Al-Mustaqbal University
College of Science

KNOWLEDGE REPRESENTATION

1st CLASS / AI BRANCH

Department of Artificial Intelligence
Lecture 1
Assistant Lecturer: Hadi Salah

Knowledge Representation

Knowledge representation (KR) is a part of the Artificial Intelligence which responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real-world problems such as communicating with humans in natural language.

It is also a way which describes how we can represent knowledge in Artificial Intelligence. Knowledge representation is not just storing data into database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human. There are many methods can be used for knowledge representation and some of them can be described as follow:-

1. Propositional Logic
2. Predicate Logic
3. Semantic Network
4. Conceptual Graph
5. Frame Representation
6. Script Representation

1- Prepositional Calculus (Logic)

Propositional Calculus Symbols

The symbols of propositional calculus are {P, Q, R, S, …}
Truth symbols: **{True, False}**
Connective tools: **{**∧, ∨, ┐, →, ≡**}**

Propositional symbols denote ***propositions*** or **statements** about the world that may be either true or false, Propositions are denoted by uppercase letters near the end of the English alphabet letters.

For example:
P: It is sunny today.
Q: The sun shines on the window.
R: The blinds are down.
(P→Q): If it is sunny today, then the sun shines on the window
(Q→R): If the sun shines on the window, the blinds are brought down.
(┐ R): The blinds are not yet down.

Propositional Calculus Sentence

- Every propositional symbol and truth symbol is a sentence.

For example: **True, P, Q,** and **R** are sentences.

- The ***negation*** of a sentence is a sentence.

For example: ┐**P** and ┐**False** are sentences.

- The ***conjunction***, **AND**, of two sentences is a sentence.

For example: **P** ∧ ┐**P** is a sentence.

- The ***disjunction***, **OR** of two sentences is a sentence.

For example: **P** ∨ ┐**P** is a sentence.

- The ***implication*** of one sentence from another is a sentence.

For example: **P** → **Q** is a sentence.

- The ***equivalence*** of two sentences is a sentence.

For example: **P** ∨ **Q** ≡ **R** is a sentence.

- Legal sentences are also called ***well-formed formulas*** or ***WFF****s.*

In expressions of the form **P ∧ Q, P** and **Q** are called the ***conjuncts***. In **P ∨ Q**, **P** and **Q** are referred to as ***disjuncts***. In an implication, **P → Q, P** is the ***premise*** and ***Q***, the ***conclusion*** or ***consequent.***

In propositional calculus sentences, the symbols **( )** and **[ ]** are used to group symbols into sub-expressions and so to control their order of evaluation and meaning.

For Example:

(P ∨ Q) ≡ R is quite different from **P ∨ (Q ≡ R)** as can be demonstrated using truth tables. An expression is a sentence, or well- formed formula, of the propositional calculus if and only if it can be formed of legal symbols through some sequence of these rules.

For Example: **((P ∧ Q) → R)** ≡ ┐**P** ∨ ┐**Q** ∨ **R** is a well-formed sentence in the propositional calculus because:

P, **Q**, and **R** are propositions and thus sentences.

P ∧ **Q**, the conjunction of two sentences, is a sentence.

(P ∧ Q) → R, the implication of a sentence for another, is a sentence.

┐**P** and ┐**Q**, the negations of sentences, are sentences.

┐**P** ∨ ┐**Q** the disjunction of two sentences is a sentence.

┐**P** ∨ ┐**Q** ∨ **R**, the disjunction of two sentences, is a sentence.

((P ∧ Q) → R) ≡ ┐**P** ∨ ┐**Q** ∨ **R**, the equivalence of two sentences, is a sentence.

This is our original sentence, which has been constructed through a series of applications legal rules and is therefore "***well formed***".

Propositional Calculus Semantics

An ***interpretation*** of a set of propositions is the assignment of truth value, either **T** or **F**, to each propositional symbol. The symbol **True** is always assigned **T**, and the symbol **False** is assigned **F**.

The interpretation or truth value for sentences is determined by:

- The truth assignment of ***negation***, ┐**P**, where **P** is any propositional symbol, is **F** if the assignment to **P** is **T**, and **T** if the assignment to **P** is **F**.
- The truth assignment of ***conjunction***, ∧, is **T** only when both **conjuncts** have truth value **T**; otherwise, it is **F**.
- The truth assignment of ***disjunction***, ∨, is **F** only when both **disjuncts** have truth value **F**; otherwise, it is **T**.
- The truth assignment *of* ***implication***,→, is **F** only when the premise or symbol before the implication is **T** and the truth value of the consequent or symbol after the implication is **F**; otherwise, it is **T**.
- The truth assignment of ***equivalence***, ≡, is **T** only when both expressions have the same truth assignment for all possible interpretations; otherwise, it is **F**.

- The truth assignments of compound propositions are often described by ***truth tables***. A **truth table** contains all possible truth value assignments to the atomic propositions of expression and gives the truth value of the expression for each assignment. Thus, a truth table enumerates all possible worlds of interpretation that may be given to an expression.
- **For Example**, the truth table for **P ∧ Q, Fig. (A)** , lists truth values for each possible truth assignment of the operands.

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

- **P ∧ Q** is true only when both **P** and **Q** are both **T. Or (∨), not(┐),** **implies (→)**, and **equivalence (≡)** is defined in a similar fashion. The construction of these truth tables is left as an exercise. Two expressions in the propositional calculus are equivalent if they have the same value under all truth value assignments.
- This equivalence may be demonstrated using truth tables. **For example**, a proof of the equivalence of **P → Q** and ┐ **P ∨ Q** is given by the truth table **Fig. (B).**

P	Q	¬P	P→Q	¬P∨Q	(¬P∨Q) ≡(P→Q)
T	T	F	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

Fig. (B) : Truth table demonstrating the equivalence of (┐ P∨Q) ≡ (P→Q)

By demonstrating that two different sentences in the propositional calculus have identical truth tables, we can prove the following equivalences. For propositional expressions **P**, **Q**, and **R**:

- The double negation law : $\neg(\neg P) \equiv P$
- The implication in terms of $\vee$ law: $P \rightarrow Q \equiv \neg P \vee Q$
- The contrapositive law: $(P \rightarrow Q) \equiv (\neg Q \rightarrow \neg P)$
- De Morgan's law: $\neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$ and $\neg(P \wedge Q) \equiv (\neg P \vee \neg Q)$
- The commutative laws: $(P \wedge Q) \equiv (Q \wedge P)$ and $(P \vee Q) \equiv (Q \vee P)$
- The associative law: $((P \wedge Q) \wedge R) \equiv (P \wedge (Q \wedge R))$
- The associative law: $((P \vee Q) \vee R) \equiv (P \vee (Q \vee R))$
- The distributive law: $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
- The distributive law: $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$

Truth table then will list all possible truth value assignments to the propositions of expression, the standard truth tables are shown in the figure below:

And

p	q	$p \cdot q$
T	T	T
T	F	F
F	T	F
F	F	F

Or

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

If . . . then

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Not

p	$\sim p$
T	F
F	T