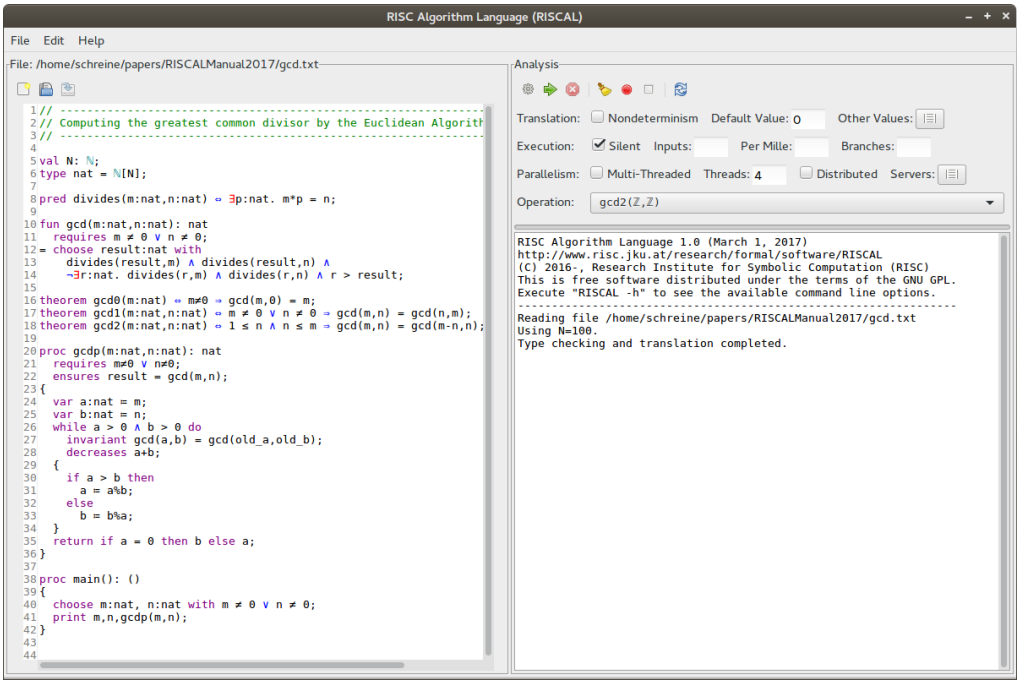
JKU LIT Project "LOGTECHEDU":
Student Projects and Bachelor/Master Theses

State: March 26, 2019

LOGTECHEDU (Logic Technology for Computer Science Education) is a project of the Johannes Kepler University (JKU) Linz Institute of Technology (LIT) on novel **logic-based software tools for education**, with focus on undergraduate university courses in computer science and mathematics. The project pursues various **research strands** each of which offers various **student projects and bachelor/master theses** for students of **computer science** or **mathematics** respectively the corresponding **teaching studies**. The project financially supports these activities by **tutorship and fellowship positions**.

LOGTECHEDU Strand: Specification and Verification Systems for Education (The RISC Algorithm Language RISCAL)



The **RISC Algorithm Language (RISCAL)** is a specification language and associated software system for describing mathematical algorithms and formally specifying their behavior based on mathematical theories. By design of the language, the specified model is of finite (but arbitrary) size; thus the correctness of algorithms, specifications, and theories can be automatically decided by model checking. See also

> http://www.risc.jku.at/research/formal/software/RISCAL
> http://www.risc.jku.at/research/formal/software/RISCAL/presentation

for more details on the software respectively a video presentation demonstrating its goals and use.

Various topics for student projects and bachelor/master theses on the use of RISCAL, its further development, and its application in education are available; examples are presented below.

Contact: Wolfgang Schreiner <Wolfgang.Schreiner@risc.jku.at>
Associate Professor, Research Institute for Symbolic Computation (RISC)
Johannes Kepler University (JKU), A-4040 Linz, Austria

Student Projects and Bachelor Theses: Specification and Validation/Verification of Algorithms

There are multiple student projects and bachelor theses available whose goal is to formalize in RISCAL problems and algorithms for various application areas, for example

- elementary data structures (arrays, lists, trees, . . .),
- string processing, image processing, geometry,
- discrete mathematics and graph theory,
- computer algebra and computational logic.

The goal of each activity is to formalize in RISCAL the basic theory, formulate the specifications of the problems, validate the specifications according to various criteria, formulate algorithms solving the problems, annotate the algorithms with meta-information (loop invariants and termination measures). A bachelor thesis is also expected to derive verification conditions for the correctness of the algorithms with respect to the specifications, and automatically check the conditions with RISCAL.

