Notes on SYNASC 2012

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

26–29 September 2012

Contents

1 26 September 2012: MICAS Workshop **3**

1.1 Trust Model Engines in Cloud Computing: Roxana-Marcela Farcasescu . . . 3

1.2 Measurements of Services: Peter Matthews . . . 3

1.3 How do you port legacy applications to the Cloud: Harvey Kuhn 4

1.3.1 problems areas addressed in case study . . . 4

1.4 Laudations . . . 5

1.4.1 Opening: Tudor Jebelean . . . 5

1.4.2 Stephen Watt on Buchberger . . . 5

1.4.3 Ioan Rus on Stefan Maruster . . . 5

1.4.4 Maruster's reply . . . 6

1.4.5 Musical Interlude . . . 6

1.4.6 e-Austria . . . 6

1.4.7 Dana Petcu, director of e-Austria . . . 6

1.4.8 Second Musical Interlude . . . 6

2 27 September 2012 **7**

2.1 The FlexiFormalist Program: Kohlhase . . . 7

2.2 Interactive vs Automated Proofs in Computational Origami: Ida 8

3 28 September 2012 **10**

3.1 Synthesis from Examples: Sumit Gulwani . . . 10

3.1.1 Bit-vector algorithms . . . 10

3.1.2 Spreadsheet macros . . . 10

3.1.3 Language . . . 11

3.1.4 Syntactic String Transformations . . . 11

3.1.5 Education: Intelligent Tutoring Systems . . . 11

3.1.6 Geometry Constructions . . . 11

3.1.7 Grading of Programs . . . 11

3.1.8 Algebra Problems . . . 11

3.1.9 Intelligent Mathematics Editor . . . 12

3.1.10 Intelligent Drawing . . . 12

3.2 Automated Certification of a Logic-Based . . . 12

3.3 Towards an optimal power-aware scheduling technique . . . 12

3.4 Automated Synthesis of Some Algorithms on Finite Sets: Dramnesc 13
3.5 : Andrei 14
3.5.1 The **K** Framework 14
3.6 Label-based Programming Language Semantics in **K** Framework with SDF: Bogdănaş 14
3.7 Program Verification in the presence of complex numbers, functions with branch cuts etc.: Davenport 14
3.8 Arithmetic with Free Algebras and Hereditarily Finite Sets: a Natural Bridge between Numeric and Symbolic Computations: Paul Tarau 15
3.9 15
3.10 Speculative Genetic Scheduling for Hadoop Environments 16
3.11 Improving The Responsiveness of Replicated Virtualized Services in Case of Overloaded Replicas Connectivity 16
3.12 Symbolic State Space Exploration of RT Systems in the Cloud . 16
3.13 A fault-tolerant distributed solution for the parallel Gaussian reduction of huge arrays 16
3.14 17

4 29 September 2012 18
4.1 What is an Equation: Watt & Marcus 18
4.1.1 Our thesis 19
4.2 Sparse Matrices 19
4.3 A computational framework for the study of partition functions and graph polynomials: Markowsky 20
4.4 On Coxeter spectral study of posets and a digraph isomorphism problem 20
4.5 Nominal Fusion Calculus 21
4.6 : Tarau 21

5 Miscellanea 22
5.1 Kohlhase 22

Chapter 1

26 September 2012: MICAS Workshop

1.1 Trust Model Engines in Cloud Computing: Roxana-Marcela Farcasescu

[JHD arrived part-way through.]

This treats each rule as a service, which (apparently) leads to better performance. Security is very important, which depends on user profiles.

Q "Security as a service is optional" — this worries me.

A It's optional in the sense that some services may not need it.

Q! But you may be in breach of Data Protection and Traceability rules if you don't have a minimum of security — you should look into what the provides do.

1.2 Measurements of Services: Peter Matthews

- Measurements of quality means different things to different people (10 seconds is fine in many places, elsewhere μs matter).
- Establishing the relationship between metrics is complex:
  - Ontological solutions work in the lab but there are few in real life;
  - SAWSDL was promoted as on way to remedy this
  - UDDI had little to offer in terms of carrying service data
  - Standards would be one way of doing this.
- Cloud computing changes the game — 23% of CIOs find that their sales teams have contracted out to cloud firms without them knowing about it.

- But there are complex measures of quality.
- CA and others produced a first cut at 'measures for services'.
- CMU has done a set of 'measures for services', based on various categories, such as accountability, agility, Assurance ....
- http://www.cloudcommons.org/about-smi describes the CSMIC Service Measurement Index. This is populated with data from cloud service providers. The Consortium includes cloud service providers[1], universities, and measurement services.
- They have a prototype tool (in Excel).

Ultimately, this is a business-focused way of measuring services. Regulatory compliance is a huge minefield, e.g. European different transpositions of directives, or Sarbanes–Oxley in the U.S.

Q How do you measure functionality?

A Largely tick-box —- "we have Sarbanes–Oxley compliance" etc.

1.3 How do you port legacy applications to the Cloud: Harvey Kuhn

This is a case study which is work in progress. The BOC group sells 'solutions' and consulting. Was a Client/sever architecture, and now has a web architecture. Should be moved to (multiple) clouds. One example is a (machine) translation service — how does this move from one cloud to another? Many of our customers are banks, and German banks require that the services be performed in Germany, etc.

1.3.1 problems areas addressed in case study

- How do we describe the requirements in a way which helps with provider selection?
- Legacy migration. Need cost and impact analyses.
- Data Location (legal reasons,

Q Which category of users are looking at?

A Our existing programmers in the beginning.

[1] Such as Mycroft. Apparently a large US provider, though JHD had never heard of them.

1.4 Laudations

1.4.1 Opening: Tudor Jebelean

We have a set of distinguished people, but to represent them as a list, we have to order them.

Institute e-Austria, whose tenth birthday we are celebrating, was formed by University of Linz, notably RISC, and the University of Timisoara.

The Rector presented Diplomas of Excellence to Professor Bruno Buchberger and Professor Stefan Maruster. Also (not on the agenda), Dana Petcu, ??, Tudor Jebelean and Vlorel Negru.

Jebelean first met Buchberger in 1990, and was impressed by his optimism. He gave TJ many of the skills for a professional career.

1.4.2 Stephen Watt on Buchberger

Founder and Director of RISC, Founder and first Editor-in-Chief of the Journal of Symbolic Computation, and Founder and still Director of SoftwarePark Hagenberg. Main research areas in Gröbner bases, which he invented in his thesis, and then the Theorema project.

But a scholar transmits knowledge also via his students, and he has directed the habilitations of 8 students, which SMW named. There are also over 40 doctoral students (whom mercifully were not all named), six of whom had the honour of graduating in the presence of the Austrian President.

SoftwarePark Hagenberg has now over 1000 employees and 1500 students. subsectionBuchberger's Reply Very grateful to life and nature for such beautiful years. TJ was the first from Timisoara to come to Linz, and he impressed us by his knowledge and hard-working attitude. He has had 30 or more followers by now. With very little seed money, e-Austria has evolved to 30+ people. BB brings greetings from Prof. Dr Karlheinz ??, the Austrian Federal Minister for Science. Mentions many others, including JHD.

So please remember the vital rôle of mathematics in software, and of software in automation and all of life.

1.4.3 Ioan Rus on Stefan Maruster

Quotes Novalis: "the real mathematician is an enthusiast". Gave a career summary. Many students, including Jebelean, Petcu, Negre etc. Discusses contributions to numerical analysis. Numerous editorial activities and four Romanian honorary degrees.

The University of Cluj-Napoca also presented him with a Diploma of Excellence at this event.

1.4.4 Maruster's reply

In 1962 was part of the team building MECIPT, a "one-off" valve computer at 50 operations/second[2], 4KB of drum memory, with paper tape I/O. Showed an original valve complex, storing one bit, physically much larger than the pre-proceedings 4GB memory stick. Compare this with PC on his desk, and Blue Gene with 1024 processor cards.

1.4.5 Musical Interlude

Violin, piano and clarinet.

