Partial evaluation of Probabilistic Logic Programs.

Context.
Partial evaluation is an automatic program transformation technique. The technique takes any program together with some information concerning the way that the program is likely to be queried. It then uses the information regarding the query to transform the program into a new program that works more efficiently for this kind of queries.

For instance, assume that you have a program that analyzes boolean expressions and that simplifies them (for instance by bringing them to a standard form). The program could have a top-procedure `analyze(Boolean_expression, Output_expression)`, where the first argument provides the program with some input expression and the program computes the output expression. We could now be interested in 'querying' this program with queries of the type `?- analyze(or(and(X, or(Y, Z)), or(U, V)), Output)`, where X,Y,Z,U and V are variables. The input position does not yet fully describe a boolean expression, because the variables X,Y,Z,U and V are placeholders for other boolean expressions that are not yet known to us. We only know part of the pattern of the expression yet.

Partial evaluation will automatically transform the initial program into a new program that expects queries of this type. The new program is likely to be much more efficient on these queries than the original program.

Partial evaluation is very well studied for declarative programming languages, in particular for Prolog. ECCE [1] is a well-know partial evaluator for Prolog.

Probabilistic Logic Programs are a new trend in A.I.-programming languages that combines Logic Programming (Prolog) with probabilistic reasoning. Logic rules are combined with probabilities to allow reasoning in uncertain situations. Problog [2] is one of the new languages of this kind.


Goal
The goal of the thesis is to experiment with idea's from partial evaluation and investigate whether they are directly applicable to Probabilistic Logic Programs.

The thesis is an initial step in the investigation of this problem. It is not expected that a full blown partial evaluation system will be obtained in the thesis. The thesis is supposed to deliver interesting examples where partial evaluation of Problog programs will lead to gains. The thesis should also develop a small prototype of a simple partial evaluator that achieves optimizations for some Problog programs.

Profile
A student taking this thesis proposal should be very experienced with programming and programming languages. Preferably the student should already be familiar with Prolog. The work will be partly theoretical, partly practical experimentation.

Contact:
Danny De Schreye, danny.deschreye@cs.kuleuven.be, Dept Computer Science building, room 1.033.