Abstract

In [12] the notion of a quantitative logic program has been introduced, and its declarative semantics explored. The operational semantics given in [12] is extended significantly in this paper - in particular, the notion of correct answer substitution is introduced and soundness and completeness results obtained. In addition, the completeness results for the and-or tree searching technique given in [12] is strengthened to be applicable to quantitative logic programs that are not well covered, thus removing one restriction in the completeness theorem obtained in [12]. In addition, the soundness and completeness results for SLDq-resolution in [12] are strengthened to apply to any nice QLP. Moreover, all these soundness and completeness results are applicable to existential queries unlike the results of [12,13] and [14] which are applicable to ground queries only. It was shown in [12] that the greatest supported model of a class of QLPs is semi-computable. In this paper, we give an explicit procedure to compute (partially) the greatest supported model, and obtain soundness and completeness results. This has applications in reasoning about beliefs.

1. Introduction

A quantitative logic program (QLP) was defined in [12] as a set of (universally closed) sentences of the form

$$A_0 : \mu_0 \leftarrow A_1 : \mu_1 \& \ldots \& A_k : \mu_k$$

where each $A_i$ is an atom, and $\mu_i \in [0, 1]$. Several different kinds of queries were (informally) introduced in [12] - we make these intuitions more formal here.

1. A ground query (or g-query, for short) is a formula of the form $\leftarrow A : \mu$ where $A \in B_Q$ and $\mu \in [0, 1]$ and $B_Q$ is the Herbrand base of the QLP $Q$.

2. Similarly, if $A$ is an atom (possibly non-ground), then $\leftarrow A : \mu$ is an existential query (or ex-query for short).

3. If $A \in B_Q$, then $\leftarrow A : ?$ is a confidence-request query (or cr-query, for short). Intuitively, given a ground atom $A$, the cr-query $\leftarrow A : ?$ asks: “What is the truth value assigned $A$ by the least model of $Q$ ?”. 

SLDq-resolution
Different classes of QLPs (e.g. nice QLPs, decent QLPs, well-covered QLPs) were defined in [12], and some soundness and completeness results were obtained (for ground queries) with respect to these classes of programs. In this paper, we:

1. Introduce the notion of answer substitution, and give an operational semantics that is both sound and complete w.r.t. computation of answer substitutions for existential queries for the class of function-free QLPs. In particular, we strengthen the completeness results of [12] by dropping the restriction that QLPs be well-covered. In addition, we prove that SLDq-resolution [12] computes correct answer substitutions for ex-queries. We also obtain a completeness result. This was not done in [12].

2. In [12], it was shown that if Q is a nice, decent QLP, then the greatest supported model of Q is semi-computable (r.e.). However, no algorithm to partially compute the greatest supported model was given there. In this paper, we explicitly develop such an algorithm, and obtain soundness and completeness results. We have already shown in [12] that the semi-computability of the greatest supported model of Q allows us to reason about beliefs. To our knowledge, no such algorithm to reason about (even a limited notion of) consistency exists in the literature.

As this paper is primarily an extension of [12], we expect the reader to be familiar with the model theoretic aspects of quantitative logic programming described there. In addition, the algorithm given there for computing ground queries with respect to well-covered QLPs is similar to the approach of Van Emden [14], and we consequently see no reason to repeat that procedure here. We will, however, give some of the other main definitions and theorems of that paper in the next section. Section 3 is concerned with describing processing of existential queries. Section 4 described how the greatest supported model can be computed.

2. QLPs - An Overview

DEF 1: The set $T$ of truth values is $[0, 1] \cup \{\top\}$. The partial ordering $\leq$ on $T$ is defined as shown in the following Hasse diagram (Fig.1):

As usual, $x \geq y$ iff $y \leq x$. Similarly, $\succ, \prec$ are the irreflexive restrictions of $\geq, \leq$ respectively. Note that according to this ordering, $0.1 \geq 0.2$. 