +, n, W$ of MLTT
- sufficient to reconstruct most/all established examples
- plausibly sufficient to reconstruct all from conjectured schmea
- Interpreted in important models

An intermediate approach is also possible.

Heuristic 1 (Folklore[1]) *Assume identity types, $\prod$-types with functional extensionality, $\sum$-types. $+$-types, 0 and 1, and a universe closed under these. Then all the inductive types of CIC can be simulated by W-types.*

A W-type has a single constructor of the form $c(a : A)(f : W_{A,B}^{B_a}) : W_{A.B}$ where A parameterises the constructors (or: is non-recursive argument of constructor) and B_1 gives recursive arguments of constructor a. The idea is that terms over a single-sorted signature ....

- Mutual inductive types can be consolidated as an inductive family indexed over n.
- Inductive families indexed over I can be consolidated as (a subtype of) a single inductive type, and recovered as fibres of a function to I.
- A constructor with multiple arguments can be replaced by a single-argument constructor over a $\sum$-type.
- Multiple recursive calls in a constructor can be consolidated using sums, and multiple constructors via a $+$-type.

2.5.1 First idea: pushout types

These easily give any on-recursive HIT. Not so obviously, they do many recursive ones (van Doorn, Kraus, Rijke). This is in HoTT/UniMath libraries. Modelled in

- sets and groupoids
- simplicial sets and similar homotopy-theoretical models, but with a size blowup (Lumsdaine–Shulman)
- Cubical sets without a size blowup (Coquand–Huber–??)

Proposition 1 (Blass) *Suppose "ZFC+a proper class of strongly compact cardinals" is consistent. Then ZF is not sufficient to prove "every infiniary equational theory has an initial algebra".*

Such initial algebaras can be presented as HITs, so (under that assumption) ZF cannot prove the set model supports these HITs. But ZF certainly does prove that the set model supports quotients.

2.5.2 Anatomy of a HIT declaration

Two essential new twists.

- path-constructor specification involves terms s, t referring to source, target of a path
- via these, later constructors can refer to earlier ones (also echoed in eliminator premises and computation rules).

Then how general should s, t be: not general enough doesn't cover, but too general means either we can't write the eliminator, or the result is inconsistent. The obvious choice is "anyting derivable in appropriate context". But these leads to a "can't write down eliminator" issue. Need s, t tobe natural (up to propsitional equality) with respect to maps preserving the earlier constructors. When more constructors are aded, one finds that the naturality paths need to be coherent. Usual ∞-categorical headaches: no finite anount of data seems to be enough.

2.5.3 Finitary inductive types: Dybjer–Moenaclaey

Sourc/target terms froma restricted grammar:

- variables from recursive calls
- earlier lower-dimensioanl constructors
- (in source/target of a 2-path) compoistion, identities and inverses.

This has a schema defined for part and 2-path constructors. Interpretationin groupoid model. But this grammar is too restrictive for free algebras for infinitary types.

2.5.4 Higher W-types

Rephrase naturality. "Natural transformations $X^2 \to X$ for algebaras X" becomes "natural elements of X, for algebaras equipped with 2 distinguished elements" becomes "elements of the free algebar on 2 elements". Extends to ∞-naturality, insettings where one can state that. It's plausibly "fully general". But it gets nasty about 3 or 4 constructors. Inconvenient to recover other HITs

2.5.5 HITs of Cubical Type Theory

Coquand–Huber-?? work-in-progress. The implementation of CTT alows wide,flexible schema of HIT declarations. Don't have size blowup. They do not claim consistency.

2.5.6

Theorem 5 (Lumsdaine–Shulman) *Let M be an excellent monad category ...*

This cell monad semantics still doesn't seem to do "all HITs". The semantic construction used to model individual HITs in an *ad hoc* way. Potential size blowup.

2.5.7 Q&A

Q Are infinitarytypes realistic? Who would type them in?

A Exa,ple of indexed family.

Chapter 3

12 July 2017

3.1 Formally Verified Approximations of Definite Integrals: Mahboubi

$m \leq \int_a^b f \leq M$ is the problem. Ideas

- Symbolic Integration might not work
- Bounds might require approximation
- Bounds might be singular.

Question from stack overflow about Goldbach conjecture. Can't do symbolically, error bounds from their numeric are not guaranteed. I have bounds on derivatives. See Helfgott's Numerical Verification paper [HP13]. Also integrals in the major arcs paper [Hel13, (6.10)].

Representations of reals —- IEEE. Notoriously difficult to specify. We use intervals.

Definition 3 *A function* $F : \mathbf{I}^n \to \mathbf{I}$ *is* isotonic *if* $y_i \subseteq x_i$ $(1 \leq i \leq n)$ *implies* $F(y) \subseteq F(x)$.

Note that intervals are closely connected to Riemann integration: $\int_a^b f \in (b-a)[m, M]$.

Use the Coq Flocq library — many options for radix, precision etc. [BoldoetalARITH'11]. Also Coquelicot which has real analysis, conservative extension of the standard library, defines elementary mathematical functions and has a theory of Riemann integration.

Interval library on $\mathbf{IR}_\perp$. Specifications related back to Coquelicot. Use SLPs rather than trees to get sharing right! Represented in a non-destructive (i.e. functions don't pop their arguments) stack format. Two evaluation views $[e]_{\mathbf{R}}$ and $[e]_{\mathbf{I}_\perp}$, also $[e]_{\mathbf{R}_\perp}$. Slide 23 claims $[e]_{\mathbf{I}_\perp}(i) \neq \perp \Rightarrow f$ continuous on i.

For small problems it works: but $\int_0^8 \sin(t+\exp(t))dt \in [0.340144, 0.354\ldots]$ after 20 levels of dichotomy. Hence we need higher-order methods, which she formalises as exact integration of approximations. A rigoorous polynomial approximation is a pair (P, Δ) with $P \in \mathbf{I}[X], \Delta \in \mathbf{I}$ such that ....

$\int_0^1 \frac{dx}{1+x^2} = \frac{\pi}{4}$. Absolute error (interval width) 10^{-18} takes 2.9 seconds with accuracy 60, degree 12, depth 5 precision 70. The above example is error 10^{-4} in 277.6 seconds with accuracy 15, degree 12, depth 12, precision 30. Octave's `quad`, `quad?`, `quad??` give off values without warnings, `quadv` off value with warning, `quadl` 9 seconds for correct.

$\int_0^1 |(x^4+\cdots)e^x|$ (note the abs introduces singularity in derivative). INTLAB gives wrong answer without warning[1]. VNODE-LP can't be used because of this.

Improper integrals means limits. $\int_B^\infty hg$ can be treated if g is integrable and h is bounded. Hence need a catalogue of suitable g. Negative exponentials, Bertrand integals $\int_b^\infty t^a \ln^b(t)dt$. $\int_1^\infty \cos x \frac{\ln x}{x^2} dx \approx 0.1595$ gets 10^{-2} in 42 seconds. Maple fails.

Another example (quite messy) shows preprint quoted is wrong (despite "extra-cautious, doubled all bounds").

3.1.1 Conclusions

- A framework
- configurable tactic
- improved libraries (interval, real analysis)
- a number of possible improvements
- Possible symbolic/numeric integrations. Note that there's the "who's in charge" question. What about syntactic differentiations with singularities.

3.1.2 Q&A

Q–FW Coq

A Standard logic, inductive talks.

Q There is a lot of work in the numerical community: don't reinvent.

A We didn't.

Q–Bas Very impressive. Can you compare with ....

A There is a lot of other work.

[1] "Fixed" by remving support for absolute value!

3.2 Metaprogramming with Dependent Type Theory: de Moura

`leanproof.github.io`.

