1^{st} class

Introduction to Artificial Intelligence

م.م سرى محمود عبدالله

Introduction to Artificial Intelligence

What is Intelligence?

Intelligence is the ability to learn about, to learn from, to understand about, and interact with one's environment.

What is Artificial Intelligence (AI)?

A.I:- Is a simply way of making a computer think.

A.I:- Is the part of computer science concerned with designing intelligent computer system that exhibit the characteristic associated with intelligent in human behavior.

This requires many processes:-

1- Learning: - acquiring the knowledge and rules that used these knowledge.

2- Reasoning: - Used the previous rules to access to nearly reasoning or fixed reasoning.

A.I Principles

1- The data structures used in knowledge representation.

2- The algorithms needed to apply that knowledge.

3- The language and programming techniques used their implementation.

What are the goals of AI?

- To make computers more useful by letting them take over tedious tasks from human.
- Understand principles of human intelligence.

- **What is problem reduction meaning?**

  Problem Reduction means that there is a hard problem may be one that can be reduced to a number of simple problems. Once each of the simple problems is solved, then the hard problem has been solved.

AI Purposes

AI has two purposes:

One is to use the power of computers to increase human thinking, just as we use motors to increase human or horse power. Robotics and expert systems are major branches of that.

The other is to use a computer's artificial intelligence to understand how humans think. In a humanoid way, If you test your programs not merely by what they can accomplish, but how they accomplish it, they you're really doing cognitive science; you're using AI to understand the human mind."

Rule-Based System Consider the following rules

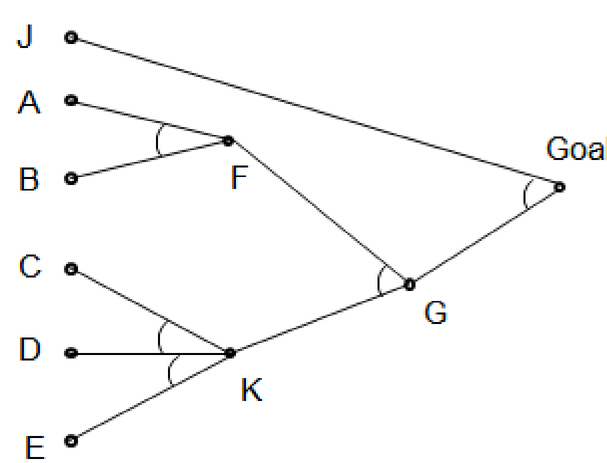
If ***A*** and ***B*** then ***F***

If ***C*** and ***D***

and ***E*** then ***K***

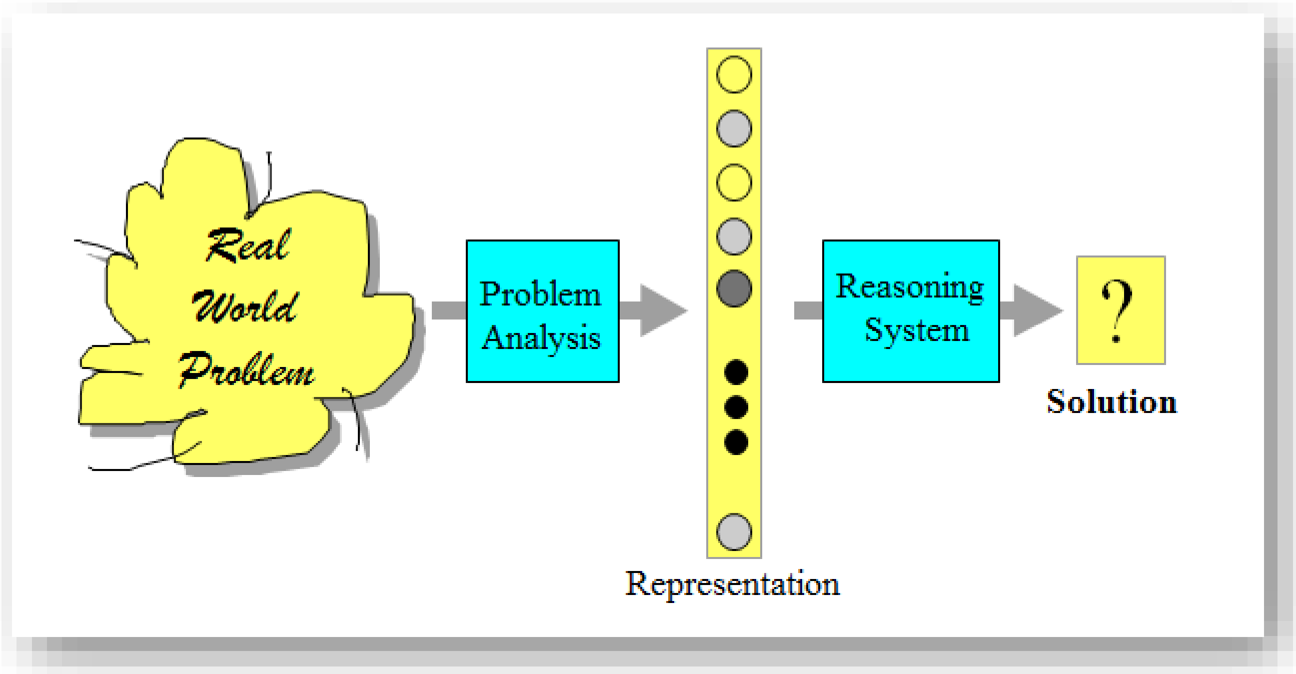
If ***F*** and ***K*** then ***G***

If ***J*** and ***G*** then ***Goal***



- We can ***Forward Chain*** from Premises to Goals or ***Backward Chain*** from Goals and try to prove them.

