Reasoning with CAS is a Dead End

An Alternative: Systems that Explain Themselves

Walther Neuper

IICM, Institute for Computer Media,
University of Technology.
Graz, Austria

eduTPS: Working Group on Education and TP Technology
at CADGME, Targu Mures, Romania
Sep. 7, 2016

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME'16
   - Survey on Technology for Math
2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle
3. TP: "Systems that explain themselves"
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME’16
   - Survey on Technology for Math
2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle
3. TP: “Systems that explain themselves”
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Keynotes of CADGME'16

with explicitly mentioning *"proof"* or *"reasoning"*:

- *". . . A technology rich environment provides a working framework for experimentation, conjecturing and automated proofs (mostly using algorithms based on computations of Gröbner bases)."*
- *". . . Computer Algebra Systems and other digital technologies supporting mathematical reasoning are being implemented in educational situations all over the world. The transformational nature of these technologies is large but the educational value is unclear."*

Keynotes of CADGME’16

with explicitly mentioning *“proof”* or *“reasoning”*:

- *“. . . A technology rich environment provides a working framework for experimentation, conjecturing and automated proofs (mostly using algorithms based on computations of **Gröbner bases**).”*
- *“. . . Computer Algebra Systems and other digital technologies supporting mathematical reasoning are being implemented in educational situations all over the world. The transformational nature of these technologies is large but the educational value is unclear.”*

Where is **the specific technology built for reasoning**?

Keynotes of CADGME'16

with explicitly mentioning *"proof"* or *"reasoning"*:

- *". . . A technology rich environment provides a working framework for experimentation, conjecturing and automated proofs (mostly using algorithms based on computations of **Gröbner bases**)."*
- *". . . Computer Algebra Systems and other digital technologies supporting mathematical reasoning are being implemented in educational situations all over the world. The transformational nature of these technologies is large but the **educational value is unclear**."*

Where is **the specific technology built for reasoning**?
". . . unclear" because essentials of math are missed?

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME’16
   - Survey on Technology for Math

2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle

3. TP: “Systems that explain themselves”
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Technology for math

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
applied sciences	numerical computation	Spreadsheets (SSH)
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	CAS
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
	numerical computation	Spreadsheets (SSH)
applied sciences		
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	CAS
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
applied sciences	numerical computation	Spreadsheets (SSH)
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	CAS
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
applied sciences	**numerical computation**	Spreadsheets (SSH)
math. education	**symbolic computation**	
	Computer Algebra Macsyma 1968	CAS
	Theorem Proving (TP) Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
	geometry	CAD/CAM DGS
mathematics		
	numerical computation	Spreadsheets (SSH)
applied sciences		
	symbolic computation	
math. education		
	Computer Algebra Macsyma **1968**	CAS
	Theorem Proving (TP) Automath **1967**	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	**CAD/CAM** DGS
	numerical computation	Spreadsheets (SSH)
applied sciences		
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	**CAS**
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	**CAD/CAM** DGS
applied sciences	numerical computation	**Spreadsheets (SSH)**
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	**CAS**
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
applied sciences	numerical computation	Spreadsheets (SSH)
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	CAS
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	**CAD/CAM** DGS
applied sciences	numerical computation	**Spreadsheets (SSH)**
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	**CAS**
	Theorem Proving Automath 1967	**TPS** **TP-based Systems**

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
applied sciences	numerical computation	**Spreadsheets (SSH)**
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	**CAS**
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM DGS
applied sciences	numerical computation	Spreadsheets (SSH)
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	CAS
	Theorem Proving Automath 1967	TPS

Survey on Math. Software

In science of ...	SW-tools for ...	... standardized as ..
mathematics	geometry	CAD/CAM **DGS**
applied sciences	numerical computation	**Spreadsheets (SSH)**
math. education	symbolic computation	
	Computer Algebra Macsyma 1968	**CAS**
	Theorem Proving Automath 1967	**TPS !?!**

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME’16
   - Survey on Technology for Math

2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle

3. TP: “Systems that explain themselves”
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Success Stories for TP

Although primarily intended for *software verification*, TPs are becoming indispensable in mathematics, too; for instance

1. Four Colour Theorem: proved within TP "Coq" by Georges Gonthier in 2005
2. Kepler Conjecture (now: Theorem) proved by TPs "HOL Light" and "Isabelle" under the leadership of Tom Hayles in 2014
3. ...