1. Testing. SAGE, PAC
2. Verification. Vcc, Boogie

Based on Z3.

Easy to use for simple properties. Scalability issues, proof stability. This bad problems for Hyper-V (hypervisor problem), Ironclad and Ironfleet, Everest. (last three on GitHub).

"The stratgey ossue in SMT Solving [deMouraPassmore]. Started Lean in 2013. Wanted to be able to extend Lean in itself, access to internal data structures of provers.

`CIC--` — no Fixpoint/Match, but add recursors, as in the 1988 paper. Unbounded recursors are only used for meta-programming. Code ⇒ equation compiler ⇒ type checker. Also compiler results ⇒ Bytecode, C++, LLVM etc. Any speedup in code generation feeds back into performance of Lean!

Has `structure` construct, with some implied contexts. Demonstrates a `has_sizeof` constructor.

3.2.1 Metaprogramming

`meta def find: expr` → list expr → tactic expr. Can look up suitable tactics. Then can write an `assumption` tactic. Indcutie type `expr` which includes meta-variables as well as usual. Two different kinds of quotations, depending on when elaborated.

Tactic `monad` interacts with the datat structure `tactic_state`, with meta-variables including open subgoals.

As in Haskell `f $ g t` is short for `f (g t)`. "When I first saw it, I thoght it was weird, but it's grown on me".

Example 1 `f x x = x p x x`. *Want to simplify with axioms* `p (f (g b) b)`, `p (f (g (g b)) (g b)) b`. *Use an* `rsimp` *tactic. Using this and axioms and hypothesis, and equality is a congruence, we can using* `rsimp` *to detect a congruence class of minimal expressions.*

SMT tactic has a state saving mechanism. Allows him to write a meta tactic `collect_implied_eqs`.

Superposition prover. 2200 LoC. Also a`transfer` tactic (200 LoC). Used for proving `dlist`, a version of `list`. Lean to SMT2 tactic. Coinductive predicates (800 LoC). Use the impredicativity of Prop, and so kernel extensions are needed.

Example 2 *Simple expression language. There's a* `nano_crush` *tactoc for this. This has a* `mk_relevant_lemmas` *tactic.*

We have a sampling-based profiler (very useful) and a debugger (implemented in Lean itself). We still want to expose more: structured trace messages, code generator extensions etc.

3.2.2 Demo

Hover over expressions to get the type. Auto-completion. Tactic state window (rather like MATLAB workspace). "Lean is not vapourware".

3.2.3 Q&A

Q Coq-style reflection? A function that evaluated to true if the certificate is good.

A You have access to the Lean data structures. But the kernel is minimalistic, without the efficiency techniques of Coq.

Q Reflection in the core logic?

A Discuss offline

Q Roll-your-own syntax as in Agda?

A Coming.

Q–FW Can I save with proofs, that don't need rechecking?

A Yes

Q General references?

A Haskell-like references in the tactoc monad, but not full ML references. Note that we have refernce-counting GC.

Q ?

A Automatically create bijections between `expr list` and `list expr` etc.

3.3 Hammers & Counterexample Generators & Beyond: Blanchette

Introducer: how on earth does one person do so much. The secret is re-use.

3.3.1 Hammers

Hammer: any interaction of automatic and interactive theorem proving.

Nipick A (counterexample) model finder for Isabelle (HOL). One can waste a lot of time trying to prove things that aren't true. Nitpick is FORL. Based on a first-order equivalent called KodKod (basically SAT). Note that it's important to translate the counterexample back into Isabelle. How does this translation work. We work with finite fixed cardinalities. Predicates are relations on $A_1 \times \cdots \times A_n$ etc. Higher-order functions encoded as functions from $\underbrace{A \times \cdots A}_{|\sigma| \text{ times}}$. Blows up, but in practice counterexample are small. You only get back a fragment of a model, but that's enough.

Universal quantifiers in positive position are a problem: many are used in idioms, e.g. axiomatisation of $\mathbf{N}$. But many things can be exported if you recognise them, and choose a subterm-closed sub-world.

Nunchaku A modular model finder for higher-order logic. Lovely annotated diagram of a nunchaku (Japanese weapon). Isabelle front-end and multiple backends. More precision by better approximation.

The pipeline is a pair of (encode,decode) functions, and OCaML just iterates down this list.

1. Monomorpohise
2. Specialise
3. Polarise
4. Encode (co)inductive predicates
5. Encode (co)recursive functions
6. Encode higher-order functions

A useful back-end is SAT-solving+narrowing.

Sledgehammer Does there exist a functions f from $\mathbf{R}$ to $\mathbf{R}$ such that $f(x + y^2) - f(x) \geq y$. JH had a proof in LOH-Light. Four lines of hint, and rest is automatic. Thre were several precursors: Otter in ACL2, Bliksem in Coq, Gandalf in HOL98 etc., and successors, MizAR for Mizar, SMT-Coq/CVC4Coq for Coq. Technical question is efficeint encoding of type symbols *where necessary*.

Supporting quotes from LP, TN and TH.

Matryoshka Higher-order automatic provers for proof assistants.

1. Extend superposition and SMT to HO. Need something that truly extends FO, and is on a "pay per use" model for HO.
2. pracical methods oand heuristics

3. stratified architectures. Base FO provers, E and veriT. How to exploit FO work to generate/guide HO instantiation.
4. Connection with proof assistants.

Leanback Lean+Number Theory idea: NWO grant proposal. Possibly p-adics, also elliptic curves.

3.3.2 Q&A

Q ML for proof search?

A We use ML for lemma filtering: e.g. 500 out of 10K. Very valuable. Also "select next clause to resolve".

3.4 The Social Machine of Mathematics: UHM

Image of piece of paper with Babbage/Lovelace doodles over Königsberg. (see `http://www.cs4fn.org/magazine/cs4fnissue20.pdf`)

Let's not forget that FlySpeck was a major engineering project. But note that many mathematicians (e.g. Clay prize winners) are talking about autamted proof. But, looking at a section of [Wil95] you see many "analogous": to what? Quotes Hardy — before the "no such thing as proof" quote, there's a lot of "mountaineering" analogies: apparently Hardy knew Mallory.

3.4.1 MathOverflow: Group Theory

Questions are 36% conecture, 28% "what is this", 14% examples, 22% "other". 90% of questions are resolved/moved forward. 56% shared referecens, 34% examples, 37% of answers contained challenges/corrections. Note that senior people (e.g. Derek Holt) put a lot of effort in.

3.4.2 PolyMath

See [GN09]. `tinyurl.com/jt75z2n` for the twin primes issue (PolyMath 8). Note that "accessible" was an important goal. Again, senior mathematicians took ownership. Requires commitment to the collective — what will your tenure committee say?

33% examples/counterexamples, 10% concepts, 20% conjecture,

3.4.3 Impact

This used to be an ordinary English word until REF. Example: QMUL REF on infrastructure networks. Analysed 209 of these. See [MM17] and clmplex linkages diagram.

Correctness versus risk: G.H. Hardy very sloppy, see also Hilbert earlier. POPL is starting to demand theorem provers: is this right?

3.5 Everything's Bigger in Texas: "The Largest Math Proof Ever": Marijn Heule

Solving and Verifying the Boolean Pythagorean problem via Cube and Conquer. This was 200TB: new proof is 3PB. SAT solvers started on verification for Dutch trains (an un-killer application!).

3.5.1 Pythagorean Triples

[Sch17] $a + b = c$ is forced monochromatic. $a^3 + b^3 = c^3$ of course has no solutions. $a^2 + b^2 = c^2$ is unknown. For small n there are non-monochromatic solutions. [1:7664] [CooperOverstreet2015]. $x_1 = T$ iff i is red. Each triple adds two clauses.