A model of knowledge-based systems development



Branches of AI

- logical AI
- Search
- Pattern Recognition
- Representation
- Inference
- Learning from Experience
- Planning
- Heuristics
- Natural language processing

Applications of AI

- Game playing
- Speech recognition
- understanding natural language
- Computer vision
- Expert systems
- Heuristic classification

Characteristics of AI

- High societal impact (affect billions of people)
- Diverse (language, vision, robotics)
- Complex (really hard)
- Better handling of information
- Relieves information overload
- Conversion of information into knowledge

Knowledge Representation

Many of the problems machines are expected to solve will require extensive knowledge about the world. Among the things that AI needs to represent are: objects, properties, categories and relations between objects; and many others. There are many methods can be used for knowledge representation and they can be described as follows:-

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

1. Propositional Logic (calculus)

- **Truth symbols**: true, false
- **Propositional symbols**: P, Q, S, ... (**atomic sentences**)
- **Wrapping parentheses**: (…)
- **Sentences are combined** by **connectives**:

$\wedge$	...and	[conjunction]
$\vee$	...or	[disjunction]
$\Rightarrow$	...implies	[implication / conditional]
$\Leftrightarrow$	.is equivalent	[biconditional]
$\neg$	...not	[negation]

<u>Laws:</u>

1. $\sim(\sim P) \equiv P$
2. $P \rightarrow Q \equiv \sim P \vee Q$
3. $P \vee Q \equiv Q \vee P$
4. $P \wedge Q \equiv Q \wedge P$
5. $\sim (P \vee Q) \equiv \sim P \wedge \sim Q$
6. $\sim (P \wedge Q) \equiv \sim P \vee \sim Q$

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

Examples

$(P \wedge Q) \vee (\neg Q \vee P)$

P	Q	P∧Q	┐Q	┐Q∨P	(P∧Q)∨(┐Q∨P)
T	T	T	F	T	T
T	F	F	T	T	T
F	T	F	F	F	F
F	F	F	T	T	T

$A \rightarrow B \quad \equiv \quad \neg A \vee B$

A	B	A→B	┐A	┐A∨B
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Homework:

1. $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$

2. $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$

Examples

- It is hot

  p

- It is not hot

  $\neg p$

- If it is raining, then will not go to mountain

  $p \rightarrow \neg q$

- The food is good and the service is good

  $x \wedge y$

- If The food is good and the service is good then the restaurant is good

  $x \wedge y \rightarrow z$

The prepositional calculus has its limitations that you cannot deal properly with general statements because it represents each statement by using some symbols jointed with connectivity tools.

Homework

Proofs in Propositional Logic the following sentences.

If it is sunny today, then the sun shines on the screen. If the sun shines on the screen, the blinds are brought down. The blinds are not down.

Is it sunny today?

2. Predicate Logic:

To solve the limitations in the prepositional calculus, you need to analyze propositions into predicates and arguments, and deal explicitly with quantification. *Predicate Logic* provides formalism for performing this analysis of prepositions and additional methods for reasoning with quantified expressions.

Ali is a man

Man(ali)

is(ali,man)

object(obj1,obj2,…..).

<u>1-Facts</u>

Maha is a girl

- girl(maha)
- is(maha,girl).

I have a book

- have (I, book).

Ali is a brave man

- is (ali , man, brave)
- man (ali,brave)
- brave (ali,man).
- man(ali) ^ brave (ali)

Ali has red car

- has (ali, car,red).
- has (ali,car) ^ color(car,red)

This is sunny day

- is(day,sunny)
- sunny(day).

Maha has 4 books

- has(maha,4,book)
- has(maha,book) ^ number (book,4)

Ali going to school now

- go(ali,school) ^ time(now)

I have one or two books

- have (I,books) ^ (number(books,1) V number(books,2))

2- Rules

-If its winter then it is cold

- is(weather,winter) → is (weather,cold)
- weather (winter) → weather (cold)

-When I’m sick, I will go to the doctor

- sick(I) → go (I,doctor)

-If student will read good, he will pass

- read(X,good) → pass(X).

-Ahmed goes to the school when he is 6 years old

- age (ahmed,6) → go(ahmed,school).

Examples

- If it is raining, tom will not go to mountain

- rain(weather)→┐go(tom,mountain)
- is(weather,rain) →┐go(tom,mountain)

-All basketball players are tall

- ∀X play(X, basketball) →tall(X)

-John like anyone who likes books

- like(X,book) →like(john,X)

-Nobody likes taxes

- ┐∃X like(X,taxes)

-There is a person who writes computer class

- ∃X write(X,computer class)

-All dogs are animals

- ∀X dogs(X) →animals(X)

-John did not study but he is lucky

- ┐study(john)∧luky(john)

-All cats and dogs are animals

- $\forall X \forall Y$ cats(X) $\wedge$dogs(Y) $\rightarrow$animals(X)$\wedge$animals(Y)

- Everyone passing their AI exam and winning the lottery is happy. But everyone who studies or lucky can pass all their exams. John did not study but he is lucky. Everyone who is lucky wins the lottery. Prove that John is happy.

- $\forall$X pass(X,AI exam)$\wedge$win(X,lottery)$\rightarrow$happy(X)
- $\forall$Y$\forall$E study(Y)$\vee$lucky(Y)$\rightarrow$pass(Y,E)
- $\neg$study(john)$\wedge$lucky(john)
- $\forall$Z lucky(Z)$\rightarrow$win(Z,lottery)
- happy(john)?

Homework: ***Using predicates logic to represent the following sentences***

All people that are not poor and smart are happy. Those people that read are not stupid. John can read and wealthy. Happy people have exciting life. Can anyone found with an exciting life.

