Knowledge representation

Introduction

Knowledge is the progression that starts with data which s limited utility. Data when processed become information, information when interpreted or evaluated becomes knowledge an understanding of the principles of embodied with the knowledge is wisdom this shown in the following figure:

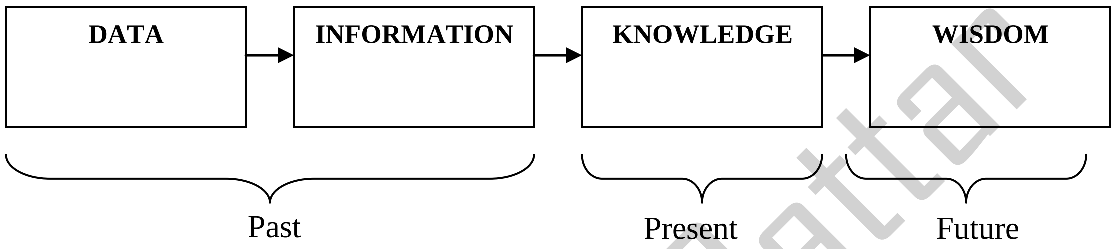


Figure Relation ship between data, information knowledge and wisdom.

Let us tabulate the differences between data-base and knowledge-base

data-base	knowledge-base
1. It is defined as a collection of data representing facts	**1.** It has information at higher level of abstraction
2. it is larger than a KB	**2.** it is smaller than a DB
3. changes are fast	**3.** changes are gradual
4. all information needs to be stared explicitly	**4.** it has the power of inference
5. It is maintained for operational for purposes	**5.** It is used for data analysis and planning
6. knowledge is represented by relational, network or hierarchical model	**6.** Knowledge is represented by logic or rules.

What are knowledge representation schemes?

In AI, there are four Basic categories of representation schemes: ***logical, procedural***, ***network*** and ***structured*** representation schemes.

1. **Logical representation**: uses expression in formal logic to represent its knowledge base, predicate Calculus is the most widely used representation scheme.
2. **Procedural representation**: represent knowledge as set of instructions for solving problem. These are usually if then rules we used in rule based system.
3. **Network representation**: capture knowledge as a graph in which the nodes represent the objects or concepts in the problem domain and the arcs represent the relations or association between them.
4. **Structured representation:** extend network representation schemes by allowing each node to have complex data structure named slots with attached values.

A good knowledge representation system should possess the following properties:-

1. **Representation adequacy:** it is defined as an ability to represent the required knowledge.
2. **Inferential adequacy:** it is defined as an ability to manipulate the knowledge representation to produce new knowledge corresponding to that inferred from the original.
3. **Inferential Efficiency**: it is defined as an ability to direct the inferential mechanisms into the most productive direction by string appropriate guides.
4. **Acquisitioned Efficiency:** it is defined as the ability to acquire new knowledge using automatic methods where possible rather than reliance on human intervention.

Some terminologies related to knowledge

Let us define certain terms that will be used again and again here. They are as follows:

1. Knowledge and data: A doctor has both knowledge and data. Here, data is the patient's record whereas knowledge is what he has learned in the medical college.

2. Belief: It is defined as essentially any meaningful and coherent expression that can be represented. It may be true or false.

----

3. Hypothesis: It is defined as a justified belief that is not known to be true. It is packed up with some supporting evidences but it may still be false.
4. Knowledge: It is true justified belief.
5. Meta knowledge: it is the knowledge about the knowledge
6. Epistemology: it is the study of the nature of knowledge.

Knowledge types

Major classification of knowledge is as follows:
Tacit (or procedural) knowledge and Explicit or declarative knowledge
Let us tabulate the differences between them now.

Tacit knowledge or Procedural	Explicit knowledge Declarative
1. It is an ***embodied*** knowledge that exist within the human being	**1.** It is an ***embodied*** knowledge that exist outside the human being
2. It is difficult to articulate formally	**2.** It can be easily articulate formally
3. It is difficult to share or communicate this knowledge.	**3**. This knowledge can be shared, copied, processed and stored.
4. This knowledge is drawn from experience, action, subjective insight.	**4.** This knowledge is drawn from artifact of some type as principle, procedure, process and concepts.
5. It is a procedural knowledge – about know to do something.	**5.** It is a declarative knowledge – that someth9ng is true or false.
6. It focuses on tasks that must be performed to reach on particular objective and goal *Examples:* procedures, rules, strategies, agendas, models	**6.** It refers to the representations of object and events, knowledge about facts and relationships *Examples:* concepts, objects, facts, propositions, assertions, semantic net, logic and descriptive model
7.It is hard to debug	**7.** It easy to validate
8. Representation in form of set of rules.	**8.** Representation in form of production system

The Propositional Calculus

The propositional calculus and, in the next subsection, the predicate calculus are first of all languages. Using their words, phrases, and sentences, we can represent and reason about properties and relationships in the world. The first step in describing a language is to introduce the pieces that make it up: its set of symbols.

D E F I N I T I O N

PROPOSITIONAL CALCULUS SYMBOLS

The *symbols* of propositional calculus are the propositional symbols:

P, Q, R, S, T,…

truth symbols:

true, false

and connectives:

$\wedge$, $\vee$, $\neg$, $\rightarrow$, $\equiv$

The list of connectives in propositional logic and their meaning is tabulated below.

Operators	**Notations**
AND	**Λ**
OR	**V**
Negation	**¬, ~**
If p then q	**p → q**
If p then q and if q then p	**p ↔ q**
Implication	**⇒**
Bi-directional Implication (IFF)	**⇔**
Identity	≡
Logical entailment	|=
Derivability	|-

It should be noted that AND and OR operators are sometimes referred to as **conjunction** and **disjunction** respectively. It may further be added that the provability and implication symbols have been used in an interchangeable manner in this book. The author, however, has a strong reservation to use implication symbol in place of if-then operator and vice versa [3]**.** *The symbol* “x |- y ” *implies that y has been derived from x by following a proof procedure*. The *logical entailment relation*: “x |= y ” *on the other hand means that y logically follows from x.*

Propositional symbols denote *propositions*, or statements about the world that may be either true or false, such as “***the car is red***” or “***water is wet***.” Propositions are denoted by uppercase letters near the end of the English alphabet. Sentences in the propositional calculus are formed from these atomic symbols according to the following rules:

D E F I N I T I O N
PROPOSITIONAL CALCULUS SENTENCES
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*, or *and*, of two sentences is a sentence.
For example: P∧ ¬ P is a sentence.
The *disjunction*, or *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 **WFFs**.
In expressions of the form P∧ Q, P and Q are called the *conjuncts*. In P ∨ Q, P andQ are referred to as *disjuncts*. In an implication, P→ Q, P is the *premise* or *antecedent* and Q, the *conclusion* or *consequent*.
In propositional calculus sentences, the symbols ( ) and [ ] are used to group symbols into subexpressions 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 of legal rules and is therefore “well formed”.

Set of Definitions
1) A **proposition** is a statement or its negation or a group of statements and/or their negations, connected by AND, OR and If-Then operators.
For instance,

p ,
it-is-hot, the-sky-is-cloudy ,
it-is-hot $\wedge$ the-sky-is-cloudy,
it-is-hot $\rightarrow$ the-sky-is-cloudy
are all examples of propositions.