Theorem 6 ([HKM16]) *[1,7824] can be bicoloured but not 7825.*

NB: at 7284 about 10% of the cells are "either", which all die at the next point.

3.5.2 Sat Solving

Guess until a conflict. This is very effective on easy problems, but very hard to parallelise. Lookahead tries to find a small binary search tree. IIt's effective on small hard problems, but is expensive. CDCL is typically parallelised by portfolio, possibly with clause sharing. Cube+conquer is to do some look-ahead to plit the tree, then solve each branch independently (no sharing as values different).

N=number of leaves. On Schur triples problem (22608 clauses and 708 variables). On a single core, N=1 gives 1000sec, N=10 takes 2500 N=100 900 secs, N=10,000 is 200 seconds, with cube time just noticeable. 250 seconds at N=100,000, most of the time being cube.

For our problem, can use thousands of cores. Total time 4 CPU years, but less that 2 days wall time with 800 cores. Best off-the-shelf is Treengeling,which seems to be two orders of magnitude slower, and speeds up less well with parallelism.

3.5.3 Linear Speedups

3.5.4 Bugs

[BrummayerBiere2009,Brummayeretal2010]. Proofs are now mandatory for the annualSAT competition. UNSAT must be verifiable. Hardware Model problem hasn't yet gone that far. So what is a proof on UNSAT. C is *satisfiability-preserving* if F and $F \wedge C$ are both either SAT or UNSAT. Add such clauses repeatedly until $\bot$. For PT needed 10^{12} such clauses. 16,000 hours.

1. encode For PT this is 6 lines of C.

2. re-encode (including triple elimination if there's a singleton,and repeat). This is actually common in SAT solvers. Symmetry-breaking: most common group coloured red.

3. split to 10^6 on laptop, then down to $7 \cdot 10^10$ in parallel 250Khours.

4. solve For each ϕ_i we solved $F_{7824} \wedge \phi_i$, 13,000 CPU hours. One such is satisfiable, hence a backbone of 2304 literals. $x_{5180}, x_{???}$ forced red, two more forced blue. contradicting at 7825.

5. prove: Proofs verified in parallel [HeuleBiere2015].

3.5.5 Media

Headline in Nature was nothing like what he said in the inyerview! Mathematicians have this fanstasy that every problem has a short proof. There might not be: for example Ramsey Theory. What can we say about "questions with no easy proofs".

3.5.6

Weakly Human Much huam effort + some computer, e.g. 4CT

Weakly Alien A giant humanly-generated case split, e.g. min(given) = 17 in Sudoku.

Alien A giant case split that mysteriously avoids expoenential, as in vdW(6)=1132

Strongly Alien as in this.

Let's go beyond NP. Trying to solve Collatz this way.

Chapter 4

13 July 2017

4.1 Scaffolds and frames: the MathComp formal algebra library: Gonthier

Introducer (Jeremy Avigad): I thought the goal of Theorem Proving was to reproduce the paper way, but GG thought that CS had taught us a lot about large complex objects. I now think there's a half-way house.

14 co-authors, 5 in this room. My aim in buidling this was to make it as unremarkable as possible. We normally think of a library as being the equivalent of a frame that holds the house up — vital but unexciting. Have to fit together (and computerisation makes this more precise). On the way, you need scaffolding: also framing, but you intend to throw that away. There are also tools, that let builders put the scaffolding and frame together. But tools can be very specialised and personal.

4.1.1 Q&A

Combinatorics arithmetic, lists sets, graphs and functions, divisibility bigops.

Undergraduate algebra structures, polynomials,matrices[1], spaces, Galois theory, group and character theory.

Graduate Group theory modular representations, solvalbel, Frobenius, special groups, exceptional characters, Feit–Thompson.

Constructive mathematics[2] Quantifier elimination, algebraic numbers.

But not any calculus.

Showed Lagrange Theorem: there's actually a slides worth of variants. Naming conventions for lemmas. This *is* hard to do. In practice most of the proof lines go on a few "hard" theorems. Needs subsets like "monic polynomials".

[1] Note that these two interoperate: matrices of polynomials and v.v.

In traditinal mathematics, one doesn't state intentions. Notation is concise for efficiency, but overloaded and ambiguous. proof argument patterns aren't generally textbook logic. But formalisations have to deal with these issues.

4.1.2 Wrong Turns

- Dependent Type Family Theory. It took me seven iterations to define "path" for 4CT. The inductive defintion is actually very hard to reason with, becaus eusing "path" given a new object, even tough you might know things about it.
- Dependent Type. Inductive path, Fixpoint last: this needs re-inventing the whole theory of lists. Also having several relations on the set of paths is impossible.

The correct solution was a soft type, letting one use list operations. Coq's implicit argument passing means that "proof that P is a path" carries all the information needed.

- Type theory for groups. But this doesn't let you have the same object in multiple groups. Note that "group theory" should really be called "subgroup theory"!

```
Variable gT : fin GroupType
Definition group set (G: {set gT}) :=
     (1 \in G) && (G*G \subseteq G)
```

NB His Coq uses $\leq$ to mean $\subseteq$, not "subgroup".

```
Defintion quotient Q := {set coset_of } := coset @* Q.
Infix "/" := quotient.
```

We use $G/H =^{def} N_G(H) < H > / < H >$.

Example [FT63, Theorem 14.7]. His page of Coq really looks rather like the original.

Parametric overloading: constants whose type is given by thetype component of the structure. Straightforward unification. There's also *ad hoc* overloading. `coset h x * coset H y`. In Coq we can extract the type component of a structure. This is equivalent in expressive power to type classes,but is better behaved: driven by conflict rather than "when the prover feels like it".

This mechanism allows to implement generic operations/lemma over big operators. Claims 80 lemmas fro big operators, and we don't want these for each of 10 big operators. ALso lots of linear and sesquilinear operations. Can push scalar operations across linear maps.

Can have a predicate "this sum is direct" — note non-extensional, as depends on the way the sum is written! Note that this reasoning is common in character theory. Build a chain of inequalities, and a cycleimplies equality everywhere.

Thre's a case of bespoke meta-reasoning in [FT63]. Integer vectors for a norm problem. Naive SMT fails, naive SAT bitblast fails, and the tailored SAT encoding is longish. But the effort required to justify the encodiing is greater than that for bespoke logic. We can note it's symmetric, but the author of the book missed it. 4CT also has some bespoke logic.

4.1.3 Q&A

Q I understood all you said, but none of the slides.

A Most mathematicians can read ASCII LaTeX

Q How much of this depends on Coq?

A–FW I didn't use *all* Coq, such as type classes. Things like generic linearity are pretty idiosyncratic: "I'm proud of it for the wrong reasons".

Q–FW So what would Lean need?

A Not much: Leo is an excellent listener. Maybe some hints.

Q Do you use the impredicativity of ??, or singleton elimination.

A Probably, in that there are a few bespoke induction rules. But not essentially.

Q Lessons for mathematics.

A The mathematics was pretty explicit, but the problem was the notation, and finding a good transposition into Coq. Also, the defintiions were sometimes too flexible, e.g. for quotients. So if $\overline{G} = G/K$, then $\overline{A}$ doesn't mean A/K, but rather the image of A under the map $G \to \overline{G}$.

4.2 Mining the Archive of Formal Proofs: Nipkow

This is a very statistical talk. So what does a lage archive look like today? AFP is an on-line library for Isabelle, all contributed by users. A theory is a sequence of definitions, lemmas and proofs.

AFP Each submission is lightly reviewed by one of four editors (TN, LCP, Klein+1). `http://isa-afp.org`. Started 2004, 362 articles, 97K lemmas/theorems. 1.7M lines. HO Logic.

