Knowledge Representation Models

What is knowledge representation?

- KR / KRR is concerned with how AI agents think and how thinking contributes to intelligent behavior of agents.
- Represents information about the real world so that a computer (an agent) can understand and utilizes this knowledge to solve the complex real world problems.
- Its not just storing data in a database, but it enables an intelligent machine to learn from that knowledge and experiences. It can behave intelligently like a human.

KRR

- The ability of machines to think and act like humans such as understanding, interpreting and reasoning constitute knowledge representation.
- It is related to designing agents that can think and ensure that such thinking can constructively contribute to the agent’s behaviour in acting like humans.

What to Represent?

- **Object:** All the facts about objects in our world domain.
- **Events:** Events are the actions which occur in our world.
- **Performance:** Describes the behavior of an agent which involves knowledge about how to do things.
- **Meta-knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.

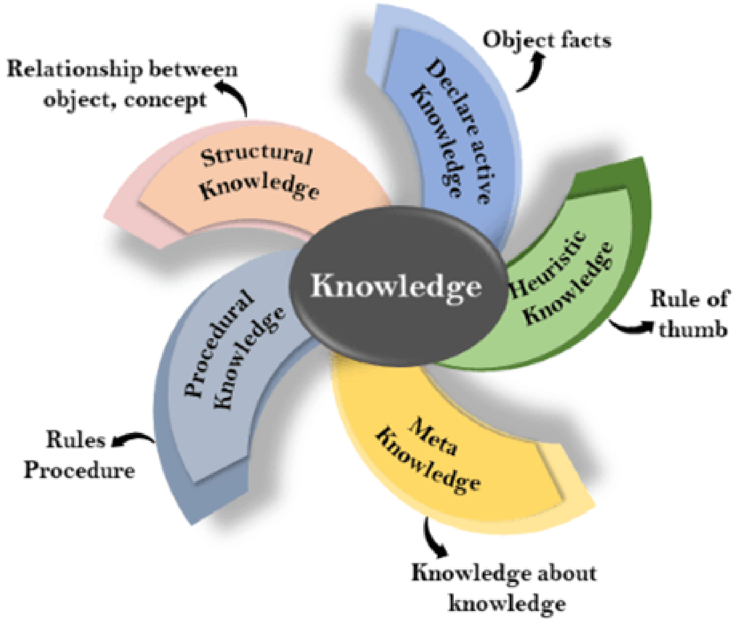
Contd

- **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB.

The Knowledgebase is a group of the Sentences.

- **knowledge:** Knowledge is awareness or familiarity gained by experiences of facts, data, and situations.

Types of knowledge



1. Declarative Knowledge

- Declarative knowledge is to know about something.
- It includes concepts, facts, and objects.
- It is also called descriptive knowledge and expressed in declarative sentences.
- It is simpler than procedural language.

2. Procedural Knowledge

- Procedural knowledge is a type of knowledge which is responsible for knowing how to do something. Also known as imperative knowledge. It can be directly applied to any task. Depends on the task on which it can be applied.
- It includes rules, strategies, procedures, agendas, etc.

3. Meta-knowledge

- Knowledge about the other types of knowledge is called Meta-knowledge.

4. Heuristic knowledge

- Heuristic knowledge is representing knowledge of some experts in a domain.
- Heuristic knowledge is based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

