Resolution in the Propositional Calculus

Outline

- A New Rule of Inference: Resolution
- Converting Arbitrary wffs to Conjunctions of Clauses
- Resolution Refutations
- Resolution Refutation Search Strategies

A New Rule of Inference: Resolution

- Literal: either an atom (positive literal) or the negation of an atom (negative literal).
- Ex: Clause is $\{P, Q, \neg R\}$ wff

  (equivalent to $P \vee Q \vee \neg R$)
- Ex: Empty clause { } is equivalent to *F*

Resolution on Clauses (1)

- From $\{\lambda\} \cup \Sigma_1$ and, $\{\neg\lambda\} \cup \Sigma_2$ we can infer

  $\Sigma_1 \cup \Sigma_2$

  - called the resolvent of the two clauses
  - this process is called resolution

- Examples
  - $R \vee P \text{ and } \neg P \vee Q \Rightarrow R \vee Q$: chaining
  - $R \text{ and } \neg R \vee P \Rightarrow P$: modus ponens

Resolution on Clauses (2)

- $P \vee Q \vee R \vee S$ with $\neg P \vee Q \vee W$ on P

  $\Rightarrow Q \vee R \vee S \vee W$

- $P \vee Q \vee \neg R$ and $P \vee W \vee \neg Q \vee R$

  - Resolving them on *Q* : $P \vee \neg R \vee R \vee W$
  - Resolving them on *R* : $P \vee \neg Q \vee Q \vee W$
  - Since $\neg R \vee R$ and $\neg Q \vee Q$ are *True*, the value of each of these resolvents is *True*.
  - We must resolve either on *Q* or on *R*.
  - $P \vee W$ is not a resolvent of two clauses.

Soundness of Resolution

- If $\{\lambda\} \cup \Sigma_1$ and $\{\neg\lambda\} \cup \Sigma_2$ both have *true*, $\Sigma_1 \cup \Sigma_2$ is *true*.
- Proof : reasoning by cases
  - Case 1
    - If λ is *True*, $\sum_2$ must *True* in order for $\{\neg\lambda\} \cup \Sigma_2$ to be *True*.
  - Case 2
    - If λ is *False*, $\sum_1$ must *True* in order for $\{\lambda\} \cup \Sigma_1$ to be *True*.
  - Either Σ_1 or Σ_2 must have value True.
  - $\Sigma_1 \cup \Sigma_2$ has value True.

Converting Arbitrary wffs to Conjunctions of Clauses

- Ex: $\neg(P \supset Q) \vee (R \supset P)$

1.	$\neg(\neg P \vee Q) \vee (\neg R \vee P)$	Equivalent Form Using ∨
2.	$(P \wedge \neg Q) \vee (\neg R \vee P)$	DeMorgan
3.	$(P \vee \neg R \vee P) \wedge (\neg Q \vee \neg R \vee P)$	Distributive Rule
4.	$(P \vee \neg R) \wedge (\neg Q \vee \neg R \vee P)$	Associative Rule

- Usually expressed as $\{(P \vee \neg R), (\neg Q \vee \neg R \vee P)\}$

Converting wffs to Conjunctions of Clauses

Ex. Convert the following wffs to conjunctions of clauses

$$\mathrm{p} \wedge (\mathrm{q} \rightarrow \mathrm{r}) \rightarrow \mathrm{s}$$

$$((p \vee q) \rightarrow r) \rightarrow p$$

$$\begin{aligned}
& p \wedge (q \rightarrow r) \rightarrow s \\
& = \sim (p \wedge (\sim q \vee r)) \vee s \\
& = \sim p \vee \sim (\sim q \vee r) \vee s \\
& = \sim p \vee (q \wedge \sim r) \vee s \\
& = \sim p \vee s \vee (q \wedge \sim r) \\
& = (\sim p \vee s \vee q) \wedge (\sim p \vee s \vee \sim r)
\end{aligned}$$

Resolution Refutations

- Resolution is not complete.
  - For example, $P \wedge R \mapsto P \vee R$
  - We cannot infer $P \vee R$ using resolution on the set of clauses $\{P, R\}$ (because there is nothing that can be resolved)
- We cannot use resolution directly to decide all logical entailments.

