Chapter 7: Logical Agents

7.1 and 7.2 Valuable Background

Knowledge-based agents use a process of reasoning over an internal representation of knowledge to decide what actions to take.

1. Contain a knowledge base made of sentences.
2. Sentences are represented using a knowledge representation language, making an assertion about the world.
3. An Axiom is a sentence not derived from other sentences.
4. We add new sentence and query our knowledge using TELL and ASK.

May involve INFERENCE: Creating new sentences form existing ones.
Inference follows that an answer comes form what has been told.
The KB Agent :

```
function KB-AGENT(percept) returns an action
   persistent: KB, a knowledge base
               t, a counter, initially 0, indicating time

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   action ← ASK(KB, MAKE-ACTION-QUERY(t))
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

How to modify the behavior ?
- Done at the knowledge level
- Tell the agents its goal
- Tell the agent what it knows

Example:
- Tell taxi about bay bridge
- Tell taxi to go from SF to Oakland
- Taxi knows to cross the bridge

At the implementation level, can be done in lots of ways.
How to Build it?
- TELL it what it needs to know
- Declarative Approach :
  TELL sentence one by one until the agent knows what to do in its environment.
- Procedural Approach
  Encode the behaviors directly to the computer code
- Best agents combine both

7.2. WUMPUS WORLD

7.3 Logic
Knowledge bases consist of sentences
Sentences follow the syntax of their well formed – Example : x + y + z = 14
Meaning of sentence – Semantics
Format of the sentence – Syntax
Semantics defines the truth of each sentence wrt each possible world model.

POSSIBLE WORLD:
Possible world are like agent environment.
Model are more abstract mathematical constructs: all possible values for a set of variables.
- Assignment fix truths

When some sentence is true in model m, then,
- m Statics alpha
- Or you could say m is a model of alpha
- M(alpha) means the set of all model of alpha

For example, consider the statement "It is raining." In one possible world, it might indeed be raining, making the statement true. In another possible world, it might not be raining, making the statement false. Possible worlds provide a way to explore different scenarios and evaluate the truth of statements within those scenarios.

ENTAILMENT
Entailment between sentences: One sentence logically follows another.

- Can be written: α |= β
- So, α entails the sentence β.
- Formally, α |= β iff for all model where α is true, β is true too.
- Another way to put it:

  $\alpha \models \beta$ if and only if $M(\alpha) \subseteq M(\beta)$.

- Example: x = 0 entails xy = 0

Entailment:

- Definition: If α entails β, it means α being true guarantees β is also true.
- Notation: $\alpha \models \beta$
- Example: "All humans are mortal" entails "Socrates is mortal."

SATISFICATION

Satisfaction in Logic:

- **Definition:** Satisfaction means a statement holds true in a given scenario or model.
- **Example:** In the equation $x + y = 5$, with $x = 2$ and $y = 3$, the statement is satisfied because $2 + 3 = 5$.
- **Importance:** It shows whether a statement is true under specific conditions, aiding logical reasoning.

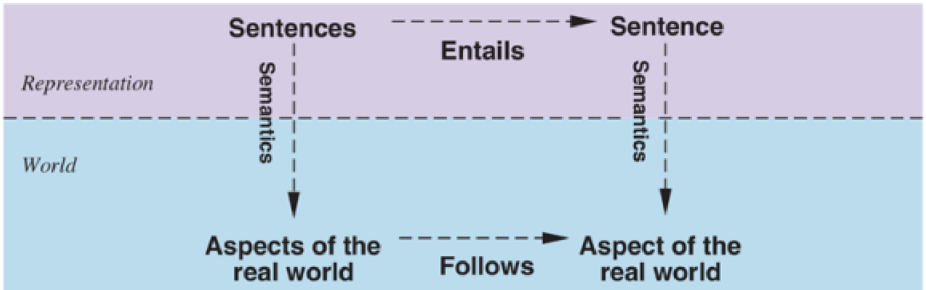
Model Checking:

- **Definition:** Model checking is a method in logical reasoning where all possible scenarios, or models, are examined to verify if a given statement holds true in each scenario.
- **Example:** In assessing whether $x + y = 5$ is true, model checking involves testing various combinations of values for x and y to ensure the statement holds in every scenario.
- **Process:** Model checking systematically evaluates all possible interpretations of variables within a given logical statement to determine its truthfulness across different scenarios.
- **Application:** It is a fundamental technique used in logical inference and verification tasks, ensuring the validity and consistency of logical statements across a range of possible situations

SOUND or TRUTH PRESERVING inference algorithm derives only entailed sentences.
- Unsound algos just makes stuff up

COMPLETE inference algos derive all entailed sentences.

Figure 7.6



Sentences are physical configurations of the agent, and reasoning is a process of constructing new physical configurations from old ones. Logical reasoning should ensure that the new configurations represent aspects of the world that actually follow from the aspects that the old configurations represent.

GROUNDING
Logical reasoning and the agents real environment.
How can we know the KB is true in reality?

Simple answer – Sensors

In Wumpus world agent has a smell sensor