Master Thesis: A Prover Interface for RISCAL

The goal of this thesis is to develop an RISCAL interface to external proving assistants such that verification conditions and other theorems (which can be checked in RISCAL over models of a fixed size) can be proved by these assistants (over models of arbitrary sizes).

This comprises the translation of the RISCAL logic language and of the associated theories to the language of the corresponding proof assistant and the actual software connection of RISCAL to the assistant; the implementation language is Java.

The result shall be a generic interface that allows to develop "plugins" for various concrete proof assistants; in the frame of the thesis the concept shall be demonstrated by providing such plugins for at least two assistants (say the RISC ProofNavigator and Isabelle/HOL).

Master Thesis: Transition Systems and Linear Temporal Logic in RISCAL

The goal of this thesis is to develop in RISCAL a framework for modeling transition systems (non-deterministic respectively concurrent systems) and analyzing their correctness with respect to temporal logic specifications. This comprises the following tasks:

- The development of a RISCAL command for the nondeterministic execution of transition systems specified by their state space, initial state condition, and transition relation (respectively non-deterministic transition function), as well as appropriate fairness conditions.
- The introduction of a (variant) of linear temporal logic (LTL) into the logic language of RISCAL.
- The extension of the RISCAL model checker to verify the execution of a system with respect to specifications expressed in LTL.

The checking mechanism is to be formally elaborated; the developments are implemented in Java as part of the RISCAL software.

Master Thesis: An Educational Concept for Self Study with RISCAL

The goal of this thesis is to elaborate on the basis of RISCAL a didactic concept that elaborates how RISCAL can be used in education, in particular for self-instructed and self-paced learning (respectively in blended learning scenarios that incorporate such elements).

The concept shall be generally applicable to subject areas relevant for the formal education of bachelor studies of computer science and/or mathematics (logic, discrete mathematics, etc) and thus be first described in general terms.

However, it shall be also concretized for a particular sample topic by the elaboration of corresponding lecture materials (RISCAL specifications, lecture notes, presentation slides), such that a course (or part of a course) in the amount of approximately 1.5 ECTS (1 semester hour) can be based on this concept and supporting material.

Furthermore, evaluation criteria shall be elaborated that allow to quantitatively judge the success of the concept when actually applied in a subsequent course (the evaluation itself is not necessarily any more in the scope of the thesis).

Ongoing Topics

Master Thesis: Application of SMT Solvers to Deciding Formulas

This thesis aims at the application of SMT solvers for deciding the validity of a formula as a (potentially more efficient) alternative to the checking by evaluation currently applied by RISCAL. For this purpose, given (a finite instance of) a RISCAL specification, a formula expressed in this specification (and all the definitions on which it depends) have to be translated to a suitable theory of the SMT-LIB standard (http://smtlib.cs.uiowa.edu) such that some SMT solver supporting this theory can be applied.

Since a specific instance of a RISCAL specification describes a finite model, such a translation is always possible; it generally involves the replacement of quantifiers by finite combinations of logical connectives and the translation of basic types and operations to corresponding counterparts of the selected SMT-LIB theory.

The translation is to be formally elaborated and implemented in Java as part of the RISCAL software.

Completed Topics

Master Thesis: Automated Generation of Validation/Verification Conditions

This thesis aims at the automated generation of conditions that validate specifications and verify the correctness of algorithms with respect to their specifications. This comprises the following tasks:

- Specification validation: generate conditions that prove that a specification is implementable, not trivial, determines the result uniquely; generate the function that is implicitly defined by the specification.
- Algorithm verification: generate verification conditions that show that for every argument that satisfies the specified precondition the algorithm indeed returns a result that satisfies the postcondition; generate conditions that show that the user-specified program annotations (loop invariants etc) are correct, and that show that the application of all operations/functions are well-defined.

The condition generator is to be formally elaborated and implemented in Java as part of the RISCAL software.

Master Thesis: Visualization of Formula Interpretation/Valuation

The goal of this thesis is to aid students in understanding why a particular predicate formula holds respectively (and more importantly) why a formula supposed to hold is actually violated.

For this purpose, a visual model for the evaluation of formulas in first order predicate logic is to be elaborated that allows to inspect the variable instantiations and evaluation paths of a corresponding formula evaluation that are relevant for determining the value of the formula (for instance, if a universally quantified formula is violated only the witness of the variable that makes the body of the formula false and the evaluation starting from that witness has to be displayed); the model shall be hierarchical such that it scales to non-trivial formulas, allowing e.g. to investigate multiple evaluation paths and the evaluation of predicates respectively functions applied in the evaluations.

The visual model is to be elaborated and validated by several sketchbook examples and then to be implemented in Java (with the help of a suitable visualization library) as part of the RISCAL software.