The Quest for Better Languages

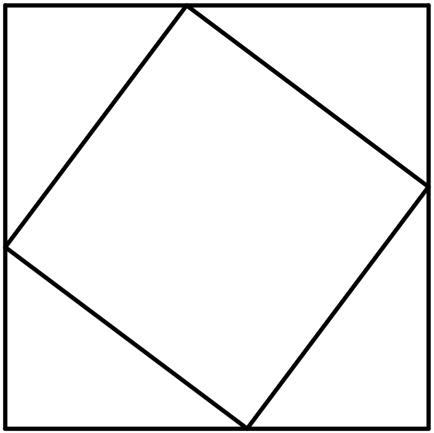
Alessio Guglielmi

University of Bath

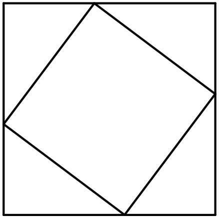
25 November 2015

Outline

The dream



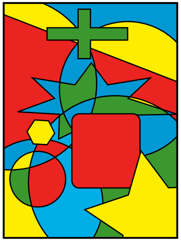
The dream (cont.)



- No syntax, no symbols, no words.
- Nothing is arbitrary here.
- An alien could understand this proof (of Pythagoras theorem of course).
- Is something like this possible for every proof? Does a natural language exist?
- Let us turn to the science of formal languages: logic.

The reality: The four colour theorem

Theorem: *Given any separation of a plane into contiguous regions, no more than four colours are required to colour the regions so that no two adjacent regions have the same colour.* [6]



```
Lemma sumt_ctree_pick_rev : forall t t', sumt (ctree_pick_rev t t') = Color0.
Proof.
move=> t' t; rewrite /ctree_pick_rev; set cs0 : colseq := seq0.
have: Color0 +c sumt cs0 = Color0 by done.
elim: t cs0 {1 3}Color0 => [t1 Ht1 t2 Ht2 t3 Ht3|lf _|] et e //.
  move=> Het /=; set cprr := ctree_pick_rev_rec.
  case Det1: (cprr _ _ _ t1) => [|e1 et1].
    case Det2: (cprr _ _ _ t2) => [|e2 et2].
      by apply: Ht3; rewrite [Color3]lock /= -addcA addc_inv.
    by rewrite -Det2; apply: Ht2; rewrite [Color2]lock /= -addcA addc_inv.
  by rewrite -Det1; apply: Ht1; rewrite [Color1]lock /= -addcA addc_inv.
by move=> Het /=; case (ctree_mem t' (etrace (belast e et))).
Qed.
```

- Proof: 100s of pieces of code such as the above! [1]
- An extraordinary achievement, but an alien would not understand: bureaucracy.

Can we free proofs and programs from the idiosyncrasies of language?

Consequences of poor languages

BUGS!

Explosion of the first Ariane 5 (1996) – Source: ESA

We are building software like in the Middle Ages they were building cathedrals. Computer ‘science’ is still not a science.

Formal languages allow us to compute

For example:

$$y = x^2 - 4$$

is a finite representation of an infinite object:

$$\begin{array}{ccc}
x & & y \\
\hline
 & \vdots & \\
-1000 & \rightarrow & 999996 \\
 & \vdots & \\
-10.5 & \rightarrow & 106.25 \\
 & \vdots & \\
\pi & \rightarrow & \pi^2 - 4 \\
 & \vdots & \\
20 & \rightarrow & 396 \\
 & \vdots &
\end{array}$$

We only can compute with finite representations.

Formal languages allow us to prove

There is someone in the pub such that, if he or she is drinking, then everyone in the pub is drinking.

Proof:

$$\cfrac{\cfrac{\mathrm{t}}{\exists x \forall y \left(\boxed{\cfrac{\mathrm{f}}{\overline{p(x)}}} \vee p(y) \right) \vee \left(\overline{p(x)} \vee \boxed{\cfrac{\mathrm{f}}{p(y)}} \right)}}{\exists x \forall y \left(\overline{p(x)} \vee p(y) \right)}$$

- This is easy, and actually not paradoxical,
- but as the Russell paradox teaches us, we need to be careful.

We need formal languages that guarantee correctness.

Computer science is logic plus complexity theory

- Turing was a logician, as tens of the founders were (and still are).
- (In the beginning at least) logicians were pure mathematicians.
- Mathematicians are more interested in the 'what' than in the 'how':

  What can we compute?

  What functions? What problems can we solve?

- The 'how' is important too:

  How long does it take to solve a problem?

  Can we easily solve the problems whose solutions we can easily check?

  The latter is the famous 'P vs. NP' problem, one of the hardest open problems in maths. If P = NP the world economy collapses!

My research: Open problems

- These problems are embarrassingly open:

  Are two given proofs the same?

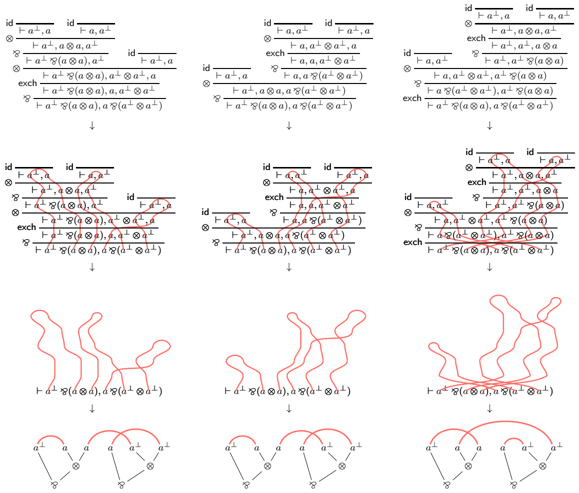
  Are two given algorithms the same?

