CHAPTER I
INTRODUCTION

In the area of inductive learning, generalization is the main operation, and the usual definition of induction is based on logical implication. Least general generalizations are of special interest since they are known to be correct whenever there exist correct generalizations. In recent years there has been a rising interest in clausal representation of knowledge in machine learning. Plotkin [8] has already described a technique for computation of least general generalizations of clauses under θ-subsumption. Unfortunately, the results obtained by this technique are sometimes too general with respect to implication.

The difference between θ-subsumption and logical implication is important since almost all inductive learners that use clausal representations perform generalization under θ-subsumption rather than generalization under implication. The main reason is that there is a well known and reasonably efficient technique to compute least general generalizations under θ-subsumption, but not under implication.

Plotkin himself realized that generalization under θ-subsumption was incomplete for a certain type of clause known as a recursive clause. Recursion is an important program structure in logic programming. The ability to learn recursive clauses is therefore crucial when using a clausal representation. In Muggleton [6] it is shown that the incompleteness of generalization under θ-subsumption only concerns one type of generalization of recursive clauses, which is often called an indirect root.

Idestam-Almquist [2] described the well known technique to compute least general generalizations under θ-subsumption and the most important theoretical results connected with it. He studied the theory of generalization under implication, and noted that implication between clauses is undecidable. He therefore introduced a stronger form of implication, called T-implication, which is decidable between clauses. He showed that for every finite set of clauses there exists a least general generalization under T-implication. Finally, he presented a technique, called recursive anti-unification, to compute indirect roots of clauses. Recursive anti-unification is a generalization of anti-unification, which is the usual technique used to compute least general generalizations under θ-subsumption.

However there exist cases for which recursive anti-unification does not work, cases where at least one of the clauses contains a structure called a cross connection. We will describe a technique, called the J-algorithm, to compute indirect roots of clauses. By this technique we can even compute roots of clauses containing cross connections, which are guaranteed to be generalizations under implication.

In this work we will only consider Horn clauses. In Chapter II background on logic and generalization under θ-subsumption of clauses is given, and in Chapter III generalization under implication and θ-proof are discussed. Finally, in Chapter IV we give our technique, the J-algorithm, and some concluding remarks.