

## Extra Exercise Sheet 2

Due: Tuesday, 7. July 2009, 23:59:59 via SVN

For help, contact [alp-staff@lists.iai.uni-bonn.de](mailto:alp-staff@lists.iai.uni-bonn.de) (staff only) or  
[alp-course@lists.iai.uni-bonn.de](mailto: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!

For the following assignments, submit

1. the implementation of each predicate,
2. the Java code on which you tested your predicates and
3. the excerpt of your Prolog Console session that documents your successful tests.

The Java program should be such that it provides examples for all the relevant cases. You may use any existing program of your own or from open source repositories. If you already submitted the program to your SVN as part of a previous assignment, you do not need to resubmit it. Just, check-in a file that specifies the repository location of the program.

Your Java program should contain **no directly or indirectly recursive calls**.

See <http://sewiki.iai.uni-bonn.de/research/jtransformer/> for the documentation of JTransformer's program element facts that you need.

**Task 1.** *Direct change of a field* (4 Points) – 10 minutes

Write a predicate *changing\_method(?Field, ?Method)* that succeeds if *Field* is the identity of a field that is assigned within a method with identity *Method*.

**Task 2.** *Direct call of a method* (2 Points) – 10 minutes

Write a predicate *direct\_calling\_method(?Callee, ?Caller)* that succeeds if *Callee* is the identity of a method that is called somewhere inside the body of the method with identity *Caller*.

**Task 3.** *Enclosing hull of a method* (6 Points) – 10 minutes

Write a predicate *calling\_method(?Method, ?Caller)* that succeeds if the method represented by *Method* may be invoked through the execution of the method represented by *Caller*. That is *Caller* contains a call that either invokes *Method* itself or invokes some other method that again contains some code through which *Method* can be called.

It suffices that a call is contained inside a body of a method. It is not relevant whether it is actually executed at runtime. You don't have to implement the halting problem.

You also can assume that the program is recursion-free.

**Task 4.** *Calling class* (4 Points) – 10 minutes

Write a predicate *calling\_class(?Method, ?Class)* that succeeds if *Class* is the identity of a class within which there is a call to the method with identity *Method*.

**Task 5.** *Field changing classes*(4 Points) – 10 minutes

Write a predicate *changing\_class(?Field, ?Class)* which succeeds, if *Class* is an identity of a class that may change the state of the field represented by *Field*.