Lecture 17

Inference

Lusi Li
Department of Computer Science
ODU

Reading for This Class:
Chapter 7, Russell and Norvig

Review

- **Last Class**
  - **Knowledge-based Agents**
  - **Propositional Logic**
- **This Class**
  - **Inference**
- **Next Class**
  - **Inference II**

Entailment and Inference

- **KB $\models \alpha$**
  - **If KB entails α, then all models (assigning ‘true’ or ‘false’ values to symbols) that evaluate the KB to True also evaluate α to True.**
- **KB $\vdash \alpha$**
  - **Inference is a procedure for deriving a new sentence α from KB following some inference approach.**

- **The inference approach is sound if it derives only sentences that are entailed by KB. <u>It can only prove true things.</u>**
  - **If KB $\vdash \alpha$, then KB $\models \alpha$**
  - **Contrapositive of soundness: if KB $\not\models \alpha$, then KB $\nvdash \alpha$**

- **The inference algorithm is complete if whatever can be entailed by KB can also be inferred from KB. <u>It can prove all true things.</u>**
  - **If KB $\models \alpha$, then KB $\vdash \alpha$**
  - **Contrapositive of completeness: if KB $\nvdash \alpha$, then KB $\not\models \alpha$**

Logical Inference Problem

- **Given:**
  - **KB: a set of sentences**
  - **α: a sentence**

- **Does a KB semantically entail α? KB $\models \alpha$?**
  - **In other words, in all models where KB is true, α is also true?**

Logical Inference Approaches

- **KB |= α ?**

- **Three approaches:**
  - **Truth-table approach**
  - **Inference rules**
  - **Resolution algorithm**

Logical Inference Approaches

- **KB $|= \alpha$?**

- **Three approaches:**
  - **Truth-table approach**
  - **Inference rules**
  - **Resolution algorithm**

Truth-Table Approach

- **KB |= α ?**
- **A two steps procedure:**
  - **Generate table for all possible models (n symbols $\rightarrow$ 2^n entries)**
  - **Check whether the sentence α is true whenever the sentences in KB are true**

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\quad \alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	*True*	*True*				
True	*True*	*False*				
True	*False*	*True*				
True	*False*	*False*				
False	*True*	*True*				
False	*True*	*False*				
False	*False*	*True*				
False	*False*	*False*				

Truth-Table Approach

- **KB |= α ?**
- **A two steps procedure:**
  - Generate table for all possible models (n symbols $\rightarrow 2^n$ entries)
  - Check whether the sentence α is true whenever the sentences in KB are true

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	True	True	True	True	True	True
True	True	False	True	True	True	True
True	False	True	True	False	False	True
True	False	False	True	True	True	True
False	True	True	True	True	True	True
False	True	False	False	True	False	True
False	False	True	True	False	False	False
False	False	False	False	True	False	False

Truth-Table Approach

- KB |= α ?
- A two steps procedure:
  - Generate table for all possible models (n symbols → 2^n entries)
  - Check whether the sentence α is true whenever the sentences in KB are true

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	True	True	True	True	True	True
True	True	False	True	True	True	True
True	False	True	True	False	False	True
True	False	False	True	True	True	True
False	True	True	True	True	True	True
False	True	False	False	True	False	True
False	False	True	True	False	False	False
False	False	False	False	True	False	False

Truth-Table Approach

- **KB |= α !**
- **The truth-table approach is sound and complete for the propositional logic!**

Example: $KB = (A \lor C) \land (B \lor \neg C)$ $\alpha = (A \lor B)$

A	B	C	$A \lor C$	$(B \lor \neg C)$	KB	α
True	True	True	True	True	True	True
True	True	False	True	True	True	True
True	False	True	True	False	False	True
True	False	False	True	True	True	True
False	True	True	True	True	True	True
False	True	False	False	True	False	True
False	False	True	True	False	False	False
False	False	False	False	True	False	False

Limitations of Truth-Table Approach

- **KB |= α ?**

- **What is the computational complexity of the truth-table approach?**

  - **Truth table is exponential in the number of the proposition symbols**

2^n **rows in the table has to be filled**

But typically only for a small subset of rows the KB is true

How to make the process more efficient?

Solution: check only entries for which KB is True.

