The Future of Mathematics:
A Personal View and Comments on Math Education

Keynote at TIME 2014, Krems, Austria
July 2, 2014

Dedicated to Helmut Heugl

Bruno Buchberger
RISC (Research Institute for Symbolic Computation)
Johannes Kepler University, A4040 Linz, Austria

Society

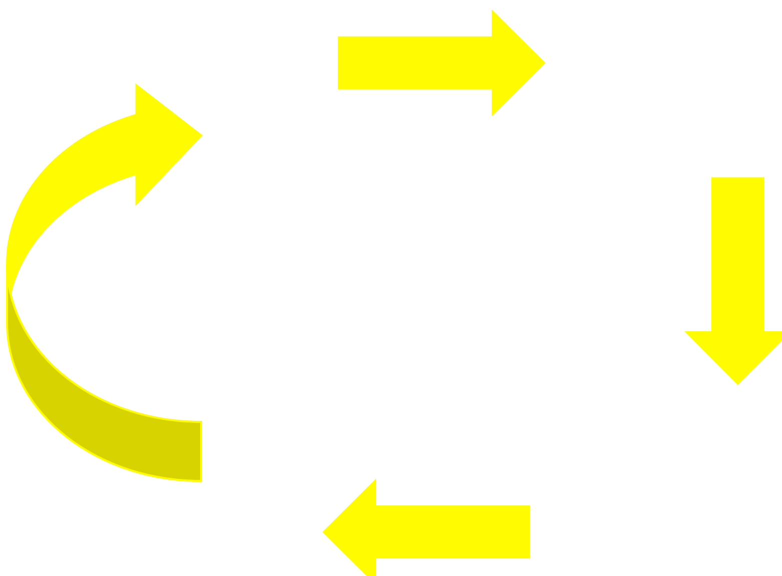
Science & Technology

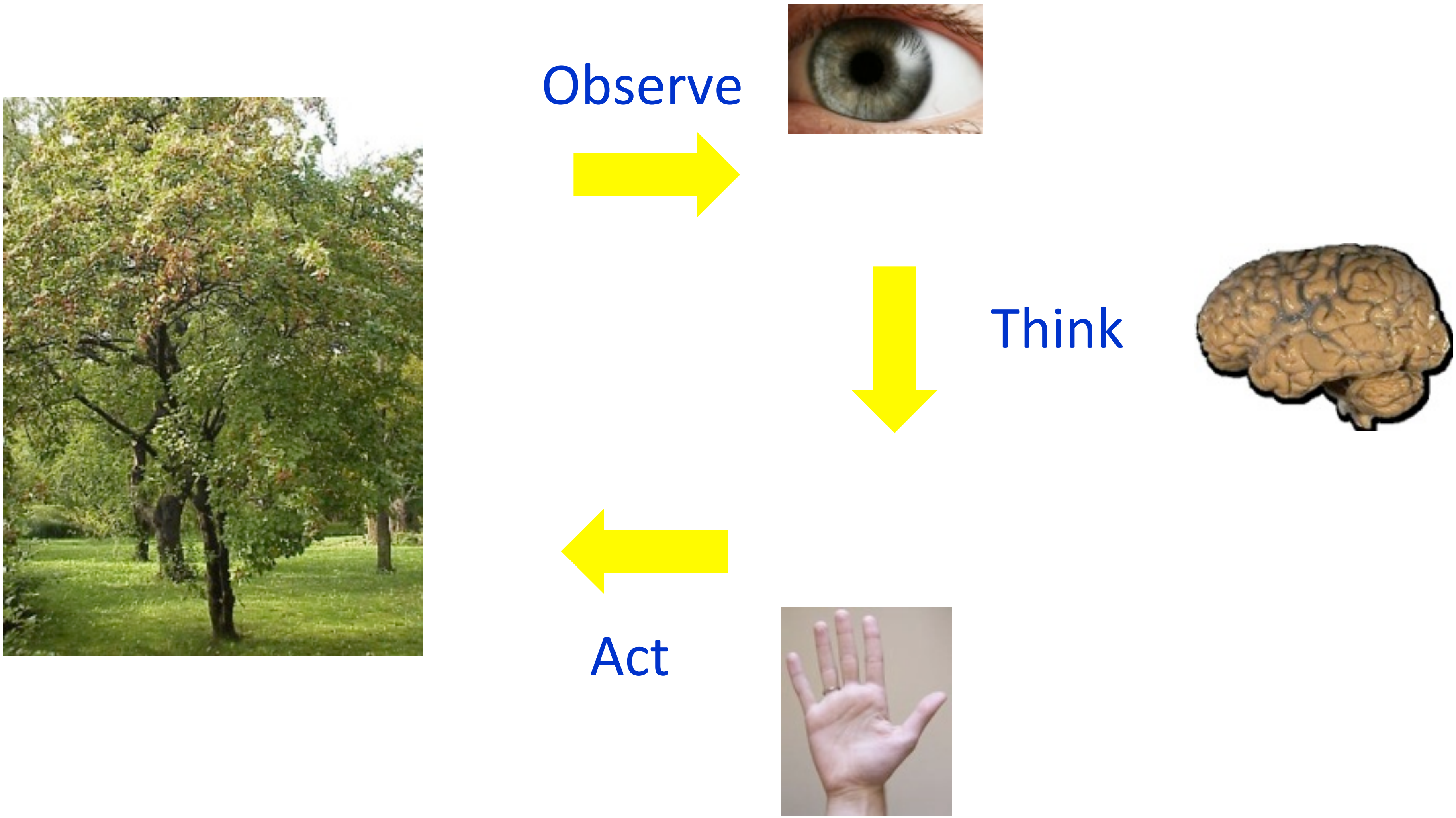
Mathematics

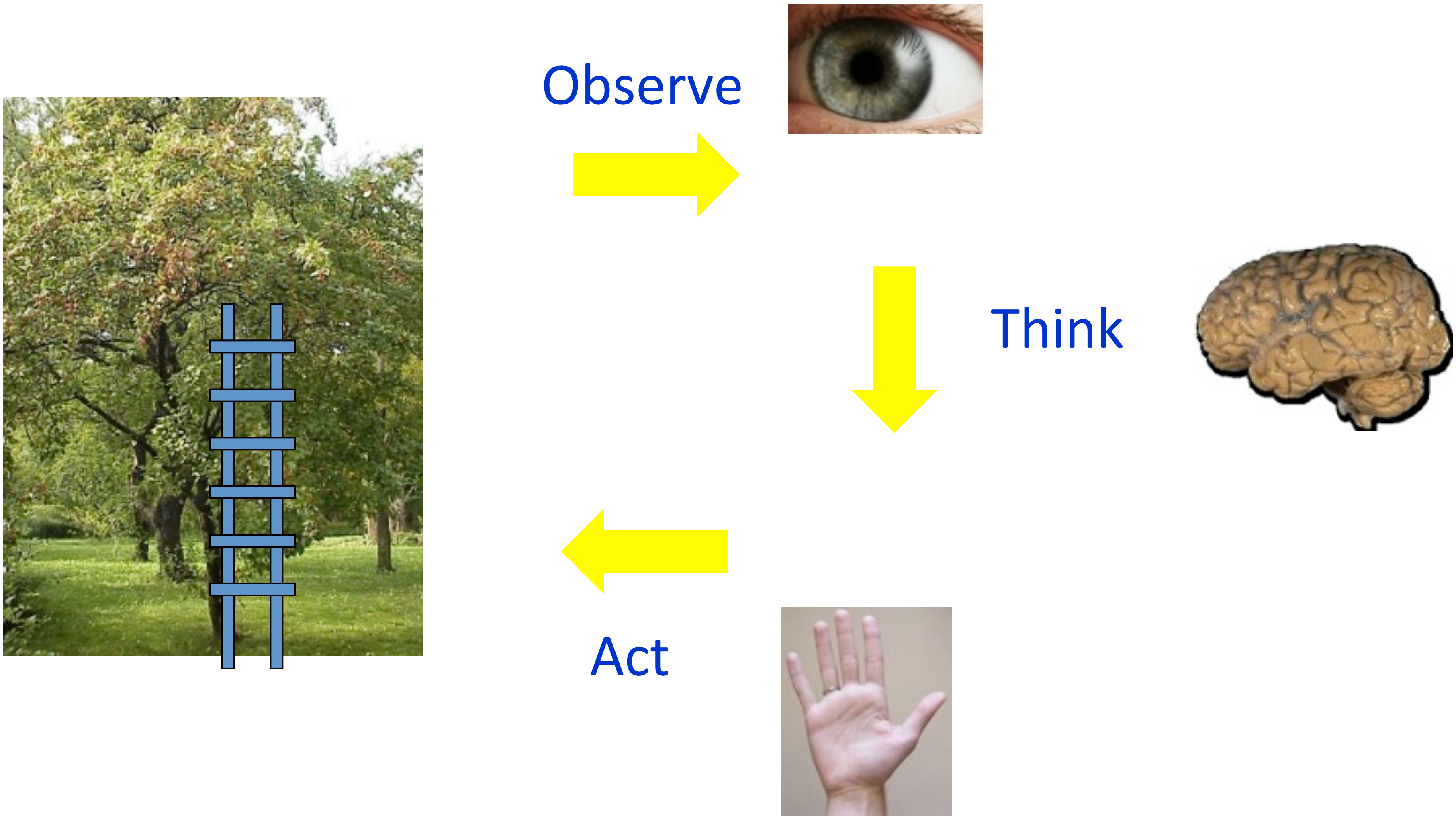
Math Edu

Where does this go?

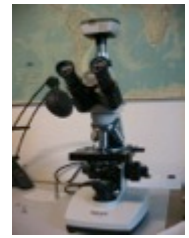
Start from (before) Adam and Eve …



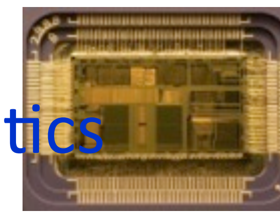

Observe
Think
Act


Observe
Think
Act

Science

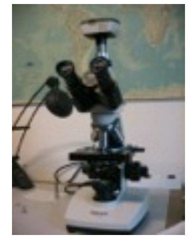


Mathematics



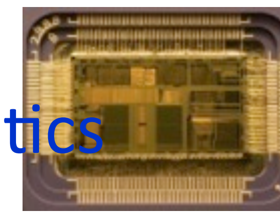
Technology

Science



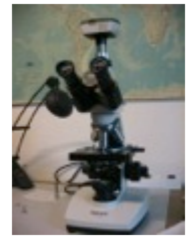
Intersubjective Verifiability

Mathematics



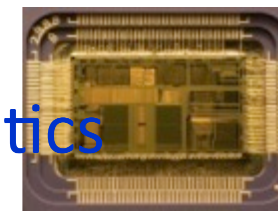
Technology

Science

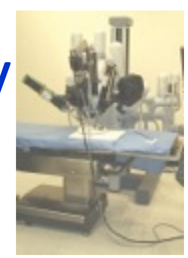


Anonymous
Peer
Reviewing

Mathematics



Technology



Science

Mathematics

“Of course”, a good scientist (mathematician, engineer, teacher, student, …, university, school, …) tries to live all three aspects equally well.

Technology

Science

Mathematics

However, conceptually,
the three arrows are different!

