How Theorem-Prover Technology Advances Educational Math Software

Lessons Learned from Preparation of a Grant Proposal

Walther Neuper

Institute for Softwaretechnology
Graz University of Technology

5th Workshop on Formal and Automated Theorem Proving and Applications
February 3-4, 2012
Belgrade, Serbia

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

A call's text

[1] **Technology Enhanced Learning systems endowed with the capabilities of human tutors.**
Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

[1] FP7-ICT-2011.8.1

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.
Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.

Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.
Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors. Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.
Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.
Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.

Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

A call's text

Technology Enhanced Learning systems endowed with the capabilities of human tutors.
Research should advance systems' capabilities to react to learners' abilities and difficulties, and provide systematic feedback based on innovative ways of interpreting the user's responses – particularly in relation to deep/shallow reasoning and thinking. Research should advance systems' understanding and use of the appropriate triggers (praise, constructive comments, etc.) influencing learning. The systems shall improve learners' meta- cognitive skills, understand and exploit the underlying drivers of their learning behaviours. Solutions should exploit advances in natural language interaction techniques (dialogues), in rich and effective user interfaces and should have a pedagogically sound, smart and personalised instructional design.

Can TPS specifically tackle these requirements ?

. . . react to learners' abilities . . . provide systematic feedback
. . . interpreting the user's responses . . .

TPS check user-input **automatically** !!!

. . . deep/shallow reasoning . . .

TPS **cover the whole** problem solving process !!!

. . . improve learners' meta-cognitive skills . . .

TPS are **self-explaining models** of mathematics !!!

. . . rich and effective user interfaces . . .

??? Yes !!!

Can TPS specifically tackle these requirements ?

. . . react to learners' abilities . . . provide systematic feedback
. . . interpreting the user's responses . . .

TPS check user-input **automatically** !!!

. . . deep/shallow reasoning . . .

TPS **cover the whole** problem solving process !!!

. . . improve learners' meta-cognitive skills . . .

TPS are **self-explaining models** of mathematics !!!

. . . rich and effective user interfaces . . .

??? Yes !!!

Can TPS specifically tackle these requirements ?

. . . react to learners' abilities . . . provide systematic feedback
. . . interpreting the user's responses . . .

TPS check user-input **automatically** !!!

. . . deep/shallow reasoning . . .

TPS **cover the whole** problem solving process !!!

. . . improve learners' meta-cognitive skills . . .

TPS are **self-explaining models** of mathematics !!!

. . . rich and effective user interfaces . . .

??? Yes !!!

Can TPS specifically tackle these requirements ?

...react to learners' abilities ...provide systematic feedback
...interpreting the user's responses ...

TPS check user-input **automatically** !!!

...deep/shallow reasoning ...

TPS **cover the whole** problem solving process !!!

...improve learners' meta-cognitive skills ...

TPS are **self-explaining models** of mathematics !!!

...rich and effective user interfaces ...

??? Yes !!!

Can TPS specifically tackle these requirements ?

. . . react to learners' abilities . . . provide systematic feedback
. . . interpreting the user's responses . . .

TPS check user-input **automatically** !!!

. . . deep/shallow reasoning . . .

TPS **cover the whole** problem solving process !!!

. . . improve learners' meta-cognitive skills . . .

TPS are **self-explaining models** of mathematics !!!

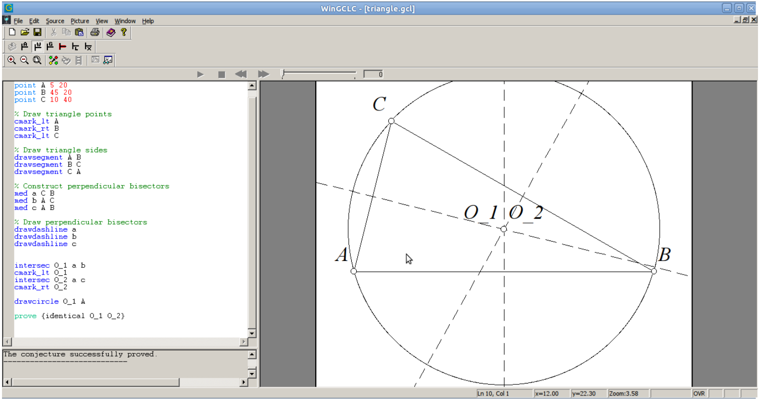
. . . rich and effective user interfaces . . .

??? Yes !!!

Outline

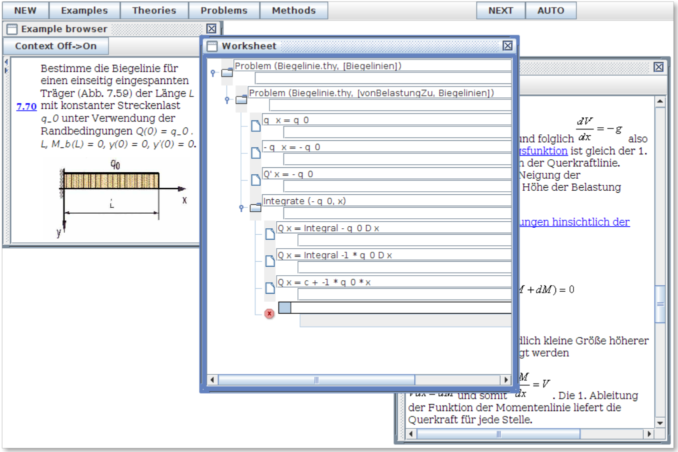
1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Check "automatically" . . .



. . . prove {identical O_1 O_2}

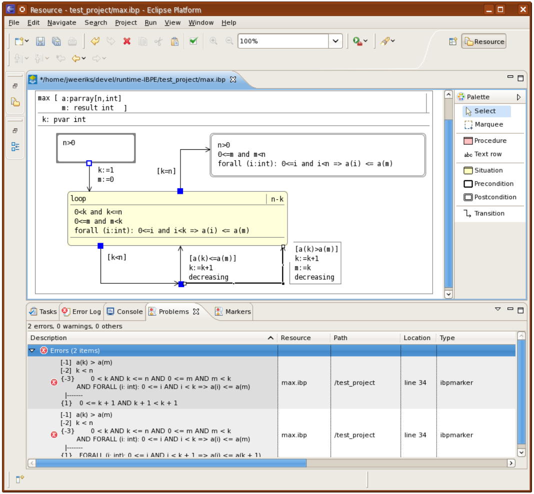
Check “automatically” . . .



$$\ldots \int -q_0 \frac{d}{dx} = c - q_0 \cdot x, \ldots \longrightarrow ok,$$

$$\ldots \int -q_0 \frac{d}{dx} = c - \frac{q_0^2}{2}, \ldots \longrightarrow not\ ok$$

Check “automatically” . . .



$$\ldots \forall (i : int) : 0 \le i \wedge i < n \Rightarrow a(i) \le a(m) \longrightarrow ok$$

$$\ldots \forall (i : int) : 0 \le i \wedge i < n \Rightarrow a(i) < a(m) \longrightarrow not\ ok$$

Check *automatically* . . .

. . . means, with respect to the three examples:

- The checks are most general and most reliable at the state-of-the-art.
- **No** specific code for a specific input.
  Rather, **one** (comprehensive collection of) theory suffices to check **all** reasonable inputs . . .
- . . . **general** code instead of (lots of) *specific* code in systems *not* TP-based.

So, the generality of TPS radically simplifies to

- provide systematic feedback when
- interpreting the user's responses.

Finally, TPS require less effort than multiple choice tests.

Check *automatically* . . .

. . . means, with respect to the three examples:

- The checks are most general and most reliable at the state-of-the-art.
- **No** specific code for a specific input.
  Rather, **one** (comprehensive collection of) theory suffices to check **all** reasonable inputs . . .
- . . . **general** code instead of (lots of) *specific* code in systems *not* TP-based.

So, the generality of TPS radically simplifies to

- provide systematic feedback when
- interpreting the user's responses.

Finally, TPS require less effort than multiple choice tests.

Check *automatically* ...

...means, with respect to the three examples:

- The checks are most general and most reliable at the state-of-the-art.
- **No** specific code for a specific input.
  Rather, **one** (comprehensive collection of) theory suffices to check **all** reasonable inputs ...
- ...**general** code instead of (lots of) *specific* code in systems *not* TP-based.

So, the generality of TPS radically simplifies to

- provide systematic feedback when
- interpreting the user's responses.

Finally, TPS require less effort than multiple choice tests.

Check *automatically* . . .