Logical Inference Approaches

- **KB |= α ?**

- **Three approaches:**
  - **Truth-table approach**
  - **Inference rules**
  - **Resolution algorithm**

Inference Rules Approach

- **Inference rules capture patterns of sound inference**
  - **Once established, don’t need to show the truth table every time**
  - **Can be used to derive new (sound) sentences from the existing ones**

- **A rule is sound if its conclusion is true whenever the premise is true**

- **Alternate notation for inference rule α |- β**

$$\frac{\alpha}{\beta}$$

premise

conclusion

If α is given, then β can be inferred by some inference rule.

“If we know α, then we can conclude β ”
(where α and β are propositional logic sentences)

Inference Rules

- **We are particularly interested in**

$$\frac{KB}{\beta} \quad \text{or} \quad \frac{\alpha_1, \alpha_2, \ldots}{\beta}$$

Inference Rules

- We are particularly interested in

$$\frac{KB}{\beta} \quad \text{or} \quad \frac{\alpha_1, \alpha_2, \ldots}{\beta}$$

- Inference steps

$$\frac{KB}{\beta_1} \rightarrow \frac{KB, \beta_1}{\beta_2} \rightarrow \frac{KB, \beta_1, \beta_2}{\beta_3} \rightarrow \ldots$$

- So we need a mechanism to do this!
- Inference rules that can be applied to sentences in our KB

Proof

The proof of a sentence α from a set of sentences KB is the derivation of α by applying a series of sound inference rules.

Inference Rules

❖ **Modus Ponens rule: (From the implication and the premise of the implication, you can infer the conclusion)**

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

- **If both sentences in the premise are true, then conclusion is true.**
- **Truth table can demonstrate the soundness of modus ponens rule**

α	β	$\alpha \Rightarrow \beta$
False	*False*	*True*
False	*True*	*True*
True	*False*	*False*
True	$\boxed{\textit{True}}$	*True*

Inference Rules

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

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

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

Inference Rules

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

$$\frac{\alpha_1, \alpha_2, \dots, \alpha_n}{\alpha_1 \land \alpha_2 \land \cdots \land \alpha_n}$$

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

Inference Rules

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

$$\frac{\boldsymbol{\alpha_1}}{\boldsymbol{\alpha_1} \lor \boldsymbol{\alpha_2} \lor \cdots \lor \boldsymbol{\alpha_n}}$$

- **If a sentence in the premise is true, then its disjunction with anything else is true.**

Inference Rules Summary

Modus Ponens rule:

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

And-Elimination rule:

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

And-Introduction rule:

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

Or-Introduction rule:

$$\frac{\alpha_1}{\alpha_1 \vee \alpha_2 \vee \cdots \vee \alpha_n}$$

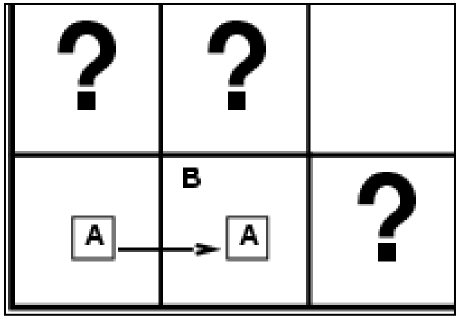
Review Logical Equivalence

- **Two sentences are logically equivalent iff they are true in same models: α ≡ ß iff α ╞ β and β ╞ α**

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{De Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{De Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Example: Wumpus World

- **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: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$**



- **$\alpha : \neg P_{1,2}$**

- **KB |= α ?**
- **Can we prove $\neg P_{1,2}$?**

Example: Wumpus World

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: $B_{2,1} \Leftrightarrow$
  $(P_{1,1} \vee P_{2,2} \vee P_{3,1})$

$\alpha : \neg P_{1,2}$

• Proof:

- 1. Apply biconditional elimination to R3 to obtain:
  - R7: $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
- 2. Apply And-Elimination to R7 to obtain:
  - R8: $(P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$
- 3. Apply logical equivalence for contrapositives to R8:
  - R9: $\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})$
- 4. Apply Modus Ponens with R9 and R2, to obtain:
  - R10: $\neg(P_{1,2} \vee P_{2,1})$
- 5. Apply De Morgan's rule to R10 to obtain:
  - R11: $\neg P_{1,2} \wedge \neg P_{2,1}$
- 6. Apply And-Elimination to R11 to obtain:
  - R12: $\neg P_{1,2}$

Example: Inference Rules

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

- ***Proof:***

  1. $P \wedge Q$
  2. $P \Rightarrow R$
  3. $(Q \wedge R) \Rightarrow S$
  4. P — *Apply And-Elimination rule to 1*
  5. Q — *Apply And-Elimination rule to 1*
  6. R — *Apply Modus Ponens rule to 2, 4*
  7. $Q \wedge R$ — *Apply And-Introduction rule to 5,6*
  8. S — *Apply Modus Ponens rule to 7, 3*

Proved: S

Example: Inference Rules

(1) If it is Saturday today, then we play soccer or basketball.