1.4.6 e-Austria

`www.ieat.ro` Founded formally on 5 October 2002, after much hard work, especially by Buchberger. Jebelean was the 'go-between", being viewed in Austria as an Austrian who went to Romania, and *vice versa*.

1.4.7 Dana Petcu, director of e-Austria

"I like challenges", and this is no exception.

2001-2 startup.

2003-5 Austrian support for start-up phase.

2005-7 Stand-alone and recognition.

2008-12 Stability.

It is a non-profit association between RISC, CS at UVT and CS at Polytechnic University of Timisoara. Main activities are R&D, Knowledge Transfer, training, conferences . EU accounts for 1/3 of the projects, but half the funding.

1.4.8 Second Musical Interlude

Piano and soprano; piano and baritone, piano and both.

[2] In a visit on 28th to the MECIPT museum, JHD reaslised that this was because instructions wer read off the drum, which was 3000rpm.

Chapter 2

27 September 2012

JHD's notes on today are sparse, as he also gave nearly hours of MPI tutorials (using NAG's material, for which he is very grateful).

2.1 The FlexiFormalist Program: Kohlhase

Note that this is a development of Hilbert's Formalist Program (1920).

Mathematics in important (and not just because it's fun!). There are three levels of documents available.

digitised replaces inter-library loan, but these images can only be read by a human.

presentational Encoded text (OCR) interspersed with presentational markup.

semantic encoded text with functional markup for the meaning.

Transforming up, or even down, requires human skills, or AI.

AI, Philosophy and Mathematics identify formalist with correctness.. Hilbert's program was:

successful in that ZFL+AoC is a foundation [Gödel1930];

disappointing as regards completeness.

Definition 1 *A formal system* $S := \langle \mathcal{L}, \mathcal{M}, \mathcal{C} \rangle$ *consists of*

- *a computable formal language* $\mathcal{L}$
- *a system of models* $\mathcal{M}$
- $\mathcal{C}$ *(some form of reasoning system?).*

Q That's a very informal definition.

A Yes — in a flexiformalist system, one could click on the components to see their definition, and so on.

almost all mathematics documents are informal in four ways.

- the foundation is underspecified
- the language is informal
- the formulae are informal, e.g. **N** (does that include 0?), and use presentation
- context references are typically under-specified, and theories and re-use is not marked up at all.

There are four reasons why formalisation hasn't happened:

1. Propaganda — "mathematics is done with pen and paper"
2. tedium — de Bruijn factor ≈ 4 (JHD: see [Wie00])
3. inflexibility: formalism requires commitment to a formal system
4. proof verification is useless as peer reviewing works.

hence the claim is that we should do *stepwise formalism*. We need to match the returns to the investment.

The solution he proposed is the "Active Documents Paradigm". Excel is a very good example of an active document: there is layout, data and reasoning rules.

An attempt at formalising robotics — "do not run over people" etc. — led to a complex multi-dimensional document structure: code, theorems, project data[1] etc.

Q It is accepted that "rigour is obtained at the expense of meaning" (e.g. Thom): is this an answer to this challenge.

A I would accept "don't formalise too early".

2.2 Interactive vs Automated Proofs in Computational Origami: Ida

"How do we convince the public that we are concerned with security and safety?"

Notes that there is a loss of rigour in geometry.

In Origami, a line can be

- The superposition of two points

[1] "Reviewed by XXX on NNN", approved by . . . .

- superposition of a point and a line;
- superposition of two lines
  - distinct
  - the same (i.e. construct the perpendicular).

* Hence the general concept of a *superposition* of two geometric objects.

Notation 1 $P \updownarrow Q$ *the segment joining* P *and* Q*.*

Hurita's method H]??

The Origami numbers are are square and cure root generated, and hence trisection of angle is possible this way [Wanzel1837]. A formal proof was given in [RomanosPaulson2010]. This needs a mutual inductive definition of 'point' and 'line'. The transformations between geometry and algebra are clearly defined, but the diagram does not commute [BeesonADG2012]. Shows a regular pentagon via Origami, and the commands his system needs to do this.