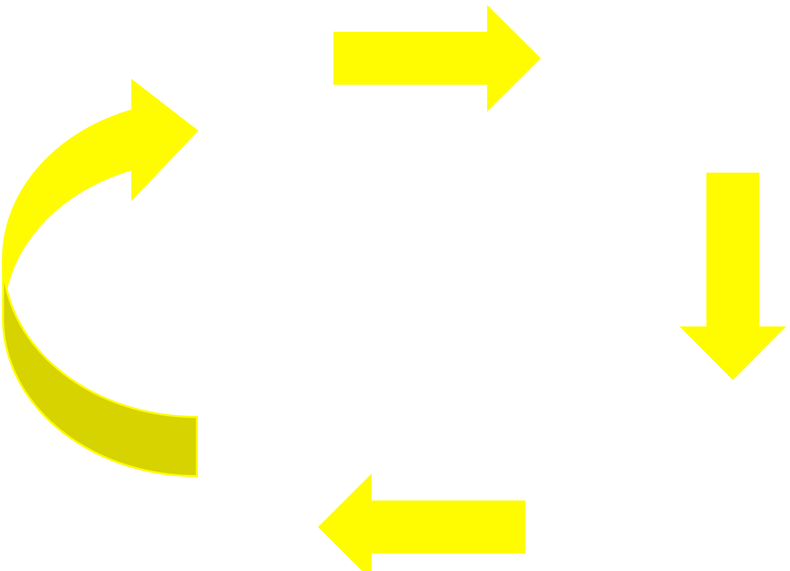
Technology

Science

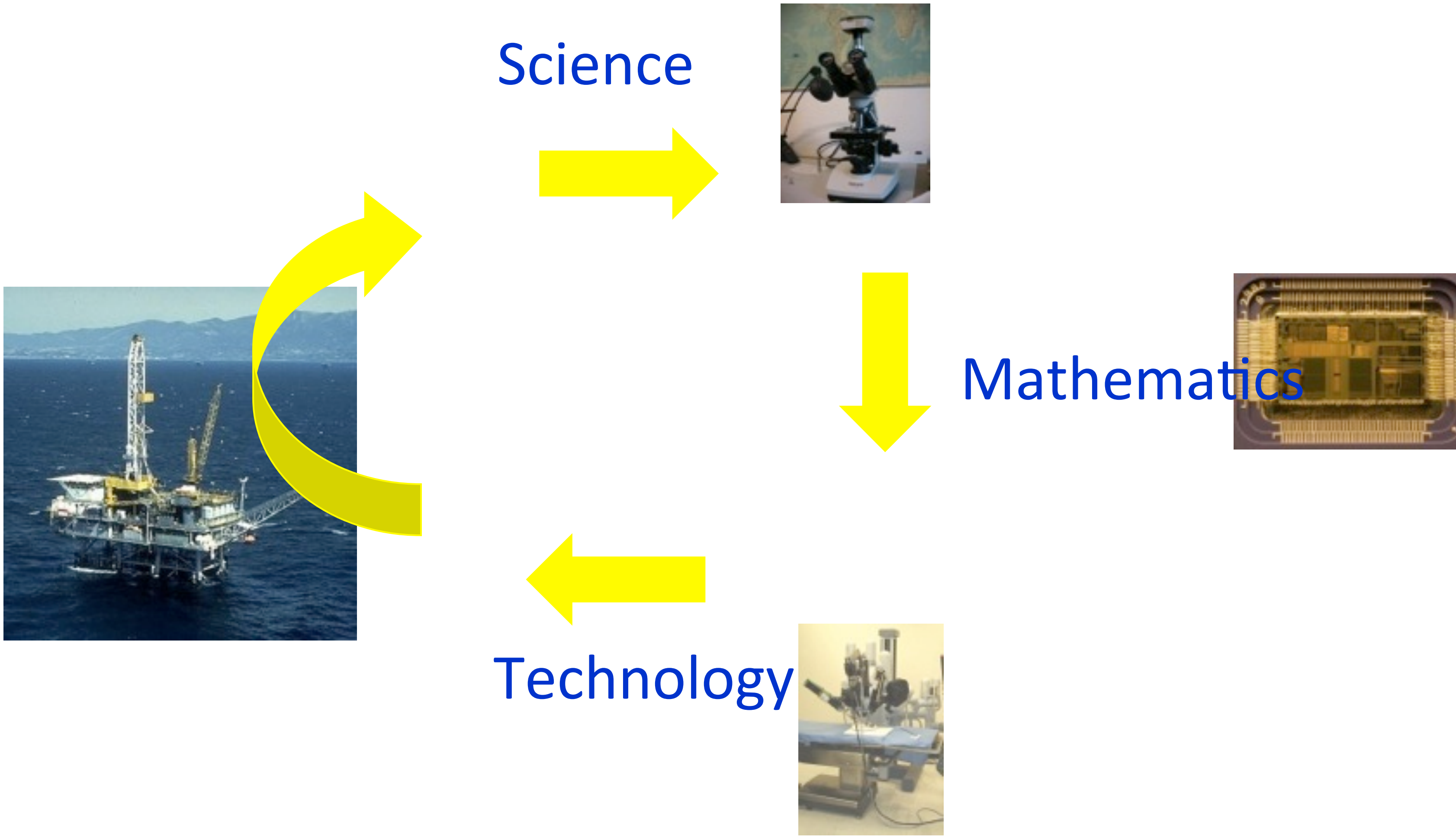
Thus, now, mathematics is essentially the art of gaining knowledge and solving problems by thinking (reasoning, ...)

