Exercise Sheet 8 (due 26.06.2013)

Prof. Dr. Jürgen Giesl

Carsten Otto

Notes:

- To solve the programming exercises you can use the Prolog interpreter **SWI-Prolog**, available for free at `http://www.swi-prolog.org`. For Debian and Ubuntu it suffices to install the `swi-prolog` package. You can use the command "`swipl`" to start it and use "`[exercise8].`" to load the clauses from file **exercise8.pl** in the current directory.
- Solve these exercises in **groups of three**! For other group sizes **less points** are given!
- The solutions must be handed in **directly before (very latest: at the beginning of)** the exercise course on Wednesday, 26.06.2013, in lecture hall **AH 2**. Alternatively you can drop your solutions into a box which is located right next to Prof. Giesl's office (this box is emptied **a few minutes before** the exercise course starts).
- Please write the **names** and **immatriculation numbers** of all (three) students on your solution. Also please staple the individual sheets!

Exercise 1 (Arithmetic in Prolog): (4+6=10 points)

Important: In addition to handing in the solution on paper, please also mail your the solutions for this exercise to **lp13-hiwis@i2.informatik.rwth-aachen.de**. Indicate your immatriculation numbers in the subject of the mail and inside the Prolog file.

a) The number $n > 1$ is a prime number if and only if for each number $\{m \mid 1 < m < n\}$ the division n/m has a non-0 remainder. Follow this (simple) definition and implement the predicate `isPrime/1` in Prolog. You may make use of the predefined predicates `=`, `unify_with_occurs_check`, `is`, `=:=`, all relations (`>`, `<=`, ...), and predefined operations (`+`, `mod`, ...).

For this task you may not use the cut (`!`) or pre-defined negation (`\+`). These will be presented in later lectures!

Hints:

- Test your program, especially before you start solving the second part of this exercise.
- It suffices to use three clauses.
- The first prime numbers are $2, 3, 5, 7, 11, 13, 17, ...$

b) Please give a graphical representation of the (finite) SLD tree for the query `?- isPrime(3).`

Exercise 2 (Equalities): (5 points)

In Prolog there are the following five built-in predicates of arity 2 computing some kind of equality:

- `=`
- `==`
- `=:=`
- `is`
- `unify_with_occurs_check`

For each combination of two of these predicates, give an example of two terms where the application of the first predicate succeeds while the application of the second predicate fails or results in an error. If this is not possible for some combination, please explain why.