How much rigour is needed, and where?

Q How many Origami theorems are there?

A There wasn't a clear answer. Buchberger intervened, saying this was an interesting question.

Chapter 3

28 September 2012

3.1 Synthesis from Examples: Sumit Gulwani

Synthesise a program, query, sequence etc. Billions of non-experts have access to intelligent devices, but essentially can't program. We can use better search algorithms, and multi-core machines (the search algorithms *are* efficiently parallelisable). The state of the art is that we can synthesise programs of size 10–20. Proposes to look at various examples.

3.1.1 Bit-vector algorithms

An SLP with arithmetic and logical operations.

Example 1 *Turn off right-most one bit.* Note that a formal specification of this is quite messy.

Naïve solutions involves loop. But `Z&(Z-1)` will do.

Example 2 *Turn off right-most contiguous sequence of one bits.*

`Z&(1+(Z|Z-1))` is apparently the answer. Suppose user says "map 01001 to 01000". There are at least two programs that do this: `(x+1)&)x-1)` and ??, but they differ on 00000, which is a distinguishing example. So the tool asks this, and uses it to disambiguate, and continues.

3.1.2 Spreadsheet macros

Example 3 *Convert* `ddddddddd` *into* `ddd-dd-dddd` *(U.S. Social Security Numbers).*

Does this on the screen. User can ask the tool to highlight rows which are distinguishing examples.

3.1.3 Language

in which he synthesises, apparently.

Guarded Command G:= Switch((b_1, e_1),...)

Boolean Expression $b := c_1 \wedge \cdots c_n$

Atomic predicate $c := Match(v, k, r)$

Regular Expression := ...

3.1.4 Syntactic String Transformations

In POPL 2011, and is "flash fill' feature of Excel 2012! Shows this, how essentially an example with the right numbers gets converted into the appropriate lookups to find these numbers elsewhere in the spreadsheet.

Example 4

3.1.5 Education: Intelligent Tutoring Systems

Solution Generation, Automated Grading.

3.1.6 Geometry Constructions

Example 5 *Construct circumcircle. [GulwanietalPDLI2011].*

These constructions are essentially SLPs (10 line sin this case), but in a rather different base language than above. Three key ideas:

- Reduce symbolic to concrete example;
- Use high-level abstractions such as "angle bisector";
- use goal-directed search.

3.1.7 Grading of Programs

Example 6 *Reverse an array.*

The professor produces an error model (JHD? "buggy rules" as in [YO81]?), such as "off by one in array indexing".

3.1.8 Algebra Problems

Example 7 (Example problem) $(\sec x + \cos x)(\sec x - \cos x) = \tan^2 x + \sin^2 x$. *Can then generate a variety of related problems, e.g.* $(\csc x + \cos x)(\csc x - \cos x) = \cot^2 x + \sin^2 x$.

Example 8 (Example problem) $\sum_{i=1}^{n} \frac{i^2+i+1}{2} = 1/6\ (n+1)^3 + 1/3\, n - 1/6$ *generated a variety of other sums (apparently only of quadratics).*

This "correctness" of this is based on the theory of randomised algorithms.

3.1.9 Intelligent Mathematics Editor

Claims there is low entropy, hence predictive editors work. Has some statistics, but no details of the experiments. "Effectiveness of learning algorithm 52%".

3.1.10 Intelligent Drawing

It wasn't clear what this was — JHD thought it was 'line straightening' etc.

3.2 Automated Certification of a Logic-Based

Generation of the verification conditions for a program is performed by program — are *they* correct?

- The complexity of the soundness proofs depends on the choice of semantics and definition of terminations.
- The complexity plays a crucial rôle, especially is one aims at automation.

Our goal is to find a minimal set of axioms and logical inferences necessary for formulating and proving in a computer-assisted manner a correct collection of methods for program verification. The object theory (which is application-specific) is FOL with equality. We have a partial correctness meta-function, and one for termination conditions. Allow assignments (including recursive calls), `while`, conditionals and "abrupt statements" (`return`, `break`). Also annotations.

