Saccà, Domenico


Summary: From the introduction: The notion of (total) stable model represents an interesting solution to the problem of providing a formal semantics to logic programs where rules contain negated goals. Although multiple stable models may exist for the same program, a deterministic stable model semantics can be enforced using two main approaches: the possibility (or credulous or brave) semantics (a ground atom is true if it is in any stable model), and the certainty (or skeptical or cautious) semantics (a ground atom is true if it is in every stable model).

Certainty semantics has the severe drawback that, for a program with no stable models, every ground atom is inferred to be true so that what is not ‘possible’ may eventually become ‘certain’! To remove such an anomaly, we propose definite semantics which coincides with certainty semantics for the programs admitting stable models, but it infers no ground atom, rather than every ground atom for all other programs. A surprising result is that definite stable model semantics for programs on finite universes has an expressive power higher than certainty semantics. In fact, whereas bound (i.e., binary) DATALOG\neg queries (i.e., queries on function-free logic programs with negation) are known to express coNP decision problems under certainty semantics, we show that definite semantics enables bound DATALOG\neg queries to express decision problems in the class D^p, i.e., problems that can be defined as the conjunction of a problem in NP and a problem in coNP. (Note that, unless NP = coNP, D^p is different from NP \cap coNP, for it properly contains NP \cup coNP.).

MSC:
68N17 Logic programming
68Q55 Semantics in the theory of computing

Keywords:
complexity class; relational database; DATALOG\neg; expressive power; logic programming; stable model; unique and critical SAT

Software:
Datalog

Full Text: DOI

References:


[19] Schlipf, J.S., The expressive powers of the logic programming semantics, (), 196-204

[20] Schlipf, J.S., A survey of complexity and undecidability results in logic programming, (), 143-164

[21] Ullman, J.D., ()

[22] Vardi, M.Y., The complexity of relational query languages, (), 137-146

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.