This is a theoretical essay. It tries to build a bridge between offers of new technology and psychology of mathematics education. For that purpose mathematics is regarded a language. The technology is an upcoming generation of software based on (computer) Theorem Proving (TP). TP appears capable of developing "systems that explain themselves" for interactively solving problems in mathematics and of representing complete, transparent and interactive models of mathematics. That means they represent mathematics such that one can learn by "speaking" with the models within their formal language – the same way one can learn playing chess by trying and watching a chess program. The essay aims at collaborative clarification of these offers in terms of psychology of mathematics education.

INTRODUCTION

The intention of this essay, pursued by considering mathematics as a language, is technology driven. Technology and concepts from (computer) Theorem Proving (TP) allow for building a new generation of software tools as shown, for instance, in a workshop series called ThEdu, Theorem Proving Components for Educational Software. There also bibliographies on various approaches to TP-based educational software can be found as well as descriptions of planned and already existing prototypes of such software.

The new generation envisages educational tools which provide, hitherto unimaginable, a representation as complete, transparent and interactive models of mathematics – this aim will be concern of the subsequent section on “self-explanatory systems built on TP”.

Before going in these more technical details it is necessary to say, that this essay does not complete a bridge from technology to psychology of mathematics education; rather it builds a bridge head on the side of technology, while a bridge head on the side of psychology of mathematics education is left to future work. The bridge is prepared by a presentation of TP as a technology with principal affinity to learning mathematics and by a description of features of TP relevant for building TP-based educational software.

Since development of the new TP-based software generation is in an early stage and only prototypes are available, respective research on education has not yet started; the bridge head on this side is up to future research. The author of this essay hopes to stipulate such research in collaboration with experts in psychology of math education.
SELF-EXPLANATORY SYSTEMS BUILT ON TP

Serious introduction of a new software generation based on TP technology (which is still unfamiliar even to academic mathematicians, in spite of success stories with mechanical proof of the “Four Colour Theorem” or the “Kepler Conjecture”) requires more space than given within PME’s limits. So, readers are kindly asked to look up further details, figures and examples in (Krempler & Neuper 2018, abbreviated as [1] below). German speaking readers might look up these details in (Neuper 2018, §2). Subsequent headlines are marked by letters for later reference.

(a) Transparent Derivations from Elementary Notions

If one explains a notion new to somebody, not only in mathematics, she or he uses simpler notions in the explanation. This is implemented in TP to an extreme extent. For instance, Fig.1 shows a simple calculation $12345 \times 67 + 12345 \times 33$ in the TP Isabelle (Nipkow, Paulson & Wenzel 2002).

![Fig. 1: Precision and expressiveness of formulas in TP-based systems](image)

The calculation (identified by lemma) is surrounded by boxes which only appear by mouse click: A click on $+$ shows the respective definition in terms of logic together with laws required for this operation. A click on $\text{int}$ shows the usual mathematical definition of integer numbers by a product of natural numbers $\text{nat} \times \text{nat}$ (and not some type of bit pattern as in Computer Algebra Systems).

Further clicks would access simpler notions than natural numbers, the end of the deductive chain are the axioms of logic – and the deduction steps can be presented in a human readable way (Wenzel 1999) as well. So a TP-based system can show all underlying mathematics knowledge, it can be considered a transparent model of mathematics.
(b) Traditional Solving of Engineering Problems

The calculation in Fig.1 shows the way a TP is working: The user supplies a proposition, i.e. the calculation together with the result, and the system assists the user to prove it correct (in this case the theorem \texttt{distrib_left} is involved) – but this (coming up with the result at the beginning) is not what a student expects from a system when stepwise constructing solutions!

There are several systems modelling the usual way of mathematical problem solving, for instance (Back, Grundy & Wright 2007) or (Krempler & Neuper 2018). The latter present the construction of a differential equation as shown in Fig.2.

![Fig. 2: Construction of a differential equation for a physical model.](http://demonstrations.wolfram.com/OscillationOfTwoMassesConnectedBySprings/)

This format corresponds closely to the traditional format of problem solving by paper and pencil, in particular the steps of calculation at the left margin (starting with 2122; the line numbers are just for reference in the paper), whereas the justifications at the right margin are an added value by TP: each formula at the left is justified by a so-called tactic on the right; this is called “forward reasoning” in TP and leads to results correct by construction.