5. Structural knowledge

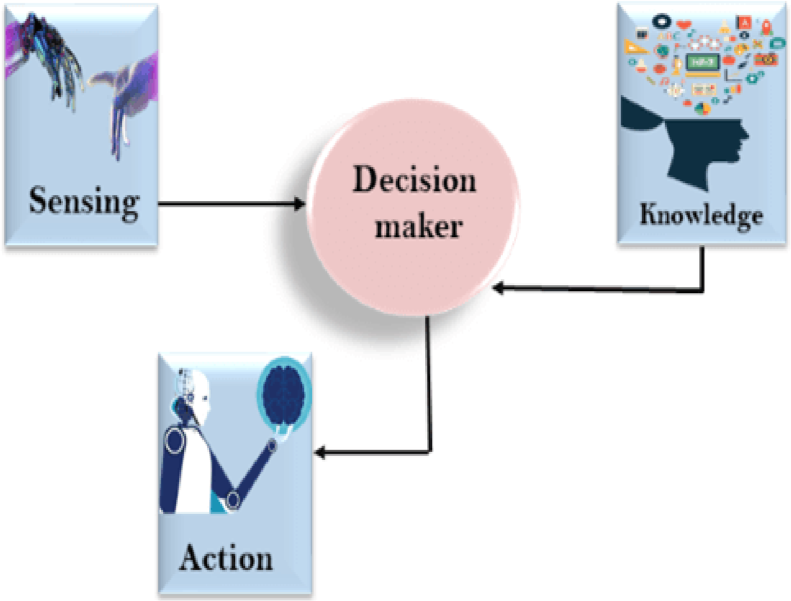
- Structural knowledge is basic knowledge to problem-solving.It describes relationships between various concepts such as kind of, part of, and grouping of something.
- It describes the relationship that exists between concepts or objects.

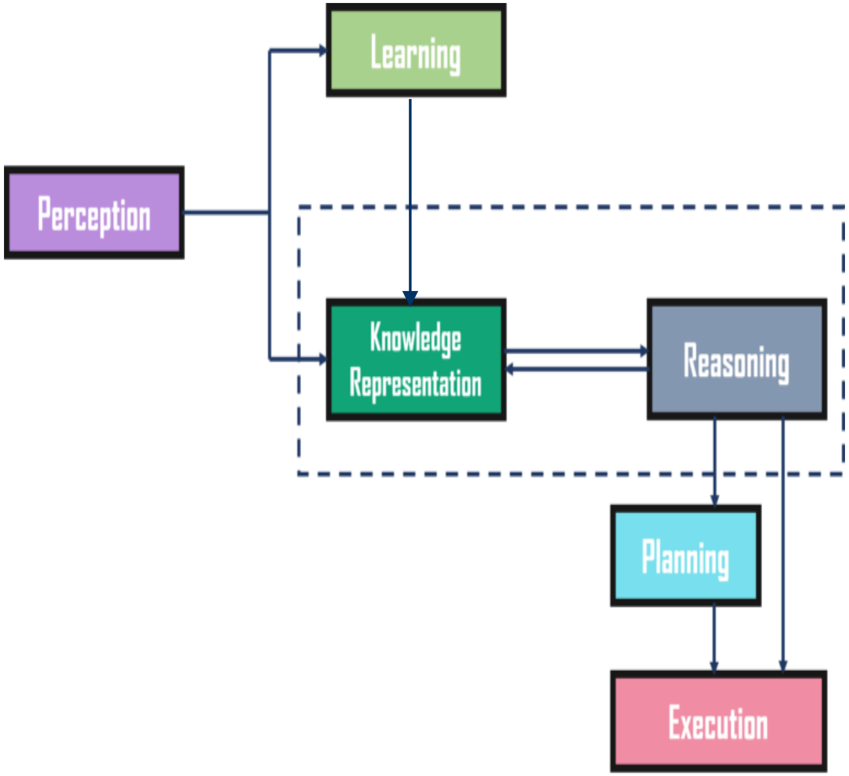
AI knowledge cycle

AI agent consists of various components to display their intelligent behavior

- Perception
- Learning
- Knowledge Representation and Reasoning
- Planning
- Execution

