Defeasible Datalog

DESCRIPTION

Datalog is a declarative logic programming language that uses propositional logic for querying. Defeasible reasoning allows us to reason with contingent statements. Datalog does not possess the expressivity to handle defeasible reasoning.

OBJECTIVES

Enhance the expressivity of Datalog to allow for defeasible reasoning - i.e. Defeasible Datalog. Ensure that Defeasible Datalog is theoretically sound. Implement Defeasible Datalog, using RDFox.

METHOD

We approached Defeasible Datalog in four parts. Firstly, we enhanced Datalog with the syntax and semantics we required in order to use the rational closure algorithm. Secondly, we adapted the rational closure algorithm for Datalog. Thirdly, we showed our adapted rational closure algorithm satisfied the KLM properties - ensuring it can be used for defeasible reasoning. Lastly, we used the enhanced Datalog syntax and semantics as well as the adapted rational closure alogirhtm to implement Defeasible Datalog.

KLM PROPERTIES

A relation satisfying these properties have been shown to correctly handle defeasible reasoning. Thus, we had to show our adapted rational closure algorithm satisfied these properties.

$$\textbf{(LLE)}\ \frac{A(x) \equiv B(x),\ \mathcal{K} \mid\!\sim A(x) \leadsto C(x)}{\mathcal{K} \mid\!\sim B(x) \leadsto C(x)}$$

$$\textbf{(RW)}\ \frac{\mathcal{K} \mid\!\sim A(x) \leadsto B(x),\ B(x) \models C(x)}{\mathcal{K} \mid\!\sim A(x) \leadsto C(x)}$$

$$\textbf{(And)}\ \frac{\mathcal{K} \mid\!\sim A(x) \leadsto B(x),\ \mathcal{K} \mid\!\sim A(x) \leadsto C(x)}{\mathcal{K} \mid\!\sim A(x) \leadsto B(x) \wedge C(x)}$$

$$\textbf{(Or)}\ \frac{\mathcal{K} \mid\!\sim A(x) \leadsto C(x),\ \mathcal{K} \mid\!\sim B(x) \leadsto C(x)}{\mathcal{K} \mid\!\sim A(x) \vee B(x) \leadsto C(x)}$$

$$\textbf{(CM)}\ \frac{\mathcal{K} \mid\!\sim A(x) \leadsto B(x),\ \mathcal{K} \mid\!\sim A(x) \leadsto C(x)}{\mathcal{K} \mid\!\sim A(x) \wedge B(x) \leadsto C(x)}$$

$$\textbf{(RM)}\ \frac{\mathcal{K} \mid\!\sim A(x) \leadsto C(x),\ \mathcal{K} \not\mid\!\sim A(x) \leadsto \neg B(x)}{\mathcal{K} \mid\!\sim A(x) \wedge B(x) \leadsto C(x)}$$

RATIONAL CLOSURE

The Rational Closure algorithm was chosen as a way to allow for defeasible reasoning. We adapted rational closure to be used with Datalog syntax and semantics.

Algorithm 2: `RationalClosure`

Input: A knowledge base $\mathcal{K}$ and a defeasible rule $A(x) \leadsto B(x)$
Output: **true**, if $\mathcal{K} \mid\!\sim A(x) \leadsto B(x)$, and **false**, otherwise

1 $(R_0, \ldots, R_{n-1}, R_\infty, n) :=$ `BaseRank`$(\mathcal{K})$;
2 $i := 0$;
3 $R := \bigcup_{i=0}^{j<n} R_j$;
4 **while** $R_\infty \cup R \models \neg A(x)$ **and** $R \neq \emptyset$ **do**
5 $\quad R := R \setminus R_i$;
6 $\quad i := i + 1$;
7 **return** $R_\infty \cup R \models A(x) \rightarrow B(x)$;

WORKED EXAMPLE

Defeasible reasoning in the 'can penguins fly' example, along with KB and resultant RC

```
IMPORTING THE TBox (i.e. the datalog/.dlog file)...

THE CLASSICAL DATALOG RULES ARE:
        animal:Bird(?X) :- animal:Penguin(?X) .
        animal:Bird(?X) :- animal:Robin(?X) .
THE DEFEASIBLE DATALOG RULES ARE:
        ability:Fly(?X) :- animal:Bird(?X) .
        neg:False(?X) :- animal:Penguin(?X), ability:Fly(?X) .

----------------------------------------------------------------
RANKING THE RULES...

RULES HAVE BEEN RANKED AS FOLLOWS:
Level ∞:
        animal:Penguin(?X) :- animal:Bird(?X)
        animal:Robin(?X) .:-
Level 1:
        neg:False(?X) :- animal:Penguin(?X), ability:Fly(?X) .
Level 0:
        ability:Fly(?X) :- animal:Bird(?X) .

----------------------------------------------------------------
DOING QUERY...

DOES 'ability:Fly(?X) :- animal:Penguin(?X)' ENTAIL FROM THE KNOWLEDGE BASE?
NO

DOES 'ability:Fly(?X) :- animal:Penguin(?X)' ENTAIL FROM THE KNOWLEDGE BASE?
YES
```

```
Penguins --> Birds
Robin    --> Birds

Birds    ~~> Fly
Penguins ~/~> Fly
```

→

```
Rank ∞:
   Penguins -->  Birds
   Robin    -->  Birds
Rank 1:
   Penguins ~/~> Fly
Rank 0:
   Birds    ~~>  Fly
```

→ Thus, Penguins don't fly!

CONCLUSION

We were able to enhance Datalog to allow for defeasible reasoning. This was done using an adaptation of the rational closure algorithm. We were able to show that our adaptation is theoretically sound. We were also able to implement our adaptation using RDFox and showed its correctness with test cases.

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

Computer Science Department
University of Cape Town
Private Bag X3
Rondebosch
7701

JOSHUA ABRAHAM
(joshuasl8a@gmail.com)
Practical Implementation

THOMAS POWNALL
(pwntho001@myuct.ac.za)
Theoretical Implementation

PROF. THOMAS MEYER
(tmeyer@cs.uct.ac.za)
Supervisor

CAIR CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH