Expert Systems

(Resolution Systems)

Lecture 003

Sandro Spina

Introduction to Resolution

- The resolution rule of inference (Robinson 65) is commonly implemented in theorem proving AI programs.

- Resolution is the primary rule of inference in PROLOG

- Instead of many different inference rules of limited applicability such as modus ponens, etc., PROLOG uses the one general purpose inference rule of resolution.

Normal Forms

- Before resolution can be applied, the wff must be in normal or standard form.
- Three main types of normal form include:
  - Conjunctive normal form
  - Full Clausal form (Kowalski) and
  - Its Horn clause subset
- The basic idea of normal form is to express wffs in a standard form that uses only the ∨, ∧, and possibly the ¬.
- The resolution method is then applied to normal form wffs in which all other connectives and quantifiers have been eliminated.
- This conversion is necessary because the resolution method is an operation on pairs of disjuncts which produces new disjuncts, which simplifies the wff.

Conjunctive Normal Form

- The following illustrates a wff in conjunctive normal form, which is defined as the conjunction of disjunctions which are literals.
- $(P1 \vee P2 \vee \ldots) \wedge (Q1 \vee Q2 \vee \ldots) \wedge \ldots (Z1 \vee Z2 \vee \ldots)$
- Terms such as Pi must be literals, which mean that they contain contain no logical connectives such as the conditional and biconditional, or quantifiers.
- A literal is an atomic formula or a negated atomic formula, for example the following wff:
  - $(A \vee B) \wedge (\neg B \vee C)$ is in conjunctive normal form
  - The terms within the parenthesis are clauses.

Clausal Form

- A full clausal form expression is generally written as follows:
  - A1, A2, ... An → B1, B2, ... Bm
  - Which is interpreted as saying that is all subgoals A1, A2, ... An are true, then one or more of B1 or B2 ... Bm are also true. Using standard predicate notation we get:
  - A1 Λ A2 … An → B1 V B2 ... Bm
- This can be expressed in disjunctive form as the disjunction of literals using the equivalence p → q = ¬p V q, so
  - A1 Λ A2 … An → B1 V B2 ... Bm
  - = ¬(A1 Λ A2 … An) V (B1 V B2 … Bm)
  - = ¬A1 V ¬A2 ... ¬An V B1 V B2 ... Bm (using de Morgan's Law)

Horn Clause

- PROLOG uses a restricted type of clausal form, the Horn clause, in which only one head is allowed. There we get:
- A1, A2, ... An → B
- That is like saying that B is true only when A1 through to An are true.

Resolution Basic Goal

- The basic goal of resolution is to infer a new clause, the resolvent, from two other clauses called parent clauses.
- The resolvent will have fewer terms than the parents. By continuing this process of resolution, eventually a contradiction will be obtained or the process is terminated because no progress is being made. A simple (very simple) example of resolution is shown in the following argument.
  - $A \vee B$ (Parent Clause)
  - $A \vee \neg B$ (Parent Clause)
  - A (Resolvent)
  - How does the conclusion follow:
  - $(A \vee B) \wedge (A \vee \neg B) = A \vee (B \wedge \neg B) = A$
  - Using the Axioms of Distribution

Clauses and Resolvents

Parent Clauses	Resolvent	Meaning
p → q, p (or) ¬p V q, p	q	Modus Ponens
p → q, q → r (or) ¬p V q, ¬q V r	p → r (or) ¬p V r	Chaining or Hypothetical Syllogism
¬p V q, p V q	q	Merging
¬p V ¬q, p V q	¬p V p (or) ¬q V q	TRUE (a tautology)
¬p, p	nil	FALSE (a contradiction)

Resolution Systems

- Given wffs A_1, A_2,...A_n and a logical conclusion of theorem C, we know:
  - $A_1 \wedge A_2 \wedge \ldots \wedge A_n \vdash C$ is equivalent to stating that:
  - $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow C \;\equiv\; \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \vee C$
  - $\equiv\; \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n \vee C$

    (this should be a tautology !!)

  - Let us now take the negation as follows:
  - $\neg\,[\,A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow C\,] \;\equiv\; \neg[\neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \vee C]$
  - $\equiv\; [\neg\neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \wedge \neg C]$
  - $\equiv\; A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \neg C$
  - (and this a contradiction !!!)

  - Both are equivalent ways of proving that a formula C is a theorem. In the first the first case we have to see if it is true in all cases. Equivalently, for the the second formula we prove a theorem by showing it leads to a contradiction !!

An example using a resolution refutation tree

- Condiser the argument:
  - $A \rightarrow B$
  - $B \rightarrow C$
  - $C \rightarrow D$

- To prove that the conclusion $A \rightarrow D$ is a theorem by resolution refutation, we first convert it to disjunctive form using the equivalence: $p \rightarrow q \equiv \neg p \lor q$.
  - We get $\neg A \lor D$
  - And its negation $\neg(\neg A \lor D) \equiv A \land \neg D$
  - The resolution method can now be applied to
    $(\neg A \lor B) \land (\neg B \lor C) \land (\neg C \lor D) \land A \land \neg D$

The resolution refutation tree