Lecture 18

Inference II

Lusi Li
Department of Computer Science
ODU

Reading for This Class:
Chapter 7, Russell and Norvig

Review

- **Last Class**
  - **Inference**
    - **Truth Table Approach**
    - **Inference Rule Approach**
- **This Class**
  - **Resolution Algorithm**
- **Next Class**
  - **First Order Logic**

Limitations of Inference Rules Approach

- The inference rules can be applied whenever suitable premises are found in the KB
- The conclusion of the rule must follow regardless of what else is in the KB
- The inference rules covered so far are sound, but if the available inference rules are inadequate, then it's not complete
- Now, we introduce a single inference rule, resolution, that gives a complete inference algorithm when coupled with any complete search algorithm

Logical Inference Approaches

- **KB |= α ?**
- **Three approaches:**
  - **Truth-table approach**
  - **Inference rules**
  - **Resolution algorithm**
    - **Proof by contradiction**

Resolution Algorithm

- **KB |= α ?**

- **Two inference rules:**
  - **unit resolution rule**
  - **full resolution rule**

- **Conversion to conjunctive normal form (CNF)**

- **Satisfiability (SAT) problem**

- **Resolution algorithm**

Resolution Algorithm

- KB $|= \alpha$?

- Two inference rules:
  - unit resolution rule
  - full resolution rule

- Conversion to conjunctive normal form (CNF)

- Satisfiability (SAT) problem

- Resolution algorithm

Complementary Literals

- A **literal** is an either an atomic sentence or the negated atomic sentence, e.g.:

  P or $\neg P$

- Two literals are **complementary** if one is the negation of the other, e.g.:

  P and $\neg P$

- A **clause** is a disjunction of literals, e.g.,

  $P \lor \neg Q \lor R$

  $\neg P \lor Q$

- A **unit clause** is a disjunction of a single literal, e.g.,

  P

  $\neg P$

Inference Rules-Unit Resolution

❖ **Unit Resolution rule: (From a clause, if one of the literal is false, then you can infer the other one is true)**

$$\frac{\alpha \vee \beta, \neg\beta}{\alpha} \quad \text{or equivalently} \quad \frac{(\alpha \vee \beta) \wedge \neg\beta}{\alpha}$$

- **If l_i and m are complementary literals, then**

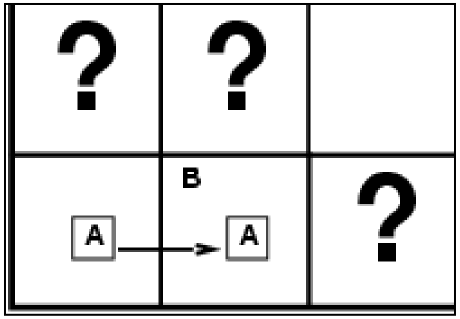
$$\frac{l_1 \vee \cdots \vee l_k, \quad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \ldots \vee l_k}$$

- **If both sentences in the premise are true, then conclusion is true.**

Inference Rules-Unit Resolution

KB:

- **R1: $\neg P_{1,1}$**
- **R2: $\neg B_{1,1}$**
- **R3: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$**
- **R4: $\neg P_{2,1}$**
- **R5: $B_{2,1}$**
- **R6: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$**
- **…**
- **R13: $P_{1,1} \vee P_{2,2} \vee P_{3,1}$**
- **R14: $P_{2,2} \vee P_{3,1}$**



Apply unit resolution rule to R1 and R13

Inference Rules-Resolution

- **Resolution rule: (From two clauses, a new clause is produced containing all the literals of the two original clauses *except* the one pair of complementary literals. Or equivalently, implication is transitive.)**

$$\frac{\alpha \lor \beta, \quad \neg\beta \lor \gamma}{\alpha \lor \gamma} \quad \text{or equivalently} \quad \frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

- **If l_i and m_j are complementary literals, then**

$$\frac{l_1 \lor \cdots \lor l_k, \quad m_1 \lor \cdots \lor m_n}{l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \ldots \lor l_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \ldots \lor m_n}$$

- **If both sentences in the premise are true, then conclusion is true.**

Inference Rules-Resolution

Note: we can resolve only one pair of complementary literals at a time.
E.g., we can resolve P and $\neg P$ to deduce

$$\frac{P \vee \neg Q \vee R, \quad \neg P \vee Q}{\neg Q \vee Q \vee R}$$

Or, we can resolve Q and $\neg Q$ to deduce

$$\frac{P \vee \neg Q \vee R, \quad \neg P \vee Q}{P \vee R \vee \neg P}$$

Then if the new clause contains complementary literals, it is discarded (as a tautology) since it is always-true.

