An Abductive Procedure for the CMS/ATMS

Katsumi Inoue

ICOT Research Center
Institute for New Generation Computer Technology
1-4-28 Mita, Minato-ku, Tokyo 108, Japan
inoue@icot.or.jp

Abstract

This paper concerns procedural semantics for a variety of ATMSs. Reiter & de Kleer view an ATMS as a kind of abduction in which the best explanation of a formula is defined as a minimal conjunction of hypotheses that explain the formula. However, they do not give any algorithm to compute such minimal explanations of a formula in their CMS that is a generalization of de Kleer's basic ATMS. In this paper, we use the notion of characteristic clauses to make precise definitions of the CMS and the ATMS and to produce a sound and complete abductive procedure based on an extension of linear resolution. By means of this abductive procedure, we give the CMS algorithms for computing minimal explanations in the interpreted approach and for updating them in the compiled approach. We then present algorithms for generating and updating labels of nodes in an extended ATMS that accepts non-Horn justifications and literal assumptions. Finally, how a variation of the abductive procedure can be used to answer queries for circumscription of ground theories is presented.

1 Introduction

An assumption-based truth maintenance system (ATMS) [4] has been widely used when problems require reasoning in multiple contexts. However, this basic ATMS can only handle the restricted form of formulas, and is described algorithmically rather than declaratively or model-theoretically, and no proof of its correctness is given, so it is not obvious how to generalize or refine it. The motivation for this research was the desire to formalize generalizations of the ATMS within simple model and proof theories.

Recent investigations such as those of Reiter & de Kleer [22] and Levesque [16] show that there are strong connections between an ATMS and a logical account of abduction or hypothesis generation [20, 3, 9, 19]. An ATMS can be characterized by the following type of abduction:

Definition 1.1 Let \( W \) be a set of formulas, \( A \) a set of ground literals (called the assumptions), and \( G \) a closed formula. A conjunction \( H \) of elements of \( A \) is an explanation of \( G \) from \((W, A)\) if (i) \( W \cup \{H\} \models G \) and (ii) \( W \cup \{H\} \) is satisfiable.

An explanation \( H \) of \( G \) from \((W, A)\) is minimal if no proper sub-conjunct of \( H \) is an explanation of \( G \) from \((W, A)\), that is, if no sub-conjunct \( H' \) of \( H \) satisfies \( W \cup \{H'\} \models G \).
The ATMS is precisely intended to generate all and only minimal explanations [11]. In the ATMS terminology, the set of minimal explanations of a node \( G \) from the justifications \( W \) and the assumptions \( A \) is called the label of \( G \), which is consistent, sound, complete and minimal. The basic ATMS [4] is restricted to accepting only Horn clause justifications and atomic assumptions. In the above declarative conditions for an ATMS, justifications can contain non-Horn clauses, and assumptions are allowed to be literals, so that this generalization covers de Kleer's various extended versions of the ATMS [5, 6, 7], Dressler's extended basic ATMS [8], and Reiter & de Kleer's clause management system (CMS) [22].

In spite of its usefulness in a wide range of applications, the algorithms for the ATMS in [4, 5, 6] have not yet been proved to be correct with respect to the declarative semantics. Although the CMS is well defined and the basic connection between resolution and the CMS processing is given in [22], there has not yet been any complete algorithm for computing labels of a formula for non-Horn theories in terms of popular and useful resolution methods. One of the problems is that although linear resolution is widely used and contains several restriction strategies, it is incomplete for consequence-finding [18] so that it cannot be directly used as an ATMS procedure.

The goal of this paper is to provide a sound and complete abductive procedure which solves the above problems for the CMS and ATMSs. In the remaining sections, we describe abduction as the problem of finding the characteristic clauses [1, 24] that are theorems of a given set of clauses and that belong to a distinguished sub-vocabulary of the language. We will give an extension of propositional linear resolution procedures which is complete for characteristic-clause-finding, then show ways in which to implement the CMS and the extended ATMS described above for both label generating (the interpreted approach) and label updating (the compiled approach). Since this extended ATMS can accept literal assumptions and non-Horn clauses, the methods described in this paper can also be applied to better implementations of theorem provers for closed world assumptions [1] and circumscription [21, 10] of ground theories, based on abductive procedures [13]. Unless otherwise specified, proofs for theorems and propositions are shown in Appendix.

2. Characteristic Clauses

We begin with some definitions and notations that will be used throughout this paper. We shall assume a propositional language with finitely many propositional symbols \( \mathcal{A} \) and with logical connectives. The set of literals is defined as: \( \mathcal{A}^\pm = \mathcal{A} \cup \neg \mathcal{A} \), where \( \neg \mathcal{S} \) means the set formed by taking the negation of each element in \( S \). A clause is a finite set of literals, understood disjunctively; the empty clause is denoted by \( \Box \). A conjunctive normal form (CNF) formula is a conjunction of clauses. Let \( C \) and \( C' \) be two clauses. \( C \vdash C' \) denotes a clause whose literals are those in the difference of \( C \) and \( C' \). \( C \) is said to subsume \( C' \) if every literal in \( C \) occurs in \( C' \) (\( C \subseteq C' \)). In logical notation, \( C \) subsumes \( C' \) if \( \models C \supseteq C' \). For a set of clauses \( \Sigma \), by \( \mu \Sigma \) or \( \mu [\Sigma] \) we mean the set of clauses of \( \Sigma \) not subsumed by any other clause of \( \Sigma \).

**Definition 2.1** Let \( \Sigma \) be a set of clauses.

1. A clause \( C \) is an implicate of \( \Sigma \) if \( \Sigma \models C \). The set of implicants of \( \Sigma \) is denoted by \( Th(\Sigma) \).
2. The prime implicants of \( \Sigma \) are: \( PI(\Sigma) = \mu Th(\Sigma) \).