10-1-2000

Classes of Logic Programs which Possess Unique Supported Models

Anthony K. Seda

Pascal Hitzler
pascal.hitzler@wright.edu

Follow this and additional works at: http://corescholar.libraries.wright.edu/cse

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation
http://corescholar.libraries.wright.edu/cse/82

This Conference Proceeding is brought to you for free and open access by Wright State University's CORE Scholar. It has been accepted for inclusion in Computer Science and Engineering Faculty Publications by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
Classes of Logic Programs which Possess Unique Supported Models

Pascal Hitzler¹ and Anthony Karel Seda
Department of Mathematics
University College, Cork, Ireland
{phitzler,aks}@ucc.ie, http://maths.ucc.ie/~{pascal,seda}/

Logic programming is concerned with the use of logic as a programming language. The main manifestation of this computing paradigm is in the various versions of Prolog which are now available, in which computation is viewed as deduction from sets of Horn clauses, although there is also growing interest in the related form known as answer set programming, see [10]. The reference [2] contains a good survey of the growth of logic programming over the last twenty five years both as a stand-alone programming language and as a software component of large information systems.

One advantage a logic program $P$ has over conventional imperative and object oriented programs is that it has a natural machine-independent meaning, namely, its logical meaning. This is often referred to as its declarative semantics, and is usually taken to be some “standard” model canonically associated with $P$. Unfortunately, it is often the case that there are many possible choices for the standard model, some even taken in many-valued logic, which do not in general coincide and all of which have a claim to be “the natural choice” depending on one’s view of non-monotonic reasoning [6, 7, 11]. However, most two-valued semantics which are proposed are refinements of a particular model called the supported model semantics or Clark completion semantics [5], which we study in this paper. Indeed, we focus on programs which have a unique reading under this semantics or, in other words, programs which have a unique supported model and which are called uniquely determined programs. Several classes of programs with this property have been studied in the literature including acceptable programs [1], which are of great importance in connection with termination, and the acyclic and the locally hierarchical programs [4] studied from the point of view of denotational semantics. In general, such classes as these are defined by restricting their syntax using well-founded orders.

Starting from these results, our discussion will revolve around the following aspects of classes of programs, which we call unique supported model classes or USM classes, each member of which has a unique supported model.

1) By means of operators in three-valued logic, it is possible to obtain a unifying framework for the USM classes mentioned above, see [8]. Using this observation, results which apply to these classes separately can be generalized to a considerable extent. This approach gives rise to the USM class of $\Phi$*-accessible* programs. This class is remarkable in that not only does each program in it have a unique supported model, but in addition it is computationally adequate in the sense that every partial recursive function can be implemented, under Prolog, by such programs. This does not hold, for example, for the subclass of acceptable programs which always terminate under the Prolog selection rule.

2) The general classes arrived at in (1) are susceptible to a semantical analysis using fixed-point theorems from general topology. The application of these theorems yields strong results on how to obtain the supported-model semantics by a process involving

¹The first named author acknowledges financial support under grant SC/98/621 from Enterprise Ireland.
limits in spaces which carry a notion of distance. Results of this kind make it possible to study logic programs in the context of neural networks [9] or chaotic dynamical systems [12].

(3) It is reasonable to expect that many of the standard semantics will coincide for uniquely determined programs, so that they can be viewed as semantically unambiguous. A formal analysis shows that this is the case for the $\Phi'$-accessible programs, for example, and our results provide some interconnections between the various semantics which have been proposed, although this issue has yet to be fully understood and is much investigated in the current literature.

References