Semantic Net

It is consisting of a set of nodes and arcs; each node is represented as a rectangle to describe the objects, the concepts and the events. The arcs are used to connect the nodes and they divided to three parts:-

Is a: → for objects & types

Is → To define the object or describe it

Has a

→ To describe the properties of objects or the actions that the object can

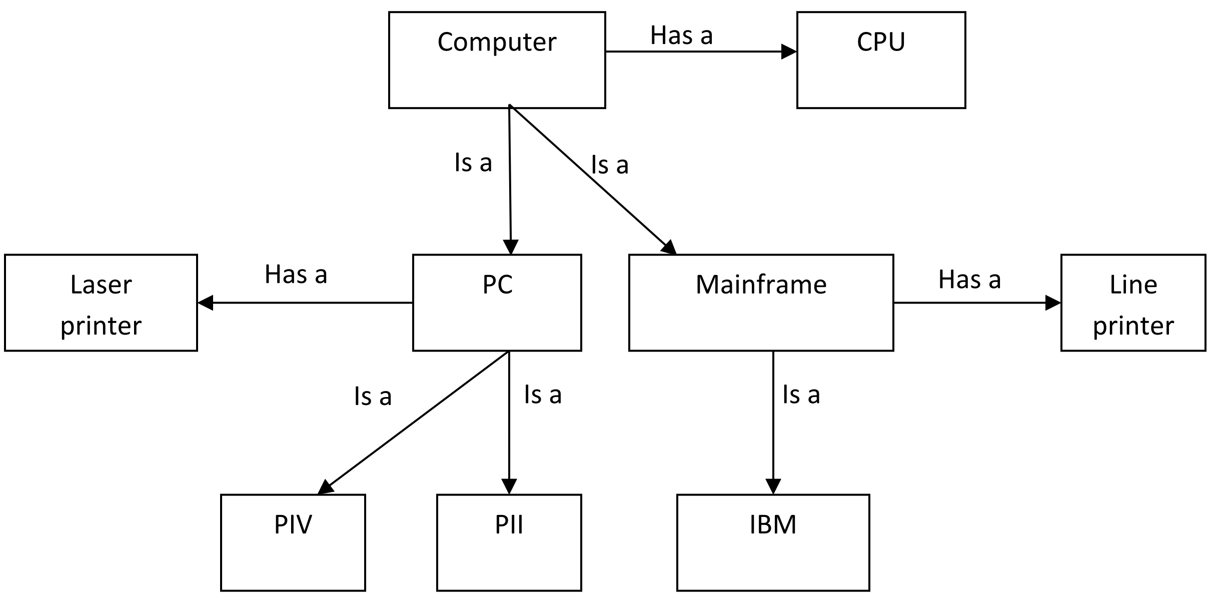
Can

لتمثيل الأفعال والأحداث والكائنات

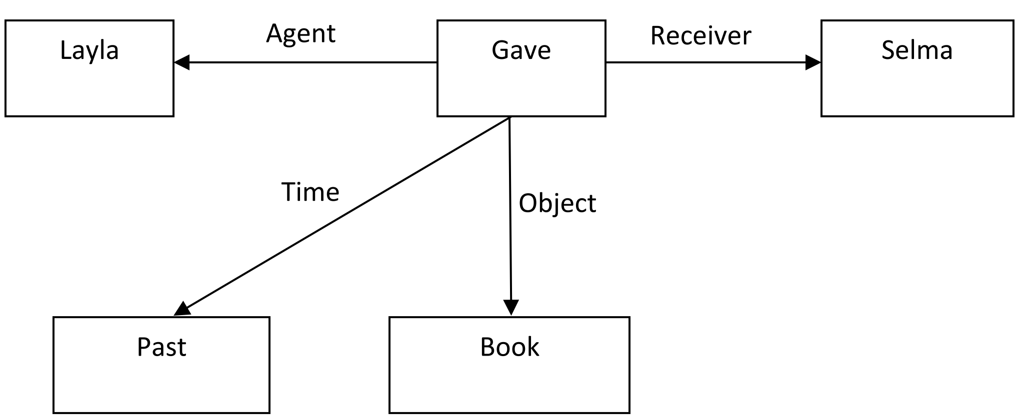
→ لتمثيل العلاقة بين الكائنات

في وصف اللغات الطبيعية فان arcs تخرج من الفعل لتوضح او لتشير الى الفاعل (agent) والمستقبل (Receiver) والكائن (object) كما تشير الى وقت حدوث الفعل أي في الماضي , الحاضر او المستقبل.

