Creation and automation in proofs

Gilles Dowek
1. Sam’s famous carrot cake
Preheat oven to 350 degrees F. Grease and flour an 8x12 inch pan.

In a medium bowl, sift together flour, baking soda, salt and cinnamon. Set aside.

In a large bowl, combine eggs, buttermilk, oil, sugar and vanilla. Mix well. Add flour mixture and mix well.

In a medium bowl, combine shredded carrots, coconut, walnuts, pineapple and raisins.

Using a large wooden spoon or a very heavy whisk, add carrot mixture to batter and fold in well.

Pour into prepared 8x12 inch pan, and bake at 350 degrees F for 1 hour. Check with toothpick.

Allow to cool for at least 20 minutes before serving.
Making a carrot cake for the first time

Combining shredded carrots, coconut, walnuts, pineapple and raisins: a creative action

This creative action activates: shredding the carrots: an automatic action
Deciding to repeat the main theme, or to transpose it up by one semitone: a **creative** action

This creative action activates: transposing the theme: an **automatic** action
Going for a walk in Kyoto

Deciding to go from Ginkaku-ji to Nanzen-ji: a creative action

This creative action activates: putting one foot in front of the other: an automatic action
Creative and automatic actions

Creative: conscious, intentional, ...
Automatic: non-conscious, non-intentional, ...

Different parts of the brain involved

May be relative: making a carrot cake automatic after a while (activated by: organizing a tea party)

What it is not: algorithmic / non-algorithmic
Rather: an algorithm we can explain / an algorithm we cannot
Creative and automatic actions in mathematical proofs

Prove that 221 is composite

Find a divisor: 13

Check that 13 is a divisor of 221: divide 221 by 13
In predicate logic

No distinction between finding and checking
between proving and computing

Everything flattened: creative by default
II. From creative actions to automatic actions in proof search
Associativity

How can you prove

$$(a+(b+(c+d)))+((e+f)+g) = (a+(b+c))+((d+e)+(f+g))$$

using the associativity of addition

$$\forall x \forall y \forall z \ ((x + y) + z = x + (y + z))$$

?
Paramodulation

Each time we have a term of the form \((x + y) + z\) replace it by \(x + (y + z)\) (and vice-versa) until \(t = t\)

Around 10 possibilities for the first step

Around 10 possibilities for the second step

... 

Around \(10^{10}\) possibilities to explore
Knuth-Bendix 1970

**Transform** the axiom

\[ \forall x \forall y \forall z ((x + y) + z = x + (y + z)) \]

into a rewrite rule

\[ (x + y) + z \rightarrow x + (y + z) \]

and rewrite the proposition to be proved

\[ a + (b + (c + (d + (e + (f + g)))))) = a + (b + (c + (d + (e + (f + g)))))) \]
When reducing

- always replace \((x + y) + z\) by \(x + (y + z)\) and not the converse

- chose any subterm of the form \((x + y) + z\) and never try another

Still: if \(t_1 = t_2\) provable it rewrites to \(t = t\) thanks to confluence and termination

Replace a creative activity by an automatic one
Reducing in a first step does not work with quantifiers

Prove $\exists x \ (x + c = a + (b + c))$

Nothing to reduce

Yet, $x + c$ and $a + (b + c)$ not unifiable

(no term $t$ such that $t + c$ identical to $a + (b + c)$)
In the proof with the associativity axiom: first substitute, yielding 

\[(a + b) + c = a + (b + c)\]

then apply associativity

Unification must be performed modulo associativity:

There exists \( t \) s.t. \( t + c \) and \( a + (b + c) \) same reduced form

Plotkin: exchange the associativity axiom with a replacement of 
unification by equational unification
III. From creative actions to automatic actions in proofs
Deduction modulo theory

How can we prove the completeness of Plotkin’s method?

\[ \text{assoc} \vdash A \iff A \text{ a proof by Plotkin’s method} \]
Deduction modulo theory

How can we prove the completeness of Plotkin’s method?

\[ \text{Assoc} \vdash A \iff \vdash_{\text{Assoc}} A \iff A \] a proof by Plotkin’s method

What is \( \vdash_{\text{Assoc}} A \)?
What is $\vdash_{Assoc} A$?

