Predicate invention using meta-interpretive learning

Unit: Machine Learning
Supervisor: Hendrik Blockeel
Daily supervisor: Sebastijan Dumancic
Contact: hendrik.blockeel@cs.kuleuven.be

Context:
Inductive logic programming is a domain that studies how to learn logic programs from examples of their input/output behavior. It has been shown that, using ILP, one can for instance give a set of examples of unsorted lists and their sorted version, and have the ILP system learn a program for sorting lists. However, it is typically only possible to learn a sorting method if all the auxiliary methods needed for it are defined in advance. Essentially, the ILP system needs the right building blocks, and just learns how to combine them in the right way.

If the auxiliary methods are not given, the system needs to invent them itself. In ILP, this is called predicate invention. This is a very hard task. However, recent research has shown that so-called meta-interpretive learning (MIL) may have potential for solving this task.

Goal:
The goal of this thesis is to investigate to what extent MIL can solve problems that require predicate invention.

Approach:
The student will read a few research papers on MIL. Next, he/she will download or program an implementation of MIL, and experiment with it in order to become familiar. Finally, it will be investigated how difficult it is to learn non-trivial predicate definitions (which corresponds to method definitions in other programming languages) using MIL. The focus will be on relatively simple problems: predicate definitions of the kind that one might find in textbooks on logic programming, but which require at least one auxiliary predicate.

Profile:
Suitable for a student in the ECS option who is theoretically strong and has a good background in formal logic or logic programming. A strong interest in machine learning is expected.