Success Stories for TP

Although primarily intended for *software verification*, TPs are becoming indispensable in mathematics, too; for instance

1. Four Colour Theorem: proved within TP “Coq” by Georges Gonthier in 2005
2. Kepler Conjecture (now: Theorem) proved by TPs “HOL Light” and “Isabelle” under the leadership of Tom Hayles in 2014
3. . . .

Success Stories for TP

Although primarily intended for *software verification*, TPs are becoming indispensable in mathematics, too; for instance

1. Four Colour Theorem: proved within TP “Coq” by Georges Gonthier in 2005
2. Kepler Conjecture (now: Theorem) proved by TPs “HOL Light” and “Isabelle” under the leadership of Tom Hayles in 2014
3. ...

Success Stories for TP

Although primarily intended for *software verification*, TPs are becoming indispensable in mathematics, too; for instance

1. Four Colour Theorem: proved within TP “Coq” by Georges Gonthier in 2005
2. Kepler Conjecture (now: Theorem) proved by TPs “HOL Light” and “Isabelle” under the leadership of Tom Hayles in 2014
3. . . .

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME'16
   - Survey on Technology for Math
2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle
3. TP: "Systems that explain themselves"
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Theorem Prover “Isabelle”

- Have a look at `http://isabelle.in.tum.de/`
- and see: (almost) all mathematics is mechanised `http://isabelle.in.tum.de/dist/library/HOL/`
- or download & install just now !

- Short demonstration of “Isabelle”

Theorem Prover “Isabelle”

- Have a look at `http://isabelle.in.tum.de/`
- and see: (almost) all mathematics is mechanised `http://isabelle.in.tum.de/dist/library/HOL/`
- or download & install just now !

- Short demonstration of “Isabelle”

Theorem Prover “Isabelle”

- Have a look at
  `http://isabelle.in.tum.de/`
- and see: (almost) all mathematics is mechanised
  `http://isabelle.in.tum.de/dist/library/HOL/`
- or download & install just now !

- Short demonstration of “Isabelle”

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME'16
   - Survey on Technology for Math

2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle

3. TP: "Systems that explain themselves"
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Advances of TP technology

In software based on TP technology ...

- each (formal) *element* (in a proof or a calculation) is **deduced** from first principles (axioms, definitions)
- each (formal) *operation* is **justfied** by axioms or proved properties
- **semantics** of formal objects (elements, operations) is completely mechanised in human readable format
- **interactive** manipulation provides **experience** while correctness is checked automatically
- so mechanical operation+deduction+justification provide "systems that explain themselves" (where *formal logic* replaces *common sense*).

Advances of TP technology

In software based on TP technology ...

- each (formal) *element* (in a proof or a calculation) is **deduced** from first principles (axioms, definitions)
- each (formal) *operation* is **justfied** by axioms or proved properties
- **semantics** of formal objects (elements, operations) is completely mechanised in human readable format
- **interactive** manipulation provides **experience** while correctness is checked automatically
- so mechanical operation+deduction+justification provide “systems that explain themselves” (where *formal logic* replaces *common sense*).

Advances of TP technology

In software based on TP technology …

- each (formal) *element* (in a proof or a calculation) is **deduced** from first principles (axioms, definitions)
- each (formal) *operation* is **justfied** by axioms or proved properties
- **semantics** of formal objects (elements, operations) is completely mechanised in human readable format
- **interactive** manipulation provides **experience** while correctness is checked automatically
- so mechanical operation+deduction+justification provide “systems that explain themselves” (where *formal logic* replaces *common sense*).

Advances of TP technology

In software based on TP technology . . .

- each (formal) *element* (in a proof or a calculation) is **deduced** from first principles (axioms, definitions)
- each (formal) *operation* is **justfied** by axioms or proved properties
- **semantics** of formal objects (elements, operations) is completely mechanised in human readable format
- **interactive** manipulation provides **experience** while correctness is checked automatically
- so mechanical operation+deduction+justification provide “systems that explain themselves” (where *formal logic* replaces *common sense*).