Steps in Fig.2 can be expanded in case there are intermediate steps, for instance the normalisation from line 2128 to 212a would take several steps as required for work by hand.

Last not least: the student always can start from a meaningful situation – if given “real world” problems he or she can understand at his personal level. For instance, the calculation in Fig.2 solves a problem which can be explained by animated figures, [http://demonstrations.wolfram.com/OscillationOfTwoMassesConnectedBySprings/](http://demonstrations.wolfram.com/OscillationOfTwoMassesConnectedBySprings/).

(c) Automated Checks of User Input

Correctness of a calculation like in Fig.2 is ensured by automated TP (usually abbreviated as ATP): Given a logical context, input of a formula by a student creates
a proof situation – the formula must be (automatically) derived from the logical context. Such a context is initialised by a so-called formal specification, which covers a class of problems (for instance, differential equations); data specific to a problem are hidden behind the textual problem description. The specification for the problem solution in Fig.2 (there collapsed) is shown in Fig.3.

A specification comprises a Model (line 2111-21114) and References to methods solving the problem (here collapsed, also not shown are the collections presented to students for selection). A model comprises input items as in 21111, preconditions checking validity of input in 21112 (omitted), output items in 21113 and a postcondition in 21114. Further details and another example are given in [1]§3.4.

(d) Generation of Concrete Examples and of Variants

Once there is logical context, TP can provide powerful tools: automated generation of concrete examples for (abstract) definitions, for instance for the definition of integers in Fig.1, or even counter-examples for wrong propositions, see [1] end of §3.2.

A specific extension of TP is Lucas-Interpretation [1]§2.2, which allows to propose a next step in a problem solution. Combined with a dialogue component, this allows the system to give hints for the next step (for instance a partial formula) if the user gets stuck. So there arise dialogues on equal terms: both have some knowledge (and tell it on request), both can check the partner’s steps, both know how to do a next step and both can change roles any time. This is like playing chess with a computer. So one can have interactive models of mathematics.

(e) Complete, Transparent and Interactive Models of Mathematics

Section (a) and (d) showed, that TP-based systems can represent transparent and interactive models of mathematics, respectively. Last not least, TP-based systems can be considered also complete models for two reasons: first the chain of deduction from any item of mathematics knowledge is complete down to the axioms of logic as shown in (a), second TP-based systems can cover all phases of problem solving: modeling a problem (stated by text and/or figures, see (b)) in formulas, referencing
specifications and methods (see (c) above) and then constructing solutions (see (b) above), which might involve modeling and specifying sub-problems recursively.

The notion of “mechanised models” triggers warnings and questions: Don’t end up such models with mechanical drill & practice? What about creativity and intuition which can not (as we believe) be implemented by machines?

A quick answer is: Creativity and intuition are required for creating new mathematical concepts or for applying mathematical knowledge within novel contexts. However, as soon as mathematical theory is finished, it can be mechanised completely by TP (see subsequent section) – but educational curricula do not comprise creating new mathematical theories, not even in academic engineering studies. So the above questions reduce to: Can TP-based models of mathematics be designed such, that they sufficiently meet students’ needs as identified by psychology of mathematics education? Which needs at which age?

“SPEAKING MATHEMATICS” IN TP-BASED SYSTEMS

First let us identify similarities and differences between natural language and the language of formal mathematics (to be precise: the latter does not address prose in the context of mathematics, but formulas and structures like formal specifications) as depicted in Fig.4.

Language transports meaning; in our context meaning has to be created in two directions, towards human intuition on the green side of Fig.4 left and towards formal mathematics on the gray right. Meaning postulates understanding, and in case this concerns complicated matters, it also involves abstraction, which collects notions to a new notion and forgets details as appropriate; respective relations between green and gray will be discussed later in subsection “Abstraction as a Language Feature”.

Fig. 4: Natural language versus formal language of mathematics
Beforehand we want to address processes which establish and confirm meaning, because this is a primary focus in education. The bottom left of Fig. 4 is a focus of theories in mathematics education. As already mentioned, this essay throws educational analysis into reverse, takes the bottom right into the focus and asks: How is meaning established in formal mathematics? How can students experience meaning by doing, how can students create meaning in this context?