- First formulated by Hilbert in 1900 [5].
- How many mathematical structures do you know for which 'sameness' is not defined? This is about the 'how'.
- Answering those questions means finding better languages.
- In order to answer, building (engineering) is not enough:

  We need to understand and discover (science).

We improve logic by introducing geometry

We remove bureaucracy from proofs and programs so that we can compare shapes:



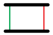
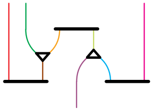
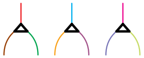
$\vdash a^\perp \parr (a \otimes a), a \parr (a^\perp \otimes a^\perp)$

Picture taken from [4]

We improve logic by reducing complexity

We control complexity by better composition mechanisms via deep inference [2]

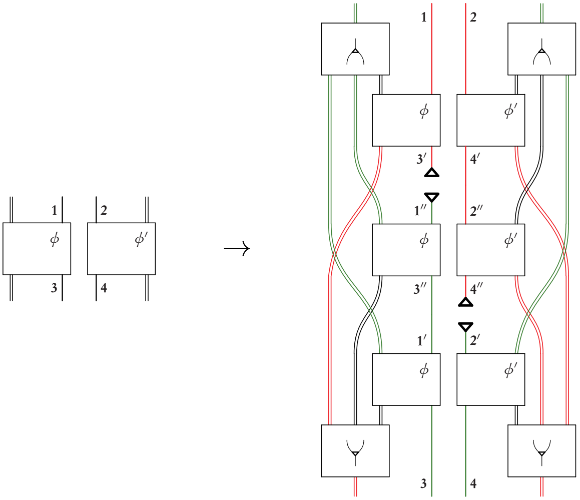
$${}_{\mathsf{s}}\frac{{}_{\mathsf{m}}\dfrac{\dfrac{\mathsf{t}}{a\vee\bar a}}{[a\vee\mathsf t]\wedge[\mathsf t\vee\bar a]}}{\left[{}_{\mathsf{s}}\dfrac{[a\vee\mathsf t]\wedge\bar a}{\dfrac{a\wedge\bar a}{\mathsf f}\vee\mathsf t}\quad\vee\quad\mathsf t\right]}
\qquad
=\frac{\left({}_{\mathsf{s}}\dfrac{a\wedge\left[\bar a\vee\dfrac{\mathsf t}{\bar a\vee a}\right]}{\dfrac{a\wedge\dfrac{\bar a\vee\bar a}{\bar a}}{\mathsf f}\quad\vee\quad\dfrac{a}{a\wedge a}}\quad\wedge\quad\bar a\right)}{a\wedge\dfrac{a\wedge\bar a}{\mathsf f}}
\qquad
{}_{\mathsf{m}}\frac{\dfrac{a}{a\wedge a}\vee\dfrac{b}{b\wedge b}}{[a\vee b]\wedge[a\vee b]}\quad\wedge\quad\frac{a}{a\wedge a}$$

What in the syntax is artificially related becomes geometrically independent.

We transform proofs and programs by manipulating shapes

Syntactic properties become geometric properties [3]:



What should we expect in twenty years?

We will build software as reliably as we build bridges (and cathedrals) now, but there will be other benefits. Just an example:

All of maths ($\approx$ 100,000,000 pages) represented as a semantic database.

We could:

- trust proofs (because they are automatically verified);
- access proofs at different abstraction levels (detail, just the idea, etc.);
- produce proofs by delegating routine tasks to the computer (with artificial intelligence?);
- …

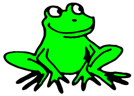
All fields of science will benefit.

Conclusion

“Computer science is no more about computers than astronomy is about telescopes.” (Edsger Dijkstra?)

“Be curious and ambitious, and do not limit yourself to building stuff – try also to understand and discover. And do not do it for the money but do it for the fun.” (me)

“Electric lamps were not invented by improving candles” (Carlo Rubbia?)



References

[1] Georges Gonthier (2008): *Formal Proof—The Four-Color Theorem.*
Notices of the American Mathematical Society 55(11), pp. 1382–1393.
Available at http://www.ams.org/notices/200811/tx081101382p.pdf.

[2] Alessio Guglielmi: *Deep Inference.*
Web site at http://alessio.guglielmi.name/res/cos.

[3] Alessio Guglielmi & Tom Gundersen (2008): *Normalisation Control in Deep Inference Via Atomic Flows.*
Logical Methods in Computer Science 4(1), pp. 9:1–36, doi:10.2168/LMCS-4(1:9)2008.
Available at http://arxiv.org/pdf/0709.1205.pdf.

[4] Lutz Straßburger (2006): *Proof Nets and the Identity of Proofs.*
Technical Report 6013, INRIA.
Available at http://hal.inria.fr/docs/00/11/43/20/PDF/RR-6013.pdf.

[5] Rüdiger Thiele (2003): *Hilbert's Twenty-Fourth Problem.*
American Mathematical Monthly 110, pp. 1–24, doi:10.2307/3072340.

[6] Wikipedia: *Four Color Theorem.*
Available at https://en.wikipedia.org/wiki/Four_color_theorem.