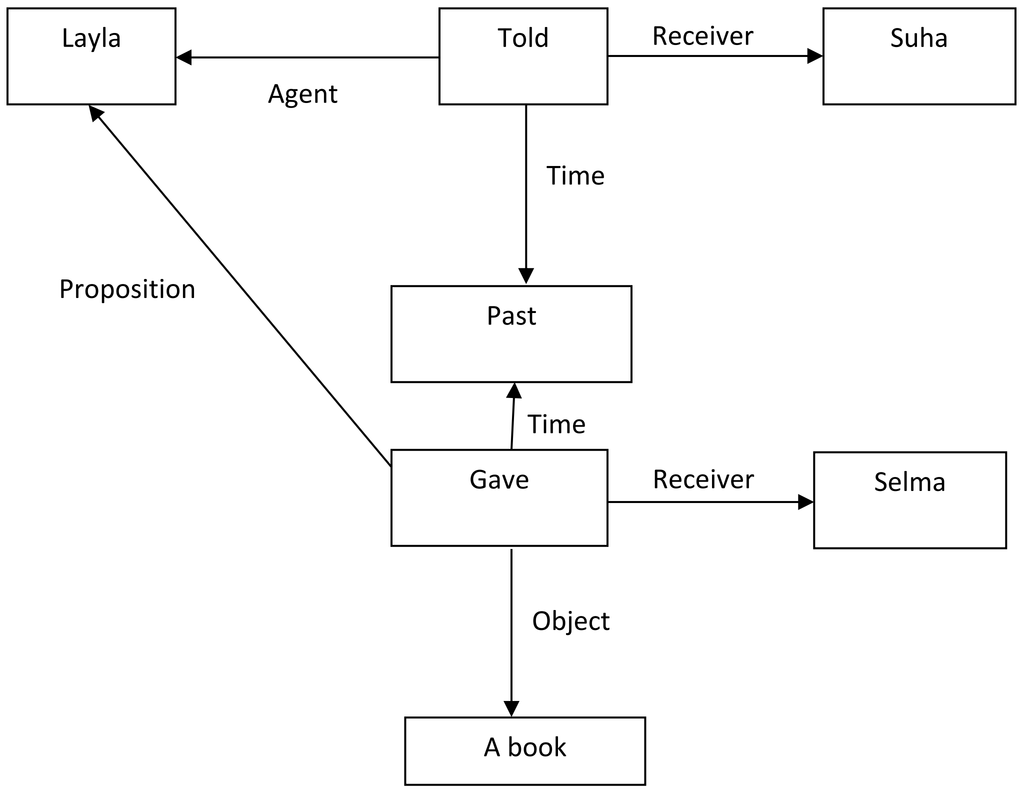
Example 1: Computer has many parts like a CPU and the computer divided into two types, the first one is the mainframe and the second is the personal computer, Mainframe has line printer with large sheet but the personal computer has laser printer, IBM as example to the mainframe and PIII and PIV as example to the personal computer.



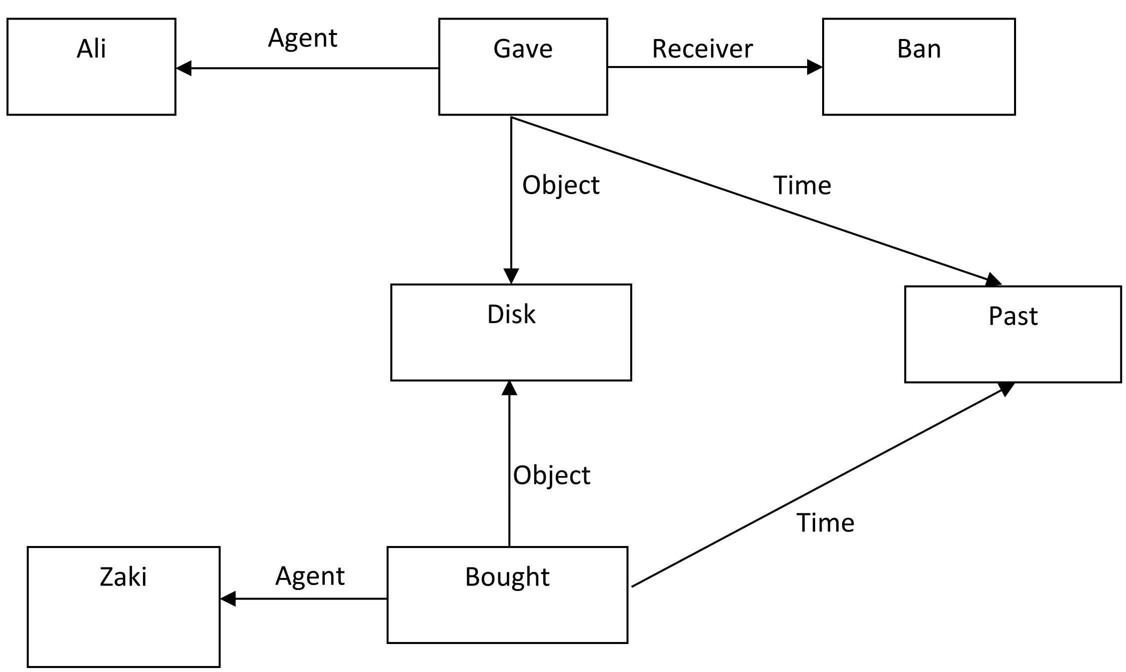
Example 2: Layla gave Selma a book



Example 3: Layla told Suha that she gave Selma a book



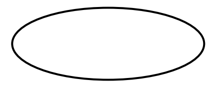
Example 4: Ali gave Ban a disk which is Zaki bought



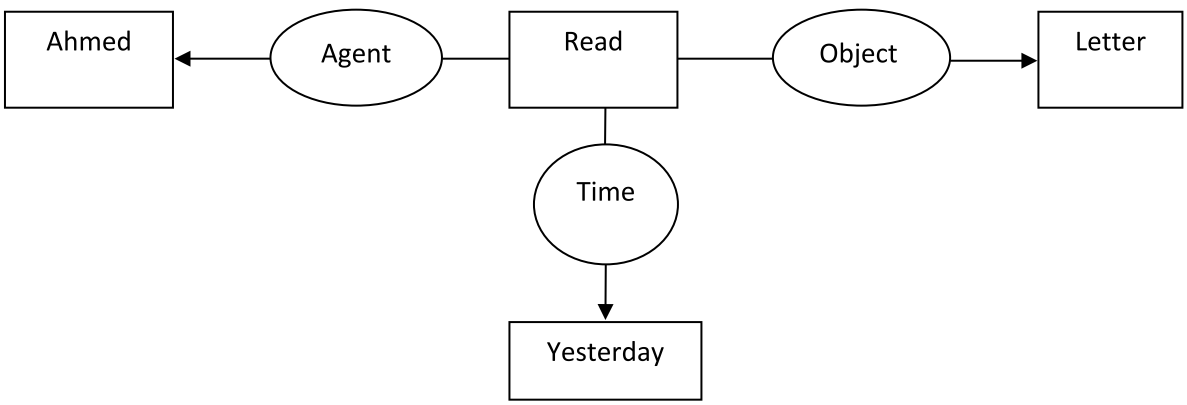
4. The Conceptual Graph

وهي طريقة لتمثيل المعرفة مشابهة لطريقة Semantic Net وتتكون من جزئيين:

