Artificial Intelligence
Chapter 14
Resolution in the Propositional Calculus

Biointelligence Lab
School of Computer Sci. & Eng.
Seoul National University

Outline

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

14.1 A New Rule of Inference: Resolution

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

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

14.1.2 Resolution on Clauses

- 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
  - $R \vee P$ and $\neg P \vee Q \Rightarrow R \vee Q$: chaining
  - R and $\neg R \vee P \Rightarrow P$: modus ponens
  - Chaining and modus ponens is a special case of resolution
  - $P \vee Q \vee R \vee S$ with $\neg P \vee Q \vee W$ on $P \Rightarrow Q \vee R \vee S \vee W$

14.1.2 Resolution on Clauses (Cont'd)

- $P \vee Q \vee \neg R$ 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.

14.1.3 Soundness of Resolution

- To show soundness of an inference rule *R*,
  - Show that $\Delta \vdash_R w$ implies $\Delta \models w$.
- Since $\{\lambda\} \cup \Sigma_1$ and $\{\neg\lambda\} \cup \Sigma_2 \vdash_{res} \Sigma_1 \cup \Sigma_2$,
  - Show $\{\lambda\} \cup \Sigma_1$ 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*, Σ_2 must *True* in order for $\{\neg\lambda\} \cup \Sigma_2$ to be *True*.
  - Case 2
    - If λ is *False*, Σ_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.

14.2 Converting Arbitrary wffs to Conjunctions of Clauses (CNF)

- Any wff in propositional calculus can be converted to an equivalent CNF.
- Ex: $\neg(P \supset Q) \vee (R \supset P)$

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

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

14.3 Resolution Refutations

- Resolution is not complete.
  - For example, $P \wedge R \nvdash P \vee R$
  - 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.

Proof by Contradiction

- Reductio by absurdum
  - If the set Δ has a model but $\Delta \cup \{\neg w\}$ does not, then $\Delta \models w$.

- If we can show that $\Delta \cup \{\neg w\} \vdash_{res} \{\}$ (equivalent to *F*), then $\Delta \models w$.

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 $\Delta \vdash \omega$.
  - 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 $\Delta \nvdash \omega$,
  - Then the resolution refutation procedure will terminate without producing the empty clause.

A Resolution Refutation Tree

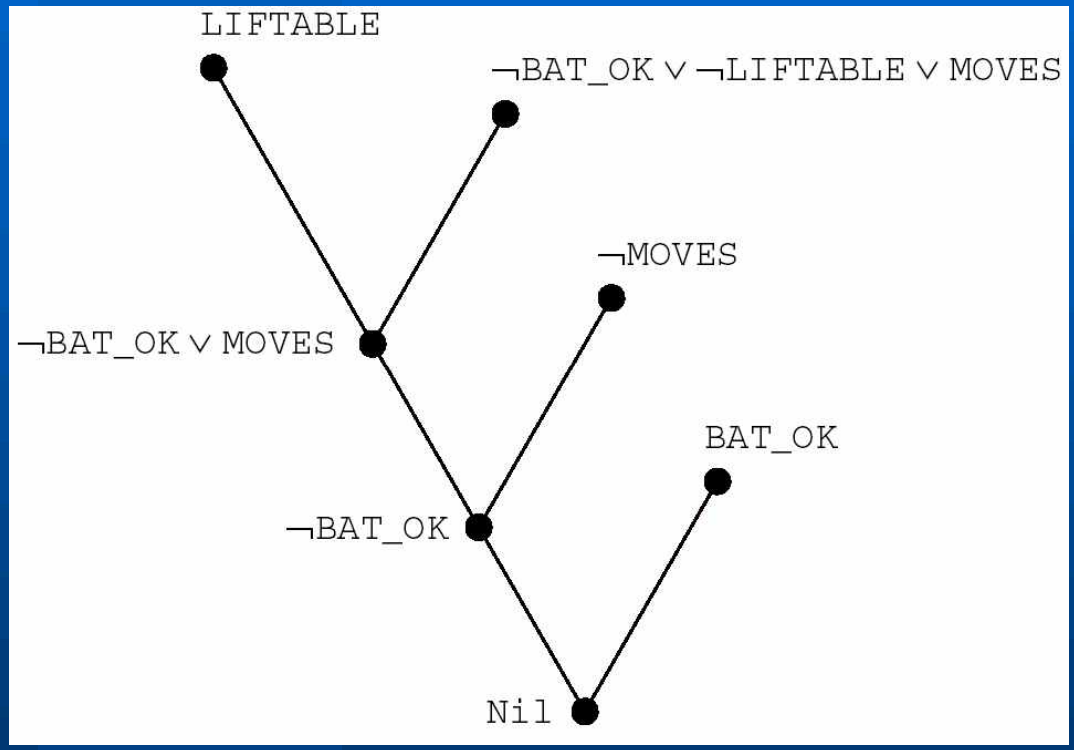

Figure 14.1 A Resolution Refutation Tree

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)

14.4 Resolution Refutations Search Strategies

- Ordering strategies
  - Which resolutions should be performed first?
  - 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

- Permit only certain kinds of resolutions to take place at all.
- 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

14.5 Horn Clauses

- A Horn clause: a clause that has at most one positive literal.
- Ex: P, $\neg P \vee Q$, $\neg P \vee \neg Q \vee R$, $\neg P \vee \neg R$
- Three types of Horn clauses.
  - A single atom: called a "fact"
  - An implication: called a "rule"
  - A set of negative literals: called "goal"
- There are linear-time deduction algorithms for propositional Horn clauses.