Thus, we cannot resolve on both P and Q at once to infer R!

Inference Rules-Resolution

- **Soundness of resolution inference rule:**
  - **If l_i is true, then m_j is false**
    - **hence $(m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \ldots \vee m)$ must be true**
  - **If l_i is false, then m_j is true**
    - **hence $(l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \ldots \vee l_k)$ must be true**
  - **no matter what**
    - $(l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \ldots \vee l_k) \vee (m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \ldots \vee m)$ **is true**

- **In the inference rule approach, the resolution rule can also be applied.**

- **Any complete search algorithm, applying only the resolution rule, can derive any conclusion entailed by any knowledge base in propositional logic.**

- **A resolution-based theorem prover can, for any sentences α and β in propositional logic, decide whether α ⊨ β**

Resolution Algorithm

- KB |= α ?

- Two inference rules:
  - unit resolution rule
  - full resolution rule

- Conversion to conjunctive normal form (CNF)

- Satisfiability (SAT) problem

- Resolution algorithm

Resolution and CNF

- **Resolution inference rule applies only to clauses (disjunctions of literals)**
- ***Every sentence of propositional logic is logically equivalent to a conjunction of disjunctions of literals***
- **A sentence expressed as a conjunction of clauses is said to be in conjunctive normal form or CNF:**
  - **conjunction of clauses**
  - **E.g.,** $(\boldsymbol{A} \vee \neg \boldsymbol{B}) \wedge (\boldsymbol{B} \vee \neg \boldsymbol{C} \vee \neg \boldsymbol{D})$

$$
\begin{aligned}
CNFSentence &\rightarrow Clause_1 \wedge \cdots \wedge Clause_n \\
Clause &\rightarrow Literal_1 \vee \cdots \vee Literal_m \\
Literal &\rightarrow Symbol \mid \neg Symbol \\
Symbol &\rightarrow P \mid Q \mid R \mid \ldots
\end{aligned}
$$

Procedure for Obtaining CNF

1. **Replace $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$**
2. **Replace $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$**
3. **Move $\neg$ inwards: $\neg(\neg \alpha)$, $\neg(\alpha \vee \beta)$, $\neg(\alpha \wedge \beta)$**
4. **Distribute $\vee$ over $\wedge$, e.g., $(\alpha \wedge \beta) \vee \gamma$ becomes $(\alpha \vee \gamma) \wedge (\beta \vee \gamma)$**
5. **Flatten nesting: $(\alpha \wedge \beta) \wedge \gamma$ becomes $\alpha \wedge \beta \wedge \gamma$**

Example: Conversion to CNF

Convert the sentence $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ into CNF:
The steps are as follows:

1. **Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.**

   $$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

2. **Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.**

   $$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$

3. **Move $\neg$ inwards using De Morgan rule:**

   $$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$$

4. **Apply distributivity law ($\vee$ over $\wedge$) and flatten:**

   $$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

The original sentence is now in CNF, as a conjunction of three clauses. It is much harder to read, but it can be used as input to a resolution procedure.

Resolution Algorithm

- KB |= α ?