Mathematics

Technology



Self-Application



Self-application:

the intelligence of nature,

the nature of intelligence.

By self-application,
the science / techno / eco spiral
gains speed, sophistication, …

The history of science
is the history of

how much these arrows become explicit.

If one understands this

then one can look (forward) to the future.

“Ancient” Mathematics: “see the truth”

- observe (“mathematical”) objects in reality
- and “see” a (general) truth.

"Ancient" Mathematics: "see the truth"

No clear distinction between

"seeing" = observing
and
"seeing" = thinking

“Modern” Mathematics: “prove new truth from observed truth”

Clear distinction between

“seeing” = observing
and
“seeing” = thinking (reasoning, … proving)

(observe)

(a+b) (a+b) = c.c + 4.(a.b)/2 =

(think)

= a.a + 2.a.b + b.b

Mathematics of 20th Century

mathematical logic:

proof = finite sequence of thinking steps
on finite language objects

result: “all mathematical truth can be built up
from simple truth”
in finitely many proof steps (Gödel, “Bourbaki”)

result: “all mathematical methods can be expressed
in the language of logic (“computer”)

Note:

In the 60 volumes of Bourbaki,
the word "computer" does not occur !

Unfortunately, in 20th century, mathematics split into

"pure" mathematics and
"computational" mathematics.

Computational mathematics was “only” concerned with “approximate” problems using “approximate” numbers:

$x^2 + b\,x + c = 0, \quad x = ?$

in case: b = 3, c = 5:

x = -1.5… ± 1.65831… i

Around 1950: “Symbolic Computation”:

x = 1/2 (-b ± Sqrt (b^2 - 4 c)).

If you want, plug in b = 3, c = 5.

How far can “symbolic computation” go?

Answer: Very far !
But may be very difficult.
Why difficult? Or even impossible?

1950 – now: Symbolic algorithms
for traditional mathematical problems
(e.g. general non-linear systems,
integration,
differential equations, …)

A common misunderstanding:

(Numeric or symbolic) algorithms are just a „silly" iteration of simple steps.

The truth:

Algorithms need „deeper" mathematics than „pure mathematics".

Mathematics of 21st Century

Of course, rich „normal math" will go on.

However, math of 21th century will be „self-application" of math to itself. In other words, „symbolics" will be the essence.

Symbolics 1st floor: Invent new mathematics for symbolic algorithms for traditional math problems.

