Lecture 16

Knowledge-based Agents II

Lusi Li
Department of Computer Science
ODU

Reading for This Class:
Chapter 7, Russell and Norvig

Review

- **Last Class**
  - **Knowledge-based Agent I**
- **This Class**
  - **Knowledge-based Agent II**
- **Next Class**
  - **Inference**

KB Agents - Summary

- Intelligent agents need knowledge about the world for making good decisions.
- The knowledge of an agent is stored in a knowledge base in the form of sentences in a knowledge representation language.
- A knowledge-based agent needs a knowledge base and an inference mechanism. It operates by storing sentences in its knowledge base, inferring new sentences with the inference mechanism, and using them to deduce which actions to take.
- A representation language is defined by its syntax and semantics, which specify structure of sentences and how they relate to world facts.
- The interpretation of a sentence is the fact to which it refers. If this fact is part of the actual world, then the sentence is true.

Review Entailment

$$\alpha \models \beta \text{ iff } M(\alpha) \subseteq M(\beta)$$

- α entails β
- β logically follows from α
- Under all interpretations in which α is true, β is true as well
- All models of α are models of β
- Whenever α is true, β is true as well
- When α is false, β can be either true or false.

$$KB \vdash_i \alpha$$

- Sentence α is derived from KB by the inference procedure I
- An inference algorithm that derives only entailed sentences is **sound**
- An inference algorithm that derives any sentence that is entailed is **complete**

Propositional Logic: Syntax

- Propositional logic is the simplest logic
- Syntax defines allowable sentences.
- Atomic sentence = a single proposition symbol
  - Each symbol stands for a proposition that can be true or false

- Complex sentences are combined by connectives (operators) :
  - If P is a sentence, $\neg$P is a sentence (negation)
    - Literal: atomic sentence or negated atomic sentence
  - If P and Q are sentences, P $\wedge$ Q is a sentence (conjunction)
    - P and Q are conjuncts
  - If P and Q are sentences, P $\vee$ Q is a sentence (disjunction)
    - P and Q are disjuncts
  - If P and Q are sentences, P $\Rightarrow$ Q is a sentence (implication)
    - P is premise and Q is conclusion. Implications are rules or if-then statements
    - No requirement of any causation or relevance between P and Q
  - If P and Q are sentences, P $\Leftrightarrow$ Q is a sentence (biconditional)
    - E.g., the sentence $W_{1,3} \Leftrightarrow \neg W_{2,2}$ is a biconditional

Grammar Summary

A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedence, from highest to lowest.

$$
\begin{aligned}
Sentence &\rightarrow AtomicSentence \mid ComplexSentence \\
AtomicSentence &\rightarrow True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence &\rightarrow (\ Sentence\ ) \mid [\ Sentence\ ] \\
&\mid \neg\ Sentence \\
&\mid Sentence \land Sentence \\
&\mid Sentence \lor Sentence \\
&\mid Sentence \Rightarrow Sentence \\
&\mid Sentence \Leftrightarrow Sentence \\
\\
\textsc{Operator Precedence} &: \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{aligned}
$$

- Counter examples:
  - $(A \land \Rightarrow R)$
  - $(A \lor (\neg C)$
  - $A \Rightarrow B \Rightarrow C$

Correct examples:

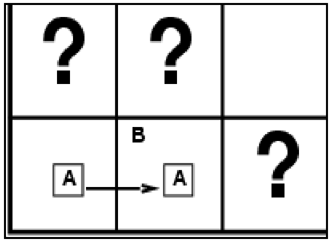
$(A \Rightarrow B) \Rightarrow C$

$A \Rightarrow (B \Rightarrow C)$

Propositional Logic: Semantics

- **The semantics define the rules for determining the truth of a sentence with respect to a particular model.**
  - **Specify how to compute the truth value of any sentence, given a model**

- **Atomic sentences are easy.**
  - ***True* is true in every model and *False* is false in every model.**
  - **The truth value of every other proposition symbols must be specified directly in the model. E.g., m1 = {$P_{1,2}$= false, $P_{2,2}$ = true, $P_{3,1}$ = false}**



- **Rules for evaluating truth in any model *m*:**
  1) **$\neg P$ is true iff P is false**
  2) **$P \wedge Q$ is true iff P is true and Q is true**
  3) **$P \vee Q$ is true iff P is true or Q is true**
  4) **$P \Rightarrow Q$ is true unless P is true and Q is false**
  5) **$P \Leftrightarrow Q$ is true iff $P \Rightarrow Q$ is true and $Q \Rightarrow P$ is true**

- **Simple recursive process evaluates an arbitrary sentence, e.g.,**
  - **$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$ = *true* $\wedge$ (*true* $\vee$ *false*) = *true* $\wedge$ *true* = *true*, given *m1***

Model of Propositional Logic

- **Assignment of a truth value – true or false – to every atomic sentence in KB**
- **Examples:**
  - **Let A, B, C, and D be the propositional symbols in KB**
  - **m = {A=true, B=false, C=false, D=true} is a model**
  - **m′ = {A=true, B=false, C=false} is not a model**

- **With n propositional symbols, one can define 2^n possible models**

- **A model of a KB is a “possible world” (assignment of truth values to propositional symbols) in which each sentence in the KB is True.**