Example 9 *Consider*

```
while (i<n) do
    if (e=a[i]) then break
    i:=i+1
```

with a suitable partial correctness annotation. Proof of correctness is three lemmas and a concluding theorem.

Q Base semantics?

A Theorema.

3.3 Towards an optimal power-aware scheduling technique

Reducing energy consumption has become vital in the real-time embedded systems community. We assume:

- non-pre-emptive scheduling

- tasks and single-instance
- set of jobs known in advance.
- uni-processor architecture (for the time being).

A job $J := (c, d)$ where c is the cost and d the deadline. Use a variant of EDF (Earliest Deadline First), but replace idle time by decreasing the clock frequency to reduce power. We use $P_{\text{dyn}} = h \cdot p^\alpha$ where α is hardware-dependent. With two jobs ($d_1 < d_2$)

$$f(x) = \frac{c_1^2}{x} + \frac{c_2^2}{d_2 - x}$$

and there is an analytic solution, but messy (and piecewise). Note, however, that it only needs to be pre-computed.

Q These assumptions don't really match a mobile 'phone.

A True: we are looking at industrial systems. "non-pre-emptive" is the hard one.

Q What about ordering constraints on the job?

A Not considered at first, but shouldn't be difficult.

Q Computational complexity?

A [Some confusion here]

3.4 Automated Synthesis of Some Algorithms on Finite Sets: Dramnesc

This is also a program synthesis problem, but starting from a specification. Given G a function on sets (or W a predicate on sets), find an algorithm for sets represented as monotone lists: use $<$ on objects and $a \prec A$ to mean $a < b \forall b \in A$, and $\ll$ on lists. Has a collection of 7(?) strategies to find a solution. Shows an example (looked like Theorema, though this wasn't stated until the end). It did look relatively readable. They have deduced membership, inclusion, union, difference and intersection.

Q–MK How efficient are these algorithms? Quoted slowsort example (where an automatic synthesis program, given the specification "$(a_1, \ldots, a_n)$ is sorted if $a_1 <$ least of sorted($(a_2, \ldots, a_n)$) and $(a_2, \ldots, a_n)$ is sorted", synthesised a correct, but exponential, sort).

A "these are linear" — JHD is not sure what this means in terms of intersection/union.

3.5 : Andrei

3.5.1 The K Framework

A framework[1] for defining operational semantics for programming languages. IMP, SiMPLe (typed and untyped), C, SSRISC and Verilog in the assembly area, Scheme and Haskell. A configuration is a nested sequence of cells.

IMP has (unbounded) integers and Booleans, expressions of both types, statements and (untyped) variables. But often hard to specify languages in **K**.

Let L be a programming language,a nd G a grammar for L, with P_G the parser from G. Then annotate G to G' such that $P_{G'}$ produces an AST. Need (not sure why, but it's apparently important) to eliminate non-productive rules.

Future challenges include handling ambiguities.

3.6 Label-based Programming Language Semantics in K Framework with SDF: Bogdănaş

SDF - Syntax Definition Formalism. A parser generator. We already had a syntax for Java in SDF, so we wished to use is for Java semantics. The **K**-framework gives us executable semantics. One cell is "state", the other is "k" (probably the equivalent of "stack"). Apparently SDF parses `2` as `"2"`, so we need some conversion rules.

3.7 Program Verification in the presence of complex numbers, functions with branch cuts etc.: Davenport

The paper is [Dav12].

How to make yourself deeply unpopular: stand up at a computer algebra conference and say "we can't solve quadratics". Challenge: invert the injection $f : z \mapsto \frac{1}{2}\left(z + \frac{1}{z}\right)$, as an injection from $\{z : \Im(z) > 0\}$ to $\mathbf{C}$.

For a solution, see `http://staff.bath.ac.uk/masjhd/Slides/SYNASC-2012.pdf`.

Q Given that you can *verify* some of these, is there a use for synthesis?

A Very good point.

[1] `http://k-framework.org`.

