Executable first-order queries in the logic of information flows

Heba Aamer
Universiteit Hasselt

The logic of information flows (LIF) has recently been proposed as a general framework in the field of knowledge representation. The general aim of LIF is to model how information propagates in complex systems. LIF allows machine independent characterizations of computation; in particular, it allows tasks of a procedural nature to be modeled in a declarative fashion.

In this talk, we focus on the task of query processing under limited access patterns, where an information source is said to have a limited access pattern if it can only be accessed by providing values for a specified subset of the attributes; the source will then respond with tuples giving values for the remaining attributes. The problem of processing queries under limited access patterns is well-studied in the database literature. One of the proposed approaches is the "executable" fragment of first-order logic defined by Nash and Ludäscher. Executable FO is a syntactic fragment of FO in which formulas can be evaluated over information sources in such a way that the limited access patterns are respected. Furthermore, the syntactical restrictions are not very severe and become looser the more free variables are declared as input.

We show that LIF is well-suited for modeling the same task. Toward this goal, we introduce a variant of LIF called "forward" LIF, in a first-order setting. We define FLIF^{io}, a syntactical fragment of forward LIF, and show that it corresponds exactly to executable FO. FLIF^{io} is an io-disjoint fragment of FLIF which is defined in terms of input and output variables that are inferred for expressions. We establish inertia and input-determinacy properties for FLIF expressions which are instrumental in proving our equivalence between io-disjoint expressions and executable FO, but are also interesting in their own right.

An advantage is that our language, FLIF, provides a new, navigational perspective on query processing with limited access patterns. Another advantage of io-disjoint FLIF is that it is very clear how expressions in this language can be evaluated by plans. As we will show, the structure of the evaluation plan closely follows the shape of the expression, and all joins can be taken to be natural joins; no attribute renamings are needed.

This is joint work with Bart Bogaerts (Vrije Universiteit Brussel), Dimitri Surinx (Universiteit Hasselt), Eugenia Ternovska (Simon Fraser University), and Jan Van den Bussche (Universiteit Hasselt).