- Two inference rules:
  - unit resolution rule
  - full resolution rule

- Conversion to conjunctive normal form (CNF)

- Satisfiability (SAT) problem

- Resolution algorithm

Inference Problem and Satisfiability

- **Inference procedures based on resolution work by using the principle of proof by contradiction or proof by refutation.**
- **A sentence is satisfiable if it is true in some model, e.g., $A \vee B$, C**
- **A sentence is unsatisfiable if it is true in no models, e.g., $P \wedge \neg P$**
- **Entailment is connected to inference via the Deduction Theorem:**

$KB \models \alpha$ if and only if ($KB \Rightarrow \alpha$) is valid
(note: $KB \Rightarrow \alpha$ is ($\neg KB \vee \alpha$))

- **Satisfiability is connected to inference via the following:**

$KB \models \alpha$ if and only if ($KB \wedge \neg\alpha$) is unsatisfiable

- **Thus proof by contradiction:**

proving α given KB by checking the unsatisfiability of ($KB \wedge \neg \alpha$)

Satisfiability (SAT) Problem

- **Determine whether a sentence in the conjunctive normal form (CNF) is satisfiable (i.e. can evaluate to true)**

$$(P \lor Q \lor \neg R) \land (\neg P \lor \neg R \lor S) \land (\neg P \lor Q \lor \neg T) \land \cdots$$

- **Variables:**
  - **Propositional symbols (P, Q, R, T, S)**
  - **Values: *True, False***
- **Constraints:**
  - **Every conjunct must evaluate to true, at least one of the literals in every conjunct must evaluate to true**

- **Thus, to show (KB $\land \neg$ α) is unsatisfiable is to show no models evaluate CNF to true.**

$$\ldots \land P \land \neg P$$

Resolution Algorithm

- **KB |= α ?**
- **Two inference rules:**
  - **unit resolution rule**
  - **full resolution rule**
- **Conversion to conjunctive normal form (CNF)**
- **Satisfiability (SAT) problem**
- **Resolution algorithm**

Resolution Algorithm

- **To show that KB $\models \alpha$, we show that (KB $\wedge \neg\alpha$) is unsatisfiable**

1. (KB $\wedge \neg\alpha$) is converted into CNF

2. Apply iteratively the resolution rule to the resulting clauses

- Each pair that contains complementary literals is resolved to produce a new clause, which is added to the set if it's not present

3. Stop when:

- **Contradiction (empty clause {}) is reached:**
  - **E.g., P, $\neg$ P $\Rightarrow$ {}**
  - **Prove the entailment**
- **No more new clauses can be derived**
  - **Reject the entailment**

- **{ } is a disjunction of no disjuncts is equivalent to False, thus the contradiction**

Resolution Algorithm

- **Proof by contradiction, i.e., show that KB $\wedge \neg\alpha$ is unsatisfiable**

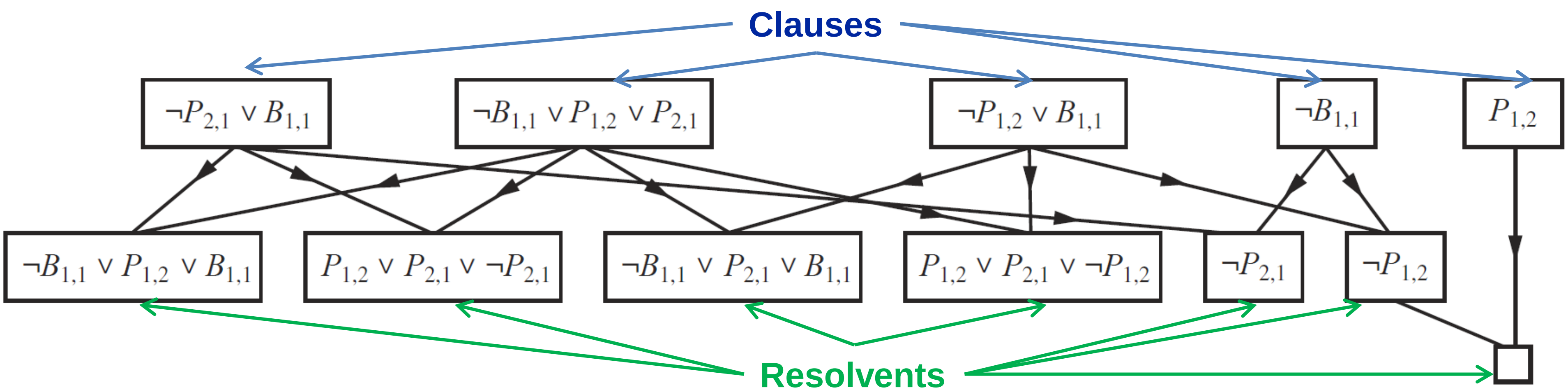
function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  $clauses \leftarrow$ the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  $new \leftarrow \{\ \}$
  loop do
    for each pair of clauses C_i, C_j **in** $clauses$ **do**
      $resolvents \leftarrow$ PL-RESOLVE(C_i, C_j)
      if $resolvents$ contains the empty clause **then return** *true*
      $new \leftarrow new \cup resolvents$
    if $new \subseteq clauses$ **then return** *false*
    $clauses \leftarrow clauses \cup new$

