

Cognitive Science meets Computer Mathematics. A Case Study in "Next-Step-Guidance"

Gabriella Daróczy and Walther Neuper

June 8, 2012

Keywords:

Cognitive Science Stepwise Problem Solving Error Patterns
Mathematics Education Next Step Guidance Learning by Trial and Error
Computer Theorem Proving

Abstract:

Cognitive Science (CS) successfully derives practical advice for the design of learning scenarios, also for mathematics. Cognitive Science also plays a prominent role in the design of educational software. However, CS's contributions to the design of educational math software appears sparse.

The emergence of educational systems based on Theorem-Proving (TP) increases the potential of contributions from CS: TP-based systems

1. cover the whole range of problem solving from specification, stepwise construction of a solution to final (automated) checks
2. represent all underlying math knowledge in a human readable format, which allows context-sensitive access by the learner
3. perform automated check of user input in a general and reliable way if given an appropriate logical context
4. and last not least are able to propose a next step towards a solution generated automatically by so-called Lucas-Interpretation.

TP-based systems allow to model a significant wider range of mathematical problem solving, so educational systems design becomes an issue. This talk particularly addresses the last of the above points called "Next-Step-Guidance (NSG)" — if a system is able to automatically propose a next step, this ability calls for intelligent exploitation and raises questions like:

- Which parts of the next step shall be presented to the learner, the rule to apply or the formula ?
- Which part of a formula or a rule should be omitted (error patterns) ?
- In which situations is the learner allowed to request NSG ? (Not in exams!)
- ... etc.

A prototype implementation will be presented which answers some of these questions. The usability of the prototype suggests a new learning model analogous to learning chess by chess programs: learning happens in interaction by trial and error, by watching the system's steps — just instead of pursuing checkmate NSG supports interactive construction of a problem solution in cooperation with the system.