2) When a statement cannot be logically broken into smaller statements, we call it **atomic.**
For example, p, q, the-sky-is-cloudy are examples of atomic propositions.
3) A proposition can assume a **binary valuation space**, i.e., for a proposition p, its valuation space v (p) $\in \{0,1\}$.
4) Let r be a propositional formula, constructed by connecting atomic propositions p, q, s, etc. by operators. An **interpretation** for r is a function that maps v (p), v (q) and v (s) into true or false values that together keep r true.
For example, given the formula: p $\wedge$ q. The possible interpretation is v(p) = true and v (q) =true. It may be noted that for any other values of p and q the formula is false. There may be more than one interpretation of a formula. For instance,
the formula: $\neg p \vee q$ has three interpretations given below.
Interpretations:
{v (p) = true, v (q) = true}, {v (p) = false, v (q) = false}, and
{v (p) = false, v (q) = true}.
5) Propositional formula is called **satisfiable** if its value is true for some interpretation
For example the propositional formula p $\vee$ q is satisfiable as it is true for some interpretations {v (p) = true, v (q) = true}, {v (p) = false, v (q) = true} and {v(p) = true, v (q) =false}.
Generally, we use |= p to denote that p is satisfiable.
6) A propositional formula is **unsatisfiable** or **contradictory** if it is not satisfiable, i.e., for no interpretation it is true.
7) Propositional formula is called valid or tautology, when it is true for all possible interpretations. For example, $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ is a tautology, since it is true for all possible v (p), v (q) and v (r) $\in \{0,1\}$. Here we have 8 possible interpretations for the propositional formula, for which it is true.

Tautologies in Propositional Logic

The tautologies [1] may be directly used for reasoning in propositional logic.
For example, consider the following statements.
p1 = the-sky-is-cloudy, p2 = it-will-rain, and p3 = if (the-sky-iscloudy) then (it-will-rain) $\equiv$ p1 $\rightarrow$ p2.
"p1" and "p2" above represent premise and conclusion respectively for the if then clause. It is obvious from common sense that p2 directly follows from p1 and p3. However to prove it automatically by a computer, one requires help of the following tautology, the proof of which is also given here.
p3 $\equiv$ p1 $\rightarrow$ p2
$\equiv \neg(p1 \wedge \neg p2)$, since p1 true and p2 false cannot occur together.
$\equiv \neg p1 \vee p2$ (by De Morgan's law)

List of tautologies in propositional logic

1. $\neg\neg p \equiv p$
2. $p \wedge q \equiv q \wedge p$

3. $p \lor q \equiv q \lor p$
4. $(p \land q) \land r \equiv p \land (q \land r)$
5. $(p \lor q) \lor r \equiv p \lor (q \lor r)$
6. $p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
7. $p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$
8. $\neg(p \land q) \equiv \neg p \lor \neg q$
9. $\neg(p \lor q) \equiv \neg p \land \neg q$
10. $p \lor p \equiv p$
11. $p \land p \equiv p$
12. $p \land q \rightarrow p$
13. $p \land q \rightarrow q$
14. $p \rightarrow p \lor q$
15. $q \rightarrow p \lor q$

The Semantics of the Propositional Calculus

In the last section presented the syntax of the propositional calculus by defining a set of rules for producing legal sentences. In this section we formally define the *semantics* or "meaning" of these sentences. Because AI programs must reason with their representational structures, it is important to demonstrate that the truth of their conclusions depends only on the truth of their initial knowledge or premises, i.e., that logical errors are not introduced by the inference procedures. A precise treatment of semantics is essential to this goal.

A proposition symbol corresponds to a statement about the world. For example, P may denote the statement "it is raining" or Q, the statement "I live in a brown house." A proposition must be either true or false, given some state of the world. The truth value assignment to propositional sentences is called an *interpretation*, an assertion about their truth in some *possible world*.

Formally, an interpretation is a mapping from the propositional symbols into the set {T, F}. As mentioned in the previous section, the symbols true and false are part of the set of well-formed sentences of the propositional calculus; i.e., they are distinct from the truth value assigned to a sentence. To enforce this distinction, the symbols T and F are used for truth value assignment.