MML Mizar ML is 3MLoC, 50K lemmas. This is maintained by the developers. Set theory.

Coq There are two Coq repositories¿ GitHub and OPAM. Started 1992/3. 1.5M lines, 54K lemmas. 168 entries. Largely maintained by Coq developers. Type theory.

AFP is used for published articles. Hence mostly applications, rather than foundations. There is no imposed structure. Note there is also an Isabelle distribution, which unclodes 500KLoC of basic library. This is maintained by the developers. Isabelle and AFP release in sync. Updates to AFP come form boththe Isabelle developers and from authors. Each change to Isabelle or AFP triggers an incremental test run. So in a release version, all AP versions work, wheras witha development version,most work most of the time. This can make changes too costly, which has pro/con. Growth over time: authors/LoC/afrticles all grow similarly. Average submission 4600LoC. Half on 0–2K, then tails off sharply. loglog plot looks like s atrainght line (power law). Similar to file sizes etc. Largest is 77KLoC: a model of Java Threads/memeory model!

Main topics are programming languages, program analysis, model checking, data structures. Also Flyspeck, and some combinatorics.

9% definitions, 17% lemma statements, 60% proofs, 14% other (includes type classes). [Wiedijk2007] says ISA is 8/21/50, Coq 10/12/60, HOL Light 1/14/62; Mizar 3/9/84. Some of the paucity of definitions lines in HOL Light/Mizar are because they are pure maths. Lemmas/definition is 3.6 in AFP, 3.3 on OddOrder Theorem. Proofs in Isabelle are block-structured. Most proofs are short, and an exponential decay.

25% of the lines have changes, varying from 45% from the 2004 articles (some cosmetic, some substantial) to abou tthe same in 2012 articles, then tailing off. Open question: how much is development and how much is maintenance? TN's guess is 50:50. 362 nodes, of which 146 are isolated. 217 edges (excluding transitive ones) and maximum depth 9. Max out-degree is 10, in-degree is 6. Most popular library theory is `Multiset` (35), then `Code_Target_Numeral`. 237 out of 362 articles are never used. How does this compare? Not clear: quotes the 90% figure from [Meh07].

In CNF, 600 lemmas have 0 clauses (tautologies!). 75779 are one clause, up to 23 with 101–200 clauses.

Took a benchmark suite of 1250 goals. In 2010 could prove 46%, in 2015 it went to 65%, or 75% with machine learning. We have better translations to FOL for the provers. Now seeing CVC4 54.8%, and the worst, Verit is 51.3%. Portfolio gets 63.3%.

4.2.1 Conclusions

Seems in line with other citation data. But it's good to replace anecdotal with hard data.

4.2.2 Q&A

JHD For 90% figure, see [Rem14].

4.3 Formal Verification of Financial Algorithms: Passmore

What we do at Aesthetic Integration is take modern techniques and apply them to the very complicated area of modern financial algorithms. We walk to get themembeeded in regulation, as has happened in avionics. Even highly automated techniques are hard to sell!

Showed his firm's sales video. Cited major flash crash, but in fact minor ones happen all the time. Lack of transparency is a major problem. For over a decade the industry has been required to supply the specifications to the regulators (e.g. NYSE Byelaws), but typcially a 500-page PDF of a mixture of text and examples. "Financial markets are safety-critical".

4.3.1 Goals

To convince you that what we do is necessary.

Orders habe a "market type", e.g. "market order". Then "limit order", and now a huge proliferation of venues.

Venues Big ones, but now 66 in USA alone, each of which has its array of types. New venues come on line regularly. Venues compete for business. There's a chicken/egg issue. The incentive is developing an order type that does what you want. These become essentially programs.

Matching logic We see these as the instruction set of the market.

Matching microstructure

Smart order router sits above venues.

Regulations fairness (Reg ATS-N), best execution(Reg NMS). Differnet in different jurisdictions.

Almost all anaysis papers start "we assume all orders are market orders". We can now build, in Isabelle, models of other order types. We have a stack of financial models.

Q What is unfairness?

A An order needs to carry around information about who it's from. Bank's private dark pools (private venues) are often fined for giving clients priority.

4.3.2 Our Stack

Venue Each has its own matching logic. Also banks have their internal "dark pools" (and these have different transparency rules, especially over liquidity). Note that a bank sending an oredr out is revealing its cards, and incurring transaction fees.

Smart Order Routers These take large orders (that might have marlet impact) and split them up. Requires knowledge of semantics. Requirement that the client must never get a worse price.

Trading Algo[(rithm)s

Algo containers

Inventory Management

Collateral Trading

Moving from high to low frequenct, discrete nonlinear to low-frequency. These have connectivity protocols, generally FIX (Financial Information Exchange). 74-page spec. for NYSE one, for example. Have a DSL for these. We

UBS had been fined $14M by SEC for unfairness. This took four years, and UBS made its regulatory filings public. One 28-page one, informal specification. Our model found a much more fundamental flaw. There's a ranking function defined that wasn't actually transitive. This was how we won a contest (620 competitors) on "banking efficiency". An order book (for a given stock) is a pair of ordered lists (buy and sell). Order should be price then time, but in practice more complicated. But at one point the UBS documentation had (price,quantity,time).

4.3.3 Imandra

Very influenced by ACL2. Pricniple "Computing with counterexamples". We think of our technique as symbolci execution modulo sign invariance, a CAD-like system. We get region decompositions of input space.

Shows model of a venue's matching (written in OCaML). `type order = {` a 16-field record. Code for comparison function. Asks Imandra to prove transitivity, and Imandra produces a counterexample. This is itself a valid OCaML object so can be experimented with.

We have to formaulise the regulations to verify compliance. Example of a MFID2-compliance contract: 15 clauses in English becomes 114 formal goals.

Swiss Stock Exchange manual is 216 pages: traders must pas an examination. Mostly "proof by example" - manual connection to specification. Formalise this, and a function `step` that moves a trade on. `match_price` aLgorithm. Runs `decompose` on this, to get a decomposition into CAD-like regions. Have a printer that prints the region, and an example-generator. Can export documentation derived from this.

We are still at Venues and next two levels, and really can't get to next layers yet. Claims this is the "killer app" for formal methods.

4.3.4 Q&A

Q Next step?

A Team of 14, so need to be selective. We've open-sourced our DSL for FIX-specs. Working with a FIX provider.

Q Is this dangerous?

A Yes,there are black-hat issues here. There's currently an arms race between industry and regulators.

Q Are judges prepared to rely on your stuff rather than the PDF.

A No precedent. We should look at other safety-critical industries. FAA standards work well. Investigations are currently retrospective.

Q Multiple venues?

A Yes, for smart-order routers?

Q Converting natural language?

A A major issue.

4.4 Accessible Reasoning with Diagrams — Ontology Debugging: Jamnik

Diagrams, analogy, symmetry, but mostly "visual reasoning". Project started with Noia wanting to formalise privacy properties, across CS, lawyers etc. Ontologies are suually expressed symbolically, generally in DL (OWL), which is not user-friendly (to say the least). Visual tools are often informal, e.g. UML for OWL. Other (hard) diagramatic notations.

We use `conceptdiagrams`, which are formal, and supported by empirical studies. An extension of Euler diagrams, which differs from Venn in that you are allowed A to be a subset of B to be represented by concentric circles. Arrows for surjective relations (dashed for partial). Shading is an upper bound on cardinality (so shading + nothing less is empty). An *incoherent* ontology is one that implies an empty set (?). An *inconsistent* ontology is one which can have no model.

4.5 Machine Learning for Interactive Theorem Proving, Revisit, Reuse and Recycle your Proofs; Komendantskaya

Q What sort of ML would you like in your theorem prover?