. . . means, with respect to the three examples:

- The checks are most general and most reliable at the state-of-the-art.
- **No** specific code for a specific input.
  Rather, **one** (comprehensive collection of) theory suffices to check **all** reasonable inputs . . .
- . . . **general** code instead of (lots of) *specific* code in systems *not* TP-based.

So, the generality of TPS radically simplifies to

- provide systematic feedback when
- interpreting the user's responses.

Finally, TPS require less effort than multiple choice tests.

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Cover the whole problem solving process

…deep/shallow reasoning …happens to address the coincidence of two key points: reasoning is …

1. the key point of mathematical thinking technology
2. the key point of Theorem-Prover based Systems (TPS)

This coincidence enables TPS to cover the **whole** problem solving process – see the "7 foundamental capabilities" ⟶

Cover the whole problem solving process

. . . deep/shallow reasoning . . . happens to address the coincidence of two key points: reasoning is . . .

1. the key point of mathematical thinking technology
2. the key point of Theorem-Prover based Systems (TPS)

This coincidence enables TPS to cover the **whole** problem solving process – see the “7 foundamental capabilities” ⟶

Cover the whole problem solving process

. . . deep/shallow reasoning . . . happens to address the coincidence of two key points: reasoning is . . .

1. the key point of mathematical thinking technology
2. the key point of Theorem-Prover based Systems (TPS)

This coincidence enables TPS to cover the **whole** problem solving process – see the “7 foundamental capabilities” $\longrightarrow$

7 foundamental capabilities

in PISA's competence model for mathematics:

1. **Communication**: *"... perceiving the existence of some challenge and recognizing a problem situation ..."*
2. **Mathematising**: *"... transforming a problem defined in the real world to a strictly mathematical form ..."*
3. **Representation**: *"... selecting, interpreting and using a variety of representations to capture a situation ..."*
4. **Reasoning** and argument: *"... logically rooted thought processes that check a justification that is given, ..."*
5. Devising **strategies** for solving problems: *"... critical control processes that solve problems ..."*
6. Using **symbolic**, formal and technical language and **operations**: *"... within a mathematical context ..."*
7. Using mathematical **tools**: *"... being able to make use of various tools that may assist math activity ..."*

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

7 foundamental capabilities . . .

. . . respectively **covered by TPS**:

1. Communication: not *specifically* addressed by TP
2. Mathematising: **formalisation** is a prerequisite for TPS support — can be prepared and hidden from student
3. Representation: various **specifications** can be offered — and tried out using next-step-guidance
4. Reasoning: every operation in TPS has a **mechanized justification** — can be hidden and handled on request
5. Strategies: various solving **algorithms** can be offered — and tried out using next-step-guidance
6. Symbolic operations: **all** TPS operations have a **symbolic** representation — next-step-guidance helps.
7. Tools: TPS address the other capabilities above

Doesn't all that overstrain students ? Not necessarily !

Cover the whole problem solving process

. . . deep/shallow reasoning . . . happens to address the coincidence of two key points: reasoning is . . .

1. the key point of mathematical thinking technology
2. the key point of Theorem-Prover based Systems (TPS)

This coincidence enables TPS to cover the **whole** problem solving process – see the “7 foundamental capabilities” ⟶

. . . we have ⟵

the whole process supported within **one coherent framework** which provides automated checks of user input.

However, pedagogical design of TPS is a novel challenge:

- *How gradually expose students to increasing formality ?*
- *How provide flexible next-step-guidance ?*
- *. . . ?*

Cover the whole problem solving process

. . . deep/shallow reasoning . . . happens to address the coincidence of two key points: reasoning is . . .

1. the key point of mathematical thinking technology
2. the key point of Theorem-Prover based Systems (TPS)

This coincidence enables TPS to cover the **whole** problem solving process – see the "7 foundamental capabilities" ⟶
. . . we have ⟵
the whole process supported within **one coherent framework** which provides automated checks of user input.

However, pedagogical design of TPS is a novel challenge:

- *How gradually expose students to increasing formality ?*
- *How provide flexible next-step-guidance ?*
- *. . . ?*

Cover the whole problem solving process

. . . deep/shallow reasoning . . . happens to address the coincidence of two key points: reasoning is . . .