**Operations on Items of Mathematical Language**

If considering mathematics as a language, (1) what language items are at disposal to the human partner in a TP-based system, (2) what is achieved by these items and (3) what for are they checked automatically? These are the three-fold answers:

- **Specification (1),** see Fig.3: determines input- and output-data and the logical context (2). *Check for appropriateness by the preconditions and by data hidden behind the textual or figurative description of examples (3).*
- **Worksheet,** see Fig.2: formulas tactics and sub-problems serve construction of a problem solution. *Check if the solution meets the post-condition by ATP according to (c)*.
- **Formula in a Worksheet,** see Fig.3: step in a problem solution. *Check if derivable from the logical context (in particular, from the previous formula).*
- **Tactic in a Worksheet,** see Fig.3: determines next step towards a solution. *Check if applicable to logical context by Lucas-Interpretation.*
- **Specification or Method in a Collection,** see (c): select for current Worksheet. *Check appropriateness for logical context.*
- **All Formulas** (also if presented partially by the system): input, edit, delete. *Check for syntax and type correctness.*

All these items and respective operations can be handled equally by both partners, the student and the system – they are in a dialogue on equal terms. And if a student gets stuck, the system can suggest a next step ((c) above), while the dialogue component ((d) above) has to balance activity such that a student is neither over- nor under-challenged. Every interaction has an observable effect: either a step forward towards a problem solution or a feedback rejecting this step. Interactions and effects are connected such, that students need no introduction, neither in software handling nor in active problem solving (since the system always can help with a next step and the system is transparent); early field tests (Neuper & Dürnsteiner 2007, Reitinger & Neuper 2008) have confirmed these expectations.

**Boundaries between Natural Language and Mathematical Language**

Language transports meaning, so mathematics must be meaningful (Brown & Walter 2005). “Speaking mathematics” is meaningful, if the problem to be solved is understood by the student; and this can be guaranteed by systems, that cover all phases of problem solving, beginning with a problem stated as text or figure, see (b).
Such meaning is usually confirmed in *natural* language (green area left in Fig.4). As soon as the student enters problem solving in the system, there is a switch to *formal* language (gray area right in Fig.4) using the above language items – while the student keeps mostly thinking in natural language, of course.

The novelty in learning with TP-based systems is: the complete, transparent and interactive (software-)model of mathematics ((e) above) provides active experience open for interplay with human thinking in the realm of natural language.

It is like playing chess with a computer: the human player acts according to the rules (i.e. in the “formal/mechanical language”) of chess and can learn by interacting with the machine: go back a few steps and try another variant, etc -- but the user thinks completely different from the computer”: investigating such interplay appears an interesting challenge for collaborative research with psychology of math education.

**Abstraction and Reflection in Formal Language**

Both, abstraction and reflection, are interesting aspects of language in general, are important issues of mathematics education, and this essay again focuses formal language, leaving human learning on the green side of Fig.4 to future work.

Abstraction has been related to TP-based software by (Neuper 2017) already: building abstraction appears supported by freely switching from survey to detail (by expanding/collapsing parts of calculations), by switching aspects (specification, solution, sub-problem) and by switching from construction to analysis according to (c). The striking feature of TP is access to get by mouse click definitions, related theorems, etc for arbitrary (abstract) notions.

Reflection first has to be weakened to self-referentiality: TP implements several layers of formal language, where the meta-level provides semantics for the object-level. Said more simply: formulas describe formulas. The challenge for future research is to find out, how such self-referentiality in technology might foster human reflections; are here new approaches to constructivism in education in the line of (Taber 2016)?

**SUMMARY AND CONCLUSIONS**

This essay tried to build a bridge from TP-based technology with features affine to learning on the one side to psychology of mathematics education on the other side.

The bridge head on the technology side has been described to an extent which might motivate educators to draw their attention to the promises of a new generation of “systems that explain themselves” for education in mathematics and respective applications in engineering studies. The bridge itself is borne by the idea to let students start from meaningful learning situations (according to (b)) and to let them learn by “speaking” with “complete, transparent and interactive models of math” within their formal language (according to (e)).
TP-based systems are restricted to formal semantics and do not rely on intuitive understanding. The essay presents this restriction as an novel support at the transition from intuition-based mathematics at high-schools to the formality in academic mathematics: There students need to drop specific intuitions for the sake of “economy of thinking” and rely on abstract concepts and on abstract operations on these concepts – this need in academic use of mathematics appears to be underexposed in present didactics of mathematics.

However, the bridge head on the educational side is missing. It is the primary aim of this essay to stipulate joint research addressing structures of learning mathematics and relating the new offers of technology to theories in psychology of mathematics education.

**References**


