Infinity in reasoning about data and knowledge

Description for the general public

A popular slogan is that, unlike mathematics, computer science is about finite objects: because both modelled reality and available storage is finite, there is neither need nor space for infinity. This simplistic view is far from being true. In fact, infinity features quite prominently in computer science, both as a challenge and a means.

Indeed, while objects considered in computer science might be finite, they are of unbounded size. This is the case already for the simplest algorithmic tasks: an input to a decision problem is finite, but any algorithm solving the problem has to deal with arbitrarily large inputs. For example, a query evaluation algorithm always examines a finite database instance, but it must work for all of them. For traditional computer architectures this is an inherent part of design. But a recent trend in database hardware is to use so-called FPGAs, which are very fast but accept input of small fixed size. Using them to process unbounded data is one of the challanges this project will tackle. Specifically, we will investigate ways of combining FPGAs with external memory, without compromising efficiency.

Harder computational tasks are not about examining an object, but about deciding if there exists an object with given properties expressed in some formal way. Many problems related to databases have this form: Is there a database instance in which a given query selects at least one record? Do the two given queries select the same records in all database instances? As these objects we are looking for can be arbitrary large, this seems to require searching through an infinite set of possibilities. This not only looks hard, but sometimes is actually impossible. For instance, no computer program can aswer if there is a database instance from which a given database query written in SQL selects at least one record. In this project we aim to find cases that can be solved. Importantly, our goal is not to find specific properties, but rather design universal formalisms that can be used to express multiple such properties, useful form the point of view modern database applications, involving graph-structured data (like social networks or flight connections) and tree-structured data (like XML or JSON files, used for instance in communication between computer systems).

A relatively new ingredient in industrial scale informations systems are knowledge bases. While in databases one stores an object (necessarily finite) and asks queries about it, in knowledge bases one stores properties of an object and tries to infer new properties from the stored ones. A new property can be inferred if there is no object that satisfies the stored properties and does not satisfy the new property. Because the object itself does not need to be stored, a knowledge base can be used to reason about infinite objects. Thus, we might be interested in finding an infinite object with certain formally expressed properties (more precisely, we want to decide if it exists). Surprisingly, this task is sometimes easier that finding a finite object with the same properties, because in constructing infinite objects we have more freedom. We plan to exploit this. We will also investigate when such objects can be turned back into finite ones, as this is sometimes easier then looking directly for finite objects. We will focus on properties expressed in so-called description logics, which form the core of the W3C standards OWL 1 and OWL 2, the most popular frameworks for knowledge base systems.