A What are the options.

Speaker I'll show you some options.

Machine learning for ITP is

- assisting the human user to complete the proof;
- assisting the automated search for a proof.

Revisit Can we mine existing proof-patterns

Reuse Can we generate new proofs from a given set

Recycle can these mined patterns be used to generate new ones by analogy.

Proof General [KHG13] had proof engines (in practice just Coq) and ML engines (MATLAB, Weka). Need to extract features from proofs: about 100 of them. All clustering algorithms did much the same thing. Took three disjoint SSReflect libraries (Linear Algebra, Combinatorics and Persistent Homology — JH's contribution). 758 lemmas in all. This tried the bigops part of SSReflect.

Example 3 *Define nilpotent matrices.* $(I - M)\sum_{0 \leq i < n} M^i = I$. *Do the "obvious" Coq things, and assume 'stuck' on inductive proof. Tool suggested the Fundamental Theorem of Persistent Homology. Suggestion 2 is from BigSum library. The tool was very good at ignoring dofferences in notation. Suggestion 3 was "*M *nilpotent* $\Rightarrow \exists N : N(1-M) = 0$*" — looks very different, but actually useful. Note that keyword search on bigsum would provide 250+ suggestions.*

But paper rejected. Coq people said "what is interesting", and ML community said "what is your evaluation".

Example 4 (Talking to Moore) *Industrial project to model (subset of) JVM in ACL2 (or Coq), and then verify correctness of JVM programs. Try to help the team, e.g. with proving (tail recursive) factorial is correct. Can we use the output of the statistical side of ACL2(ml) to help.*

TT *Target Theorem.*

ST *Source Theorem*

SL *A source lemma suggested to prove ST.*

37% of clusters can be used directly. 19% os clusters comtain basic theorems whose proof are similar . . . 14% of clusters not really useful.

Paths in Coq HoTT libray showed with EPIA 95% hit rate, but JVM project showed 12%. Average 19%. Our new technique took the JVM to 58%, though paths dropped to 91%.

Generating proof terms is much harder. Data mining Ltac is not profitable. Data mining CompCerrt library and clustering proof terms worked: 180new proofs (on top of the 420 found by SEPIA) and shortened a further 80.

4.5.1 Q&A

Q Explain "Generating proof terms is much harder".

A It's exponential.

Q Feature extraction??

A Looks at the parse tree, and clusters, foe example, $\forall$ close to $\exists$ (but was used to recognise that associativity of append was the same as associativity of plus.

Q Availability?

A Weka is free.

Q

A Clustered 1404 lemmas from SSReflect, and got 126 clusters, 45% homegeneous (only from one sublibrary of SSReflect).

4.6 FAbstracts Discussion

FW opened the meeting, which had been suggested by JU. We should look for "Hard to express" ideas, e.g. "this proof is correct up to a set of measure 0".

TH recapitulated the aim of FAbstracts. Conceptually simple. Take every published paper, the statement of the man theorem, and write it down in a way that can be understood by a proof assistant, and by a human. This will require stating the definitions. Clearly we won't get all, at least at first. Hence this is a call for challenge problems. Yang–Mills mass gap problem was one of these. The more I learn, the more I realise this is the ultimate challenge. It's on the edge of what mathematics can currently capture (never mind formal mathematics).

Question about definitions and reuse: how about "forcing model of set theory".

MK: I challenge methodology. Clay problems maximise impact. We should aim to maximise impact/formalisation cost.

Are we trying to produce a list of 100 problems? FW: not quite.

What would the FAbstract of the Kepler conjecture look like? TH: $\forall P, \#(P \cap B_r(0)) \ldots$

Livestream (Kevin Buzzard) via SMW: long question about theorems with a great deal of apparatus: FLT. Floor: he didn't actually mean FLT itself, more things like TSW conjecture, needed *en route*.

Questions about flows on 3D-sphere: Siefert conjecture and Kuperberg's counterexample.

FW: Windmill problem (mini-Polymath problem 1, and indeed all of these)

A lot of statements in quantum physics could do with formalisation, but how?

GG had stated that one needed experts even to do this.

PDFI stated that quantum cryptography was one area: the experts say it's linear algebra but it isn't. Bell's Theorem, FreeWill Theorem.

There are statements where choice of formalisation matters, e.g. Reverse-Math.

Papers like "Here is a method for computing X which sometimes works".

MK again: The challenge is the social engineering. Aren't we compiling a list of reasons *not* to do it.

Two criteria: important theorem, and hard to formalise. I am in universal algebar, and we have examples (Birkhoff's HSP Theorem, Malcev's Congruence Generation Theorem) of important theorems used all the time.

Ad MK: Look at Berkeley problems in Mathematics. Series like GTM. Greatest articles in Wikipedia for example.

The easiest abstracts to formalise are one's own, so we need a gold star user reward mechanism. E.g. a search engine in which your work is more discoverable.

It is possible to interleave the formal abstract with the text: Literate Programming style.

JHD: Cryptographic papers of the form "Protocol X is secure against a chosen ciphertext attack" where the formal statement is "The advantage of the attacker in game ... [massive definition] tends to 0 as κ tends to infinity".

FW: Suppose my paper is "I have formalised Euclid".

There are cases where two theorems have different-looking statements, but are the same.

"The integral of X is Y" — which integral? Lebesgue?

Weyl conjectures combine a lot of areas of mathematics. ABC conjecture (or rather its proof)

There's a mailing list circulating.

A book on large cardinals. Also "Consequences of the Axiom of Choice" website.

"Pathological modular forms" is a hot topic now, on an intersection of NT, Topology etc.

Rather than GTM, how about Bourbaki? But it was said that this had been done.

Classification of semi-simple Lie algebras.

"I have been looking at my papers, and can't suggest one that is doable."

Q Do I need to redefine the reals?

TH We only need one.

Floor You'll never get away with that.

The strength of an FAbstracts would be such that it can be eother formalised or (formally refuted).

Chapter 5

14 July 2017

5.1 Explanation in Mathematical Practice: Pease

"How do people do mathematics": necessary to understand the gap between tools and mathematicians. Gowers: "If we wish to teach computers how to find proofs, we should understand how we find proofs outselves". Atiyah on a proof he didn't understand, and a quote from Bourbaki. A work in AI divides explanations as follows:

Trace explanations reveal the execution trace

Strategic explanations ...

Deep explanations take into account the audience.

DARPA's "deep explanations" project. We wanted to look at empirical data, "everyday" mathematics rather than big names, collaborative mathematics and "backstage" mathematics.

5.1.1 Hypotheses

H1 Explanations do exist

H2 Answer to a why questions?

H3 Primarily an appeal to a higher level (2% route 1, 23% route 2)

H4 T/S/D categorisation. (69% T, 24% S 6% D)

H5 Explanations contain purposive elements.

H6 Explanations occur in other contexts than proof.

H7 Explanations are social.

5.1.2 MiniPolymath

Note that Polymath has led to published results, and is a substantial corpus. See [Gowers2009]. Note also IMO: 14 of 36 Fields medallists were IMO participants. MiniPolymath has had four examples of IMO problems.

2009 81–100 participants (depending on view of "anonymous")

2010 28 participants, 128 comments

2011 43–56 participants

2012 43–48 participants

One could ask if these data are typical, but Inglis challenges the "assumption of homogeneity". However, the users believe that there is an answer, the variety of collaborators may be wider, and the online mediu means no diagrams, scribbles, gestures etc.

We only asked "is this text concerned with explanation", not a binary classifier.