AI knowledge cycle





Approaches to KR

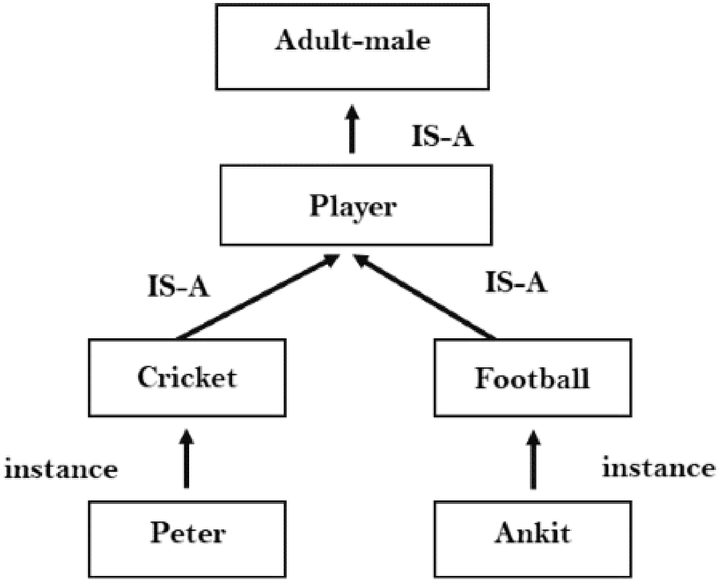
1. Simple relational knowledge

- Simplest way of storing facts which uses the relational method
- Used in database systems where the relationship between different entities is represented.
- Has little opportunity for inference.

Player	Weight	age
1	65	23
2	58	18
3	75	24

2. Inheritable knowledge:

- All data stored into a hierarchy of classes.
- Application of inheritance propei - Elements inherit values from other members of a class.
- Inheritable knowledge shows a relation between instance and class
- Every individual frame can represent the collection of attributes and its value.



3. Inferential knowledge

- Represents knowledge in the form of formal logics. Can be used to derive more facts.
- It guarantees correctness.

Example:

– Marcus is a man	**man(Marcus)**
– All men are mortal	**∀x man (x) ---> mortal (x)**

4. Procedural knowledge

- Uses small programs and codes which describes how to do specific things and how to proceed.
- Use of I**f-Then rule**.
- Use of coding languages like LISP and Prolog
- Can represent heuristic or domain-specific knowledge.

Fundamental types of Knowledge Representation

- **Logical Representation**
- Production Rules
- Semantic Network Representation
- Frame Representation

Logical representation

- Its a language with some concrete rules which deals with propositions
- Logical representation means drawing a conclusion based on various conditions.
- It consists of precisely defined syntax and semantics which supports the sound inference.
- Each sentence can be translated into logics
- The knowledge acquired by logical agents will be definite which means it will either be true or false

Syntax & Semantics

- Syntaxes are the rules which decide how we can construct legal sentences in the logic.

  - which symbol can be used in knowledge representation & how to use those symbols.

- Semantics are the rules by which we can interpret the sentence in the logic.

  - involves assigning a meaning to each sentence.

Contd

- Logical representation consists of two logics:
  - Propositional Logics
  - Predicate logics

Advantages of logical representation:

- Enables us to do logical reasoning.
- Basis for the programming languages.

Disadvantages of logical Representation:

- Have restrictions, Not very natural, inference not be so efficient.

Propositional Logic

- Syntax
  - Propositions, e.g. “it is wet”
  - Connectives: and, or, not, implies, iff (equivalent)

    $\wedge \quad \vee \quad \neg \quad \longrightarrow \quad \longleftrightarrow$
  - Brackets, T (true) and F (false)
- Semantics (Classical AKA Boolean)
  - Define how connectives affect truth
    - “P and Q” is true if and only if P is true and Q is true
  - Use **truth tables** to work out the truth of statements

contd

- Propositional logic combines atoms
  - An atom contains no propositional connectives

  Eg . The Sun is cold (False fact)

  - Have no structure (today_is_wet, john_likes_apples)

Eg. if it is raining then carry an umbrella

p -> q

Where p = it is raining

q = carry an umbrella

Predicate Logic

- **Predicates** allow us to talk about objects
  - Properties: is_wet(today)
  - Relations: likes(john, apples)
  - True or false
- In predicate logic each atom is a predicate
  - e.g. first order logic, higher-order logic
- More expressive logic than propositional

First Order Logic

- **Constants** are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): $\forall$X. likes(X, apples)
  - Exists at least one object (Existential): $\exists$X. likes(X, apples)

Example: FOL Sentence

- “Every rose has a thorn”

$$\forall X.(rose(X) \rightarrow \exists Y.(has(X,Y) \wedge thorn(Y)))$$

- For all X
  - if (X is a rose) then there exists Y (X has Y) and (Y is a thorn)

Example: FOL Sentence

- “On Mondays and Wednesdays I go to John’s house for dinner”

$$\forall X.((is\_mon(X) \vee is\_wed(X)) \rightarrow eat\_meal(me, houseOf(john), X))$$

