Overview of disjunctive logic programming*

Jack Minker

* Dedicated to Chitta Baral, José Alberto Fernández, Jorge Lobo and Arcot Rajasekar.

The field of disjunctive logic programming started approximately in 1982 and has reached its first decade. The first result in the field was the development of the Generalized Closed World Assumption (GCWA). Major results have been made in this field since 1986. An overview is presented of the developments that have taken place, which include model theoretic, proof theoretic and fixpoint semantics for disjunctive, and extended normal disjunctive theories including alternative forms of negation.

1. Introduction

The field of disjunctive logic programming started approximately in 1982 and has reached its first decade. The first result in the field appears in a paper by Minker [89] in which he develops a consistent theory for negation in disjunctive theories. The paper also set forth the concept of minimal models for computing answers both for positive and negative atoms. The work extended the Closed World Assumption (CWA) [116], which deals with Horn theories. Reiter had shown, in his paper on the CWA, that for disjunctive theories, the CWA might lead to an inconsistency.

Following 1982, renewed interest arose on disjunctive logic programming in 1986 as a consequence of a Workshop on Foundations of Deductive Databases and Logic Programming, organized by Minker [90]. Following the workshop, Minker and Rajasekar [94, 97], developed fixpoint, model theoretic and proof theoretic semantics. This work led to many theoretical developments in disjunctive and normal disjunctive logic programs, including theories of negation and disjunctive deductive databases. Much of this work was performed at the University of Maryland and is the subject of a research monograph, Foundations of Disjunctive Logic Programming [77]. The theoretical results contained in that monograph extend what is known in the theory of logic programs as developed in the monograph by Lloyd [73, 74] and the paper by Apt [4]. In this paper, an overview is presented of the developments that have taken place in disjunctive logic programming over the past twelve years.

Before providing an overview of the field, I would like to express my great appreciation to my students, Chitta Baral, José Alberto Fernández, Jorge Lobo,
and Arcot Rajasekar. This paper is dedicated to them in appreciation for their joint work with me that led to many of the results described here. I have learned a great deal from them and benefited from the interactions that we had both at the personal and professional level. The research on disjunctive logic programming and disjunctive deductive databases that was developed at Maryland could not have been accomplished without them.

2. Background

It is assumed that the reader has a background in logic programming as described in [66, 130], or other books on logic programming, and is familiar with the major theoretical results given in [4, 73, 74]. A personal perspective on the development of disjunctive logic programming is given in [92]. Kowalski [67] provides a personal perspective of the history of the field of logic programming. A sketch of the major results in logic programming is given in [92, 93], including some topics not covered in [73, 74]. Theoretical results in disjunctive logic programming are given in [77]. Theses by Baral [6], Fernández [38], Lobo [75], and Rajasekar [112] have relevant theoretical results. The intent of this paper is to provide an overview of the field, sketching the results and relevant research.

Throughout the paper we will refer to different classes of clauses. These are definite Horn clauses, normal clauses, extended clauses, disjunctive clauses, normal disjunctive clauses, and extended disjunctive clauses. Associated with each type of clause is a class of logic programs. These are termed, respectively, definite logic programs, normal logic programs, extended logic programs, disjunctive logic programs, normal disjunctive logic programs, and extended disjunctive logic programs. These logic programs are defined as follows.

A definite logic program consists of definite Horn clauses, defined as follows

\[ C \leftarrow B_1, \ldots, B_M, \]  

where \( C \) and the \( B_i, 1 \leq i \leq M, \) are atoms and \( M \geq 0. \)

A normal logic program consists of normal clauses, defined as follows

\[ C \leftarrow B_1, \ldots, B_M, \text{not } A_1, \ldots, \text{not } A_N, \]  

where \( C, \) the \( B_i, 1 \leq i \leq M, \) and the \( A_j, 1 \leq j \leq N \) are atoms and \( M, N \geq 0. \) By \text{not}, we mean negation by default, rather than logical negation.

An extended logic program consists of extended clauses, defined as follows

\[ L \leftarrow B_1, \ldots, B_M, \text{not } A_1, \ldots, \text{not } A_N \]  

where \( L \) is a literal, that is, either an atom, or an atom preceded by the operator \(-\), and the \( B_i, 1 \leq i \leq M, \) the \( A_j, 1 \leq j \leq N, \) are literals and \( M, N \geq 0. \) By \(-, \) we mean classical negation.