Mathematical Foundations of Computing

Propositional Calculus : 6

N Geetha
AM & CS
PSG College of Technology

Resolution

- **Resolution** is a valid inference rule producing a new clause implied by two clauses containing *complementary literals*
  - A literal is an atomic symbol or its negation, i.e., P, ~P
  - A clause is a disjunction of literals
- Amazingly, this is the only interference rule you need to build a sound theorem prover
  - Inference never produces any contradictions: sound
  - Resolution is not complete
  - Based on proof by contradiction and usually called resolution refutation
- The resolution rule was discovered by Alan Robinson (CS, Univ. of Syracuse) in the mid 60s

Why is logic so hard?

- A large collection of facts (predicates)
- A large collection of possible transformations (rules)
  - Some of these rules apply to a single fact to yield a new fact
  - Some of these rules apply to a pair of facts to yield a new fact
- So at every step you must:
  - Choose some rule to apply
  - Choose one or two facts to which you might be able to apply the rule
    - If there are n facts
      - There are n potential ways to apply a single-operand rule
      - There are n * (n - 1) potential ways to apply a two-operand rule
  - Add the new fact to your ever-expanding fact base
- The search space is huge!

Resolution

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.(our Hypotheses)
- To use resolution, put KB into *conjunctive normal form* (CNF), where each sentence written as a disjunction of (one or more) literals
- Use only one rule
- Facts to KB grows linearly

Tautologies

$(A \rightarrow B) \leftrightarrow (\sim A \vee B)$

$(A \vee (B \wedge C)) \leftrightarrow (A \vee B) \wedge (A \vee C)$

Example : KB: [$P \rightarrow Q$, $Q \rightarrow R \wedge S$]

- KB in CNF: [$\sim P \vee Q$, $\sim Q \vee R$, $\sim Q \vee S$]
- Resolve KB(1) and KB(2) producing: $\sim P \vee R$ *(i.e., $P \rightarrow R$)*
- Resolve KB(1) and KB(3) producing: $\sim P \vee S$ *(i.e., $P \rightarrow S$)*
- New KB: [$\sim P \vee Q$, $\sim Q \vee \sim R \vee \sim S$, $\sim P \vee R$, $\sim P \vee S$]

- Note: Sometimes we use the notation of a set for a clause: e.g. $\{P, Q, \neg R\}$ corresponds to the clause ($P \vee Q \vee \neg R$); the empty clause sometimes written as Nil or {} is equivalent to False (also use a box [];

Soundness of the resolution inference rule

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
<u>*False*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>
<u>*True*</u>	<u>*False*</u>	<u>*False*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>
<u>*True*</u>	<u>*False*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>
True	*True*	*False*	*True*	*False*	*True*
<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>

From the rightmost three columns of this truth table, we can see that

$$(\alpha \vee \beta) \wedge (\neg\beta \vee \gamma) \leftrightarrow (\alpha \vee \gamma)$$

is valid (i.e., always true regardless of the truth values assigned to α, β and γ

Proof by Contradiction using resolution

Steps:

1. Convert the hypotheses to clauses
2. Convert the conclusion to clause form;
3. Assume the NEGATION of the conclusion is T.
4. Take the first parent clause to be Neg(Conclusion) and another parent clause.
5. Resolve the parent clauses and get the resolvent
6. Continue this process until we get an empty resolvent. Then a contradiction has been found. If and when we find a contradiction, we have proved the clause.

- If a contradiction exists, it will eventually be found.
- If no contradiction exists, it is possible that the procedure will never terminate.
- In Predicate logic, the situation is more complicated since we must consider all possible ways of substituting values for the variables.

E.g. Suppose our statements are:

$$(A \vee B \vee D) \wedge (B \vee \neg D \vee G) \wedge (A \vee \neg G \vee B)$$

And we want to see if $(A \vee B)$ is true.

We start with assuming the negation of what we want to prove.

1) P1: $\neg A$ $\neg B$
2) P2 : $(A \vee \neg G \vee B)$
3) Resolvent : $(\neg G \vee B)$
3) Resolve again : $(\neg G)$
4) P1 : $\neg G$ P2 : $(B \vee \neg D \vee G)$
5) Resolvent : $(B \vee \neg D)$
6) P1 : $(B \vee \neg D)$ P2: $(A \vee B \vee D)$
7) Resolvent : $(A \vee B)$
8) Apply $\neg A$ and $\neg B$
9) Empty Resolvent; So we have a contradiction!

