LOGIC AS A PATH TO ENLIGHTENMENT
(WORK IN PROGRESS REPORT)

Wolfgang Schreiner
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria
Enlightenment and Education

- **Enlightenment**: reject claims based on authority ("ipse [Aristotle] dixit")
  - Only two sources of truth acceptable:
    - Empirical evidence (observation)
    - Well-formed arguments (reasoning).
  - Stark contrast to pre- or even anti-modern views.

- **Education**: often claims accepted by authority ("ipse [the teacher] dixit")
  - Even in “rational” disciplines like mathematics or computer science.
    - Presentations of propositions, rules, methods, and algorithms (more often than not) lack proper justification.
  - Students educated to become “believers” (or, equally worse, “non-believers”) rather than “rational skepticists”.

Students should be provided a basis for rational discourse.
Logic as a Path to Enlightenment

Logic as the “science of reasoning” provides such a basis.

■ **Predicate logic:** the “modern” logic of today.
  - Starting with Frege’s “Begriffsschrift” in 1879.
  - Incorporates and supersedes Aristotle’s term logic.
  - Rich enough to capture most of mathematics and much of natural language.

■ **Construct formal models of reality** with precise meaning and reasoning rules.
  - State propositions as formal sentences.
  - Derive valid arguments that prove the propositions.
  - Judge whether such arguments are valid or not.

Should be taught as a practical “working language” for modeling and reasoning.
Educating with the Help of Logic

Do not consider logic just as a “paper and pencil” topic.

- Today much of logic can be automated by computer software.
  - Advances in computational logic (automated reasoning, model checking, satisfiability solving) may aid in many activities.

- Education may be supported by the application of such software
  - Needs to be well considered and carefully prepared.
  - May demonstrate the practical usefulness of logic.
  - May increase the motivation of students to model and to reason.

Core idea: let students actively engage with lecturing material by solving concrete problems and by receiving feedback from software.
JKU LIT Project “LogTechEdu”

Johannes Kepler University Linz Institute of Technology project “Logic Technologies for Computer Science Education” (2018–2020):

- Solver Guided Exercises
- Teaching Solver Technology
- Proof Assistants for Education (Theorema)
- Specification and Verification Systems for Education (RISCAL)
- Logic across the Subjects in Primary, Secondary and Higher Education

Joint activities of the JKU institutes FMV (Biere, Seidl) and RISC (Schreiner, Windsteiger) and supported by the Linz School of Education (Sabitzer).
The RISC Algorithm Language (RISCAL)

A language and software system for investigating finite mathematical models.

- Formulation of mathematical theories and theorems.
- Formulation and specification of (also non-deterministic) algorithms.
- Rooted in strongly typed first order logic and set theory.
- All types are finite (with sizes determined by model parameters).
- All formulas are automatically decidable by model checking.
- Correctness of all algorithms is automatically decidable by model checking.

Checking in some model of fixed size before proving in models of arbitrary size.
RISCAL Specifications

val n: N; val cn: N;
...
type Formula = Set[Clause];
type Valuation = ClauseBase with clause(value);
...
pred satisfiable(f:Formula) ⇔ ∃v:Valuation. satisfies(v,f);
pred valid(f:Formula) ⇔ ∀v:Valuation. satisfies(v,f);
fun not(f: Formula):Formula = { c | c:Clause with ∀d∈f. ∃l∈d. -l∈c };
...
theorem notValid(f:Formula) ⇔ valid(f) ⇔ ¬satisfiable(not(f));
...
multiple pred DPLL(f:Formula)
  ensures result ⇔ satisfiable(f); decreases |literals(f)|;
⇔ if f = ∅[Clause] then
  ⊤
else if ∅[Literal] ∈ f then
  ⊥
else choose l∈literals(f) in
  DPLL(substitute(f,l)) ∨ DPLL(substitute(f,-l));
The RISCAL Software

Automatic checking of theorems, algorithms, and verification conditions.
An Exercise in Formal Problem Specification
Conclusions

- Goal: logic-based **self-directed learning**
  - Teacher become “enablers” by providing basic knowledge and skills
  - Students “educate themselves” by solving problems.
    - (Voluntary) quizzes, (mandatory) assignments, possibly (graded) exams.

- Initial target: undergraduate university students.
  - Reachout both “up and down” to graduate students and to high-school students.

- Initial focus: computer science and mathematics.
  - First own courses on “Logic”, “Formal Modeling”, “Formal Methods”; later also others’ introductory courses on algorithms and software development.

Towards “enlightenment” via “rational thinking” by “self-directed learning”.