- Note the change from “and” to “or”
  - Translating is problematic

Higher Order Logic - Beyond True and False

- More expressive than first order. Much harder to reason with
- Multi-valued logics
  - More than two truth values e.g., true, false & unknown
  - Fuzzy logic uses probabilities, truth value in [0,1]
- Modal logics
  - Modal operators define mode for propositions
  - **Epistemic logics** (belief)
    - e.g. □p (necessarily p), ◊p (possibly p), …
  - **Temporal logics** (time)
    - e.g. □p (always p), ◊p (eventually p), …

Logic is a Good Representation

- Fairly easy to do the translation when possible
- Branches of mathematics devoted to it
- It enables us to do logical reasoning
  - Tools and techniques come for free
- Prolog uses logic programs (a subset of FOL)

Non-Logical Representations?

- Production rules
- Semantic networks
  - Conceptual graphs
  - Frames

- Logic representations have restrictions and can be hard to work with
  - Many AI researchers searched for better representations

Fundamental types of Knowledge Representation

- Logical Representation
- **Production Rules**
- Semantic Network Representation
- Frame Representation

Production rules

- Production rules system consist of rule set of

  <condition,action> pairs

  - “if condition then action”

It has mainly three parts:

- The set of production rules
- Working Memory
- The recognize-act-cycle

Production rules

- In production rules, agent checks for the condition and if the condition exists then production rule fires and corresponding action is carried out.
- The condition part of the rule determines which rule may be applied to a problem and the action part carries out the associated problem-solving steps.
- This complete process is called a recognize-act cycle.

Contd

- The working memory contains the description of the current state of the problem. Previous knowledge is matched and a rule fires
- If there is a new situation (state) generates, which causes multiple production rules to be fired together, then this is called conflict set.
- In this situation, the agent needs to select a rule from these sets, and it is called a conflict resolution.

Contd

- Example:
- **IF (at bus stop AND bus arrives) THEN action (get into the bus)**
- **IF (on the bus AND paid AND empty seat) THEN action (sit down).**
- **IF (on bus AND unpaid) THEN action (pay charges).**
- **IF (bus arrives at destination) THEN action (get down from the bus).**

Advantages of Production rule

- The production rules are expressed in natural language.
- The production rules are highly modular, so we can easily remove, add or modify an individual rule.
- Although production rules lack precise semantics for the rules and are not always efficient, the rules lead to a higher degree of modularity. It is the most expressive knowledge representation system.

Disadvantages of Production rule

- Production rule system does not exhibit any learning capabilities, as it does not store the result of the problem for the future uses.
- During the execution of the program, many rules may be active hence rule-based production systems are inefficient.

Fundamental types of Knowledge Representation

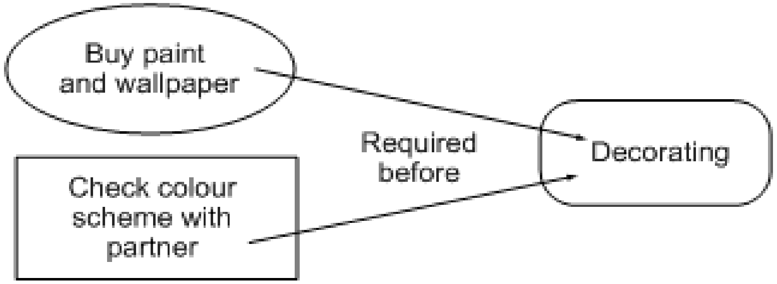
- Logical Representation
- Production Rules
- **Semantic Network Representation**
- Frame Representation

Semantic Networks

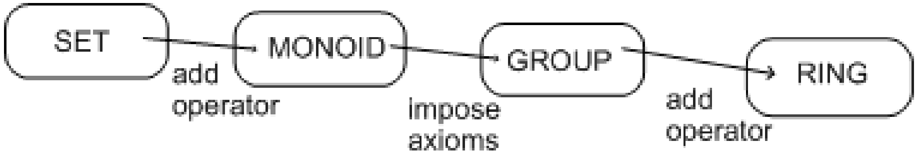
- Graphical representation of knowledge (a graph)
  - Links indicate subset, member, relation, ...
- Equivalent to logical statements (usually FOL)
  - Easier to understand than FOL
  - Specialised SN reasoning algorithms can be faster
- Example: natural language understanding
  - Sentences with same meaning have same graphs