3.8 Arithmetic with Free Algebras and Hereditarily Finite Sets: a Natural Bridge between Numeric and Symbolic Computations: Paul Tarau

We can answer positively two questions:

- can we do arithmetic directly with some "symbolic" mathematical objects such as binary trees, hereditarily finite sets?
- are these efficient enough to be practical?

Showed a Mathematica script manipulating multi-sets etc.

Definition 2 *Let σ be a signature then the free algebra A_σ is defined as the smallest set containing the the constants and $\sigma_i(a_i, \ldots, a_{\mathrm{arity}(\sigma_i)}) : a_i \in A_\sigma$.*

Definition 3 *A set with a total binary operation is a* magma.

Builds this (free magma) as a Haskell data type.

Definition 4 Peano algebra — *the "one and successor" algebra.*

Definition 5 *The "two successor" free algebra with signature $\{B/O, O/1, I/1\}$ is, among other things, the binary numbers. Note he suggests that O is $\lambda x.2x+1$ and I is $\lambda x.2x+2$.*

This leads us to the concept of representing one algebra by another. To ensure no loss of information, we insist these maps are bijections. Performance moves from $O(n)$ to $O(\log n)$.

Isomorphisms between free algebras provide bridges connecting "numeric" [in the sen of **N**, **Z** etc.] and "symbolic" objects.

Definition 6

3.9

Semantic description with power domains [Plotkin1976].

Theorem 1 *The two semantics (operational and denotational) behave the same.* Proof via Banach Fixed-Point Theorem.

3.10 Speculative Genetic Scheduling for Hadoop Environments

Hadoop is SaaS, using Large Data Sets and the MapReduce paradigm. One parameter is a "reducePercentage". The DurationReduce is this times the DurationMap. A node has maxMapSlots[2], maxReduceSlots, Power and Cost.

We propose a genetic algorithm, where 'mutation' corresponds to moving a map task from one resource to another.

Example 10 (word count) *Capacity violates the deadline by 57 seconds, but genetic works and is quite a bit faster (73 seconds under the deadline).*

3.11 Improving The Responsiveness of Replicated Virtualized Services in Case of Overloaded Replicas Connectivity

High availability is important. We encapsulate the service in a virtual machine, replicate the virtual machines, in a primary–backup configuration. But can have problems with overloading the connectivities between the replicas. REMUS replicates "in phases of same size" (?), including hard disk. We wish to integrate the HDD replication in our solution.

3.12 Symbolic State Space Exploration of RT Systems in the Cloud

Our reference model is time-based Petri nets. So a net is places, transitions, arcs and tokens; to which we add timestamps. Benchmark problem is the gas burner (which fits, almost readably, on one slide). We have (JHD wasn't quite sure why) an abstract state space as well as a concrete one. The gas burner example has 22,978 built states, and the abstract state space is nearly 15,000 states.

We use the MapReduce methodology, and a hash function to partition states, so that no communication between reducers is required.

3.13 A fault-tolerant distributed solution for the parallel Gaussian reduction of huge arrays

Situation is NASTRAN, doing non-linear analysis, which is Newton–Raphson, and therefore Gaussian. Also SPICE the circuit simulator: consider applying this to a 8GB memory stick.

[2] This looks crude, but seems (c.f. Wikipedia) to be how Hadoop actually works.

We could use a GPU, but maybe the matrix won't fit in the shared memory (limit α). Or an MPI-connected HPC cluster, but maybe the matrix won't fit in the distributed memory (limit β). So maybe MapReduce with Hadoop, but this is not trivial. Assume the initial data are "in the cloud". We will need to process the matrix in chunks. Not all nodes will be capable of handling the same-sized chunks.

We have a Byzantine agreement problem, which requires $3n + 1$ nodes. The critical point is the propagation of the pivot row. A sequential code takes 3 ticks. 4-node parallelism (no fault-tolerance) takes 0.8. 4-node fault tolerant takes 5.1 ticks, and 8 nodes 3.1 ticks.