Resolution Refutation Procedure

1. Convert the wffs in Δ to clause form, i.e. a (conjunctive) set of clauses.
2. Convert the negation of the wff to be proved, ω , to clause form.
3. Combine the clauses resulting from steps 1 and 2 into a single set, Γ
4. Iteratively apply resolution to the clauses in Γ and add the results to Γ either until there are no more resolvents that can be added or until the empty clause is produced.

Completeness of Resolution Refutation

- Completeness of resolution refutation
  - The empty clause will be produced by the resolution refutation procedure if $\Delta \mapsto \omega$.
  - Thus, we say that propositional resolution is *refutation complete*.
- Decidability of propositional calculus by resolution refutation
  - If Δ is a finite set of clauses and if $\Delta \not\mapsto \omega$,
  - Then the resolution refutation procedure will terminate without producing the empty clause.

A Resolution Refutation Tree

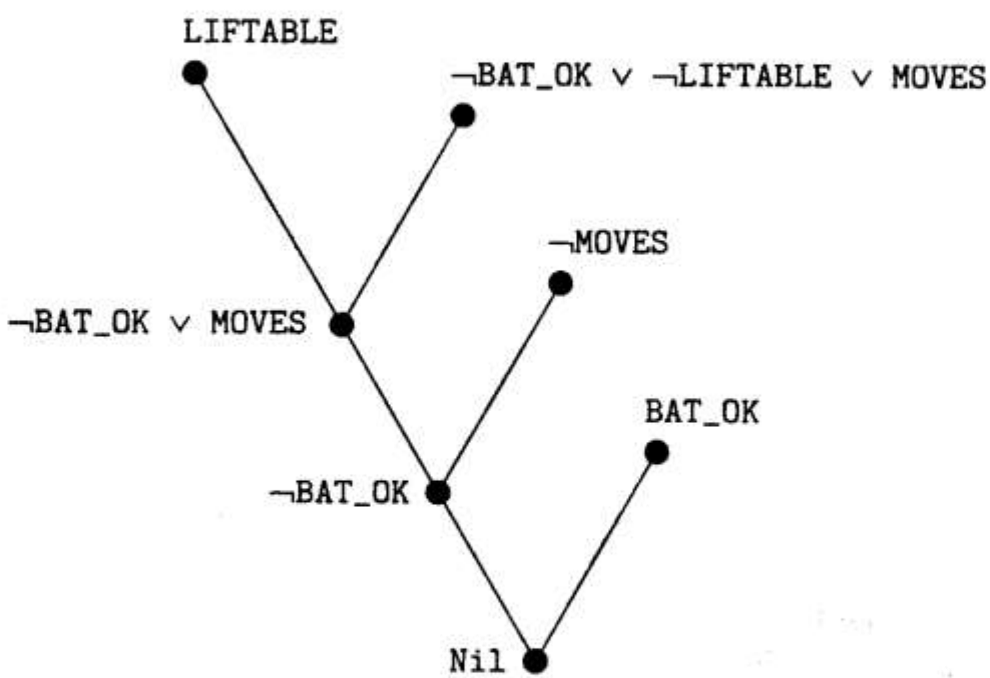

Figure 14.1

A Resolution Refutation Tree

Given:

1. BAT_OK
2. ¬MOVES
3. BAT_OK ∧ LIFTABLE ⊃ MOVES

Clause form of 3:

4. ¬BAT_OK ∨ ¬ LIFTABLE ∨ MOVES

Negation of goal:

5. LIFTABLE

Perform resolution:

6. BAT_OK ∨ MOVES (from resolving 5 with 4)
7. BAT_OK (from 6, 2)
8. Nil (from 7, 1)

Resolution Refutations Search Strategies

- Ordering strategies
  - Breadth-first strategy
  - Depth-first strategy
    - with a depth bound, use backtracking.
  - Unit-preference strategy
    - prefer resolutions in which at least one clause is a unit clause.
- Refinement strategies
  - Set of support
  - Linear input
  - Ancestry filtering

Refinement Strategies

- Set of support strategy
  - Allows only those resolutions in which one of the clauses being resolved is in the set of support, i.e., those clauses that are either clauses coming from the negation of the theorem to be proved or descendants of those clauses.
  - Refutation complete
- Linear input strategy
  - at least one of the clauses being resolved is a member of the original set of clauses.
  - Not refutation complete
- Ancestry filtering strategy
  - at least one member of the clauses being resolved either is a member of the original set of clauses or is an ancestor of the other clause being resolved.
  - Refutation complete