Eg.2:

- Start with:
  - it_is_raining $\vee$ it_is_sunny
  - it_is_sunny $\rightarrow$ I_stay_dry
  - it_is_raining $\rightarrow$ I_take_umbrella
  - I_take_umbrella $\rightarrow$ I_stay_dry
- Convert to clause form:
  1. it_is_raining $\vee$ it_is_sunny
  2. $\neg$it_is_sunny $\vee$ I_stay_dry
  3. $\neg$it_is_raining $\vee$ I_take_umbrella
  4. $\neg$I_take_umbrella $\vee$ I_stay_dry
- Prove that I stay dry:
  5. $\neg$ I_stay_dry
- Proof:
  6. (5, 2) $\neg$it_is_sunny
  7. (6, 1) it_is_raining
  8. (7, 3) I_take_umbrella
  9. (8, 4) I_stay_dry
  10. (9, 5) NIL
- Therefore, $\neg$($\neg$I_stay_dry)
  - I_stay_dry

Resolution of in Propositional logic:

① For a real world problem the following facts were identified

i) P ii) $(P \wedge Q) \rightarrow R$ iii) $(S \vee M) \rightarrow Q$

iv) M

The goal of it is to prove R

Soln

ii) $(P \wedge Q) \rightarrow R$

$\neg P \vee \neg Q \vee R$

iii) $(S \vee M) \rightarrow Q$

$(\neg S \wedge \neg M) \vee Q$

$(\neg S \vee Q) \wedge (\neg M \vee Q)$

Clauses are

i) P

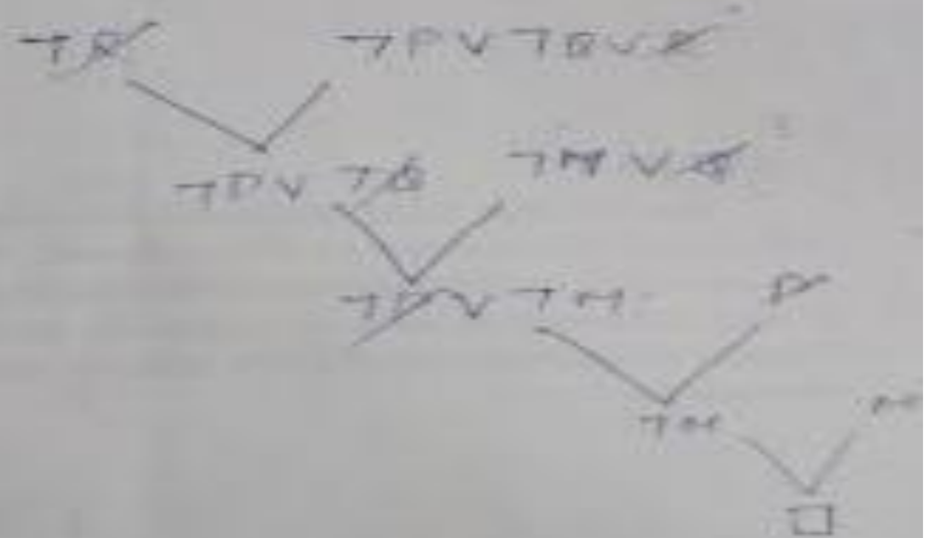
ii) $\neg P \vee \neg Q \vee R$

iii) $\neg S \vee Q$

iv) $\neg M \vee Q$

v) M

vi) R (G)



Hence goal is proved $\neg(S \rightarrow Q) \rightarrow$

$(R \rightarrow S) \rightarrow$

2) The following are the facts:

i) $(P \vee Q) \rightarrow \neg R$ ii) $P \wedge Q$ iii) $(R \wedge \neg S) \rightarrow (T \vee W)$

iv) $\neg R \rightarrow (R \wedge \neg S)$ v) $\neg T$

Prove whether W is inferable?

Soln

i) $(P \vee Q) \rightarrow \neg R$

$\neg(P \vee Q) \vee \neg R$

$(\neg P \wedge \neg Q) \vee \neg R$

$(\neg P \vee \neg R), (\neg Q \vee \neg R)$

iv) $\neg R \rightarrow (R \wedge \neg S)$

$R \vee (R \wedge \neg S)$

$R \vee R \wedge (R \vee \neg S)$

$R \wedge (R \vee \neg S)$

iii) $(R \wedge \neg S) \rightarrow (T \vee W)$

$\neg(R \wedge \neg S) \vee (T \vee W)$

$\neg R \vee S \vee T \vee W$

Facts

F_1: $\neg P \vee \neg R$

F_2: $\neg Q \vee \neg R$

F_3: P

F_4: Q

F_5: $\neg R \vee S \vee T \vee W$

F_6: R

F_7: $R \vee \neg S$

F_8: $\neg T$

F_9: $\neg W$

$\cancel{\neg W}$ $\neg R \vee S \vee T \vee \cancel{W}$

$\neg R \vee S \vee T$ $R \vee \neg S$

T $\neg T$

$\square$

∴ W is inferable

Resolution

- Any complete search algorithm applying only the resolution rule, can derive any conclusion entailed by any knowledge base in propositional logic – resolution can always be used to either confirm or refute a sentence – refutation completeness
  (Given A, it's true we cannot use resolution to derive A OR B; but we can use resolution to answer the question of whether A OR B is true.)
- Theoretical Basis of Resolution Procedure (Chang & Lee, 1973) states:1) To show that a set of clauses S is unsatisfiable, it is necessary to consider only interpretations over a particular set, called the Herbrand Universe of S
- 2) A set of clauses is unsatisfiable iff a finite subset of ground instances (all bound variables have values substituted for them) of S is unsatisfiable
- Resolution provides a way of finding a contradiction by trying a minimum number of substitutions