3.14

Convex Feasibility using the Probabilistic Vector Machine. There are direct feasibility solutions, and parallel iterative solvers. The usual choice is block-sequential and block-parallel: we are hybrid. Each block is processed by some CPUs. We improve CARP's scheme by adding weights. The update of the solution is done using only some of the constraints (update window). The update is sequentially applied for every such window. Windows might overlap. The parallel algorithm has an equivalent higher-dimensional sequential form, so converges. Equivalent to CARP when the window size is 1. Can also mimic Alternating Projection Algorithms.

Compared with GLPK, which seems to be the most stable solver on the market. Note that our density is about 0.5. It's as fast as GLPK, but has better memory requirements, and is superior to CARP in terms of network traffic.

Chapter 4

29 September 2012

4.1 What is an Equation: Watt & Marcus

A collaboration begun at SYNASC 2010. Until two years ago, SM was convinced that "equation" was the same as French's "équation" , or "ecuaţie" in Romanian. But what about "$e^{i\pi} = -1$", which is described in English as an equation, even though it involves no variables. But all dictionaries agree that in Romance languages, the equivalent word is reserved for identities involving variables. However, the root Latin is actually the same as the English meaning. It is also worth noting that "equation" is often used outside mathematics either as a metaphor for mathematics, or as "something to the solved" (or indeed "the solution", as in $E = mc^2$).

$x + y = y + x$ is "equation" or "identity" in English, but only "identité" in French.

$2 + 8 = 10$ is not the equivalent of "equation".

$2H_2 + O_2 = 2H_2O$ (a chemical equation) is not the same thing at all.

- Rhetorical stage, using words, as in medieval Islamic mathematics.
- Syncopated stage, where individual symbols are used.
- Use of the full range of symbols, following Robert Recorde (1512–1558): his "Whetstone of Witte" [Rec57] introduced algebra to England. Hence the first equation (in the sense of using the = sign) was, in modern notation, $14x + 15 = 71$.

We note that Babbage [Bab30] required that one notation should only be used for one thing, but in practice this has become impossible.

We note that Wikipedia gives different meanings in different languages.

English An equation is a mathematical statement that asserts the equality of two expressions.

French Une équation est en mathématique une égaliité contenant une ou plusieurs variables

4.1.1 Our thesis

- Involving variables.
    - Always true when variables are (suitably) universally quantified: $[\forall x, y \in \mathbf{R}]\ \sin^2 x + \cos^2 x = \sin^2 y + \cos^2 y$. "Identity" or "tautology".
    - Sometimes true. In this case, finding the solution set S is "solving" the equation. However, if we now quantify over S, this equation now becomes an identity.
    - Never true, i.e. $x^2 + 1 = 0$ over the reals.
- Not involving variables.
    - True
    - False
    - Undecidable.
- Complication: $1 + 1 = 2 + 0 \cdot x$ ostensibly involves x, but not really. However, deciding this involves zero equivalence.

Q-BB It's really about (implicit) quantification: $x + y = y + x$ is implicitly universally quantified. $x + 2 = 3$ is implicitly existentially quantified, or, rather, constructively existentially quantified in that we want to know x.

A-SM It's a semantic difference which, oddly enough, has caused little confusion in practice.

4.2 Sparse Matrices

$n \times n$ matrices with N nonzeros, Let $M := n^2$. Let $S(n)$ be the space required to store n (allowing for rounding up to byte etc. multiples).

Coordinate format Three arrays, x, y and $M_{x,y}$. The complexity of the indices is $2NS(n)$.

CSR format ?

Quadtree format For each quad, we store four bits, with the non-emptiness of each quadrant stored.

Hence an 8×8 with only $M_{1,1}, M_{1,2}, M_{2,1}$ and $M_{2,2}$ non-zero would be stored as $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, with the first quad being $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, and its quad being $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$: 12 bits of index storage.

JHD If we know that 3-bit counters suffice, then the indexing cost in Coordinate format is $2 \times 4 \times 3 = 24$ bits, but rounding to bytes would mean $2 \times 4 \times 8 = 64$ bits.

Q Do you need to transform before using?

A Yes — we have no operations that work directly on this format.