1. Used certain indicators ("Since" etc.) automated to find explanations. Then manual analysis to confirm, then detailed. 742 comments, 243 indicators, 72% explanations
2. looked at a random subset (10%) by hand. Then as above. 74, of which 13 had already been found, 21% of rest were explanations.

Detailed analysis had setps.

1. Find explanans and explanandum
2. Locate the "why" question (often implicit)
3. Determine social context and type of dialogue [Walton]: persuasion, inquiry, negotiation, information-seeking, deliberation, eristic (personal conflict). We found 62% inquiry and 22% pedagogical.

Various examples.

All hypotheses except H3 were supported. H4 should be useful fro AI.

5.1.3 Q&A

Q Disappointed by H3: in category theory that's most of what we do. What about mailing lists/wiki.

A We weren't really looking at that area. Good point about sources.

Q–FW Any cross-validation?

A Three people did the assessment with discussion and cross-validation. It's still subjective.

Q–MK There's a lot of work on argumentation.

A We use "inference anchoring theory" as a basis.

5.2 Lightweight and Heavyweight Methods for Integrating Mathematical Libraries: Kohlhase

Thesis: big proof is bad — I want to think of big knowledge via little proofs. But I do think that Big Proof needs Big Libraries. Lots of sources of mathematics: preprints, abstracts, OEIS, OAF (`http://oaf.mathhub.info`) etc. But we need translation and aggregation methods, which may well be heuristic.

120K journal articles/year. Since 1860, Zbl knows of 3.5M. Since in general has a doublimg time of 8–15 years.

0 printed

1. Digitised (over 50%). But note that image search is practically useless. If we want anything, we need to go to next level.
2. presentational markup (20%)
3. semantic ($< 0.1\%$)

Discussion abut MathSearch and challenges.

L-functions/modular forms database: point/click, or hacking the MongoDB that underpins. PDF really doesn't help: blink mathematicians prefer to read LaTeX sources on a Braille reader. We use LaTeX sources of arXiv, using BM's LaTeXML.

MMT: Theory inclusions, inclusions via renaming etc. The iport of PVS (which we insist is shallow, i.e. preserving the structure) requires extending LF with records. Multiple librraies have lots of overlap (**R**, set theory etc.). Example of OEIS, and noting that generating functions provides means of detecting relations.

OpenDreamKit is biggest MathEU project.

5.3 Proof Archeology: Fleuriot

Past ideas.

- Principia (1687), mixture of geometry and algebra + start of calculus
- Hilbert's Foundations of Geometry (1899) generally comsidered rigorous. The formalisation did reveal ambiguities.
- Proof of JCT for polygons, not "without serious difficulty", as Hilbert had claimed.

This project: Euler's *Introductio in analysin infinitorum*. Collected works 31,000 pages, which is 800 pages/year. This was intended as a textbook, in two volumes. "Functions via infinite processes". There have been few editions, and only one Englishtranslation.]Blanton, 1988]. "Aims to use ordinary a;lgebra to develop concepts" "so that the reader gradually becomes acquainted with the

idea of the infinite". ω was his usual infinitesimal. Example from $1/(1+x)^2$ and $1/(1+x)$ evaluates at -1 to prove that $-1 > \infty$, hence ∞ separated the positive from the negative. He believed that there were infinitely many infinitesimals/ infinities. Note this was contemporary with Bishop Berkeley's attack on infinitesimals.

Cites various attacks on Euler: "lacks rigor, ... blunders" [Kline].

Note also Robinson's "nonstandard analysis".

We work in Isabelle. I developed a theory of NSA years ago. ECC= "Equivalence Classes Construction". Go from $\mathbf{Z}$ to $\mathbf{Q}$ via pairs+ECC, from $\mathbf{Q}$ to $\mathbf{R}$ by Cauchy sequences+ECC, thent $\mathbf{R}^*$ via ultrapower+ECC. Then define operations etc on $\mathbf{R}^*$, also extend relations, functions and sets: if $f : \mathbf{R} \to \mathbf{R}$ then it can be extended to $f^* : \mathbf{R}^* \to \mathbf{R}^*$. Transfer principle: ψ^* is true iff ψ is true.

$x \approx y$ iff $x - y \in$ infinitesimal. Need care when multiplying by infinitesimals: $\epsilon^2 \approx \epsilon$, but we can't multiply by ϵ. Can multiply by finites, though.

Euler says we know about polynomials, so we can use $A+Bx+\cdots$ as "the best form for the mind to grasp". Hence extends Isabelle summatinto infinite sets. Hence Euler's infinite polynomials are polynomials with hypernormal degrees. People criticize Euler for lack of convergence. See his 1734 harmoic series "in fact that which will be added after infinitely namy terms will be infinitely small". Quite amazingly, we can capture this using the existing nonstandard defintion of convergence. X converges iff $\forall m \in \mathbf{N}^*.\forall n \in \mathbf{N}^* X(m) \approx X(n)$. Hence a hyper version of comparison test for sequences and sums. Central tool is a hyperbinomial theorem. Need falling facforial $x^{\underline{0}} = 1$, $x^{\underline{n+1}} = x^{\underline{n}}(x-n)$ and then get a hyperbinomial theorem.

Need $a^{1/n} \approx 1$ for $n \in \mathbf{N}^*_\infty$. WE had to redfine real exponents. This was hard, and JF used an out-of-print book!

$a^x = a^{N\epsilon} = (a^\epsilon)^N = (1+k\epsilon)^N = \sum_{j=0}^\infty \frac{N^{\underline{j}}}{j!}(k\epsilon)^j = \sum_{j=0}^\infty \frac{N^{\underline{j}}}{N*j}\frac{(kx)^j}{j!}$ (by HCT) $= \sum_{j=0}^\infty \frac{k^j x^j}{j!}$.

[McKinzieTuckey winne r of 2002 Allendoerfer Prize] tried to fill ingaps, but we found gaps in this. There's a massive controversy over the use of NSA to re-examine Euler's proofs.

"Lisez Euler, lisez Euler: C'est notre maître à tous" [Laplace].

5.3.1 Q&A

Q Looking at Euler through the lens of classical: should we ask if he were a constructivist?

A He was certainly concerned with computing. He would certainly have used Excluded Middle without noticing.

Q Wjhat if you replace Robinsor with Conway's surreal?

A Not sure?

Q–MK This is really "time travel for proof".'

A

5.4 AfterMath — Reasoning, Computing and Proof in the Postwar United States : Dick

Note that diagrams, for example, have come into/out of fashion. What was the background to computerised proof in the 1950s/60s? There was, and in, disagreement about the nature of human reasoning, and how machines might do what. Showed a lot of early anthropomorphic language from the early days.

- Logic Theory Machine (ran on Johnniac) 1955–57 Rand ("colleague"). Designed to prove [WR10]: note the special notation, which was a key aspect of the programme. Newell/Simon were principal investigators (But John Clifford Shaw did the programming). Turned to Pólya and [P45]. The LTM worked backwards by sub-problem chaining. It was the only running program at 1956 AI conference. Essentially invented the Linked List Data Structure. Diagram [Figure 7] of how to delete an element.
- Program Q 1957–59 IBM/Bell Labs ("slave")
- AURA 1965–75 ("assistant"). Early hired Larry Wos. "Inuition cannot be formulated". Even so,had to [Overbook, 1971] formulate intiotion, via weighting. Joined by J.A. Robinson: also believed computers were different, hence introduced resolution.

5.5 Panel

Vannevar Bush in 1945 "proofs as we use a chash register", 1945. Interactive can be Church– Tarski or Martin-Löf, automatic can be search or saturation. "PolyMath is Lakatos on Steroids". What we are doing is "Abstraction Engineering".