Each possible mapping of truth values onto propositions corresponds to a possible world of interpretation. For example, if P denotes the proposition "it is raining" and Q denotes "I am at work," then the set of propositions {P, Q} has four different functional mappings into the truth values {T, F}. These mappings correspond to four different interpretations. The semantics of propositional calculus, like its syntax, is defined inductively:

D E F I N I T I O N

PROPOSITIONAL CALCULUS SEMANTICS

An interpretation of a set of propositions is the assignment of a 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, $\neg$ 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 lists all possible truth value assignments to the atomic propositions of an 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, Figure 2.1, lists truth values for each possible truth assignment of the operands. P ∧ Q is true only when P and Q are both T. Or (∨), not (¬), implies (→), and equivalence (≡) are 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 of Figure

P	Q	P∧Q
T	T	T
T	F	F
F	T	F
F	F	F

Figure 2.1 Truth table for the operator **∧ .**

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

Figure 2.2 Truth table demonstrating the equivalence of **P→ Q** and **¬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:

¬(¬P) ≡ P

$(P \lor Q) \equiv (\neg P \rightarrow Q)$
the contrapositive law: $(P \rightarrow Q) \equiv (\neg Q \rightarrow \neg P)$
de Morgan's law: $\neg(P \lor Q) \equiv (\neg P \land \neg Q)$ and $\neg(P \land Q) \equiv (\neg P \lor \neg Q)$
the commutative laws: $(P \land Q) \equiv (Q \land P)$ and $(P \lor Q) \equiv (Q \lor P)$
the associative law: $((P \land Q) \land R) \equiv (P \land (Q \land R))$
the associative law: $((P \lor Q) \lor R) \equiv (P \lor (Q \lor R))$
the distributive law: $P \lor (Q \land R) \equiv (P \lor Q) \land (P \lor R)$
the distributive law: $P \land (Q \lor R) \equiv (P \land Q) \lor (P \land R)$

Resolution in Propositional Logic

The principle of resolution in propositional logic can be best described by the following theorem

Resolution theorem: For any three clauses p, q and r,

$p \lor r, q \lor \neg r \Rightarrow p \lor q$.

The resolution theorem can also be used for theorem proving and hence reasoning in propositional logic. The following steps should be carried out in sequence to employ it for theorem proving.

Resolution algorithm

Input: A set of clauses, called axioms and a goal.

Output: To test whether the goal is derivable from the axioms.

Begin

1. Construct a set S of axioms plus the negated goal.

2. Represent each element of S into conjunctive normal form (CNF) by the following steps:

a) Replace 'if-then' operator by NEGATION and OR operation.

b) Bring each modified clause into the following form and then drop AND operators connected between each square bracket. The clauses thus obtained are in conjunctive normal form (CNF). It may be noted that pij may be in negated or non-negated form.

[p 11 ∨ p12 ∨ …….. ∨ p1n] ∧
[p21 ∨ p22∨ ….. …∨ p2n] ∧
……………………………………………
[pm1 ∨ p m2 ∨ ∨ pm n]

3. Repeat

a) Select any two clauses from S, such that one clause contains a negated literal and the other clause contains its corresponding positive (non-negated) literal.

b) Resolve these two clauses and call the resulting clause the resolvent. Remove the parent clauses from S.

Until a null clause is obtained or no further progress can be made.

4. If a null clause is obtained, **then** report: "goal is proved".

Example1: consider the following information:

"The humidity is high or sky is cloudy. If sky is cloudy then it will rain. If the humidity is high then it is hot. It is not hot. "

Convert them into propositional logic formula then use resolution to prove:
"It will rain "

STEP1: convert the axioms into propositional logic

- The humidity is high or sky is cloudy.
  H V C
- If sky is cloudy then it will rain.
  C →R
- If the humidity is high then it is hot.
  H →Q
- It is not hot.
  ¬Q

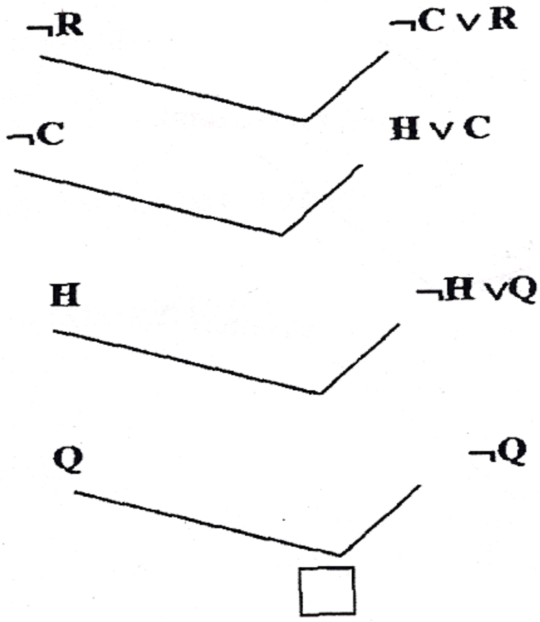
STEP2: convert the propositional logic into conjunctive normal form (CNF)

