

Formal Foundations of Computer Science 1

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

Wolfgang.Schreiner@risc.uni-linz.ac.at
<http://www.risc.uni-linz.ac.at/people/schreine>

Course Organization

- Lectures

1. **Option:** Tuesday, 8:15–9:45, HS 12.
2. **Option:** Tuesday, 18:30–20:00, HS 5.

- Information

- Web: <http://www.risc.uni-linz.ac.at/courses/formal>.
- Office: Tuesday, 17:00–18:00, KG, 4th floor, RISC-Linz rooms.

- Grades

- Written exam at end of semester.

Presence is strongly recommended.

Exercises

- **Begin: October 13 and 15.**
 - Ordered hardcopies of lecture notes.
 - Master hardcopy of first exercise sheet.
- **Late Registration**
 - email: Wolfgang.Schreiner@risc.uni-linz.ac.at
 - Firstname, Lastname, Mat.Nr., SKZ, time constraints.
- **Grades**
 - 10 exercise sheets.

Presence is mandatory.

Course Materials

- Lecture notes.
 - Web page (HTML, PostScript, PDF).
 - Hardcopy (order now: ÖS 50,-).
 - Explains details for home study.
- Course slides.
 - Web page (HTML, PostScript, PDF).
 - Print yourself and bring to lecture.
 - Explains basics; use for annotations.

Listen during lecture rather than write!

Logic Evaluator

Interpreter for executing mathematical definitions.

```
pred <(m, n) <=> and(<=(m, n), not(=(m, n)));  
> predicate </2.  
pred divides(n, m) <=> exists(p in nat(1, m): =(*(n, p), m));  
> predicate divides/2.  
pred isPrime(p) <=>  
  and(<(1, p),  
    forall(n in nat(2, -(p, 1)): not(divides(n, p))));  
> predicate isPrime/1.  
pred isNextPrime(n, p) <=>  
  and(<=(n, p), isPrime(p),  
    not(exists(q in nat(n, -(p, 1)): isPrime(q))));  
> predicate isNextPrime/2.  
formula isPrime(19);  
> true.  
formula isNextPrime(14, 19);  
> false.
```

```
formula isNextPrime(14, 17);
```

Contents

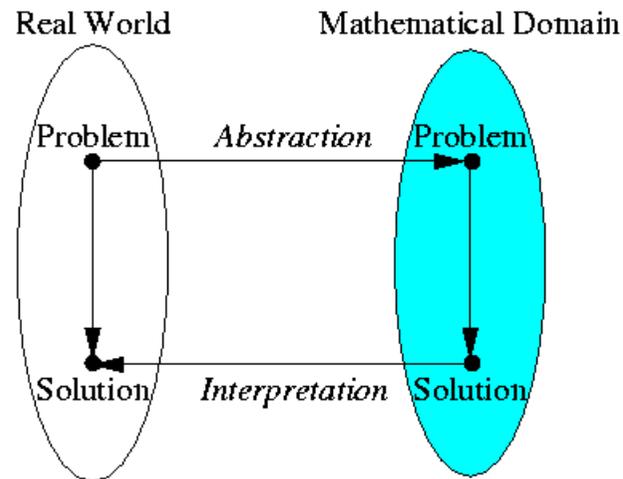
Introduction into basic mathematical domains and techniques as a foundation for computer science.

1. The Language of Logic,
2. Sets, Functions, and Relations,
3. Numbers,
4. Induction and Recursion.
5. More on Functions,
6. More on Relations.

A Defining New Notions,

B Proving Propositions.

Mathematical Domains



- **Abstraction:** construct a simpler image of the problem.
Simpler but precisely defined; operation in well understood domains possible.
- **Interpretation:** translate solution back into real world.
Effects as predicted by knowledge about mathematical domain.

Mathematical Techniques

- Precise **language**.
 - Exact definition of objects.
 - Exact statements of facts.
- Correct **reasoning**.
 - Rules for thinking.
 - Rules for arguing.

These techniques enable us to understand definitions and to judge the correctness of arguments in any scientific domain.

Our Goals

- Insight into mathematical domains.
 - Construction.
 - Properties.
 - Operation.
- Mastering of mathematical techniques.
 - Defining.
 - Proving.

In-depth understanding, not just learning by heart.