Tim Gowers My take on "why mathematicians don't . . . ". There's "interesting as an observer" and "as a participant". I am very interested as an observer. For example, I am not personally interested in formalising UG maths. These "proof assistants" are not assistants to my sort of proof. HoTT is again interesting in the first sense. A proof assistant in the sense of "graduate student" is probably decades ratherthan centuries away. Real need is reverse lookup: don't need it to be as formalised as FAbstracts. Assessing how hard something is is an intrinsic part of the process. But of course this is relative to the search:how does this relate to automated search. MK: I have this "soft search" running on my laptop now from the arXiv.

Peter Sewell `http://rems.io`. I'm really interested in the applied discrete mathematics that should (but does not) underpin computer engineering. The hardware, compilers, OS, APps are all huge, and rapidly changing, but the interfaces (ISAs, programming languages, TCP/IP) are changing less often. But verifying, say, GCC is not possible. Hence we test against

an envelope of allowed behaviour. These are large (10KLoc). Most of this is pretty simple (set theory etc.). Need to give the engineers explorable models rather than symbols with quantifiers. Butvery often these tools can only be used by "experts in prover X". Therefore we write prover-independent statements. POPL 2009: 19% partially checked, 8% completely; 2014 was 10/13%. Manual proof is very fallible, and essentially unmaintainable.

FW I think we're further than that: see Intel/JH.

PS I agree for hardware.

Q Compiling models to code is a constructive proof?

A We need to decide whether observed behaviour is compatible with a model

JHD Announces possible ICM 2018 session.

Peter Dybjer Predictions for 2051 (AI experts predicted "all human activity" by then). Will there still be two distinct professions of mathematicians and formalisers by then? At Chalmers it's very hard to have joint students. [Gonthier2008] "The approach that proved successful for this proof was to turn almost every mathematical concept into a data structure or a program in the Coq sytsem, therefore converting the entire enterprise into one of program verification."

Note that Nuprl is "extensional" type theory, but also Andromeda. Agda and Idris are "future Haskell" rather than "future Coq". But maybe the "programming language" and "theory" streams need to merge.

Catherine Lelay (UniMath) When I arrived at Princeton two years ago, we had `Nat`. Added truncation subtraction in monoids: $\texttt{truncminus}(ax, y) + y = \max(x, y)$. I formalised $\mathbf{R}_{\geq 0}$ as Dedekind cuts on $\mathbf{Q}_{\geq 0}$. Then use save extension mechanism as $\approx$ from $\mathbf{N}$.

Then use Bourbaki General Topology Chapters 1–4: ultrafilters etc. Next would be metric and normed spaces. Then differentiability, integrals.

Maria Bonacino (automation) Most recent work is Conflict-driven theory combination in SMT; CDSAT. Prior to that Conetcflict-driven reasoning in FOL: SGGS. Slso Integrating superposition in SMT: DPLL($\Gamma + \mathcal{T}$)). Hence experience with many paradigms.

Distinguish between the derivation (complete set of deduction) and the proof (those that lead to the result). In general the proof has to be reconstructed from the derivation. Particular;y true when the result was "empty clause generated". To asnwer $H \vDash \phi$ we either generate a contradiction from $H \cup \{\neg\phi\}$, or produce a model for $H \cup \{\neg\phi\}$, i.e. a counterexample.

I'd prefer to call them "machine proofs" rather than "alien proofs". Let's not offend the aliens: 3621 exoplanets w.r.t. 1 July 2017.

Stephen ? (Flyspeck) Currently a sole trader at ??. The creative process always starts with something vague and ends with a goal: in my case a formal theory.

David Tranah Faraday: three necessary steps for research: begin, end and publish. Publishers were originally involed in dissemination. I don't think papers really need validation: most stuff is never read, and a rejected paper will appear elsewhere. Publishers can do (in partnership) work on exposition, style, discoverability and stability. A statistic is that 30% of URLs are no longer valid. Where we really help is providing a brake on publishing! Maths/CS is a small fraction of STEM, and the big commercial publishers would survive without Maths/CS.

Q CUP predicted it would make a loss on [WR10].

A We still sell about 10 copies/year at £500 for the three-volume set.

Q Publishers are supposed to feed back referees' comments (who may be the only readers).

A *Behaviouir and Brain Sciences* also ublishes the referees' comments and debates.

Patrick Ion (GDML) I have seen a lot of mathematics go past. Copyright licencing has been an impediment, and a different encoding is helpful to such digital libraries. Also GDML session at ICM 2018.

Chapter 6

19 July 2017 (at Alan Turing Institute

Introduction by Jane Leeks.

David Aspinall is (what is now called) a Turing Fellow. ATI is UK's national research institute for Data Science.: four themes and seven industry verticals. ATI employs people like me, but also postdocs (15) plus research students (15). Aim to grow to 100 each, also more universities.

6.1 Passmore

JHD missed the talk.

6.1.1 Q&A

Q Might it be unsatisfiable?

A Yes, there's aphysicist at SEC who believes that one regulations violates relativity.

Q This is OCaML– not specific

A Indeed.

6.2 From Z3 to Lean: Efficient Verification: de Moura

Z3 is a collection of Symbolic Reasoning Engines. Plus heuristics to choose between. Z3 is billed as a theorem prover, but really a SAT solver. SAT with a model, or UNSAT, with a proof (or maybe certificate). Many uses: Test Case Generation, Model Checking etc. Ships woth Osabelle,Pex, SAGE, SMAL/DV,

Visual Studio. SAGE solved 5G constraints looking at Windows8+Office. We are trying to solve real-world problems, essentially irrespective of worst-case computing costs. Each MS security bulletin costs $millions, and a virus $billions. We sometimes get maisl from Russian hackers asking about Z3!.

SAGE is based on Directed Automated Random Testing. If it hasn't tested a path, it asks Z3 to find an input that will exercise the path.

In verification we have F-Star, Vcc and Boogie (and many others). Have a VCG based on annotations and Dijkstra's weakest preconditions. Easyfor siple applications. But issues of scaling, or proof stability. F-Star is based on dependent type theory, and the gulf between this and Z3 is huge.

Hence moved towards Lean: written L∃∀N. Aims to bridge the gap between automated and interactive theorem-provers. de Bruijn's principle: small trusted kernel. Dependent TYpe Theory, and parftial constructions: automatin fills the holes. Example of Metaprogramming a tactic. Note taht we're not interested in proving properties of the metaprogram. autocompletion etc. Users can ask for the formulae sent to kernel.

6.2.1 Q&A

Q What is the difference between F-Star and Lean?

A F-Star is a programming language, not a theorem prover. F-Star has an effects system,wheras Lean uses monads: not as neat (?but verifiable).

Q Why isn't Lean a programming language?

A The Meta rpgoramming is new. The I/O is very basic, for example.

Q Proof engineers refactor. Can Leab help?

A Under discussion.

Q–Melham The "white box" is what we had in Intel's Forte as well: vital.

A I'd like to see this.

6.3 Big Proofs for Programmers: DeepStatic Analysis at Facebook: Villard

Facebook logos: "move fast and break things", "only the foolish wait" etc., so how do we combine with with Verification? Claim that this is done if your analysis is compositional. Out tool is called infer. Runs on the code reported by engineers. Static analyser based on separation logic. We want it to be easy for programmers to use iner, but also fro analysis writers to write new analyses in infer. Naïve idea is nightly run: too slow. Infer analyses Java nad Objective C, but in fact via an intermediate language.

Now treat infer as another reviewer in the code review process, and the comments are treated similarly. Only reported if the warning is introduced by the diff. It runs on all mobile code bases, 10k+ diffs/month, reports 1ks of issues fixed/month (—70% fix rate). Action taken is ground truth for success. Note that "False Positive" is a theoretical result: we don't have time to work out whether a bug can actually be provoked.