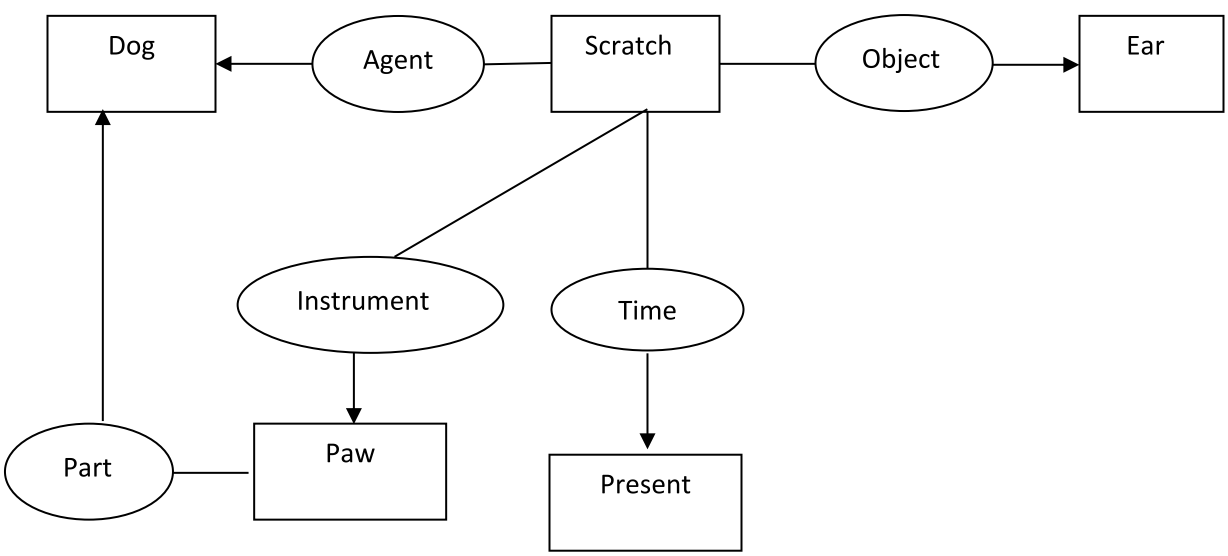
 يستخدم لتمثيل الأسماء والصفات والأفعال والثوابت

 يستخدم لتمثيل أدوات التعريف والعلاقات

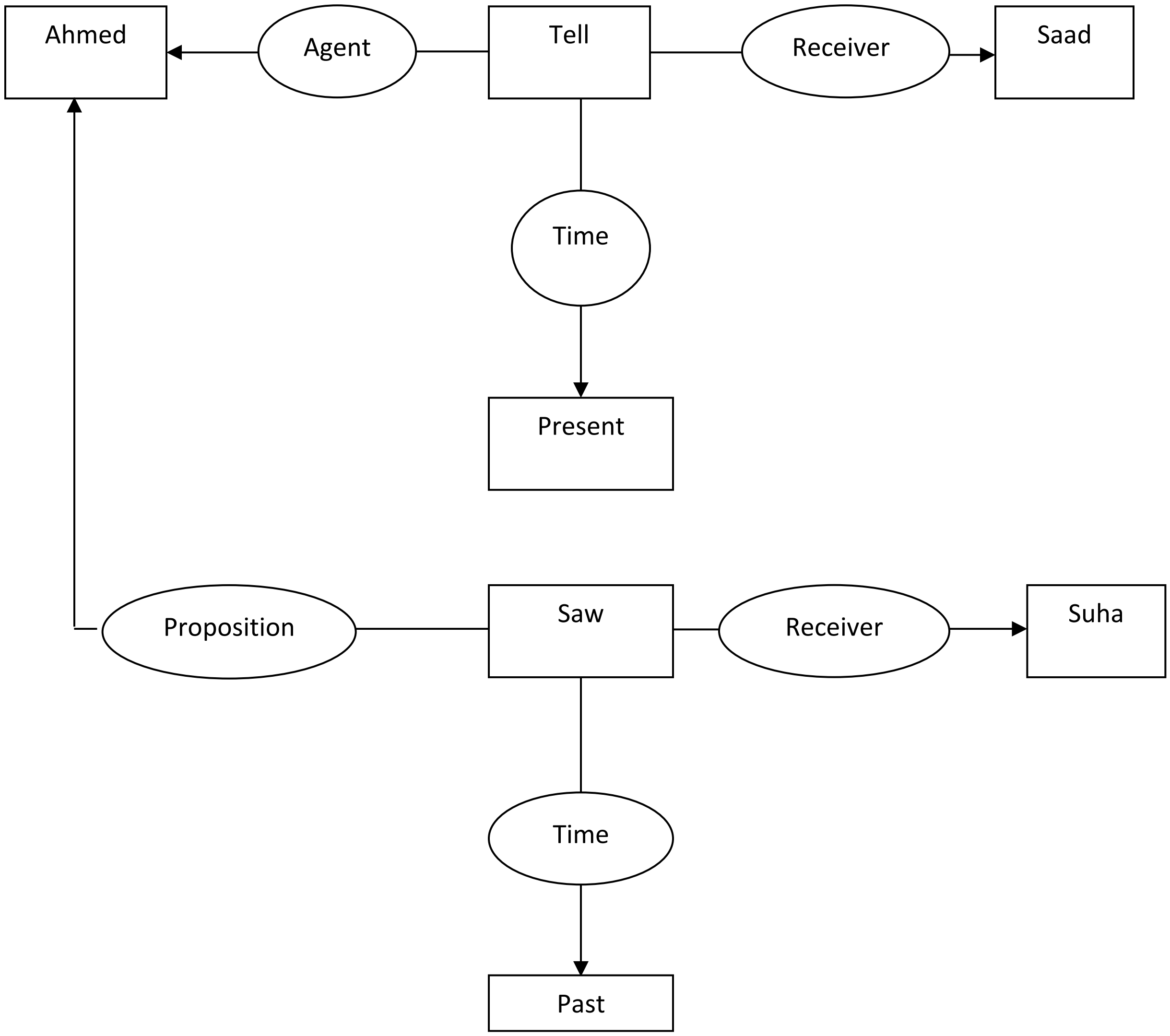
Example 1: Ahmed read a letter yesterday