1. the key point of mathematical thinking technology
2. the key point of Theorem-Prover based Systems (TPS)

This coincidence enables TPS to cover the **whole** problem solving process – see the “7 foundamental capabilities” ⟶
. . . we have ⟵
the whole process supported within **one coherent framework** which provides automated checks of user input.
However, pedagogical design of TPS is a novel challenge:

- *How gradually expose students to increasing formality ?*
- *How provide flexible next-step-guidance ?*
- *. . . ?*

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Self-explaining models of math

. . . improve learners' meta-cognitive skills . . . ("thinking about thinking") — can *not* be taught in a straight forward manner;

they *might* evolve depending on

- leisure phases in learning as opportunity for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques

⟶

Self-explaining models of math

…improve learners' meta-cognitive skills …("thinking about thinking") — can *not* be taught in a straight forward manner;

they *might* evolve depending on

- leisure phases in learning as opportunity for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques

⟶

Self-explaining models of math

…improve learners' meta-cognitive skills …("thinking about thinking") — can *not* be taught in a straight forward manner;

they *might* evolve depending on

- leisure phases in learning as opportunity for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques

⟶

Self-explaining models of math

... improve learners' meta-cognitive skills ... ("thinking about thinking") — can *not* be taught in a straight forward manner;

they *might* evolve depending on

- leisure phases in learning as opportunity for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques →

Self-explaining models of math

... improve learners' meta-cognitive skills ... ("thinking about thinking") — can *not* be taught in a straight forward manner;

they *might* evolve depending on

- leisure phases in learning as opportunity for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques →

Self-explaining models of math

…improve learners' meta-cognitive skills …("thinking about thinking") — can *not* be taught in a straight forward manner;

they *might* evolve depending on

- leisure phases in learning as opportunity for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques $\longrightarrow$

Self-explaining models of math

TPS can be “models of mathematics” with these properties:

- **complete** models
  - cover the whole problem solving process
  - comprise knowledge down to “first principles” (according to the LCF paradigm)
- **transparent** models
  - are transparent w.r.t. knowledge: all underlying knowledge is human readable; follows LCF paradigm
  - can be transparent w.r.t. operations (e.g. Coq proof language, Isar, single stepping systems)
- **interactive** models
  - check user input immediately
  - *“know the game of mathematics”* better than chess programs “know the game of chess”

Self-explaining models of math

TPS can be “models of mathematics” with these properties:

- **complete** models
  - cover the whole problem solving process
  - comprise knowledge down to “first principles” (according to the LCF paradigm)
- **transparent** models
  - are transparent w.r.t. knowledge: all underlying knowledge is human readable; follows LCF paradigm
  - can be transparent w.r.t. operations (e.g. Coq proof language, Isar, single stepping systems)
- **interactive** models
  - check user input immediately
  - *“know the game of mathematics”* better than chess programs “know the game of chess”

Self-explaining models of math

TPS can be "models of mathematics" with these properties:

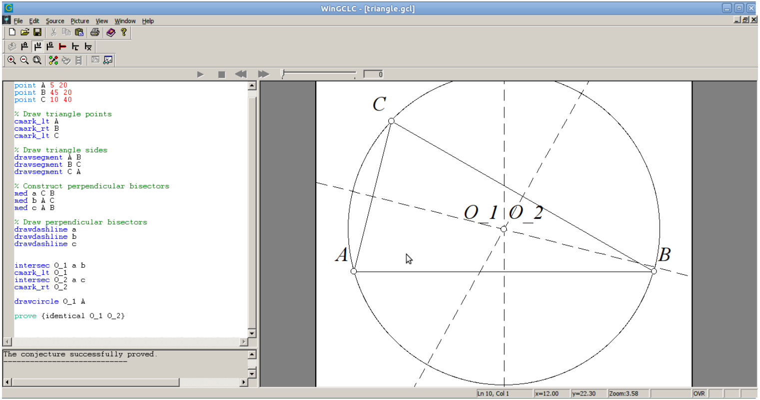
- **complete** models
  - cover the whole problem solving process
  - comprise knowledge down to "first principles" (according to the LCF paradigm)
- **transparent** models
  - are transparent w.r.t. knowledge: all underlying knowledge is human readable; follows LCF paradigm
  - can be transparent w.r.t. operations (e.g. Coq proof language, Isar, single stepping systems)
- **interactive** models
  - check user input immediately
  - *"know the game of mathematics"* better than chess programs "know the game of chess"