The usual deduction rules (sequent calculus, natural deduction, ...) but performed on propositions modulo associativity

e.g.

\[
\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}
\]

replaced by

\[
\frac{\Gamma \vdash C \quad \Gamma \vdash A}{\Gamma \vdash B} \quad \text{if } C \rightarrow^* (A \Rightarrow B)
\]
An example

\[
\forall x (x = x) \vdash_{Assoc} \forall x (x = x)
\]

\[
\forall x (x = x) \vdash_{Assoc} a + (b + c) = a + (b + c)
\]

\[
\forall x (x = x) \vdash_{Assoc} \exists y (y + c = a + (b + c))
\]

Normativity: proof-normalization = confluence
Rewriting terms and rewriting propositions

$$(x + y) + z \rightarrow x + (y + z)$$

always a one-point model: no way to prove $0 \neq 1$

$$0 = 0 \rightarrow \top$$

$$S(x) = 0 \rightarrow \bot$$

$$0 = S(y) \rightarrow \bot$$

$$S(x) = S(y) \rightarrow (x = y)$$
Rewriting terms and rewriting propositions

\[ N(x) \rightarrow \forall c \ ((0 \in c \land \forall y \ (y \in c \Rightarrow S(y) \in c)) \Rightarrow x \in c) \]
Four (times the same) examples

• $C$ defined by computation rules $C(221) \rightarrow^* \top$

\[
\begin{align*}
\vdash C(221) & \quad \top\text{-intro} \\
\vdash 13 \mid 221 & \rightarrow^* \top
\end{align*}
\]

\[
C(x) = \exists y \ (y \mid x)
\]

\[
\begin{align*}
\vdash 13 \mid 221 & \quad \top\text{-intro} \\
\vdash \exists y \ (y \mid 221) & \quad \exists\text{-intro}
\end{align*}
\]
• × and = defined by computation rules

\[(221 = 13 \times 17) \rightarrow^* \top\]

\[
C(x) = \exists y \exists z \ x = y \times z
\]

\[
\begin{align*}
\vdash 221 &= 13 \times 17 & \top\text{-intro} \\
\vdash \exists z \ (221 = 13 \times z) & \exists\text{-intro} \\
\vdash \exists y \exists z \ (221 = y \times z) & \exists\text{-intro}
\end{align*}
\]
• no computation rules

\[ \vdash \exists z \ (221 = 13 \times z) \quad \exists\text{-intro} \]

\[ \vdash \exists y \exists z \ (221 = y \times z) \quad \exists\text{-intro} \]
New normativity issues

modulo $P \rightarrow (Q \Rightarrow P)$ proof-reduction terminates

not modulo $P \rightarrow (P \Rightarrow Q)$
IV. More on proof search
Resolution modulo theory 2003, 2010

Prove $a \subseteq a$ from $x \subseteq y \iff \forall z \ (z \in x \Rightarrow z \in y)$

Resolution

\[ \neg x \subseteq y \lor \neg z \in x \lor z \in y \]

\[ f(x, y) \in y \lor x \subseteq y \]

\[ \neg f(x, y) \in y \lor x \subseteq y \]

\[ \neg a \subseteq a \]
Resolution modulo theory 2003, 2010

Prove $a \subseteq a$ from $x \subseteq y \rightarrow \forall z \ (z \in x \Rightarrow z \in y)$

Resolution modulo theory

\[ \neg x \subseteq y \lor \neg z \in x \lor z \in y \]
\[ f(x, y) \in y \lor x \subseteq y \]
\[ \neg f(x, y) \in y \lor x \subseteq y \]
\[ \neg a \subseteq a \]

$a \subseteq a$ computed to $\forall z \ (z \in a \Rightarrow z \in a)$
Resolution modulo theory 2003, 2010

First combination of literal selection based method (ordered resolution, ...) and clause selection base method (set of support, ...)

Burel 2011: 10 to 100 times faster than ordered resolution for some specific theories

∩, ∪, \, ...: definitions, just unfolded, but on demand

Similar result for Delahaye-Halmagrand tableaux method
V. Psychologism?
Where do the deduction rules come from

\[
\frac{A \land B}{A}
\]

Not from nature

Not from the structure of our brain

From the meaning of \( \land \)
But ...

Quantum logic
Quantum logic: a language to express a theory

Not more in nature than theorems of the theory of Hilbert spaces

Quantum logic could be defined (but would not be as useful) if nature were different

Same holds for the notion of computable function (Deutsch)
In the same way

The rules of Deduction modulo theory are not wired in our brains

Could be defined if our brain were different

But modelizes a difference between creative and automatic actions, that is (?) wired in our brain