

Exercise Sheet 2

Due: Sunday, 10.05.2009, 23:59:59 via SVN

For help, contact alp-staff@lists.iai.uni-bonn.de (staff only) or
alp-course@lists.iai.uni-bonn.de (staff and participants).

Please start working on the exercises early enough so that you can contact us in time in case of problems. Don't expect us to be available during weekend!

Symbolic computation: *Simplification of arithmetic expressions*

Symbolic computation is computation based on term manipulation instead of computing with numbers. Prolog excels in this domain. A standard example is the simplification of arithmetic expressions, which is our topic for this exercise sheet.

Task 1. *Designing term-based representations* (2 Points) – 5-10 minutes

Consider which terms you would need to represent arithmetic expressions with addition, difference, multiplication and division.

- a) For each operation, write down a most general function term and explain the meaning of its arguments by short keywords or self-documenting variable names.
- b) Write down the term that encodes “(a+5):5-1” in your representation.

Task 2. *Simplifying terms* (5 Points) – 20-30 minutes

- a) Write a recursive predicate, `simplify/2`, that expresses the relation between an arithmetic expression (1st argument) and its simplified form (2nd argument). Only implement the following simplification rules that you remember from school:
 - a. $a - a = 0$
 - b. $a : a = 1$
 - c. $(a + b) : c = a : c + b : c$
- b) Try your predicate, passing the term from Task 1b) as the 1st parameter. As the value for the second parameter, this test should yield the term that represents the expression, “a:5” according to the choices you made in Task 1a. If it doesn't, use the tracer (as explained on exercise sheet 1) to understand the problem. Try to fix it.