Example 2: The dog Scratch its ear with its paw



Example 3: Ahmed tell Saad that he saw Suha



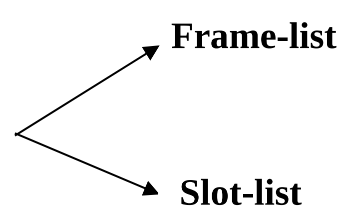
<u>Homework 1</u>: ***Using Semantic Net to represent the following statements:***

Ships are divided in two types, the first is “Ocean lines” and the second is “Oil tank”, the ships has an engine, the oil tank are specified to transfer oil therefore it has “fire tools”, the ocean lines are specified to transfer the traveler therefore it has “swimming poot”, Ibnkaldon as an example to oil tank and ship b and ship n as an example to ocean line.

<u>Homework 2:</u> ***Using Semantic Net and Conceptual graph to solve the following statements:***

a. Suha send a book to Tom.
b. Tom believe that Mustafa like cheese.
c. Monkey ema grasp the banana with hand.

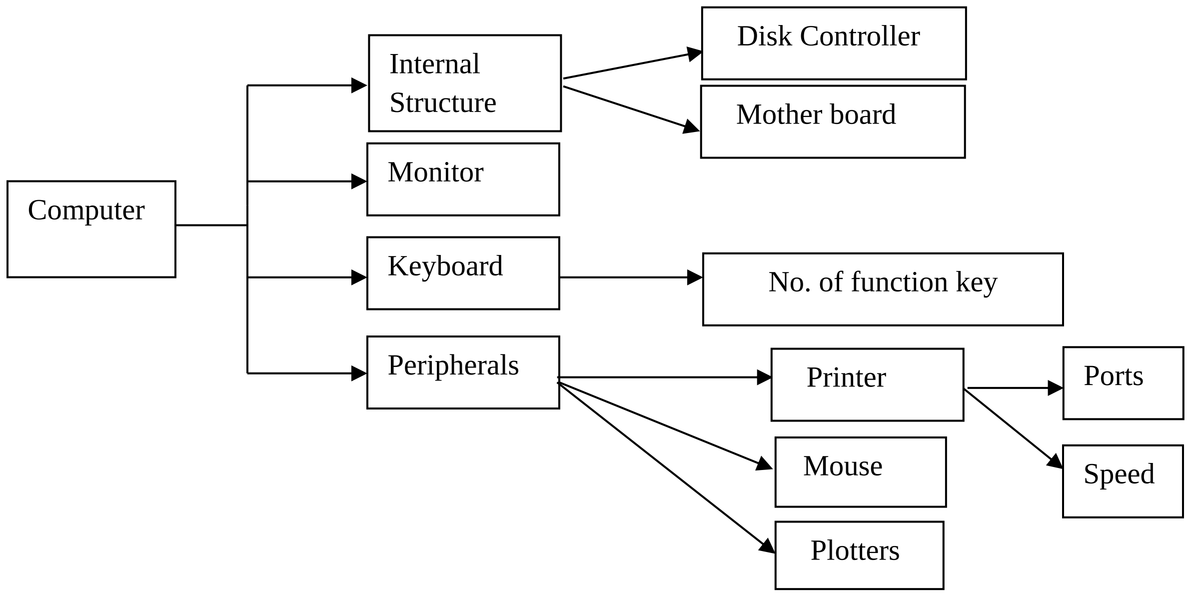
6. Frame Representation



Frame-list (node-name, parent, [child]).

Slot-list (node-name, parent).

Example: the below design should contain 5 frame-list and 8 slot-list



Frame -list (computer,_ ,[Internal structure, monitor, keyboard , Peripherals]).

Frame-list (Internal structure, computer, [disk controller, mother board]).

Frame-list (keyboard, computer, [No. of function key]).

Frame-list (peripherals, computer, [printer,mouse,plotters]).

Frame- list (printer, peripheral, [speed, ports]).

Slot-list (monitor, computer).

Slot-list (Disk controller, internal structure).

Slot-list (motherboard, internal structure).

Slot-list (No. of function key, keyboard).

Slot-list (ports, printer).

Slot-list (speed, printer).

Slot-list (mouse, peripheral).

Slot-list (plotters, peripheral).

Problem Solving

State space search

A ***state space*** is the set of all possible states of the problem under solving.

Problem Solving Operations

1. A set of states.
2. Start state.
3. Operator, possible moves, rules.
4. Goal state.
5. Determine the inference technique to inter the goal.

State Space Searches Examples:-

1. <u>Monkey and Banana Problem</u>

There is a monkey at the door in to a room. In the middle of the room a banana is hanging from the ceiling. The monkey is hungry and wants to get the banana, but he cannot stretch high enough from the floor. At the window of the room there is a box the monkey may use.

