

# D&K - Semantic Web (Lab 1)

Horn rules reasoning  
19/09/2016

The goal of this practical exercise is to make you familiar with the reasoning mechanism of a simple knowledge representation language, called Horn Rule. You are asked to code in JAVA a program that allows you to reason with a sort of simple Horn knowledge bases as seen in the lecture. A basis of the program has been provided for you to start.

Note that the implementation of the reasoning algorithms should be done in a generic way so that it can reason over any knowledge bases. This implies that the code of your reasoning algorithms cannot consider any particular features of a specific knowledge base, instead it should only rely on the interfaces predefined that characterize a knowledge base.

You need to send your **code together with a report by 25<sup>th</sup> September**. This task is **individual**, that is, no teamwork is allowed. **Please respect the requirements** specified in the following, otherwise you'll be penalized in the grade.

## Preparation

1. Download the java archive from the following link :  
[http://www.lri.fr/~ma/M2DK/sw.lab1\\_NAME.zip](http://www.lri.fr/~ma/M2DK/sw.lab1_NAME.zip)
2. Import the java project into Eclipse.
3. Rename your project (refactor in Eclipse) by **replacing NAME by yours**.

## Code structure

The useful code for you is located in the directory src/ that contains:

- The package `problem.hornRules` contains an example of using the codes to reason with the knowledge base that we have seen in the previous lecture. The class `Tutorial1.java` is the class that constructs a rule base and a fact base of the knowledge base. And the class `ReasoningHorn.java` is the one with the main method that uses a reasoner to reasoning over a Horn knowledge base.
- In the package `sw.hornRule.algorithms`, you can find the definitions of the interface for a reasoner and the algorithms that realise this interface.
- The package `sw.hornRule.models` collects different classes linked to the modelisation of a reasoning system based on Horn rules, which extend the class `Formalism.java`. For example, the class `Variable.java` defines propositional variables; and `HornRule.java` is to model a set of Horn rules (e.g. **if  $l_1$  and  $l_2$  and ...  $l_n$ , then  $l$** ).

## Questions

1. Firstly, by reading the class `Tutorial1.java`, try to understand the articulation of the various components of this modeling.
2. Complete the classes `ReasoningForwardChaining.java`, `ReasoningBackwardChaining.java`, `ReasoningForwardChainingOptimised.java`, `ReasoningBackwardChainingwithQuestions.java`.
3. Compare your results with those of the tutorial 1 to verify your programs.
4. Export your java project with the name `ws.lab1_YourName.zip`, and send it to `ma@lri.fr`