Graphical Representation

- Humans draw diagrams all the time, e.g.
  - Causal relationships

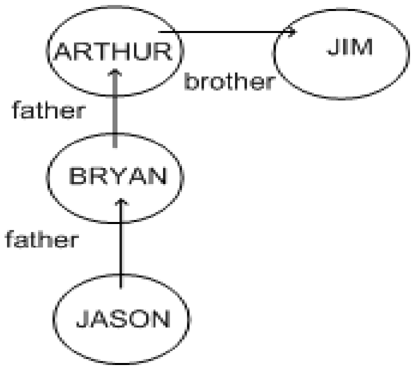


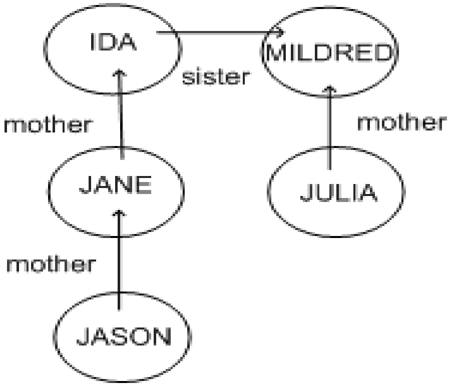
  - And relationships between ideas



Graphical Representation

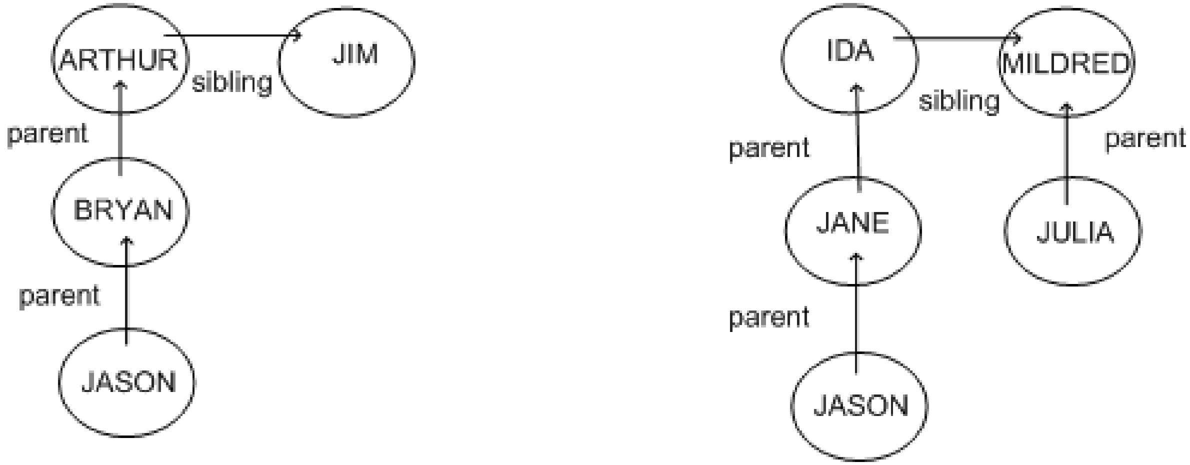
- Graphs easy to store in a computer
- To be of any use must impose a formalism





  - Jason is 15, Bryan is 40, Arthur is 70, Jim is 74
  - How old is Julia?

Semantic Networks



- Because the syntax is the same
  - We can guess that Julia’s age is similar to Bryan’s
- Formalism imposes restricted syntax

Semantic Networks

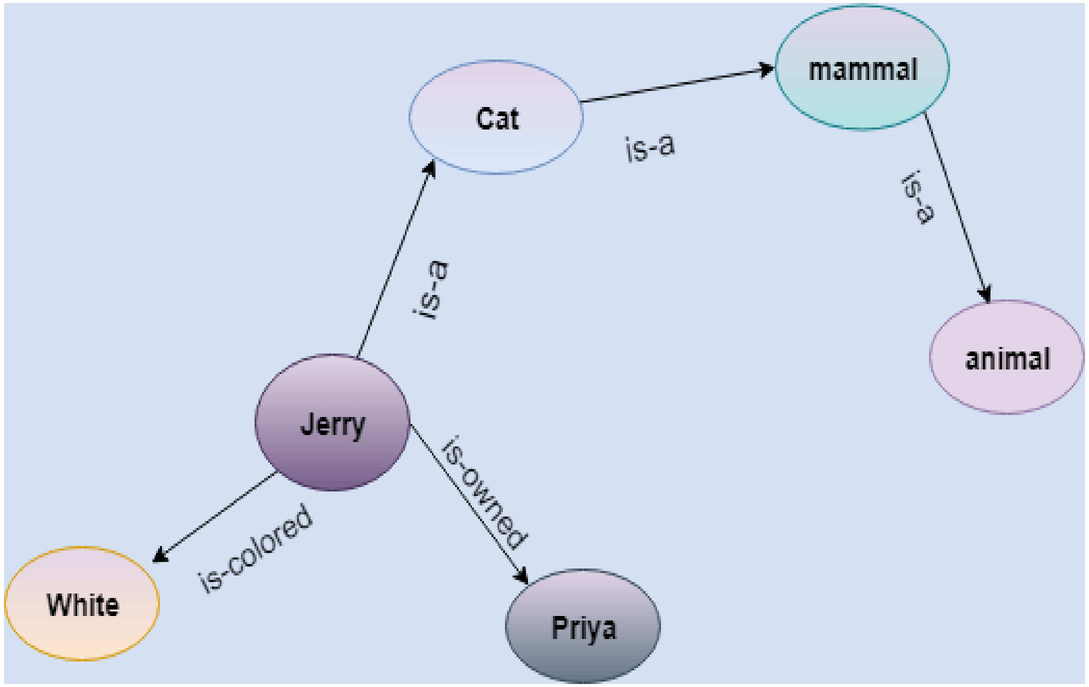
- This network consists of nodes representing objects and arcs which describe the relationship between those objects.
- Semantic networks can categorize the object in different forms and can also link those objects.
- Semantic networks are easy to understand and can be easily extended.

Contd

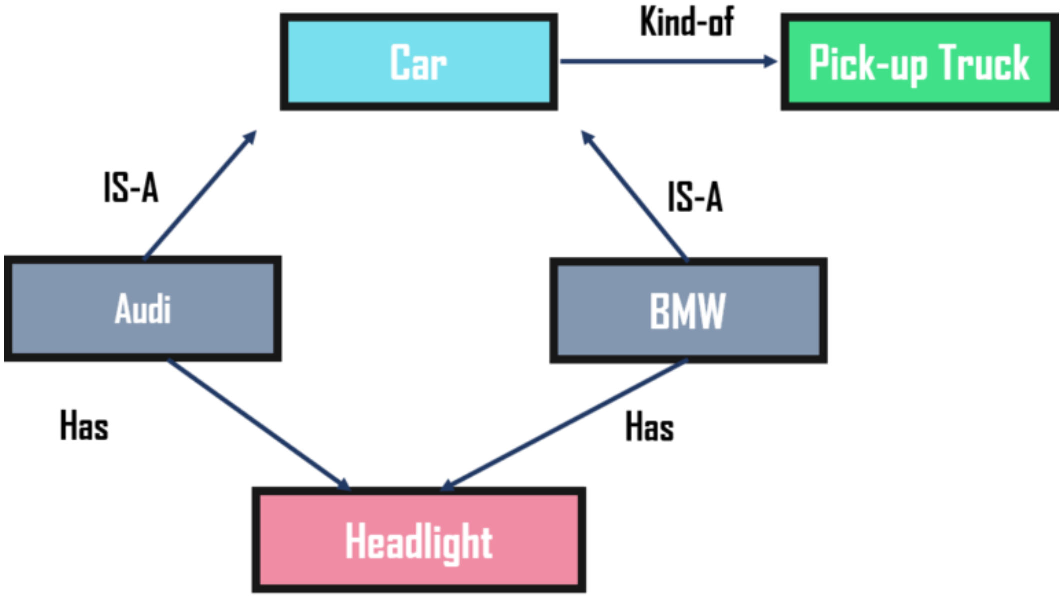
Consists of two types of relations:

- IS-A relation (Inheritance)
- Kind-of-relation
- **Example:** Statements to be represented as nodes and arcs
  - Jerry is a cat.
  - Jerry is a mammal
  - Jerry is owned by Priya.
  - Jerry is white colored.
  - All Mammals are animal.

Contd


Cat
is-a
mammal
is-a
animal
is-a
Jerry
is-owned
is-colored
White
Priya

Semantic Network



Advantages of Semantic network

- Semantic networks are a natural representation of knowledge.
- Semantic networks convey meaning in a transparent manner.
- These networks are simple and easiy to understand an implement.
- A semantic network allows to store knowledge in the form of a graphic network with nodes and arcs representing objects and their relationships.

Contd

- It could represent physical objects or concepts or even situations.
- It also has greater expressiveness than logic representation.

Drawbacks of Semantic network

- Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions.
- It might be possible in the worst case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
- Semantic networks try to model human-like memory (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.

Drawbacks - Contd

- Semantic networks try to model human-like memory (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
- These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.
- Semantic networks do not have any standard definition for the link names.
- These networks are not intelligent and depend on the creator of the system.

Fundamental types of Knowledge Representation

- Logical Representation
- Production Rules
- Semantic Network Representation
- **Frame Representation**

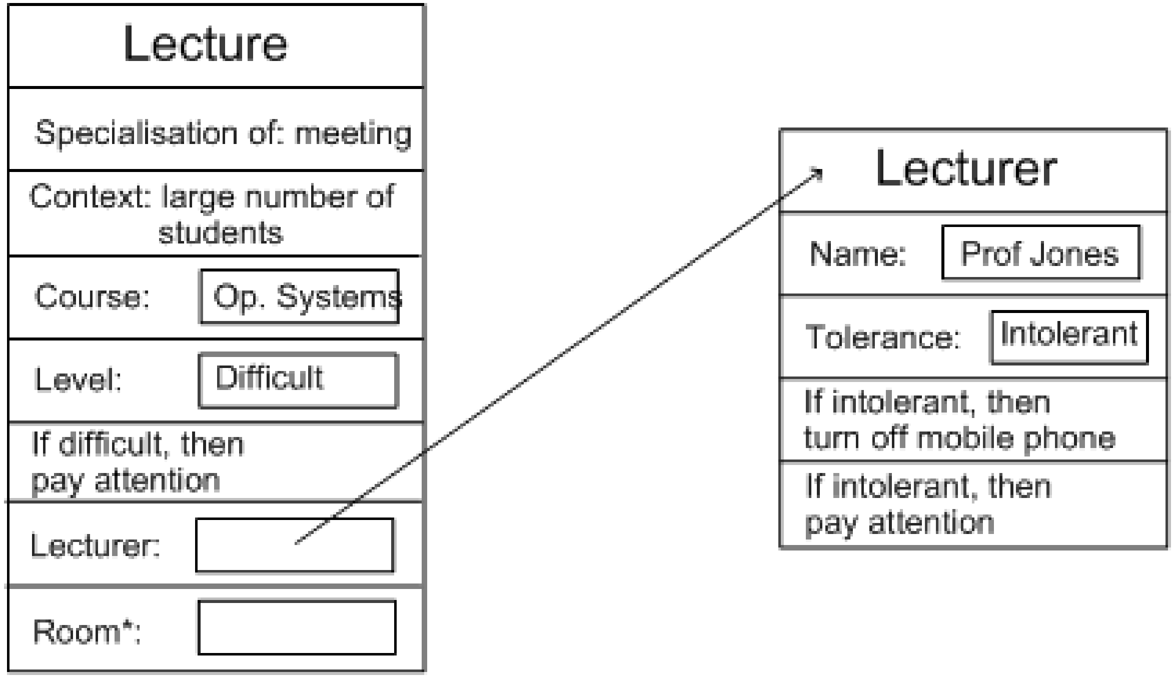
Frame Representation

- A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world.
- It consists of a collection of slots and slot values.
- Slots may be of any type and sizes.
- Slots have names and values which are called facets.

Frame Representations

- Semantic networks where nodes have structure
  - Frame with a number of slots (age, height, ...)
  - Each slot stores specific item of information
- When agent faces a new situation
  - Slots can be filled in (value may be another frame)
  - Filling in may trigger actions
  - May trigger retrieval of other frames
- Inheritance of properties between frames
  - Very similar to objects in OOP

Example: Frame Representation



Lecture

Specialisation of: meeting

Context: large number of students

Course: Op. Systems

Level: Difficult

If difficult, then pay attention

Lecturer:

Room*:

Lecturer

Name: Prof Jones

Tolerance: Intolerant

If intolerant, then turn off mobile phone

If intolerant, then pay attention

Flexibility in Frames

- Slots in a frame can contain
  - Information for choosing a frame in a situation
  - Relationships between this and other frames
  - Procedures to carry out after various slots filled
  - Default information to use where input is missing
  - Blank slots: left blank unless required for a task
  - Other frames, which gives a hierarchy
- Can also be expressed in first order logic

Facets

- Various aspects of a slot is known as **Facets**.
- Facets are features of frames which enable us to put constraints on the frames. Example: IF-NEEDED facts are called when data of any particular slot is needed.
- A frame is also known as **slot-filter knowledge representation** in artificial intelligence.

Frames - contd

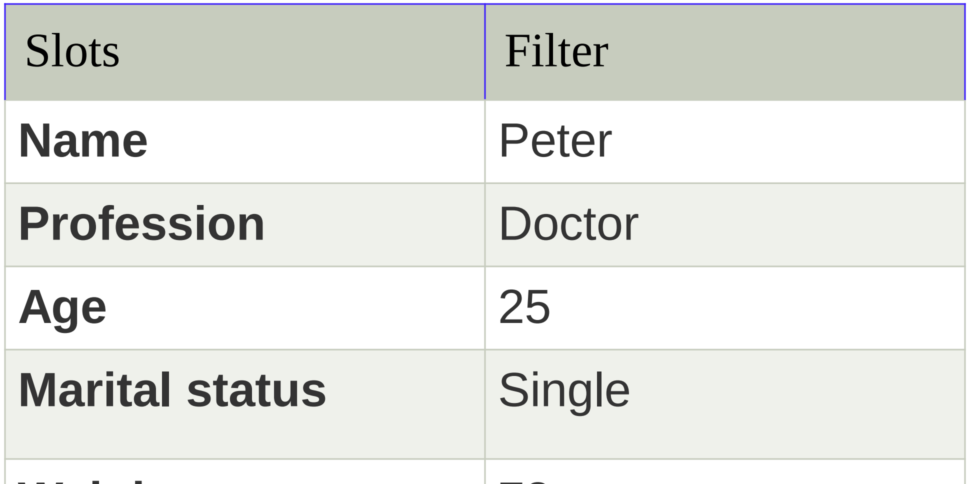
- Frames are derived from semantic networks and later evolved into our modern-day classes and objects.
- Frames system consist of a collection of frames which are connected.
- In the frame, knowledge about an object or event can be stored together in the knowledge base.
- Frames are widely used in natural language processing and machine visions.

Frame for a book

Slots	Filters
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig
Edition	Third Edition
Year	1996
Page	1152

Example 2

- Let's suppose we are taking an entity, Peter. Peter is an engineer as a profession, and his age is 25, he lives in city London, and the country is England. So following is the frame representation for this:



Slots	Filter
Name	Peter
Profession	Doctor
Age	25
Marital status	Single
Weight	78

Slots	Filter
Name	Peter
Profession	Doctor
Age	25
Marital status	Single
Weight	78

Advantages of frame representation

- makes the programming easier by grouping the related data.
- comparably flexible.
- It is very easy to add slots for new attribute and relations.
- It is easy to include default data and to search for missing values.
- Frame representation is easy to understand and visualize.

Disadvantages of frame representation

- Inference mechanism is not be easily processed.
- Frame representation has a much generalized approach.