The monkey can perform the following actions:-

- Walk on the floor
- Climb the box
- Push the box a round (if it is already at the box).
- Grasp the banana if standing on the box directly under the banana.

The question is (Can the monkey get the banana?), the initial state can be determined by:-

1- Monkey is at door.
2- Monkey is on floor.
3- Box is at Window.
4- Monkey does not have banana.

Initial state: - state (at door, on floor, at window, has not).

At door $\longrightarrow$ horizontal position of monkey

On floor $\longrightarrow$ vertical position of monkey

At window $\longrightarrow$ Position of box

Has not $\longrightarrow$ monkey has not yet grasp the banana

Goal state: - state (at box, on box, under banana, has).

state1 $\longrightarrow$ state2

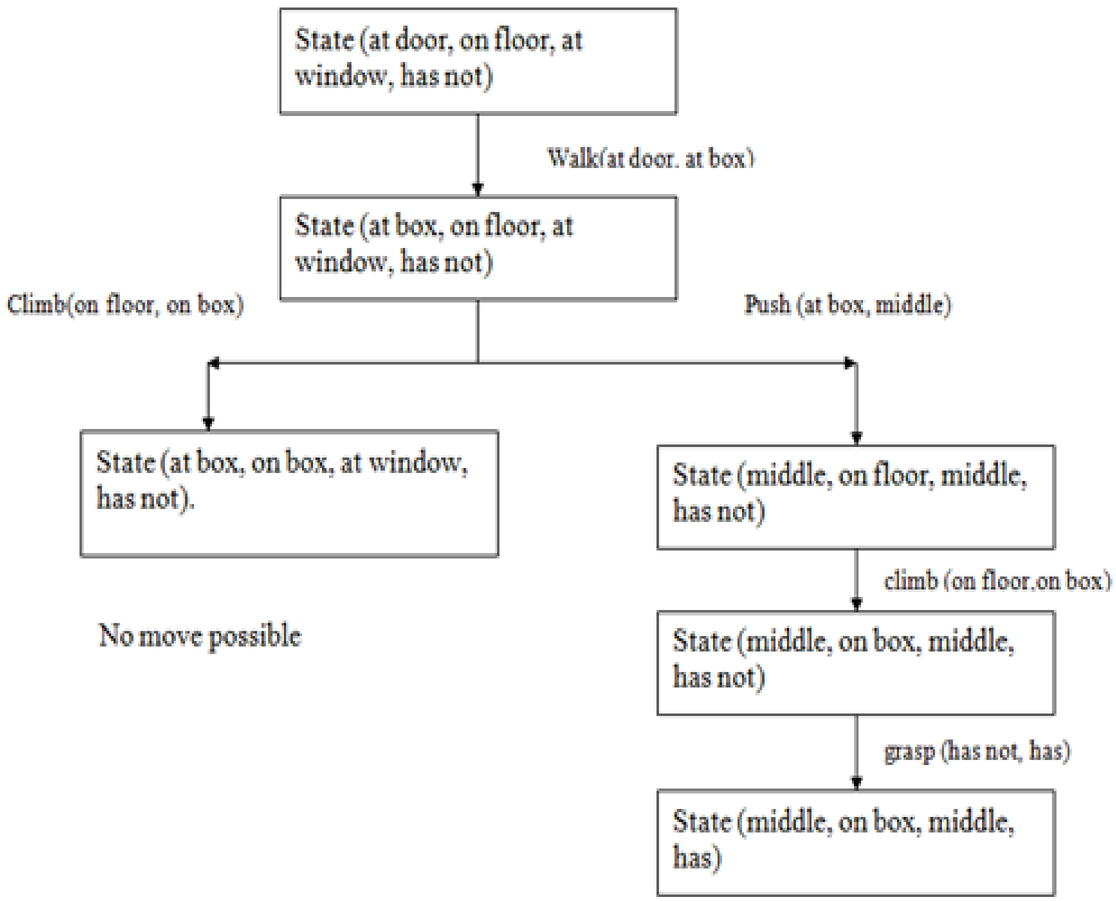
Move (state1, move, state2).

State1: is the state before the move.

Move: is the move executed.

State2: is the state after the move.

The monkey and banana problem can be represented by the following state space:-



2. <u>A water Jug Problem</u>

You are given <u>two jugs</u>, a <u>4-gallon one</u> and a <u>3-gallon one</u>. Neither has any measuring marker on it. There is a pump that can be used to fill the jugs with water. How can you get exactly <u>2 gallons</u> of water into the <u>4-gallon jug</u>?

The state space for this problem can be described as the set of ordered pairs of integers (X,Y) , such that X=0,1,2,3, or 4 and Y=0,1,2, or 3; X represent the number of gallons of a water in the 4-gallon jug , and Y represents the quantity of water in 3-gallon jug. The start state is (0, 0). The goal state is (2, n) for any value of n (since the problem does not specify how many gallons need to be in the 3-gallon jug).

<u>The state space search for the water Jug problem is:</u>

(X,Y): order pair

X: water in 4-gallons ➔ X = 0,1,2,3,4

Y: water in 3-gallons ➔ Y = 0,1,2,3

start state: (0,0)

goal state: (2,n) where n = any value

<u>The rules of the water Jug problem are:</u>

1) $(X,Y: X<4) \longrightarrow$	(4,Y)	Fill the 4 – gallon jug
2) $(X,Y: Y<3) \longrightarrow$	(X,3)	Fill the 3-gallon jug
3) $(X,Y:X>0) \longrightarrow$	(X-D,Y)	Pour some water out of the 4-gallon jug
4) $(X,Y: Y>0) \longrightarrow$	(X,Y-D)	Pour some water out of the 3-gallon jug
5) $(X,Y: X>0) \longrightarrow$	(0,Y)	Empty the 4-gallon jug on the ground
6) $(X,Y: Y>0) \longrightarrow$	(X,0)	Empty the 3-gallon jug on the ground
7) $(X,Y: X+Y>=4 \wedge Y>0) \longrightarrow$	(4,Y-(4-X))	Pour water from the 3-gallon jug into the 4-gallon jug until the 4-gallon jug is full.