Advances of TP technology

In software based on TP technology . . .

- each (formal) *element* (in a proof or a calculation) is **deduced** from first principles (axioms, definitions)
- each (formal) *operation* is **justfied** by axioms or proved properties
- **semantics** of formal objects (elements, operations) is completely mechanised in human readable format
- **interactive** manipulation provides **experience** while correctness is checked automatically
- so mechanical operation+deduction+justification provide “systems that explain themselves” (where *formal logic* replaces *common sense*).

Outline

1. Mathematics – Reasoning – Technology
   - . . . in Keynotes of CADGME’16
   - Survey on Technology for Math
2. Theorem-Prover (TP) Technology !
   - Success Stories for TP
   - Demo: TP Isabelle
3. TP: “Systems that explain themselves”
   - Advances of TP Technology
   - New Challenges for Software Development. . .

Challenges for development ...

... are to exploit the power of TP technology in order to ...

- support all phases of mathematical problem solving (phase of modelling, specifying, solving) ...
  ... all math is concerned with reasoning!
- make the system intuitive such that justifications are evident for manipulations (if asked for)
- support solution of engineering problems close to traditional paper&pencil representation
- make problem solving selfexplanatory by
  - checking steps input by students
  - generating explanations on demand
  - proposing a next step, if a student gets stuck.
- foster confidence in abstract concepts by providing experience with formal operation.

Challenges for development ...

...are to exploit the power of TP technology in order to ...

- support all phases of mathematical problem solving (phase of modelling, specifying, solving) ...
  ...all math is concerned with reasoning!
- make the system intuitive such that justifications are evident for manipulations (if asked for)
- support solution of engineering problems close to traditional paper&pencil representation
- make problem solving selfexplanatory by
  - checking steps input by students
  - generating explanations on demand
  - proposing a next step, if a student gets stuck.
- foster confidence in abstract concepts by providing experience with formal operation.

Challenges for development …

…are to exploit the power of TP technology in order to …

- support all phases of mathematical problem solving (phase of modelling, specifying, solving) …
  …all math is concerned with reasoning!
- make the system intuitive such that justifications are evident for manipulations (if asked for)
- support solution of engineering problems close to traditional paper&pencil representation
- make problem solving selfexplanatory by
  - checking steps input by students
  - generating explanations on demand
  - proposing a next step, if a student gets stuck.
- foster confidence in abstract concepts by providing experience with formal operation.

Challenges for development . . .

. . . are to exploit the power of TP technology in order to . . .

- support all phases of mathematical problem solving (phase of modelling, specifying, solving) . . .
  . . . all math is concerned with reasoning!
- make the system intuitive such that justifications are evident for manipulations (if asked for)
- support solution of engineering problems close to traditional paper&pencil representation
- make problem solving selfexplanatory by
  - checking steps input by students
  - generating explanations on demand
  - proposing a next step, if a student gets stuck.
- foster confidence in abstract concepts by providing experience with formal operation.

Challenges for development . . .

. . . are to exploit the power of TP technology in order to . . .

- support all phases of mathematical problem solving (phase of modelling, specifying, solving) . . .
  . . . all math is concerned with reasoning!
- make the system intuitive such that justifications are evident for manipulations (if asked for)
- support solution of engineering problems close to traditional paper&pencil representation
- make problem solving selfexplanatory by
  - checking steps input by students
  - generating explanations on demand
  - proposing a next step, if a student gets stuck.
- foster confidence in abstract concepts by providing experience with formal operation.

Challenges for development ...

...are to exploit the power of TP technology in order to ...

- support all phases of mathematical problem solving (phase of modelling, specifying, solving) ...
  ...all math is concerned with reasoning!
- make the system intuitive such that justifications are evident for manipulations (if asked for)
- support solution of engineering problems close to traditional paper&pencil representation
- make problem solving selfexplanatory by
  - checking steps input by students
  - generating explanations on demand
  - proposing a next step, if a student gets stuck.
- foster confidence in abstract concepts by providing experience with formal operation.

Thank you for attention ![1]

[1] PS: So, no “dead end”, but novel opportunities.