Self-explaining models of math

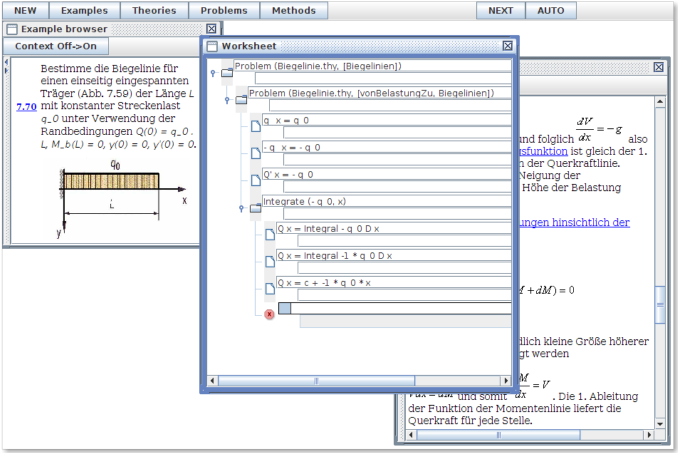
TPS can be “models of mathematics” with these properties:

- **complete** models
  - cover the whole problem solving process
  - comprise knowledge down to “first principles” (according to the LCF paradigm)
- **transparent** models
  - are transparent w.r.t. knowledge: all underlying knowledge is human readable; follows LCF paradigm
  - can be transparent w.r.t. operations (e.g. Coq proof language, Isar, single stepping systems)
- **interactive** models
  - check user input immediately
  - *“know the game of mathematics”* better than chess programs “know the game of chess”

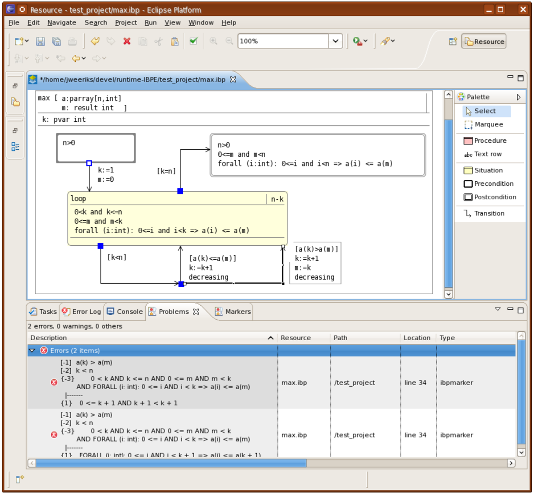
A model for geometry — construct . . . try . . . prove:



A model for applied math — specify . . . construct . . . justify



A model for sorting — specify . . . try . . . verify



Self-explaining models of math

…improve learners' meta-cognitive skills …("thinking about thinking") — can *not* be taught in a straight forward manner
and *might* evolve depending on

- leisure phases in learning which prepare for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques ⟶

Models of math, ⟵
being *complete, transparent* and *interactive*, support

- individual trials and questions at an individual pace
- exhaustive questioning within the formal world (LCF!)
- multi-faceted approaches to comprehensive questions

Self-explaining models of math

. . . improve learners' meta-cognitive skills . . . ("thinking about thinking") — can *not* be taught in a straight forward manner
and *might* evolve depending on

- leisure phases in learning which prepare for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques $\longrightarrow$

Models of math, $\longleftarrow$
being *complete, transparent* and *interactive*, support

- individual trials and questions at an individual pace
- exhaustive questioning within the formal world (LCF!)
- multi-faceted approaches to comprehensive questions

Self-explaining models of math

. . . improve learners' meta-cognitive skills . . . ("thinking about thinking") — can *not* be taught in a straight forward manner
and *might* evolve depending on

- leisure phases in learning which prepare for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques ⟶

Models of math, ⟵
being *complete, transparent* and *interactive*, support

- individual trials and questions at an individual pace
- exhaustive questioning within the formal world (LCF!)
- multi-faceted approaches to comprehensive questions

Self-explaining models of math

. . . improve learners' meta-cognitive skills . . . ("thinking about thinking") — can *not* be taught in a straight forward manner
and *might* evolve depending on

- leisure phases in learning which prepare for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques ⟶

Models of math, ⟵
being *complete, transparent* and *interactive*, support

- individual trials and questions at an individual pace
- exhaustive questioning within the formal world (LCF!)
- multi-faceted approaches to comprehensive questions

