Propositional Logic

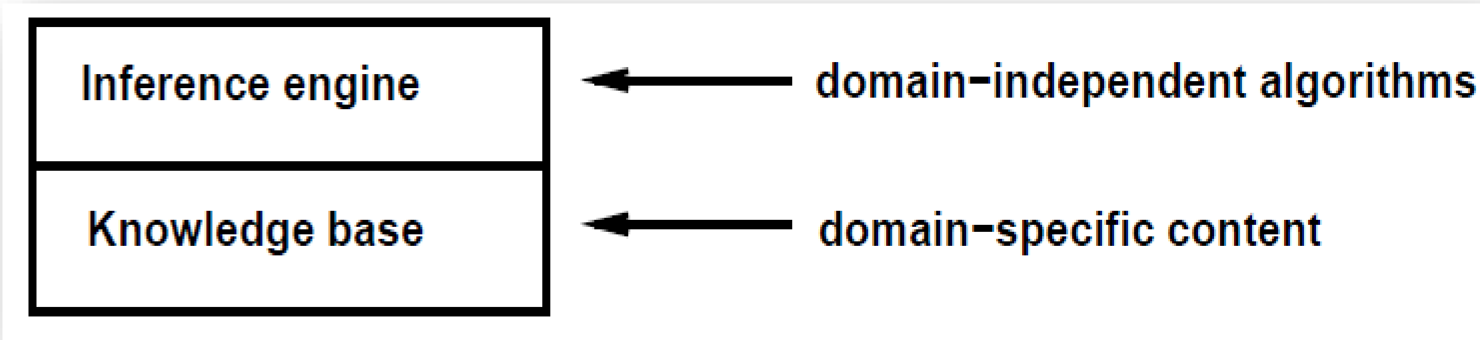
Why Do We Need Logic?

- Problem-solving agents were very inflexible: hard code every possible state.
- Search is almost always exponential in the number of states.
- Problem solving agents cannot infer unobserved information.
- We want an agent that can reason similarly to humans.

Logic in AI

- Logic can be defined as the proof or validation behind any reason provided.
- It is simply the ‘dialectics behind reasoning’.
- It was important to include logic in Artificial Intelligence because agent (system) to think and act humanly, and for doing so, it should be capable of taking any decision based on the various available options.
- There are reasons behind selecting or rejecting an option.

Knowledge Base



- Knowledge base: Set of sentences in a formal language
- Declarative approach for building an agent
  - TELL it what it needs to know
  - Then it can ASK itself what to do - answers should follow from the KB
- Agents can be viewed at the knowledge level
  - What they know, regardless of implementation
- Or at the implementation level
  - Data structures in KB and algorithms that manipulate them.

Logic In General

- Logics are formal languages for representing information, such that conclusions can be drawn
- Syntax defines the sentences in the language
- Semantics define the “meaning" of sentences, i.e., define truth of a sentence in a world
- E.g., the language of arithmetic
- $x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence
- $x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y
- $x + 2 \geq y$ is true in a world where $x=7$; $y=1$
- $x + 2 \geq y$ is false in a world where $x=0$; $y=6$

Entailment

- Entailment means that one thing **follows** from another $$KB \models \alpha$$
- Knowledge base KB entails sentence B iff B is true in all worlds where KB is true
  - E.g., the KB containing “the Giants won" and “the Reds won” entails “Either the Giants won or the Reds won“
  - E.g., x + y =4 entails 4=x + y
- Entailment is a relationship between sentences (i.e., syntax) that is based on semantics.

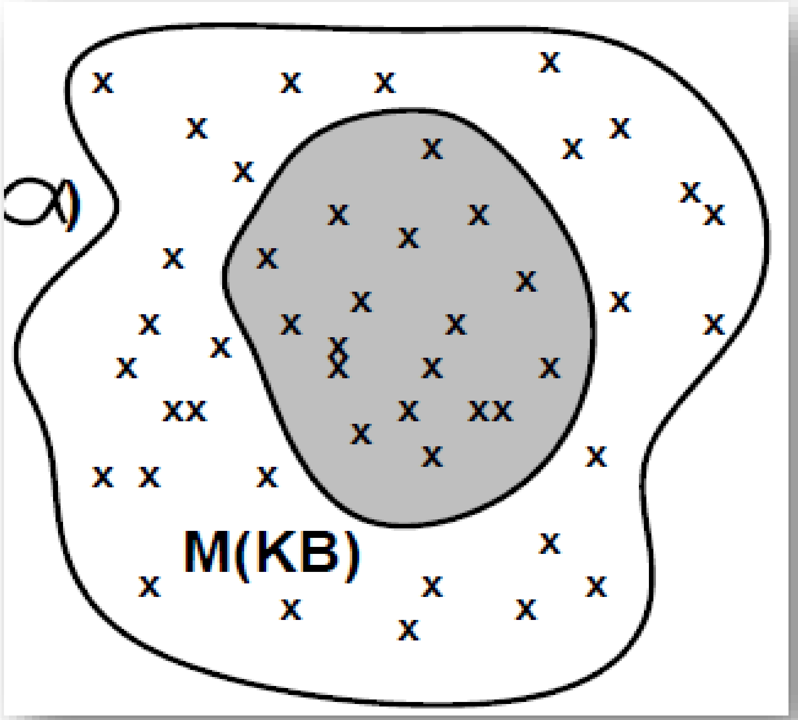
Models

- Logicians typically think in terms of **models**, which are formally structured worlds with respect to which truth can be evaluated
- We say *m* is a model of a sentence B if B is true in *m*
  - M(B) is the set of all models of B
- Then KB $\models$ B if and only if M(KB) $\subseteq$ M(B)
  - E.g. KB = Giants won and Reds won
  - B = Giants won

M(B) could be also true for worlds that are different than the worlds of KB

Models

M(B)



Types of logics in Artificial Intelligence

- Two **types of logics**: Deductive logic and Inductive logic
- In deductive logic, the complete evidence is provided about the truth of the conclusion made.
- Here, the agent uses specific and accurate premises that lead to a specific conclusion.
- Example: An expert system designed to suggest medicines to the patient because the person has so and so symptoms.
- In Inductive logic, the reasoning is done through a ‘bottom-up’ approach.
- The agent here takes specific information and then generalizes it for the sake of complete understanding.
- Example: In the natural language processing, an agent sums up the words according to their category, i.e. verb, noun article, etc., and then infers the meaning of that sentence.

