Defeasible Reasoning for Datalog

Integrating rules with exceptions into Datalog ontologies.

What is Datalog?

Datalog is an expressive database query language. Unlike many other paradigmatic database languages, such as SQL, Datalog is based on the *logic programming* paradigm, meaning that its programs are represented as sets of formulas in a formal mathematical logic:

$$\begin{gathered}\text{human}(X) \to \text{mortal}(X).\\ \text{human}(\text{Socrates}).\\ \text{mortal}(\text{Socrates})?\end{gathered}$$

Datalog as an Ontology Language

Over the last decade, extensions of Datalog have been used as *ontology* languages, i.e. as a means of expressing the relationships between concepts and objects in an application domain. For instance, a partial ontology for human families can be expressed as follows:

$$\begin{gathered}\text{human}(X) \to \exists Y \text{mother}(X, Y).\\ \text{mother}(X, Y) \to \text{human}(Y).\\ \text{mother}(X, Y), \text{mother}(X, Y') \to Y = Y'.\end{gathered}$$

Modalities in Rules

Some ontologies contain rules involving modalities, such as time ("children *eventually* become adults") or certainty ("if there's sun then it *probably* isn't raining"). We reason with modality all the time in our everyday lives.

In this project we are interested in the modality of *defeasible rules*, which express things that are normally true, but sometimes false in exceptional circumstances. For instance, birds are *normally* able to fly, but some birds such as penguins have lost the ability to do so. In our extension of Datalog, this rule would be written like so:

$$\text{bird}(X) \rightsquigarrow \text{fly}(X).$$

The KLM Framework

The KLM framework is a mathematical theory of defeasibility for propositional logic. It provides a precise description of what it means for a defeasible rule to be true, as well as how one can *reason* about defeasible rules.

The KLM interpretation of defeasible rules is based on a number of simple mathematical axioms known as the *rationality postulates*:

$$\begin{array}{rl}(\textsc{Or}) & A \rightsquigarrow C, B \rightsquigarrow C \implies A \lor B \rightsquigarrow C\\ (\textsc{And}) & A \rightsquigarrow B, A \rightsquigarrow C \implies A \rightsquigarrow B \land C\\ & \ldots\end{array}$$

Defeasible Datalog

The goal of this project is to integrate the KLM framework with expressive versions of Datalog, in order to reason about ontologies containing defeasible rules. For instance, consider the following ontology:

$$\begin{gathered}\text{penguin}(X) \to \text{bird}(X).\\ \text{bird}(X) \rightsquigarrow \text{fly}(X).\\ \text{penguin}(X) \rightsquigarrow \neg\text{fly}(X).\end{gathered}$$

This states that penguins are birds, that birds normally fly and that penguins normally don't. Our Datalog extension is able to correctly infer from this ontology that birds normally aren't penguins:

$$\text{bird}(X) \rightsquigarrow \neg\text{penguin}(X).$$

A Prototype Implementation

A prototype implementation of our Datalog extension is available here:

https://github.com/Bubblyworld/drfol-reasoner.

It is written in Haskell, and uses Microsoft's Z3 solver to perform first-order satisfaction checks internally.

Guy Paterson-Jones [1] Tommie Meyer [2] Giovanni Casini [3]

[1]guy.paterson.jones@gmail.com, [2]tmeyer@cs.uct.ac.za, [3]giovanni.casini@isti.cnr.it