Truth Table for Five Connectives

- **The five rules can also be expressed with truth tables that specify the truth value of a complex sentence for each possible assignment of truth values to its components.**

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Truth Table for Five Connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

1. **Propositional logic does not require any relation of *causation* or *relevance* between P and Q.**
   - **ODD(5) $\Rightarrow$ CAPITAL(Japan,Tokyo) True**

2. **Implication is always true when the premise is false**
   - **EVEN(5) $\Rightarrow$ CAPITAL(Japan,Norfolk) True**

Why? P => Q means "if P is true then I am claiming that Q is true, otherwise I am making no claim"

Only way for this to be false is if P is true and Q is false

Truth Table for Five Connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

The biconditional, P ⇔ Q, is true whenever both P ⇒ Q and Q ⇒ P are true.

This is often written as "P if and only if Q."

Truth Table for Five Connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Example of a truth table used for a complex sentence:

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

A Simple Knowledge Base
Wumpus World

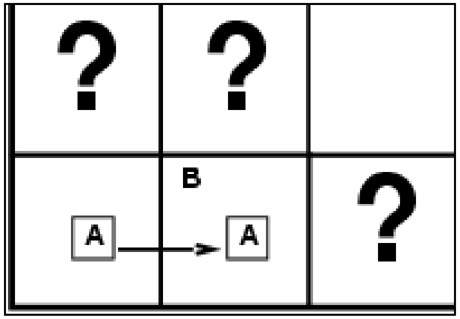
For now, we need the following symbols for each [i,j] location

- **Let $P_{i,j}$ be true if there is a pit in [i, j].**
- **Let $B_{i,j}$ be true if there is a breeze in [i, j].**
- **For [1,1]:**

  There is no pit in [1,1]:
  - **R1:** $\neg P_{1,1}$

  There is no breeze in [1,1]:
  - **R2:** $\neg B_{1,1}$

  "Pits cause breezes in adjacent squares"
  - **R3:** $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
- **For [2,1]:**
  - **R4:** $\neg P_{2,1}$
  - **R5:** $B_{2,1}$
  - **R6:** $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$



The KB consists of the above 6 sentences. It can also be considered as a single sentence – the conjunction $R1 \wedge R2 \wedge R3 \wedge R4 \wedge R5 \wedge R6$

Inference

- KB $\vdash_i \alpha$
  - sentence α can be derived from KB by the inference procedure i
- The aim of logical inference is to decide whether KB $\models$ α for some α.

ASK

- Is $\neg P_{1,2}$ entailed by KB?
- Is $\neg P_{2,2}$ entailed by KB?
- Is $\neg P_{3,1}$ entailed by KB?

- Our first algorithm for inference will be a direct implementation of the definition of entailment:
  - Model checking:

  Enumerate the models, and check that α is true in every model in which KB is true

  Sound and complete, but 2^n

Truth Tables for Inference

- KB = R1 ∧ R2 ∧ R3 ∧ R4 ∧ R5 ∧ R6
- α_1 = "square [1,2] is safe", i.e., $\neg P_{1,2}$ is true

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*false*	*false*	*true*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*false*

Inference by Enumeration

- A truth-table enumeration algorithm for deciding propositional entailment.
- Depth-first enumeration of all models is sound and complete
- For n symbols, time complexity is $O(2^n)$, space complexity is $O(n)$

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

  $symbols \leftarrow$ a list of the proposition symbols in KB and α
  return TT-CHECK-ALL($KB, \alpha, symbols, \{\ \}$)

---

function TT-CHECK-ALL($KB, \alpha, symbols, model$) **returns** *true* or *false*
  if EMPTY?($symbols$) **then**
    if PL-TRUE?($KB, model$) **then return** PL-TRUE?($\alpha, model$)
    else return *true // when KB is false, always return true*
  else do
    $P \leftarrow$ FIRST($symbols$)
    $rest \leftarrow$ REST($symbols$)
    return (TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = true\}$)
      and
      TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = false\}$))

Inference by Enumeration

- **Sound? Yes**
  - **Entailment is used directly!**

- **Complete? Yes**
  - **Works for all KB and a**
  - **Always stops**

Logical Equivalence

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

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

Logical equivalence

- **Proof Example using truth table**

$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{implication elimination}$$

α	β	$\alpha \Rightarrow \beta$
$false$	$false$	$true$
$false$	$true$	$true$
$true$	$false$	$false$
$true$	$true$	$true$

Validity and Satisfiability

- A sentence is valid if it is true in all models,
  e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$
- Validity is connected to inference via the Deduction Theorem:
  $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid
- 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., $A \wedge \neg A$
- 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: Given KB and α, establishing entailment is equivalent to proving that no model exists that satisfies KB and ¬ α.

Summary

- **Logical agents apply inference to a knowledge base to derive new information and make decisions**

- **Basic concepts of logic:**
  - **syntax: formal structure of sentences**
  - **semantics: truth of sentences wrt models**
  - **entailment: necessary truth of one sentence given another**
  - **inference: deriving sentences from other sentences**
  - **soundness: derivations produce only entailed sentences**
  - **completeness: derivations can produce all entailed sentences.**

What I want you to do

- **Review Chapter 7**