Q Where do you store the pointers?

A We don't — the matrix elements are stored in traversal order.

4.3 A computational framework for the study of partition functions and graph polynomials: Markowsky

Each is much studied, but little comparison. The chromatic polynomial $\chi(G; k)$ is the number of proper colourings of G with k functions. [Birkhoff1912] shows that this is in $\mathbf{Z}[x]$. Note also Tutte polynomial, and various knot polynomials.

4.4 On Coxeter spectral study of posets and a digraph isomorphism problem

Can use as representations:

Hasse digraph (minimal) digraph — arrow from i to j if $i \prec j$ and $\not\exists k$ $i \prec k \prec j$;

Incidence Matrix C_I (which is complete).

A poset is "one peak" if it contains one maximal element. Coxeter matrix is $\mathrm{Cox}_I = -C_i C_I^{-tr} \in M_{n\times n}(\mathbf{Z})$. The Coxeter polynomial is the characteristic polynomial of the Coxeter matrix. We are interested in $\mathbf{Z}-$*congruence*, i.e. up to multiplication of C_I by a $\mathbf{Z}$-invertible matrix.

Can classify principal one-peak posets of degree ≤ 15. Assume upper triangular C_I (topologically sort the Hasse digraph!). Can generate posets of degree n inductively from $n-1$ — easy in parallel.

Say that a connected digraph is poset-type if it is a valid Hasse digraph (no shortcuts!). Outlined an isomorphism-testing algorithm $n \notin \{6, 7, 8\}$, which JHD didn't really follow.

Q Have you compared with Mathematica's built-in?

A Not yet: this is really only a sketch, not a complete algorithm. The restriction $n \notin \{6, 7, 8\}$ is because of the large number of different Dynkin diagrams.

4.5 Nominal Fusion Calculus

Various logics.

ZF with or without AC.

ZFA ZF with atoms.

FM Fraenkel–Mostowski, used to prove independence of AC of the rest of ZF.

FM set theory delivers a model of variable symbols and α-equivalence. We use binding symbols $\forall$ and И (nominal abstraction). We say that $\text{И}a.P(a)$ is true if $P(a)$ is true for all but finitely many a.

The fusion calculus tries to extend the Π-calculus. $z\lfloor x\rfloor$ means " send x along z" (and $\lceil \ldots \rceil$ means something subtly different). $\equiv$ means something similar, but with appropriate additional rules. We wish to encode freshness conditions with И.

$$\forall x, y \text{И} z \forall P, P' \frac{P \xrightarrow{[x/y]} P'}{[z]P \xrightarrow{[x/y]} [z]P'} \qquad \text{nominal}$$

Q Have you looked at complexity, or even decidability?

A Not yet, but aim to translate into nominal Isabelle (`http://isabelle.in.tum.de/nominal/`).

4.6 : Tarau

By using pairing functions (bijections $\mathbf{N} \times \mathbf{N} \to \mathbf{N}$, we can encode ODBTs (Ordered Binary Decision Trees). This inflation/interleaving is also known as Morton code or Z-code. This encoding will also extend to binary trees with integer values.

Chapter 5

Miscellanea

5.1 Kohlhase

He went to a “spreadsheet risks” conference. One Fortune 500 company did a spreadsheet audit, thinking they had 700K in active use. Ended up a count with 13.5M!

JHD “a large spreadsheet is a mistake”.

MK “but a common one”.

Bibliography

[Bab30] C. Babbage. On notations. *Edinburgh Encyclopaedia*, 15:394–399, 1830.

[Dav12] J.H. Davenport. Program Verification in the presence of complex numbers, functions with branch cuts etc. `http://opus.bath.ac.uk/31670/`, 2012.

[Rec57] R. Recorde. The Whetstone of Witte. *London*, 1557.

[Wie00] F. Wiedijk. The De Bruijn Factor. `http://www.cs.kun.nl/~freek/notes/factor.pdf`, 2000.

[YO81] R.M. Young and T. O'Shea. Errors in Children's Subtraction. *Cognitive Science*, 5:153–177, 1981.