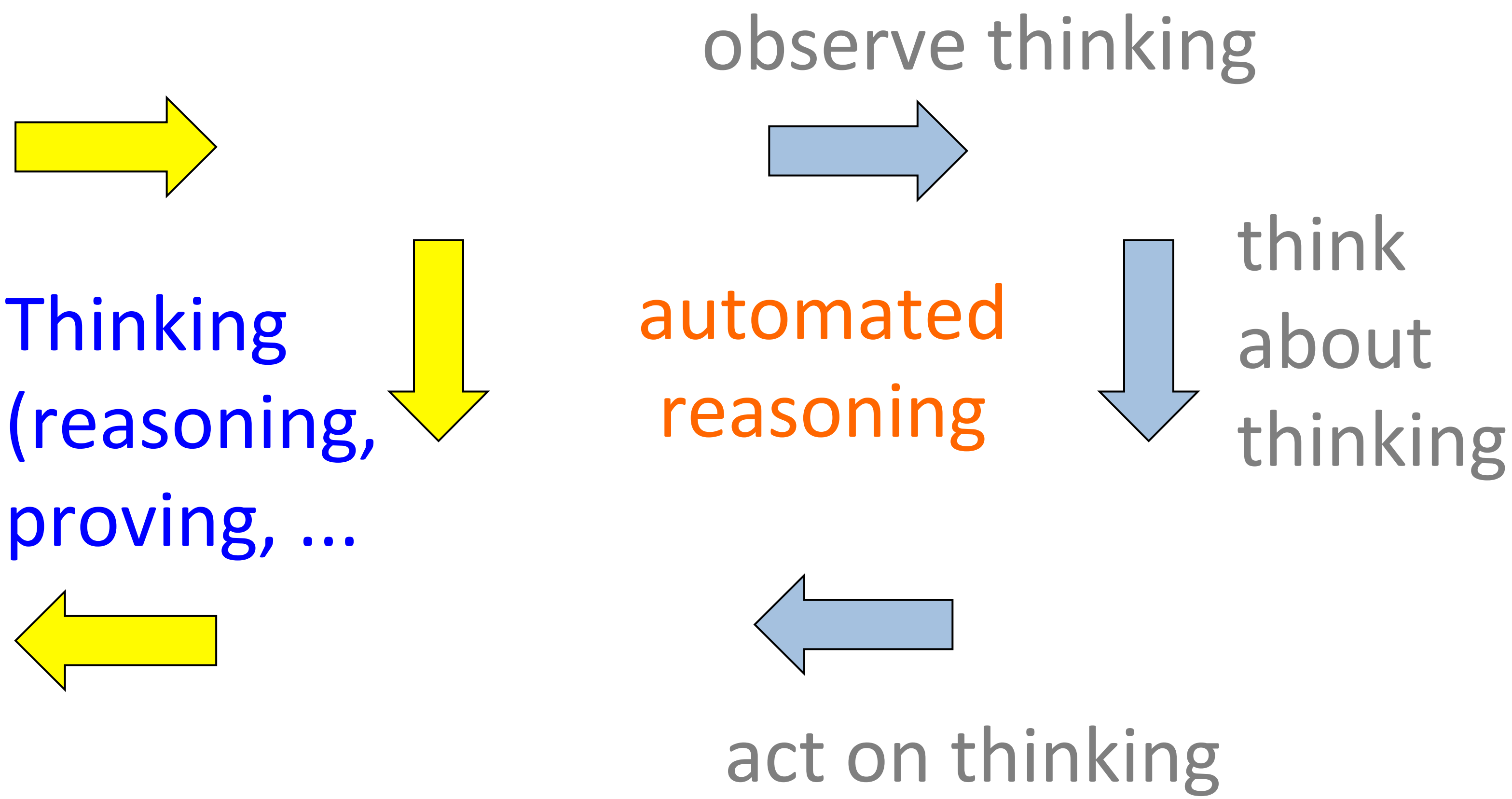
Symbolic 2nd floor: Invent algorithms for inventing and proving new mathematics.

Mathematics of 21st Century

Theory, algorithms, procedures, software,, ...

- for inventing definitions
- for inventing and proving theorems
- for inventing problems
- for inventing and proving algorithms
- for building up and managing mathematical theories in a structured way.

Self-application of math to math:



An Example of “Two Floors Math”

- „first floor“ (1965 -...): Gröbner bases
- „second floor“ (1995 - ...): Theorema (system)

- „first floor“ (BB 1965 PhD thesis -...): Gröbner bases solved a 65 years open problem (specified 1899).