Self-explaining models of math

. . . improve learners' meta-cognitive skills . . . ("thinking about thinking") — can *not* be taught in a straight forward manner
and *might* evolve depending on

- leisure phases in learning which prepare for insights
- time for individual questioning at certain phases of mental maturing
- comprehensive technology supporting independent work on math knowledge and techniques $\longrightarrow$

Models of math, $\longleftarrow$
being *complete, transparent* and *interactive*, support

- individual trials and questions at an individual pace
- exhaustive questioning within the formal world (LCF!)
- multi-faceted approaches to comprehensive questions

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

The impact expected

- Unlock the potential of the individual by a stronger and smarter adaptation and personalization of educational technologies.
- Significantly higher level of effective, personalised, ICT-based tutoring, leading to its wide-spread penetration in schools and at home.
- Higher level of engagement of youngsters in science, technology and maths, through novel educational software and opening up opportunities to access and use of laboratory equipments and virtual experiments.
- Faster, more timely and more cost-effective up/re-skilling through learning technologies and their sustained adoption by SMEs.
- Emergence of new learning models, including models invoking creativity

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Use in formal education

- for open learning scenarios in class, *independent learning* at home
- as *additional challenge* for gifted and interested students
- to *provide extra tuition* to catch up on, particularly for “slow and rigorous thinkers”.
- in engineering studies at TU Graz: we shall use one and the same TPS in basic math education (e.g. partial fraction decompositions) *and* in advanced labs (e.g. Z-Transform in Signal Processing) — *continuity between “pure” and “applied” mathematics*
- establish **continuity** between “**intuitive**” math at high-school and “**formal**” math at university.

Use in formal education

- for open learning scenarios in class, *independent learning* at home
- as *additional challenge* for gifted and interested students
- to *provide extra tuition* to catch up on, particularly for "slow and rigorous thinkers".
- in engineering studies at TU Graz: we shall use one and the same TPS in basic math education (e.g. partial fraction decompositions) *and* in advanced labs (e.g. Z-Transform in Signal Processing) — *continuity between "pure" and "applied" mathematics*
- establish **continuity** between "**intuitive**" math at high-school and "**formal**" math at university.

Use in formal education

- for open learning scenarios in class, *independent learning* at home
- as *additional challenge* for gifted and interested students
- to *provide extra tuition* to catch up on, particularly for “slow and rigorous thinkers”.
- in engineering studies at TU Graz: we shall use one and the same TPS in basic math education (e.g. partial fraction decompositions) *and* in advanced labs (e.g. Z-Transform in Signal Processing) — *continuity between “pure” and “applied” mathematics*
- establish **continuity** between “**intuitive**” math at high-school and “**formal**” math at university.

Use in formal education

- for open learning scenarios in class, *independent learning* at home
- as *additional challenge* for gifted and interested students
- to *provide extra tuition* to catch up on, particularly for "slow and rigorous thinkers".
- in engineering studies at TU Graz: we shall use one and the same TPS in basic math education (e.g. partial fraction decompositions) *and* in advanced labs (e.g. Z-Transform in Signal Processing) — *continuity between "pure" and "applied" mathematics*
- establish **continuity** between "**intuitive**" math at high-school and "**formal**" math at university.

Use in formal education

- for open learning scenarios in class, *independent learning* at home
- as *additional challenge* for gifted and interested students
- to *provide extra tuition* to catch up on, particularly for “slow and rigorous thinkers”.
- in engineering studies at TU Graz: we shall use one and the same TPS in basic math education (e.g. partial fraction decompositions) *and* in advanced labs (e.g. Z-Transform in Signal Processing) — *continuity between “pure” and “applied” mathematics*
- establish **continuity** between “**intuitive**” math at high-school and “**formal**” math at university.

Use in formal education

- for open learning scenarios in class, *independent learning* at home
- as *additional challenge* for gifted and interested students
- to *provide extra tuition* to catch up on, particularly for “slow and rigorous thinkers”.
- in engineering studies at TU Graz: we shall use one and the same TPS in basic math education (e.g. partial fraction decompositions) *and* in advanced labs (e.g. Z-Transform in Signal Processing) — *continuity between “pure” and “applied” mathematics*
- establish **continuity** between “**intuitive**” math at high-school and “**formal**” math at university.