Propositional Logic

- A proposition is a declarative statement which is either true or false.
- It is a technique of knowledge representation in logical and mathematical form.

(a) The Sun rises from West

(b) 5 is a prime number.

- In PL, symbolic variables are used to represent the logic.
- PL consists of an object, relations or function, and **logical connectives**, called logical operators.
- A proposition formula which is always true is called **tautology**, and it is also called a valid sentence.
- A proposition formula which is always false is called **Contradiction**.

Syntax of propositional logic

- There are two types of Propositions:

  Atomic Propositions

  Compound propositions

- **Atomic Proposition:** Atomic propositions are the sentences which must be either true or false.
- **Compound proposition:** Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives
  (It is raining today, and street is wet)

Propositional Logic

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)

If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)

If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Order of Precedence

If and Only If

If S1 then S2

Logical Connectives

- **Negation:** A sentence such as $\neg$ P (negation of P) is a literal can be either Positive literal or negative literal.
- **Conjunction:** A sentence which has $\wedge$ connective such as, **P $\wedge$ Q** is called a conjunction.
- **Disjunction:** A sentence which has $\vee$ connective, such as **P $\vee$ Q**. is called disjunction, where P and Q are the propositions.
- **Implication:** A sentence such as P $\rightarrow$ Q, is called an implication. Implications are also known as if-then rules. It can be represented as **If** *it is raining* (P), **then** *the street is wet* (Q), represented as P $\rightarrow$ Q
- **Biconditional:** A sentence such as **P$\Leftrightarrow$ Q** is a **Biconditional sentence,**

  If I am breathing (P)**, then** I am alive (Q)

For Implication:

P	Q	P→ Q
True	True	**True**
True	False	**False**
False	True	**True**
False	False	**True**

For Biconditional:

P	Q	P⇔ Q
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**True**

Deduction using Propositional Logic

If I am the President then I am well-known. I am the President. So I am well-known

<u>Coding: Variables</u>

a: I am the President

b: I am well-known

<u>Coding the sentences</u>:

F1: $a \rightarrow b$

F2: a

G: b

The final formula for deduction: $(F1 \wedge F2) \rightarrow G$, that is:

$((a \rightarrow b) \wedge a) \rightarrow b$

a	b	$a \rightarrow b$	$(a \rightarrow b) \wedge a$	$((a \rightarrow b) \wedge a) \rightarrow b$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

<u>Limitations of Propositional logic:</u>

Cannot represent relations like ALL, some, or none with propositional logic.

Example: **All the girls are intelligent, Some apples are sweet.**

Logical Equivalence

Two sentences are logically equivalent iff true in same models:

$\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

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

If KB is true, alpha is always true. Hence, I can say that alpha follows from KB

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

i.e., prove α by *reductio ad absurdum*

Suppose that KB=true. Then, this will become unsatisfiable only when alpha is true. Hence, I can say that alpha follows from KB

Two Famous Inference Rules

- Modus Ponens

$$\frac{P \Rightarrow Q \quad , \quad P}{Q}$$

Given that P implies Q, and I know that P is true, then I can infer Q

- Anc $P \wedge Q$

Given that P AND Q is true, I can infer that P is true, and I can also infer that Q is true.

Resolution

Conjunctive Normal Form (CNF—universal)

conjunction of $\underbrace{\textbf{disjunctions of literals}}_{\textbf{clauses}}$

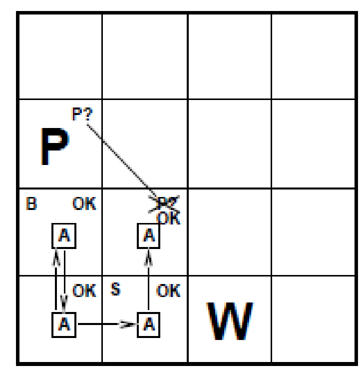
E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Resolution inference rule (for CNF): complete for propositional logic

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

where ℓ_i and m_j are complementary literals. E.g.,

$$\frac{P_{1,3} \vee P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$



Resolution is sound and complete for propositional logic

Conversion to CNF

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

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's rules and double-negation:

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

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

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

Resolution Algorithm

- Resolution rule:

$$\frac{\alpha \vee \beta \qquad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

- Resolution refutation:
  - Convert all sentences to CNF
  - Negate the desired conclusion (converted to CNF)
  - Apply resolution rule until either
    - Derive false (a contradiction)
    - Can't apply any more
- Resolution refutation is sound and complete
  - If we derive a contradiction, then the conclusion follows from the axioms
  - If we can't apply any more, then the conclusion cannot be proved from the axioms.

Example

Propositional Resolution Example

Prove R

1	P v Q
2	P → R
3	Q → R

false v R

¬ R v false

false v false

Step	Formula	Derivation
1	P v Q	Given
2	¬ P v R	Given
3	¬ Q v R	Given
4	¬ R	Negated conclusion
5	Q v R	1,2
6	¬ P	2,4
7	¬ Q	3,4
8	R	5,7
9	•	4,8