(2) If the soccer field is occupied, we don't play soccer.

(3) It is Saturday today, and the soccer field is occupied.

Prove: "we play basketball or volleyball".

First, we formalize the problem:

- P: It is Saturday today.
- Q: We play soccer.
- R: We play basketball.
- S: The soccer field is occupied.
- T: We play volleyball.

- KB: $P \Rightarrow (Q \vee R)$, $S \Rightarrow \neg Q$, $P \wedge S$
- Need to prove: $R \vee T$.

Example: Inference Rules

- **(KB: $P \Rightarrow (Q \vee R)$, $S \Rightarrow \neg Q$, $P \wedge S$**
- **Need to prove: $R \vee T$.**

- **Proof:**
- **1. $P \Rightarrow (Q \vee R)$**
- **2. $S \Rightarrow \neg Q$**
- **3. $P \wedge S$**
- **4. P** — **Apply And-Elimination rule to 3**
- **5. S** — **Apply And-Elimination rule to 3**
- **6. $Q \vee R$** — **Apply Modus Ponens rule to 1, 4**
- **7. $\neg Q$** — **Apply Modus Ponens rule to 2, 5**
- **8. R** — **Apply Unit Resolution rule to 6, 7**
- **9. $R \vee T$** — **Apply Or-introduction rule to 8**

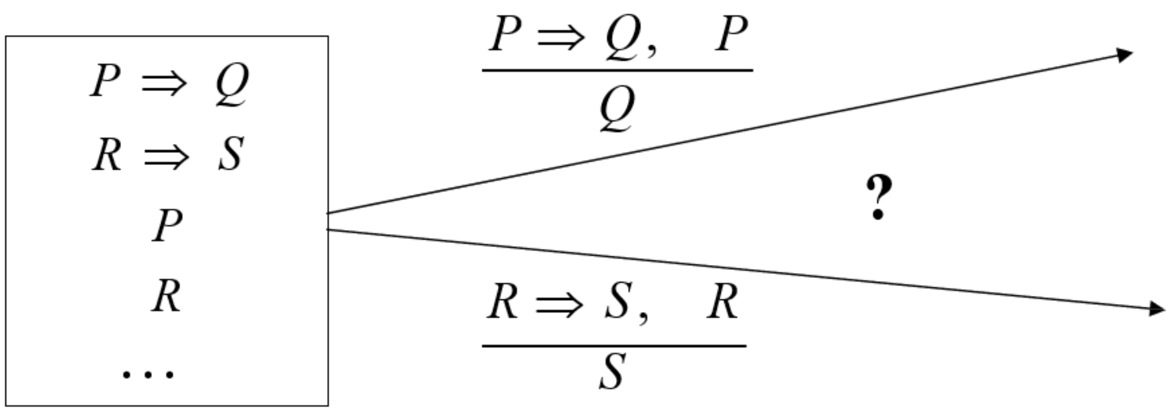
Proved: $R \vee T$

Logic Inferences and Search Problem

- **To show that a sentence α holds for a KB, we may need to apply a number of sound inference rules.**

- **Problem: possible inference rules to be applied next**

  Looks familiar?



This is an instance of a search problem:

Truth table method (from the search perspective):

- **blind enumeration and checking**

Logic Inferences and Search Problem

Inference rule method as a search problem:

- **State: a set of sentences that are known to be true**
- **Initial state: a set of sentences in KB**
- **Actions: set of actions consists of all the inference rules applied to all the sentences that match the premise of the inference rule**
- **Result: add the sentence in the conclusion of the inference rule**
- **Goal: the goal is a state that contains the sentence we are trying to prove**
- **Practically, finding a proof can be more efficient because the proof can ignore irrelevant propositions, no matter how many of them there are**

Logical Inference Approaches

- **KB |= α ?**

- **Three approaches:**
  - **Truth-table approach**
  - **Inference rules**
  - **Resolution algorithm ? Next class**

Summary

- **Logical Inference Approaches**
  - **Truth-table approach**
  - **Inference rules**

What I want you to do

- **Review Chapter 7**