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by "complete, transparent, interactive" models

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by “complete, transparent, interactive” models

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by "complete, transparent, interactive" models

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by “complete, transparent, interactive” models

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by "complete, transparent, interactive" models

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by "complete, transparent, interactive" models

Use in cross-institutional settings

- by educators to *define math competences and knowledge*
  - required as prerequisites
  - planned as goals

  for courses in schools, universities and in industry (*TPS as complete models of math*)
- by students to *independently prepare* for entry exams as well as for final exams
- by administrators to *evaluate and assess* math competences and knowledge of students

Specific impacts expected from

- support for independent learning
- coverage of math by “complete, transparent, interactive” models

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via “cloud computing”
- for *“scientific branding”* : Might TPS turn “mathematics as magic” to an attractive key for further development? If “yes”, then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via "cloud computing"
- for *"scientific branding"* : Might TPS turn "mathematics as magic" to an attractive key for further development? If "yes", then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via “cloud computing”
- for *“scientific branding”* : Might TPS turn “mathematics as magic” to an attractive key for further development? If “yes”, then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via “cloud computing”
- for *“scientific branding”* : Might TPS turn “mathematics as magic” to an attractive key for further development? If “yes”, then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via "cloud computing"
- for *"scientific branding"* : Might TPS turn "mathematics as magic" to an attractive key for further development? If "yes", then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via “cloud computing”
- for *“scientific branding”* : Might TPS turn “mathematics as magic” to an attractive key for further development? If “yes”, then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via "cloud computing"
- for *"scientific branding"* : Might TPS turn "mathematics as magic" to an attractive key for further development? If "yes", then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Use in public space

- for *new learning models* : learning by trial and error similar to learning from chess programs:
  - in math wikis make proofs and example calculations interactive
  - in museums, math spaces etc. interactively present math as a key thinking technology
  - in open price competitions address the public via “cloud computing”
- for *“scientific branding”* : Might TPS turn “mathematics as magic” to an attractive key for further development? If “yes”, then
  - announcements in academia and industry might turn from hiding math to *advertising math* in their courses — in order to attract students
  - private enterprises might use TPS to associate their brand with *scientific expertise* — in order to attract employees and customers.

Outline

1. What are the key contributions of Theorem-Prover based Systems (TPS) to math education ?
   - A call's text
   - TPS check user-input automatically
   - TPS cover the whole problem solving process
   - TPS are self-explaining models of mathematics
2. How might TPS be used in math and science education ?
   - Impact expected by the call
   - Use in formal education
   - Use in cross-institutional settings
   - Use in public space
3. Conclusion: announcement for a new generation of educational math software

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on "software which doesn't make mistakes".
  - instead of interpreting math texts on paper enjoy interactive inquiry in the "game of mathematics".
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students' independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on "software which doesn't make mistakes".
  - instead of interpreting math texts on paper enjoy interactive inquiry in the "game of mathematics".
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students' independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on "software which doesn't make mistakes".
  - instead of interpreting math texts on paper enjoy interactive inquiry in the "game of mathematics".
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students' independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on "software which doesn't make mistakes".
  - instead of interpreting math texts on paper enjoy interactive inquiry in the "game of mathematics".
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students' independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on “software which doesn’t make mistakes”.
  - instead of interpreting math texts on paper enjoy interactive inquiry in the “game of mathematics”.
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students’ independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

TPS, a new generation

- for developers of educational math software:
  - instead of free coding between pre- & post-condition have a logically coherent framework.
  - instead of struggling with lots of specific code enjoy automated checks in step-wise problem solving.
- for pupils, students and other learners:
  - instead of full responsibility for limitations of tools rely on "software which doesn't make mistakes".
  - instead of interpreting math texts on paper enjoy interactive inquiry in the "game of mathematics".
- for teachers and lecturers:
  - instead of explaining basics again and again to classes get free for discussing essentials with individuals.
  - instead of teaching for the average level of a class become a coach for your students' independent work.
- for educators and administrators:
  - forget multiple choice tests in mathematics.
  - provide the same TPS for learning and for evaluation.

Thank you for attention !

? ? ?

? ? ?

? ? ?

with the hope for further discussion in the TP community.

Thank you for attention !

? ? ?

? ? ?

? ? ?

with the hope for further discussion in the TP community.