- „second floor“ (BB 1995 Lazy Thinking -): Can generate the main idea of the 1965 PhD thesis automatically from the 1899 problem specification.

Details of Gröbner Bases and Lazy Thinking: a little bit difficult.

Emphasis in this: On the methodological significance.

First Floor: Gröbner Bases

The Gröbner Bases Problem (sloppily):

Given: F, a system of multivariate polynomials.

Find: G with same set of solutions
and all variables decoupled.

The Main Idea of Gröbner Bases Theory and Algorithmics (BB 1965):

Theorem: F is a Gröbner basis iff
all "S-polys", w.r.t. F can be reduced to 0.

S-poly(f,g) := u . f - v . g,

where u, v are such that

u . L(f) = v . L(g) = LCM (L(f), L(g)).

Algorithm: For obtaining Gröbner bases G for F,

iterate the formation of S-polys

until all of them reduce ("divide") to 0.

Second Floor: Lazy Thinking

The Algorithm Synthesis Problem:

Given: A problem specification P.

Find: An algorithm A such that

for all inputs x, P(x, A(x)).

Examples:

P(x,y) iff y.y = x.

P(L,S) iff S is a sorted version of list L.

P(f,F) iff F' = f.

P(F,G) iff G is a Gröbner bases for F.

The Lazy Thinking Method for Algorithm Synthesis:

Given: A problem specification P.

1st Step: Try out (one of finitely many) algorithm schemes A.

(Example of an algorithm scheme: "divide and conquer":

A(x) := S(x) if x is basic
M (A (L(x)), A (R(x)) if x is not basic.)

2nd Step: Try to prove (automatically): for all x, P(x, A(x)).

The proof will probably fail because nothing is known about the sub-algorithms S, M, L, R, …

3rd Step (the kernel of lazy thinking): From the failing proof, try to derive (automatically) specifications for the subalgorithms in the scheme (e.g. S, M, L, R, …) that will make the proof work.

4th Step: Iterate Lazy Thinking until subalgorithm specifications are found for which algorithms are known.

When this Lazy Thinking procedure is applied to the specification P of the Gröbner bases problem (using the "critical pair / completion algorithm scheme") then, in a couple of minutes, one obtains the main idea of Gröbner bases theory, i.e.

- the central notion of S-poly,
- the main theorem on S-polys and Gröbner bases,
- and the algorithm for computing Gröbner bases.

What are the implications of this result:

- The power of formalization and automated reasoning.

- The power of self-application.

- Is more intelligence needed for the 2nd floor than for the 1st floor? …

Conclusions

- Math will always be in the center of the automation spiral.
- Math education, as the art of ex-plaining, must be (but is not) in the center of education.
- The art of ex-plaining is the more important the higher we go in the automation spiral.

Conclusions

- Focus of 21st century math: self-application. Automated mathematical invention and verification.
- The level of sophistication in the automation of mathematics has no upper bound. (Gödel!)

Oh, happy mathematics!

Conclusions

- Researching, applying, teaching math: Please,keep all aspects together!
- Science, math, technology: Please, keep all aspects together!
- Universities, ..., schools: Please, keep all aspects together!

Conclusions

- Build-up of web-accessible formal math knowledge bases (embracing current math software systems) will replace math journals.
- A significant part of the anonymous peer reviewing process will be automated.

Conclusions

- Using technology in math edu? Math = technology!
- Algorithms (on higher and higher levels) are goal and means of math and math edu.
- Teaching math using “technology”: Give students a chance to repeat evolution! (The White-Box / Black-Box Principle, BB 1989).
- The higher the “technology”, the more important the personal teacher!

Conclusions

- Is “observing – thinking – acting” good enough for dealing with nature?

- As you go forward in evolution, learn to go back in evolution! What does this mean? See next slide!

Contact:

Bruno Buchberger
Tel.: +43 (0) 664 4211646
bruno.buchberger@risc.jku.at

Professor of Computer Mathematics (1974 -)
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria, Europe

Founder and Head (1989-2013)
of Softwarepark Hagenberg