`https://gerriot-review.googlesource.com/#/c/75724` — no-one knew if this were an actual bug, but fixed anyway.

Does it find bugs: yes! Does it help productivity: yes. Moral is that we need scalar incremental tools that are easy to extend. Compute call graph (on deamnd) does a topological sort, analyse eachprocedure once using reverse postoder scheduling Break call cycles by iterating to fixed point. Compositional in the sense that the summary for a procedure can be used by all callers. There's no gobal view. The key is that it's linear in #procedures: vital. It's incremental: easy to transition from from-scratch analysis to diff analysis.Extensible since for a new analysis we just need a new domain plus transfer function.

Givena summary of thinsg we call, how do I combine this into a summary that any of my callers can use? Interprocedural escape alanysis : $Val ::= \hat{x}|FP(x)$ Then $\hat{\sigma} \subseteq 2^{Val}$. Differs fro local variables and formal arguments. Each summary is a function of its instructions +callee summaries. Simple change propagation over call graph works. In incremental mode, never need to analyse calles of changed programs. Analyse changed procedures, and, if summary changes, callers of these procedures and recursively.

6.3.1 Q&A

Q Turnround time?

A 10 minutes is the aim. It's massively parallel.

Q For 10 seconds?

A We're not sure. It's not really a goal.

Q Call graph in the presence of reflection?

A We ignore this! And if you use this, we'll shoot you!

Q–RC As programmers battle infer, does their programming improve?

A I joined two years ago, been there longer than 75% of the company.

Q–RC You should have data over your developer population.

A There is a change in behaviour: people tell us "infer didn't find this bug".

Q Would they winge if you took it away?

A I hope so.

Q Are there ‘infer classes’?

A The respected developers respect infer, and this is transitive.

Q What about existing bugs?

A The list of things we ignore grows over time.

Chapter 7

21 July

7.1 JHD

7.1.1 Q&A

Q

A

7.2 Reasoning by Equivalence: Sangwin

I have 500 students, and 35 tutors.Experise requires practice.The Internet distributes, but we have to be careful about issues of equity an the "digital divide". "If I can automate something, I understand it better". Same applies to the assessment process.

Example 5 $\int (x-4)^3 \mathrm{d}x$. *Valid, but incorrect, answer* $(x-4)^4$.

Note that STACK [San08] separates validity (always given, even in a timed exam) from correctness. This greatly simplifies the actual marking process. Correctness based on MAXIMA. Note that there are limits (three questions by default) on the number of attempts.

Which forms of proof can be reduced to a calculation.

Example 6 (IB sample question) $\log_3(x+17) - 2 = \log_3(2x)$

Note that RE (Reasonong by Equivalence) is key to induction, and much of the real analysis he taught at Birmingham. RE is 1/3 of the marks in IB. ee [Pell'sAlgebra, 1688]].

Hence goal to extend STACK to RE. List of lines as input to STACK. Question was "solve by factoring" so CJS demended to see the factored form. Solving is "progressive transformation", "representatives of the class' and "answers in a given form".

Example 7 (147 AA* Birmingham students) *Solve* $\frac{x+5}{x-7}-5=\frac{4x-40}{13-x}$ *? 9.5% showed any cornnectives, 2 students checked their answer, 1 student explicitly considered domains of definition.*

Note that a naïve solution divides by $4x-40$*, and gets a contradiction.*

[Chrystal's Algebra 1893] first book with Argand diagrams. Question: can we use RE more. Find a such that $-x^2+ax+a-3<0$ for all x. Can solve by RE with one line of rhetoric, but end lengthy.

Q Why teach this at all?

A Good question. But how do you get the CAS to do the rewrite without the commands.

Q Chain reasoning?

A supported.

Rational expressions and rôles of domains. I have $\pm$ (rules of binding interesting), single variables inegualities. Next $|\cdot|$.

$$CA=CB \Leftrightarrow A=B \vee C=0$$

etc. When faced with $A^2=B^2$, we should say $A^2-B^2=0$ then $(A-B)(A+B)=0$.

Students complained that $\sqrt{(x-3)(x-5)}$ was wrong, but $\sqrt{x-3}\sqrt{x-5}$ was correct. Good pint. Quotes Euler's Algebra as giving $\sqrt{-1}\sqrt{-3}=6$. With $\sqrt{x-3}$ etc.STACK willcomment on the natural domains.

Note MathExpert [Bee89] and its use of assumptions.

7.2.1 Q&A

Q MOOCs?

A I worked ona Statistics one. 15k students. MCQ only. There are some good MCQs.

Q Educera?

A I've heard of it, that's all. Apparently Coq is the engine.

Bibliography

[Bee89] M.J. Beeson. Logic and Computation in MATHPERT: An Expert System for Learning Mathematics. *Computers and Mathematics*, pages 202–214, 1989.

[BM79] R.S Boyer and J.S. Moore. *A Computational Logic*. Academic Press, 1979.

[CDJW00] R.M. Corless, J.H. Davenport, D.J. Jeffrey, and S.M. Watt. According to Abramowitz and Stegun, or arccoth needn't be uncouth. *SIGSAM Bulletin 2*, 34:58–65, 2000.

[FT63] W. Feit and J.G. Thompson. Solvability of Groups of Odd Order. *Pacific J. Math.*, 13:775–1029, 1963.

[GN09] T. Gowers and M. Nielsen. Massively collaborative mathematics. *Nature*, 461:879–881, 2009.

[Hel13] H.A. Helfgott. Major arcs for Goldbach's theorem. `http://arxiv.org/abs/1305.2897`, 2013.

[HKM16] M.J. Heule, O. Kullmann, and V.W. Marek. Solving and verifying the boolean pythagorean triples problem via cube-and-conquer. In *Proceedings International Conference on Theory and Applications of Satisfiability Testing*, pages 228–245, 2016.

[HP13] H.A. Helfgott and D.J. Platt. Numerical Verification of the Ternary Goldbach Conjecture up to 8.875e30. `http://arxiv.org/abs/1305.3062`, 2013.

[KHG13] E. Komendantskaya, J. Heras, and G. Grov. Machine Learning in Proof General: Interfacing Interfaces. *10th International Workshop on User Interfaces for Theorem Provers (C. Kaliszyk and C. Lüth Eds.) EPTCS 118*, pages 15–41, 2013.

[Lak76] I. Lakatos. *Proofs and Refutations*. Cambridge University Press, 1976.

[Meh07] Lokman I Meho. The rise and rise of citation analysis. *Physics World*, 20(1):32, 2007.

[MM17] U. Martin and L. Meagher. Slightly dirty maths: the richly textured mechansism of Impact. *To appear Research Evaluation*, 2017.

[MW12] S. Marcus and S.M. Watt. What is an Equation? In *Proceedings SYNASC 2012*, pages 23–29, 2012.

[P45] G. Pólya. *How to solve it.* Princeton University Press, 1945.

[Rem14] D. Remler. Are 90% of academic papers really never cited? Reviewing the literature on academic citations. `http://blogs.lse.ac.uk/impactofsocialsciences/2014/04/23/academic-papers-citation-rates-remler/`, 2014.

[San08] C.J. Sangwin. Revisiting James Watt's linkage with implicit functions and modern techniques. *Math. Magazine*, 81:116–126, 2008.

[Sch17] I. Schur. Über die Kongruenz $x^m + y^m = z^m \pmod{p}$. *Jahresbericht der Deutschen Mathematikervereinigung*, 25:114–117, 1917.

[Wil95] A. Wiles. Modular Elliptic Curves and Fermat's Last Theorem. *Annals Math.*, 141:443–551, 1995.

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