Agent problem creates a sentence when smell sensors smells something
If the sentence is in the KB, its true in the real world.

• **Be sure to know the terms – possible words model, satisfaction, entailment, etc.**

7.4, 7.5 Propositional Logic
SYNTAX:
Syntax defines valid sentences
Atomic sentences have a single propositional symbol
Symbol represent a proportional that can be true or false.

Example – P, Q, R, W 13, and North

Two Propositional symbols have fixed meaning
True : Always true propositional
False : Always false propositional

COMPLEX SENTENCE built from simpler ones (uses parentheses and logical connectives)

Five connectives:

- $\neg$ (not). $\neg A$ is the **negation** of A.
- $\wedge$ (and). $A \wedge B$ is the **conjunction** of A and B.
- $\vee$ (or). $A \vee B$ is the **disjunction** of A and B.
- $\Rightarrow$ (implies). The sentence $(A \wedge B) \Rightarrow \neg C$ is called an **implication**.
  - The **premise** is $(A \wedge B)$ and the **conclusion** is $\neg C$
  - **Implications** are called rules or **if-then** statements.
- $\Leftrightarrow$ (if and only if). $A \Leftrightarrow B$ is a biconditional.

Here's the grammar:

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{True} \mid \textit{False} \mid P \mid Q \mid R \mid \ldots \\
\textit{ComplexSentence} & \rightarrow & (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
 & | & \neg\ \textit{Sentence} \\
 & | & \textit{Sentence} \wedge \textit{Sentence} \\
 & | & \textit{Sentence} \vee \textit{Sentence} \\
 & | & \textit{Sentence} \Rightarrow \textit{Sentence} \\
 & | & \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
 & & \\
\text{OPERATOR PRECEDENCE} & : & \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{array}
$$

Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

SEMANTICS

Semantics defines rules for determining truth of a sentence wrt some model.

In propositional logic, the model just fixes the truth value (true or false) for every propositional symbol.

One possible model could be:

m_1 = {A = false, B = false, C = true}

$2^3 = 8$ possible models here.

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 tables for the five logical connectives. To use the table to compute, for example, the value of $P \vee Q$ when P is true and Q is false, first look on the left for the row where P is *true* and Q is *false* (the third row). Then look in that row under the $P \vee Q$ column to see the result: *true*.

Truth table

- Example: from the previous model, the sentence:

  $\neg A \wedge (B \vee C)$ gives
  true $\wedge$ (false $\vee$ true) which gives
  true $\wedge$ true which gives
  true.

- Note: $\vee$ is inclusive or.
- $P \Rightarrow Q$ can be read as "P implies Q", or "if P then Q".
  - "5 is odd implies Tokyo is Japan's capital" is true in propositional logic.
  - Any implication is true whenever it's antecedent is false. Example:
    - "5 is even implies Same is smart" is true.
  - Makes sense if you say: "If P is true, then I claim Q is true", else no claim. Only false if P is true but Q is false.

- $P \Leftrightarrow Q$ is true whenever $P \Rightarrow Q$ and $Q \Rightarrow P$.
  - Often said as "if and only if" or iff
  - Wumpus World Example:
    A square is breezy if an adjacent square has a pit, and a square is breezy only if an adjacent square has a pit:

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

    And $B_{1,1}$ is a breezy square

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

Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

- A sentence is valid if it is true in <u>all</u> models.
  - Example: $P \vee \neg P$ is valid
  - AKA: **tautologies**. They are *necessarily true.*
  - A sentence true in all models is logically equivalent to True.
  - Leads to the **deduction theorem**:

    "For any setences A and B, A |= B iff the seentence (A $\Rightarrow$ B) is valid.
  - Allows verification of A |= B by checking that (A $\Rightarrow$ B) in every model.
    OR
    proving (A $\Rightarrow$ B) is equivelant to True.
  - Thus, every valid implication sentence describes a legitimate inference.

PROOF BY RESOLUTION

Inference rules so far are sound but are they complete?

Complete search algorithms will find any reachable goal.
If inference rules are not adequate the goal can not be reached and thus no proof found.
Ex. Remove a “Bridging Ruel”.
Resolution:

- Inference rule in propositional logic.
- Used to derive new conclusions from given premises.
- Complete: Can derive any logical consequence if it exists.
- Process: Iteratively apply resolution rule to resolve complementary literals in clauses.
- Widely used in automated theorem proving and AI.

- **Skip 7.5.3 and 7.5.4**

7.6 and 7.7 Skip