- H V C
- ¬C V R
- ¬H V Q
- ¬Q

STEP3: convert the goal to propositional logic

It will rain. R
Negate the goal. ¬R

Step4



Example: Consider now an example from the propositional calculus, where we want to prove a from the following axioms:

$\mathbf{b \wedge c \rightarrow a}$

$\mathbf{b}$

$\mathbf{d \wedge e \rightarrow c}$

$\mathbf{e \vee f}$

$\mathbf{d \wedge \neg f}$

we reduce the first axiom to clause form:

$\mathbf{b \wedge c \rightarrow a}$

$\mathbf{\neg(b \wedge c) \vee a}$ **by** $\mathbf{l \rightarrow m \equiv \neg l \vee m}$

$\mathbf{\neg b \vee \neg c \vee a}$ **by de Morgan's law**

The remaining axioms are reduced, and we have the following clauses:

$\mathbf{a \vee \neg b \vee \neg c}$

$\mathbf{b}$

$\mathbf{c \vee \neg d \vee \neg e}$

$\mathbf{e \vee f}$

$\mathbf{d}$

$\mathbf{\neg f}$

The resolution proof is found in Figure () First, the goal to be proved, a, is negated and added to the clause set. The derivation of indicates that the database of clauses is inconsistent.

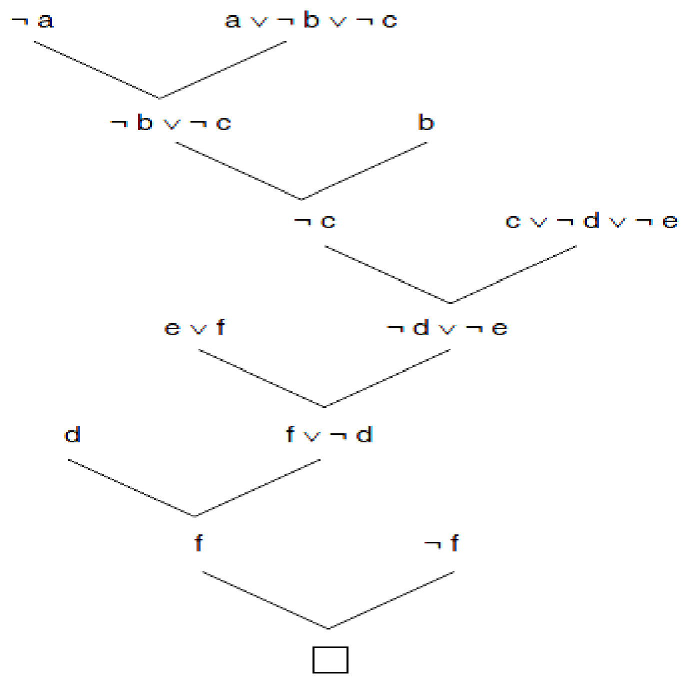


Figure () resolution proof for an example from the propositional calculus.

Example: suppose we are given the following axioms shown in the figure below and we want to prove R. First we convert the axioms as shown in the second column of the figure

Given axioms	Converted to clause form	#
p	**P**	**1**
(P∧Q)→R	¬P∨¬Q∨R	**2**
(S∨T)→Q	¬**S**∨Q	**3**
	¬T∨Q	**4**
T	**T**	**5**

Solution: we negate R producing ¬ R

¬P∨¬Q∨R ¬ R

¬P∨¬Q P

¬Q ¬T∨Q

¬T **T**

□