Resolution Example

- KB = $(B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$
- $\alpha = \neg P_{1,2}$

- (1) KB $\wedge \neg \alpha$ = $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1}) \wedge \neg B_{1,1} \wedge P_{1,2}$
  - Set of clauses: $\{(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}), (\neg P_{1,2} \vee B_{1,1}), (\neg P_{2,1} \vee B_{1,1}), \neg B_{1,1}, P_{1,2}\}$
- (2) Apply resolution rules to all clauses



Partial application of PL-RESOLUTION to a simple inference in the wumpus world. $\neg$ **$P_{1,2}$** is shown to follow from the first four clauses in the top row.

Resolution Example

- KB: $P \wedge Q$; $P \Rightarrow R$; $(Q \wedge R) \Rightarrow S$
- α : S

Resolution Example

- KB: $P \wedge Q;\ P \Rightarrow R;\ (Q \wedge R) \Rightarrow S$
- $\alpha : S$
- **Step1: convert (KB $\wedge \neg\alpha$) into CNF**
  - **$P \wedge Q \longrightarrow P \wedge Q$**
  - **$P \Rightarrow R \longrightarrow \neg P \vee R$**
  - **$(Q \wedge R) \Rightarrow S \longrightarrow \neg Q \vee \neg R \vee S$**
  - **$\neg\alpha \longrightarrow \neg S$**
  - **$(KB \wedge \neg\alpha) = P \wedge Q \wedge (\neg P \vee R) \wedge (\neg Q \vee \neg R \vee S) \wedge \neg S$**
  - **Set of clauses: $\{P, Q, (\neg P \vee R), (\neg Q \vee \neg R \vee S), \neg S\}$**

- **Step2: Apply resolution rule to on the set of clauses**
- **$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$**

$R \qquad \neg R \vee S$

S

Contradiction $\longrightarrow$ **$\{\}$**

Proved: S

Inference Rules Summary

- **All of the listed inference rules are sound.**
- **We can prove this through the truth table.**

◇ **Modus Ponens** or **Implication-Elimination**: (From an implication and the premise of the implication, you can infer the conclusion.)

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

◇ **And-Elimination**: (From a conjunction, you can infer any of the conjuncts.)

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

◇ **And-Introduction**: (From a list of sentences, you can infer their conjunction.)

$$\frac{\alpha_1, \ \alpha_2, \quad \ldots, \quad \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

◇ **Or-Introduction**: (From a sentence, you can infer its disjunction with anything else at all.)

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}$$

◇ **Double-Negation Elimination**: (From a doubly negated sentence, you can infer a positive sentence.)

$$\frac{\neg\neg\alpha}{\alpha}$$

◇ **Unit Resolution**: (From a disjunction, if one of the disjuncts is false, then you can infer the other one is true.)

$$\frac{\alpha \vee \beta, \quad \neg\beta}{\alpha}$$

◇ **Resolution**: (This is the most difficult. Because β cannot be both true and false, one of the other disjuncts must be true in one of the premises. Or equivalently, implication is transitive.)

$$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma} \quad \text{or equivalently} \quad \frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

Enumeration of Models

Similarly:

$\mathcal{P}$: Set of propositional symbols in $\{KB, \neg\alpha\}$

n: Size of $\mathcal{P}$

ENTAILS?(KB, α)

For each of the 2^n models on $\mathcal{P}$ do

If it is a model of $\{KB, \neg\alpha\}$ then return no

Return yes

Summary

- **Literal**
- **Unit Resolution Rule**
- **Full Resolution Rule**
- **Resolution algorithm**

What I want you to do

- **Review Chapter 7**