8) $(X,Y: X+Y>=3 \wedge X>0) \longrightarrow (X-(3-Y),3)$ Pour water from the 4-gallon jug into the 3-gallon jug until the 3-gallon jug is full.

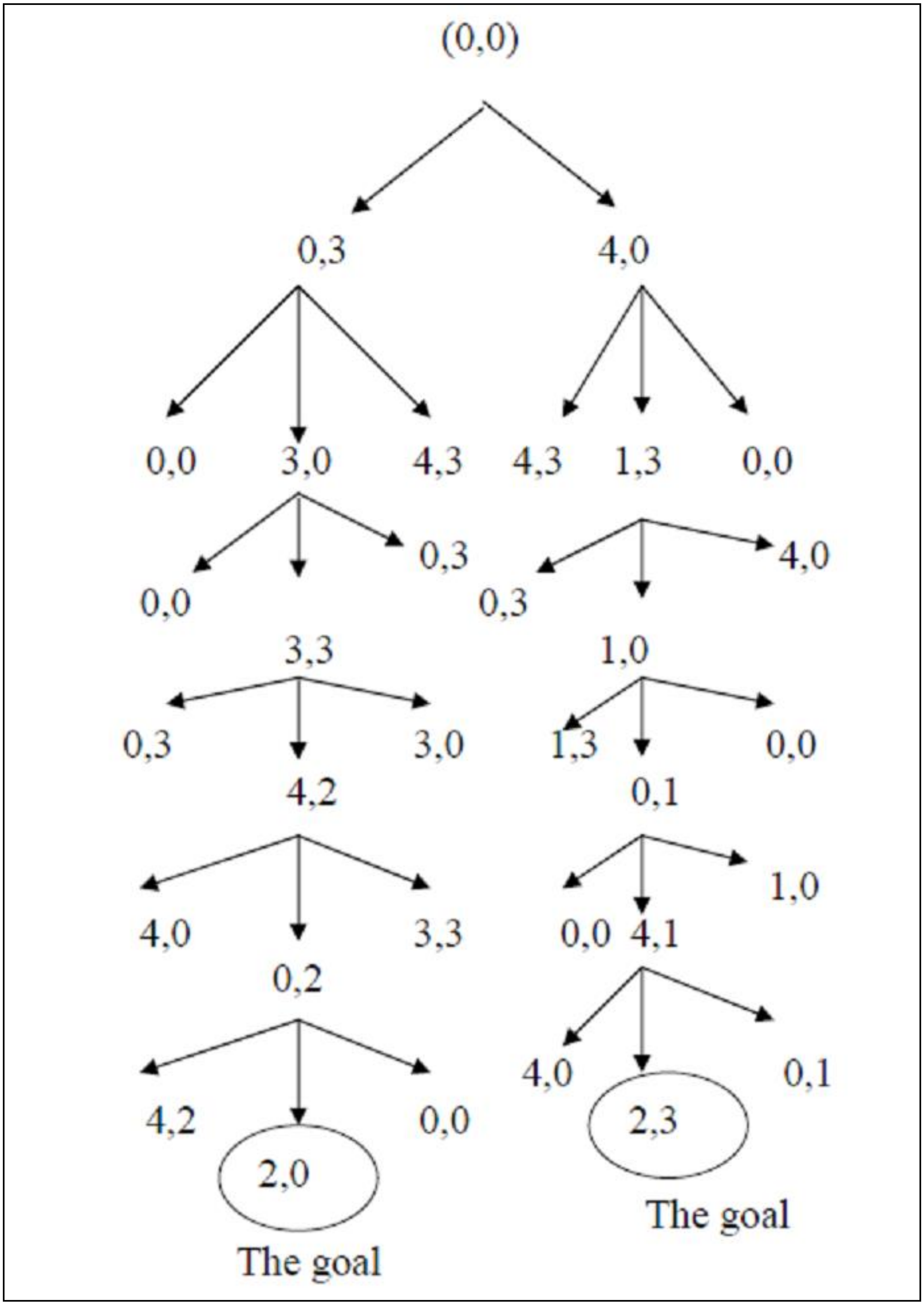
9) $(X,Y: X+Y<=4 \wedge Y>0) \longrightarrow (X+Y,0)$ pour all the water from 3-gallon jug into the 4-gallon jug.

10) $(X,Y: X+Y<=3 \wedge X>0) \longrightarrow (0,X+Y)$ pour all the water from 4- gallon jug into the 3-gallon jug.

<u>The solution of the water jug problem is:</u>

4-Gallon Jug	**3-Gallon Jug**	**Rule Applied**
0	**0**	
0	**3**	**2**
3	**0**	**9**
3	**3**	**2**
4	**2**	**7**
0	**2**	**5**
2	**0**	**9**

<u>The solution can be shown as a search tree as follows:</u>



3. Tower of Hanoi Problem

The Tower of Hanoi is a mathematical puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1- Only one disk can be moved at a time.
2- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3- No disk may be placed on top of a smaller disk.

With three disks, the puzzle can be solved in seven moves. The minimum number of moves required to solve a Tower of Hanoi puzzle is 2n+1, where n is the number of disks.

The below figure shows the trace to solve the Tower of Hanoi problem: