Knowledge Representation

Rule Based Systems (summary)

A rule-based system is used to store and manipulate knowledge to interpret information in a useful way.

A typical rule-based system has four basic components:

- **List of rules** or rule base, which is a specific type of knowledge base.
- **Inference engine**, which infers information or takes action based on the interaction of input and the rule base.
- **Working memory** (Temporary)**.**
- **User interface** through which input and output signals are received and sent.

Introduction Fuzzy Logic

Fuzzy Set Theory and Fuzzy logic

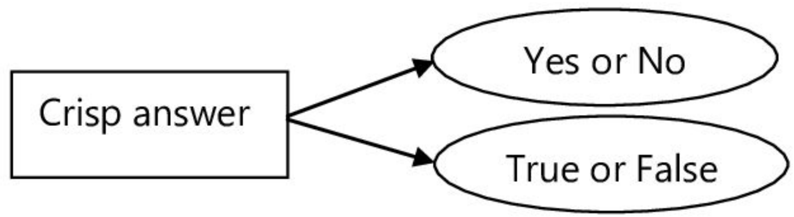
- **Fuzzy Set Theory** it deals with Fuzzy Sets. Fuzzy sets are sets whose elements have degrees of membership.
- Main components are membership function and fuzzy set operations.

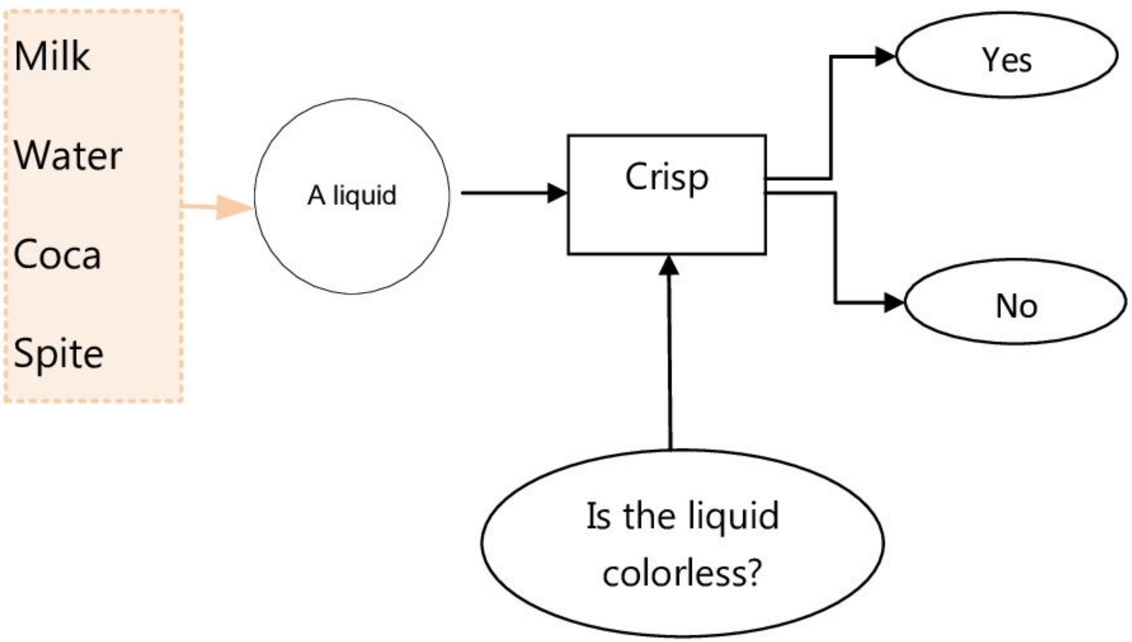
Fuzzy Set Theory and Fuzzy logic

- **Fuzzy Set Theory** it deals with Fuzzy Sets. Fuzzy sets are sets whose elements have degrees of membership.
- Main components are membership function and fuzzy set operations.

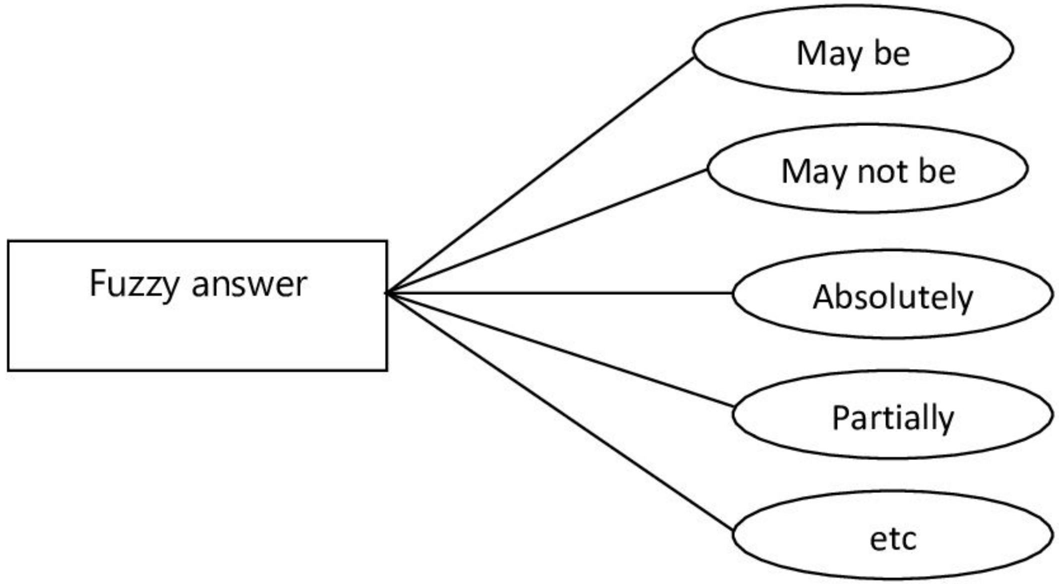
- **Fuzzy logic** it is a form of many-valued logic in which the truth value of variables may be any real number between 0 and 1.
- Note that Boolean logic, the truth values of variables may only be the integer values 0 or 1.
- Main components are fuzzy propositions and connectives.

Crisp Answer

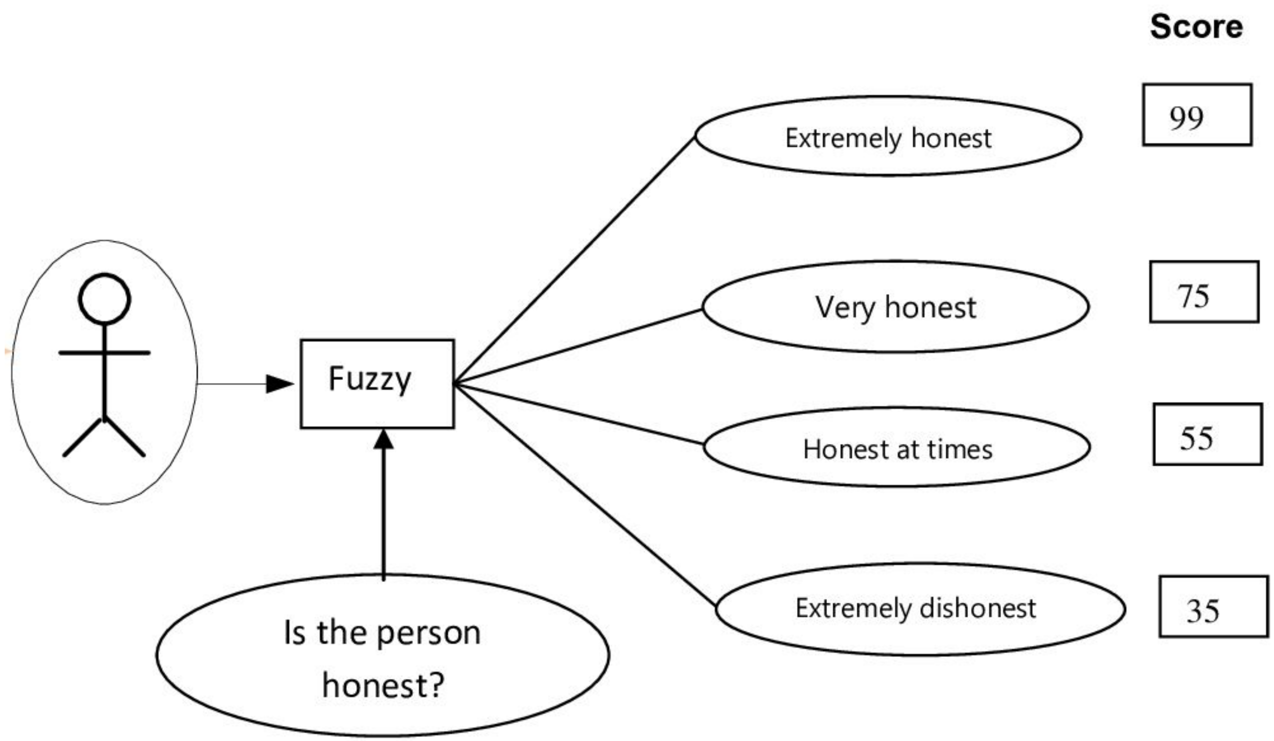




Fuzzy Answer



Fuzzy Answer



Fuzzy set

Let us discuss about fuzzy set.

X = All students in **Artificial Intelligence**

S = All **Good students**.

S = $\{ (s, g) \mid s \in X \}$ and g(s) is a measurement of goodness of the student *s*.

Example:

S = { (Rajat, 0.8), (Kabita, 0.7), (Salman, 0.1), (Ankit, 0.9) } etc.

Definition

Definition : Membership function (and Fuzzy set)

If X is a universe of discourse and $x \in X$, then a fuzzy set A in X is defined as a set of ordered pairs, that is

$A = \{(x, \mu_A(x)) | x \in X\}$ where $\mu_A(x)$ is called the membership function for the fuzzy set A.

Note:
$\mu_A(x)$ map each element of X onto a membership grade (or membership value) between 0 and 1 (both inclusive).

Fuzzy set vs. Crisp set

Crisp Set	Fuzzy Set
1. S = { s | s $\in$ X }	1. F = (s, μ) | s $\in$ X and μ(s) is the degree of s.
2. It is a collection of elements.	2. It is collection of ordered pairs.
3. Inclusion of an element s $\in$ X into S is crisp, that is, has strict boundary **yes** or **no**.	3. Inclusion of an element s $\in$ X into F is fuzzy, that is, if present, then with a degree of **membership**.

Fuzzy set vs. Crisp set

Note: A crisp set is a fuzzy set, but, a fuzzy set is not necessarily a crisp set.

Example:
$H = \{ (h_1, 1), (h_2, 1), \ldots , (h_L, 1) \}$

$Person = \{ (p_1, 1), (p_2, 0), \ldots , (p_N, 1) \}$

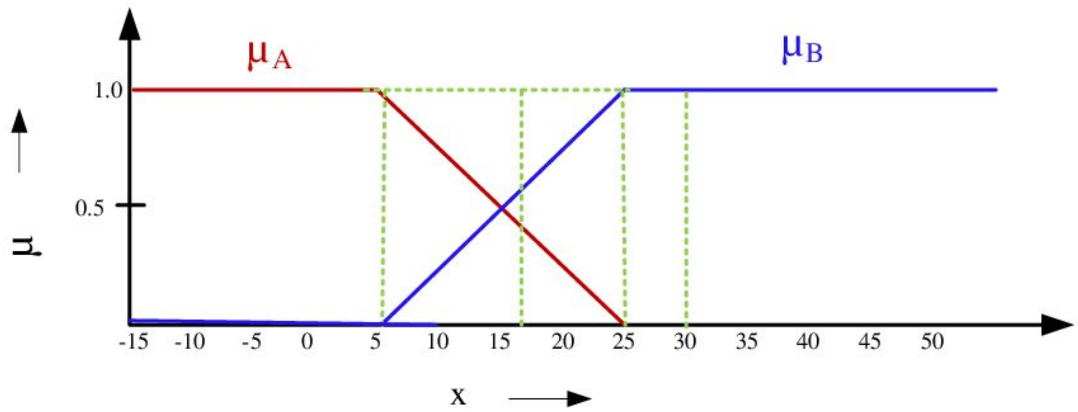
In case of a crisp set, the elements are with extreme values of degree of membership namely either 1 or 0.

Example

Two fuzzy sets A and B with membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively defined as below.

A = **Cold climate** with $\mu_A(x)$ as the MF.

B = **Hot climate** with $\mu_B(x)$ as the M.F.



Here, X being the universe of discourse representing entire range of temperatures.

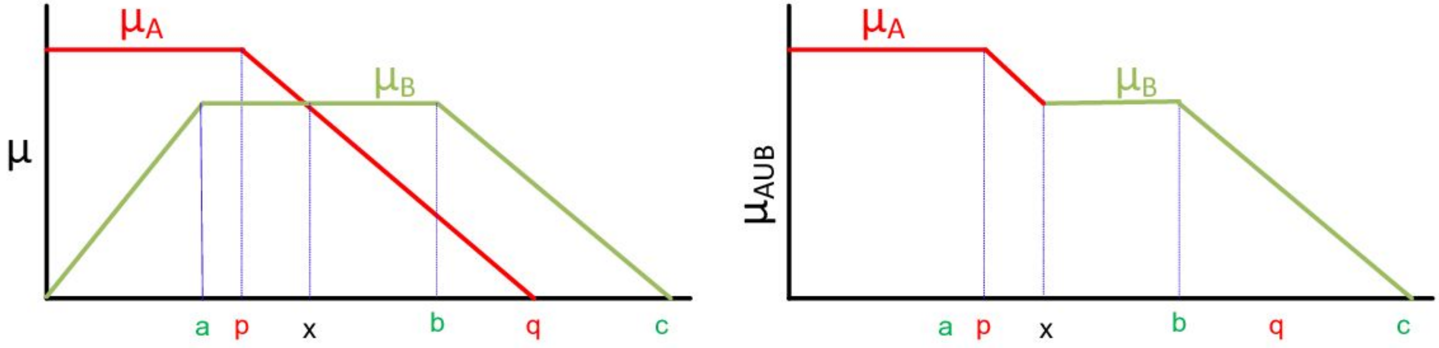
Basic fuzzy set operations: Union

Union ($A \cup B$):

$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$

Example:

$A = \{(x_1, 0.5), (x_2, 0.1), (x_3, 0.4)\}$ and
$B = \{(x_1, 0.2), (x_2, 0.3), (x_3, 0.5)\}$;
$C = A \cup B = \{(x_1, 0.5), (x_2, 0.3), (x_3, 0.5)\}$



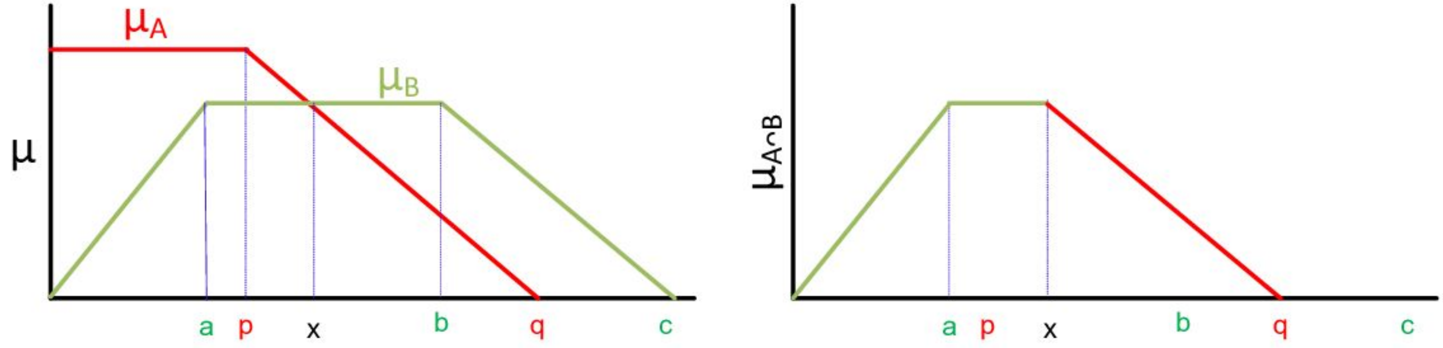
Basic fuzzy set operations: Intersection

Intersection ($A \cap B$):

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

Example:

$A = \{(x_1, 0.5), (x_2, 0.1), (x_3, 0.4)\}$ and
$B = \{(x_1, 0.2), (x_2, 0.3), (x_3, 0.5)\}$;
$C = A \cap B = \{(x_1, 0.2), (x_2, 0.1), (x_3, 0.4)\}$



Basic fuzzy set operations: Complement

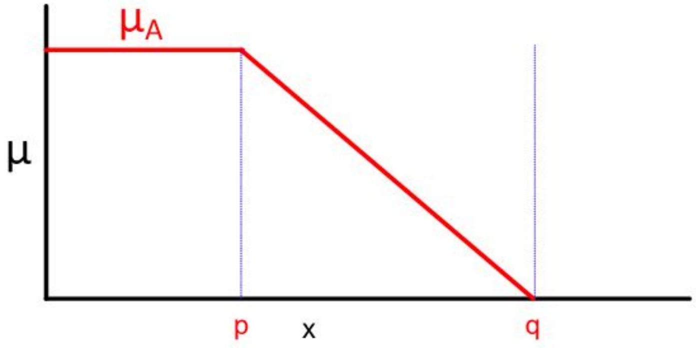
Complement (A^C):

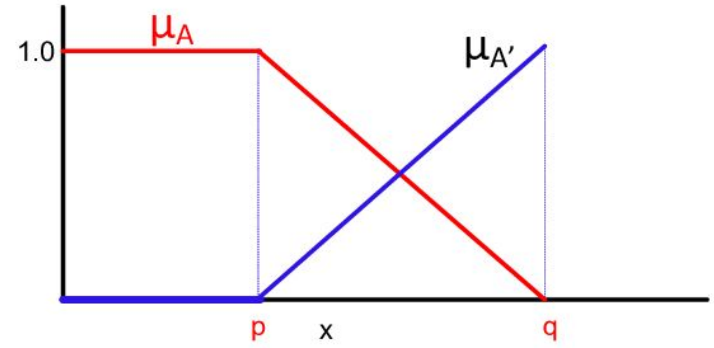
$$\mu_{A_{A^C}}(x) = 1 - \mu_A(x)$$

Example:

$A = \{(x_1, 0.5), (x_2, 0.1), (x_3, 0.4)\}$

$C = A^C = \{(x_1, 0.5), (x_2, 0.9), (x_3, 0.6)\}$





Fuzzy Logic

Fuzzy Logic

- In **crisp logic**, the truth value acquired by the proposition are 2-valued, namely true as 1 and false as 0.
- In **fuzzy logic**, the truth values are multi-valued, as absolute true, partially true, absolute false etc. represented numerically as real value between 0 to 1.

Fuzzy Logic

- A **fuzzy proposition** is a statement P which acquires a fuzzy truth value T(P).
    - Example : P: Ram is honest
    - **T**(P) = 0.8 , means P is partially true.
    - T(P) = 1 , means P is absolutely true.

Fuzzy Logic

Let P and Q are fuzzy proposition and T(P) , T(Q) are their truth values.

Connective	Symbols	Usage	Definition
Nagation	$\neg$	$\neg$ P	1 – T(P)
Disjuction	V	P V Q	Max[T(P) , T(Q)]
Conjuction	$\wedge$	P $\wedge$ Q	min[T(P), T(Q)]
Implication	$\Rightarrow$	P $\Rightarrow$ Q	$\neg$P V Q = max (1-T(P), T(Q)]

Example

- P : Mary is efficient , T(P) = 0.8 ,
- Q : Ram is efficient T(Q) = 0.65 ,
- ¬P : Mary is not efficient,
    - T(¬ P) =

Example

- P : Mary is efficient , T(P) = 0.8 ,
- Q : Ram is efficient T(Q) = 0.65 ,
- ¬P : Mary is not efficient,
    - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$

Example

- P : Mary is efficient , $T(P) = 0.8$,
- Q : Ram is efficient $T(Q) = 0.65$,
- ¬P : Mary is not efficient,
  - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$
- $P \wedge Q$: Mary is efficient and so is Ram,
  - $T(P \wedge Q) =$

Example

- P : Mary is efficient , $T(P) = 0.8$,
- Q : Ram is efficient $T(Q) = 0.65$,
- ¬P : Mary is not efficient,
  - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$
- $P \wedge Q$: Mary is efficient and so is Ram,
  - $T(P \wedge Q) = \min (T(P), T(Q)) = \min (0.8, 0.65)) = 0.65$

Example

- P : Mary is efficient , T(P) = 0.8 ,
- Q : Ram is efficient T(Q) = 0.65 ,
- ¬P : Mary is not efficient,
  - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$
- $P \wedge Q$: Mary is efficient and so is Ram,
  - $T(P \wedge Q) = \min(T(P), T(Q)) = \min(0.8, 0.65)) = 0.65$
- $P \vee Q$: Either Mary or Ram is efficient i.e.
  - $T(P \vee Q) =$

Example

- P : Mary is efficient , $T(P) = 0.8$,
- Q : Ram is efficient $T(Q) = 0.65$,
- $\neg P$: Mary is not efficient,
  - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$
- $P \wedge Q$: Mary is efficient and so is Ram,
  - $T(P \wedge Q) = \min(T(P), T(Q)) = \min(0.8, 0.65)) = 0.65$
- $P \vee Q$: Either Mary or Ram is efficient i.e.
  - $T(P \vee Q) = \max(T(P), T(Q)) = \max(0.8, 0.65)) = 0.8$

Example

- P : Mary is efficient , $T(P) = 0.8$,
- Q : Ram is efficient $T(Q) = 0.65$,
- ¬P : Mary is not efficient,
  - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$
- $P \wedge Q$: Mary is efficient and so is Ram,
  - $T(P \wedge Q) = \min(T(P), T(Q)) = \min(0.8, 0.65)) = 0.65$
- $P \vee Q$: Either Mary or Ram is efficient i.e.
  - $T(P \vee Q) = \max(T(P), T(Q)) = \max(0.8, 0.65)) = 0.8$
- $P \Rightarrow Q$: If Mary is efficient then so is Ram,
  - $T(P \Rightarrow Q) =$

Example

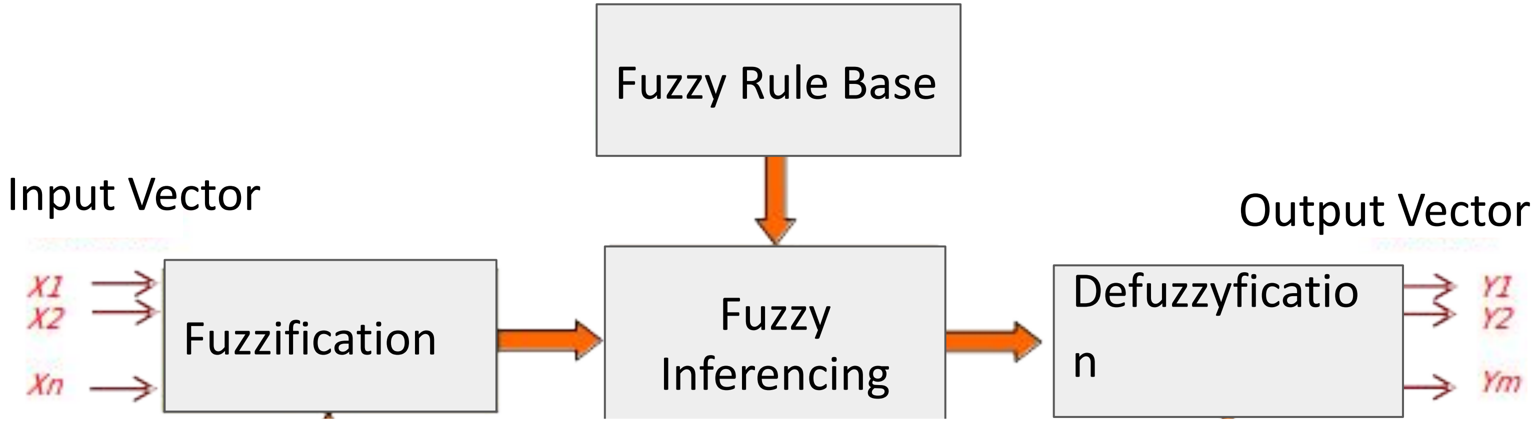
- P : Mary is efficient , $T(P) = 0.8$,
- Q : Ram is efficient $T(Q) = 0.65$,
- $\neg P$: Mary is not efficient,
  - $T(\neg P) = 1 - T(P) = 1 - 0.8 = 0.2$
- $P \wedge Q$: Mary is efficient and so is Ram,
  - $T(P \wedge Q) = \min (T(P), T(Q)) = \min (0.8, 0.65)) = 0.65$
- $P \vee Q$: Either Mary or Ram is efficient i.e.
  - $T(P \vee Q) = \max (T(P), T(Q)) = \max (0.8, 0.65)) = 0.8$
- $P \Rightarrow Q$: If Mary is efficient then so is Ram,
  - $T(P \Rightarrow Q) = \max (1 - T(P), T(Q)) = \max (0.2, 0.65)) = 0.65$

Fuzzy System Elements

Any system that uses Fuzzy mathematics may be viewed as Fuzzy system.

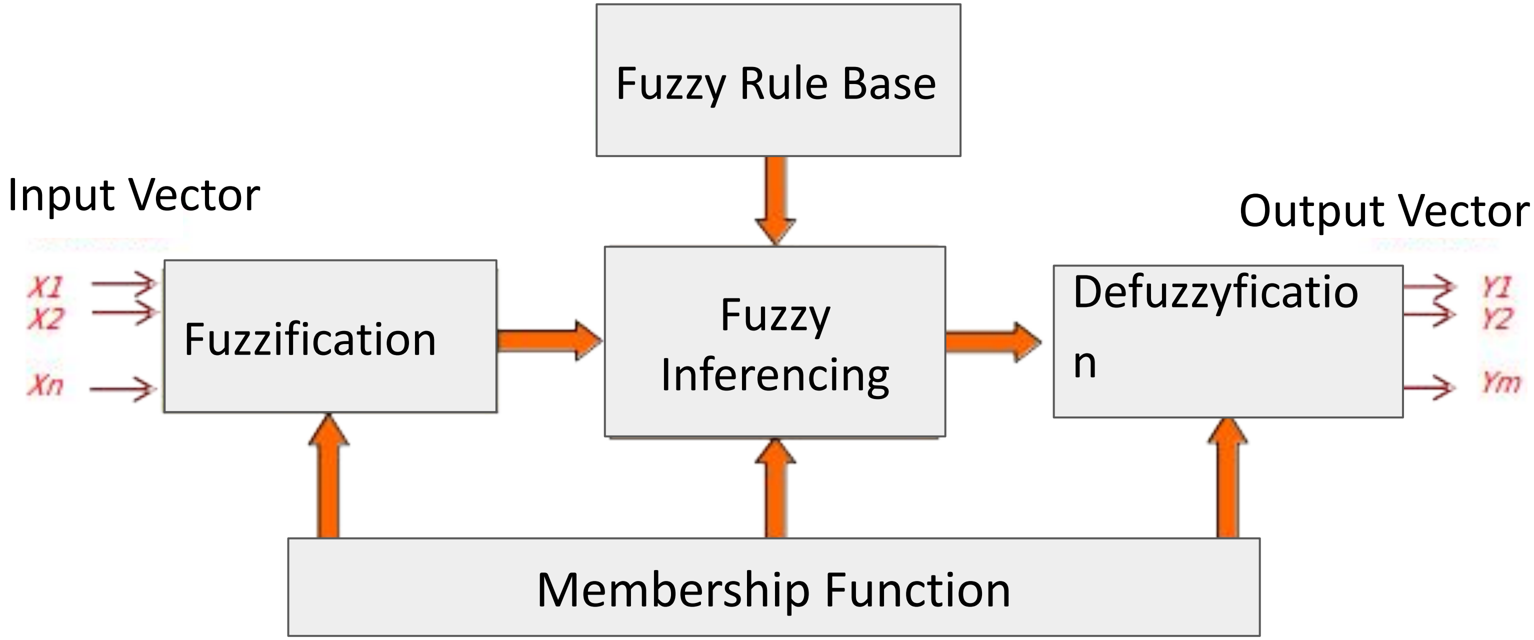
- **Fuzzification** : a process of transforming crisp values into grades of membership for linguistic terms, "far", "near", "small" of fuzzy sets.
- **Fuzzy Inferencing**: combines the facts obtained from the Fuzzification with the rule base and conducts the Fuzzy reasoning process.
- **Defuzzyfication**: Translate results back to the real world values.

Fuzzy System Elements



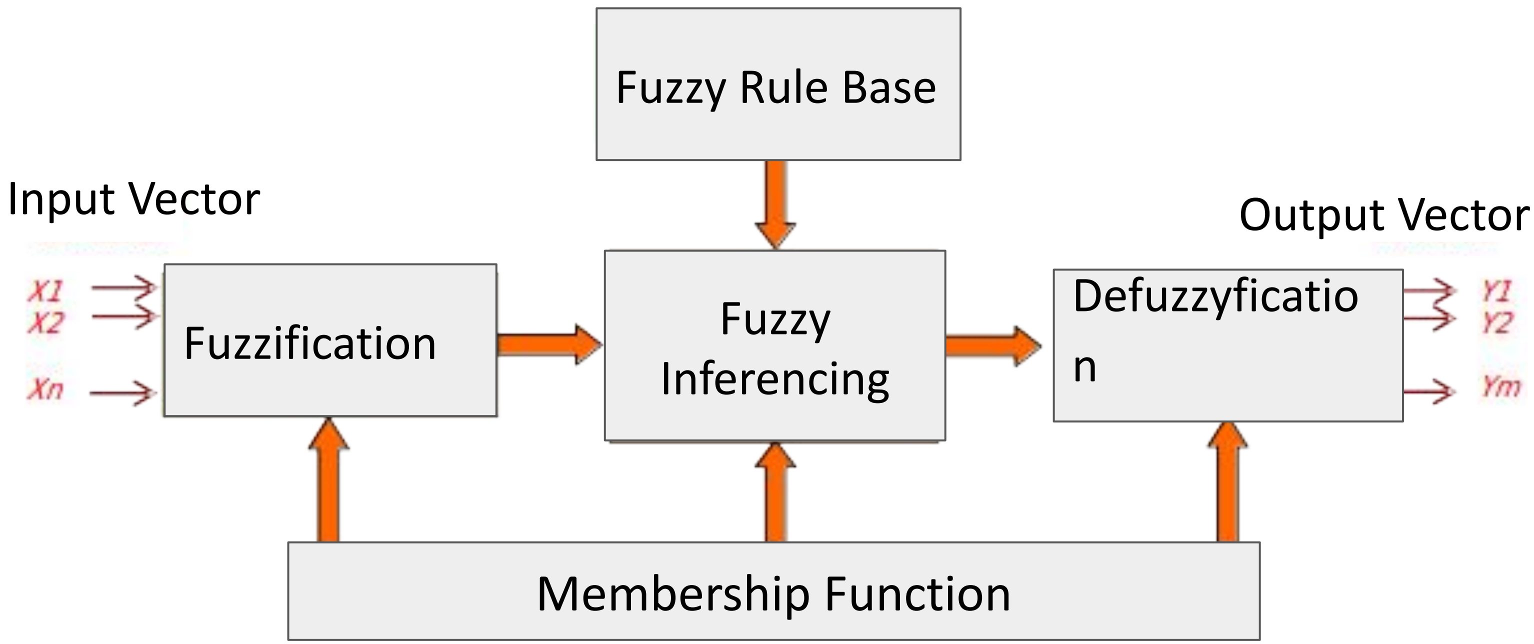
- **Fuzzy Rule base** : a collection of propositions containing linguistic variables; the rules are expressed in the form:
  - If (x is A) AND (y is B) . . . . . . THEN (z is C), where x, y and z represent variables (e.g. distance, size) and A, B and Z are linguistic variables (e.g. `far', `near', `small').

Fuzzy System Elements



- **Membership function**: provides a measure of the degree of similarity of elements in the universe of discourse U to fuzzy set.

Fuzzy System Elements



- **Input Vector :** X = [x1 , x2, . . . xn]^T are crisp values, which are transformed into fuzzy sets in the fuzzification block.
- **Output Vector** : Y = [y1 , y2, . . . ym]^T comes out from the defuzzification block, which transforms an output fuzzy set back to a crisp value.

Fuzzy System

- Fuzzy Systems can handle simultaneously the numerical data and linguistic knowledge and helps modeling of conditions which are inherently imprecisely defined.
- The applications of Fuzzy Systems are in Information retrieval systems, Navigation system, and Robot vision.

Ontology

Ontology

- **Ontology**: a set of *concepts and categories* in a subject area or domain that shows their properties and the relations between them.
- Ontology in **AI** helps in representing the knowledge about environment, events and actions that help in planning and making decisions by an AI agent.

Knowledge Engineering

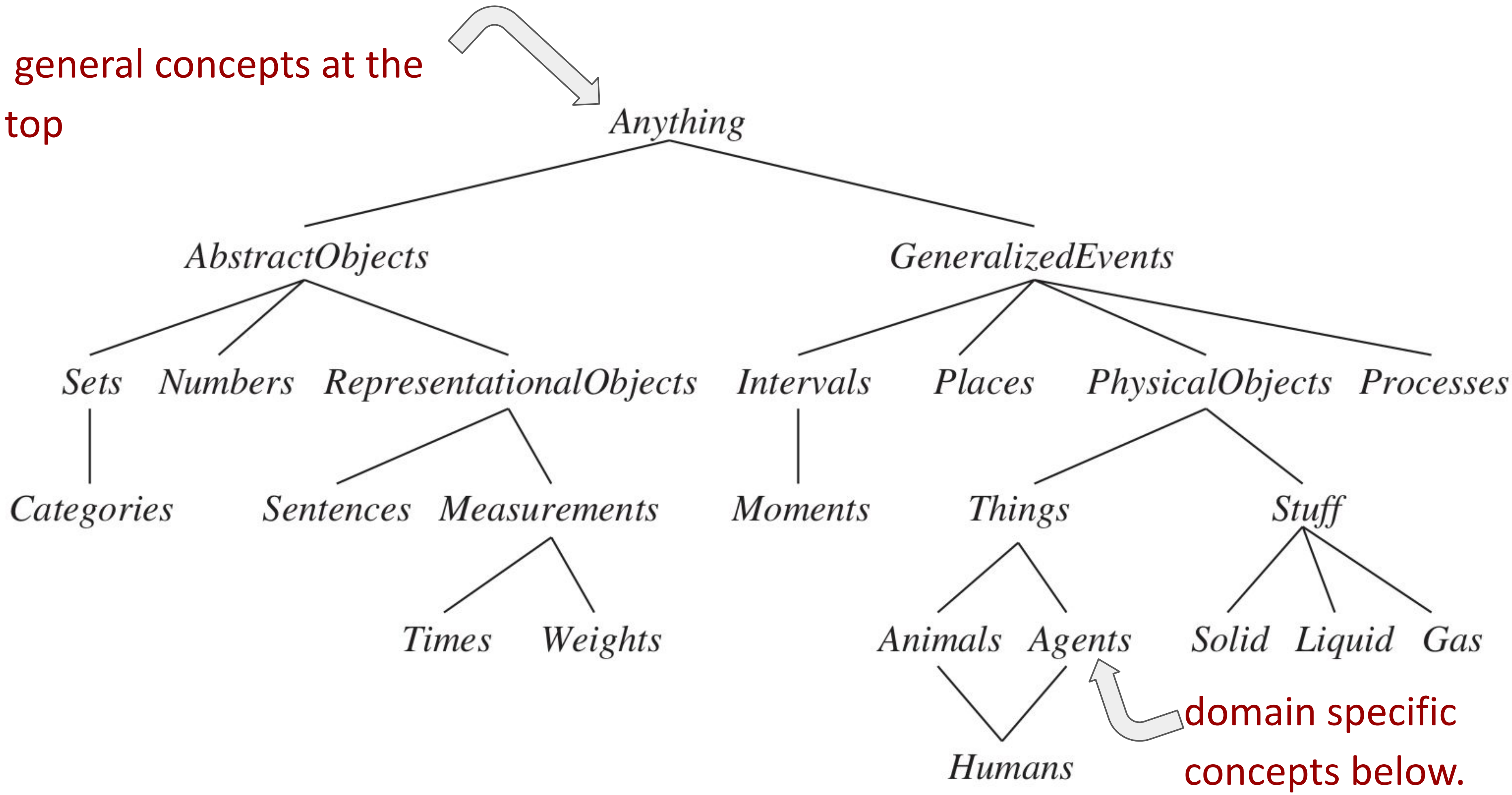
- **KE** refers to all technical, scientific and social aspects involved in building, maintaining and using **knowledge-based systems**.
  - What are the relevant facts, objects, relations ... ?
  - Which is the right level of abstraction?
  - What are the queries to the KB (inferences)?

Ontological Engineering

- The activity to build **general-purpose** ontologies Involving several areas of knowledge simultaneously
- Should be applicable in any special-purpose domain (with the addition of domain-specific axioms)
- Several attempts to build general-purpose ontologies: CYC (how the world works), DBpedia, TextRunner, etc.
- **not very successful so far**

Ontology in AI

The general framework of concepts is called an upper ontology. See example below:



Objects, Categories, Members, and Subclass

- KR requires the organisation of objects into categories
  - interaction at the level of the object
  - reasoning at the level of categories
  - ex: typically we want to buy a basketball, rather than a particular basketball instance

Objects, Categories, Members, and Subclass

- KR requires the organisation of objects into categories
  - interaction at the level of the object
  - reasoning at the level of categories
  - ex: typically we want to buy a basketball, rather than a particular basketball instance
- Categories play a role in predictions about objects
  - agent infers the presence of certain objects from perceptual input
  - infers category from the perceived properties of the objects,
  - uses category information to make predictions about the objects

Objects, Categories, Members, and Subclass

- Categories can be represented in two ways by FOL
  - predicates (ex Basketball(x)):
  - reification of categories into objects (ex Basketballs): sets
    $\Longrightarrow$ allows categories to be argument of predicates/functions
- Membership of a category as set membership
  - ex: $Member(b, Basketballs)$ (abbr. $b \in Basketballs$)
- Subcategories (aka subclasses) are (strict) subsets
  - ex: Subset(Basketballs, Balls) (abbr $Basketballs \subset Balls$)

Objects and Categories

Inheritance and Taxonomies

- A subcategory inherits the properties of the category
  - ex:
    if $\forall x.(x \in Food \rightarrow Edible(x))$, $Fruit \subset Food$, $Apples \subset Fruit$
    then $\forall x.(x \in Apple \rightarrow Edible(x))$
- A member inherits the properties of the category
  - if $a \in Apples$, then $Edible(a)$
- Subclass relation organize categories into taxonomies (aka taxonomic hierarchies)

Objects and Categories

FOL Reasoning about Categories

- FOL allows to state facts about categories:
  - an object is a member of a category
    $BB_9 \in Basketballs$
  - a category is a subclass of another category
    $Basketballs \subset Balls$
  - all members of a category have some properties
    $\forall x.(x \in Basketballs \rightarrow Spherical(x))$
  - members of a category can be recognized by some properties
    $\forall x.((Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5'' \wedge x \in Balls) \rightarrow x \in Basketballs)$
  - category as a whole has some properties
    $Dogs \in DomesticatedSpecies$

Objects and Categories

FOL Reasoning about Categories

- New categories can be defined by providing necessary and sufficient conditions for membership
  - $\forall x.(x \in Bachelors \leftrightarrow (Unmarried(x) \wedge x \in Adults \wedge x \in Males))$

Derived relations

- Two or more categories in a set s are disjoint iff they have no members in common
  - $$Disjoint(s) \leftrightarrow (\forall c_1 c_2.\ ((c_1 \in s \wedge c_2 \in s \wedge c_1 \neq c_2) \rightarrow Intersection(c_1, c_2) = \emptyset)$$
  - ex:
    $Disjoint(\{Animals, Vegetables\})$,
    $Disjoint(\{Insects, Birds, Mammals, Reptiles\})$,

Basic Relations for Categories

- $PartOf(.,.)$ relation: One object may be part of another
  - $PartOf(Bucharest, Romania)$
  - $PartOf(Romania, EasternEurope)$
  - $PartOf(EasternEurope, Europe)$
- $PartOf(.,.)$ is reflexive and transitive:
  - $\forall x.PartOf(x,x)$
  - $\forall x,y,z.((PartOf(x,y) \wedge PartOf(y,z)) \rightarrow PartOf(x,z))$

$\Longrightarrow$ $PartOf(Bucharest, Europe)$

Typical (.)

- Many categories have no clear-cut definition (ex: chair, bush, ...)
  - Ex: tomatoes are sometimes green, red, yellow, black; they are mostly round
- One useful solution: category "Typical(.)", s.t. $\mathit{Typical}(c) \subseteq c$

  $\Longrightarrow$ most knowledge about natural kinds will actually be about their typical instances
  - ex: $\forall x.(x \in \mathit{Typical}(\mathit{Tomatoes}) \rightarrow (\mathit{Red}(x) \wedge \mathit{Round}(x)))$

$\Longrightarrow$ We can write down useful facts about categories without providing exact definitions

Measurements

Quantitative Measurements

- Objects may have "quantitative" properties
  - e.g. height, mass, cost, ...
- Values that we assign to these properties are measures
- Can be represented by unit functions
  - ex $Length(L_1) = Inches(1.5) \wedge Inches(1.5) = Centimeters(3.81)$
- Conversion between units:
  - $\forall i.\ Centimeters(2.54 \times i) = Inches(i)$
- Measures can be used to describe objects:
  - ex: $Diameter(Basketball_{12}) = Inches(9.5)$
  - ex: $ListPrice(Basketball_{12}) = \$(19)$
  - ex: $\forall d.(d \in Days \rightarrow Duration(d) = Hours(24))$

Events

Events are described as instances of *event categories*

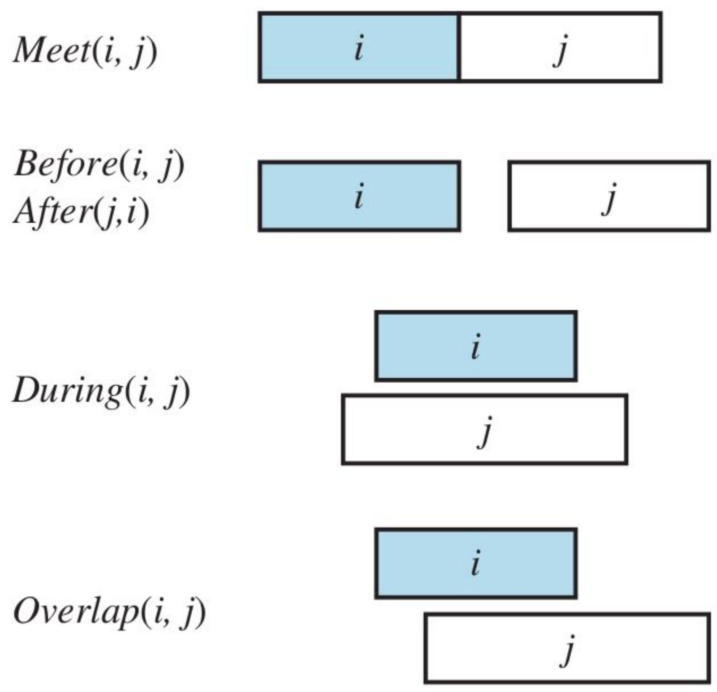
- For example, Event **E:**
    - E ∈ Flyings ∧ Flyer(E, John) ∧ Origin(E, Delhi) ∧ Destination(E, Jaipur)
- The event may or may not ongoing during a specific time t: Happens(E, t)
- The facts that are true only at specific time points are called fluents e.g., At(John, Jaipur)
    - T (True): asserts that a fluent is true at some point in time **t:** T(At(John, Jaipur), t)

Intervals

- Represent time in terms of intervals helps to represent many real world scenarios.
    - Flying(*P*, origin, destination, t) match different people P for different states as origin and destination, at different times t

Intervals

- Represent time in terms of intervals helps to represent many real world scenarios.
  - Flying(*P*, origin, destination, t) match different people P for different states as origin and destination, at different times t
- Predicates of Time interval helps to formulate the below:



Intervals

- Represent time in terms of intervals helps to represent many real world scenarios.
  - Flying(*P*, origin, destination, t) match different people P for different states as origin and destination, at different times t
- Predicates of Time interval helps to formulate the below:

$$
\begin{array}{lcl}
Meet(i,j) & \Leftrightarrow & End(i) = Begin(j) \\
Before(i,j) & \Leftrightarrow & End(i) < Begin(j) \\
After(j,i) & \Leftrightarrow & Before(i,j) \\
During(i,j) & \Leftrightarrow & Begin(j) < Begin(i) < End(i) < End(j) \\
Overlap(i,j) & \Leftrightarrow & Begin(i) < Begin(j) < End(i) < End(j)
\end{array}
$$

Objects v/s Stuff

- There are countable objects
  - e,g, apples, holes, theorems, ...
- ... and mass objects, aka stuff or substances
  - e.g. butter, water, energy, ...

$\Longrightarrow$ Intuitive meaning "an amount/quantity of..."
  - ex: $b \in butter$: "b is an amount/quantity of butter"
- Any part of stuff is still stuff:
  - ex: $\forall b, p.((b \in Butter \wedge PartOf(p, b)) \rightarrow p \in Butter)$

Objects v/s Stuff

- Can define sub-categories, which are stuff
  - ex: $UnsaltedButter \subset Butter$
- Stuff has a number of intrinsic properties, shared by its subparts
  - e.g., color, fat content, density ...
  - ex: $\forall b.(b \in Butter \rightarrow MeltingPoint(b, Centigrade(30)))$
- Stuff has no extrinsic properties
  - e.g., weight, length, shape, ...

Knowledge Representation

- Weak Slot and Filler Structures
- Strong Slot and Filler Structures

KR in Slots and Filler

- A slot is simply an attribute value pair
- A filler is a value that a slot can take

KR in Slots and Filler

- A slot is simply an attribute value pair
- A filler is a value that a slot can take

Weak Slot and Filler Structures

- A weak slot and filler structure are very general with no hard and fast rules for the representations of links between objects and relations.
- E.g., Semantic Nets and Frames

KR in Slots and Filler

Strong Slot and Filler Structures typically:

- Representation of links between objects and relations according to well defined notions.
- E.g., Conceptual Dependency and Scripts

Semantic Networks

Semantic networks is a directed or undirected graph consisting of

- Vertices: which represent concepts (e.g., category) or objects
- Edges: represent semantic relations between concepts or objects
- Labels – denoting particular objects and relations

Semantic Networks

- Two kinds of nodes:
  - Generic concepts, corresponding to categories/classes
  - Individual concepts, corresponding to individuals
- Two special relations are always present, with different names
  - IS-A, aka SubsetOf/SubclassOf (subclass)
  - InstanceOf, aka MemberOf (membership)

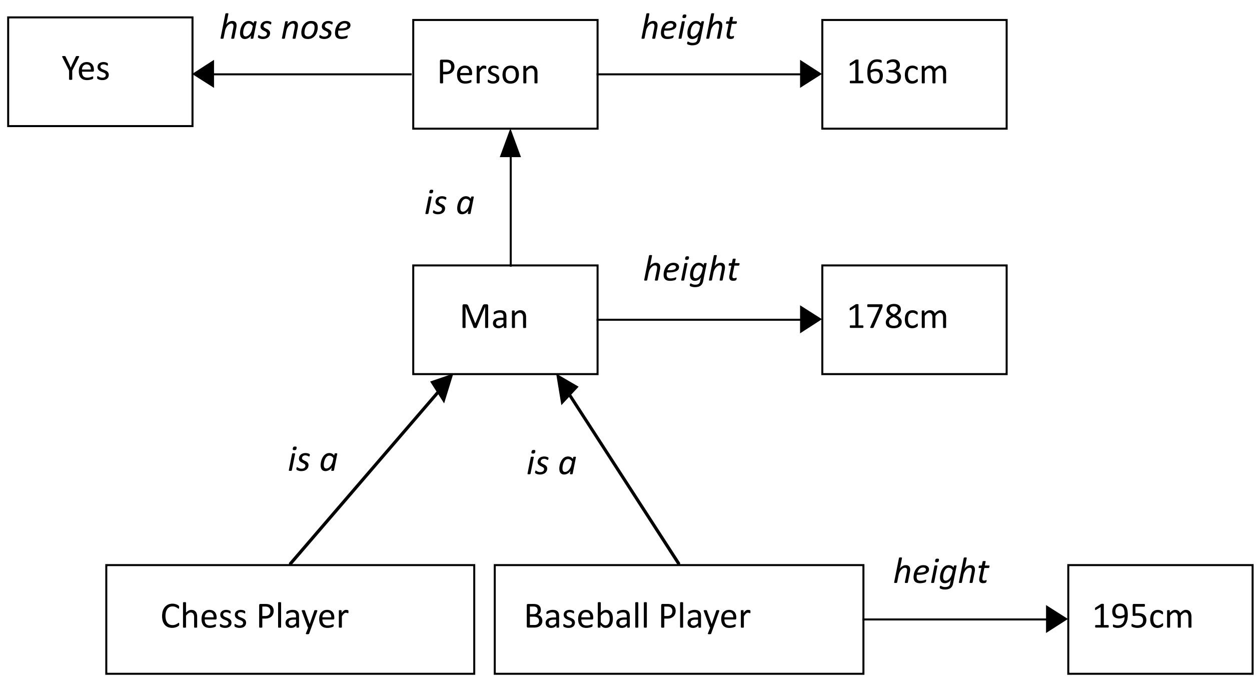
Semantic Networks

- Inheritance detection straightforward
- Ability to represent default values for categories
- Limited expressive power: cannot represent negation, disjunction, nested function symbols, existential quantification

Inheritance and Defaults

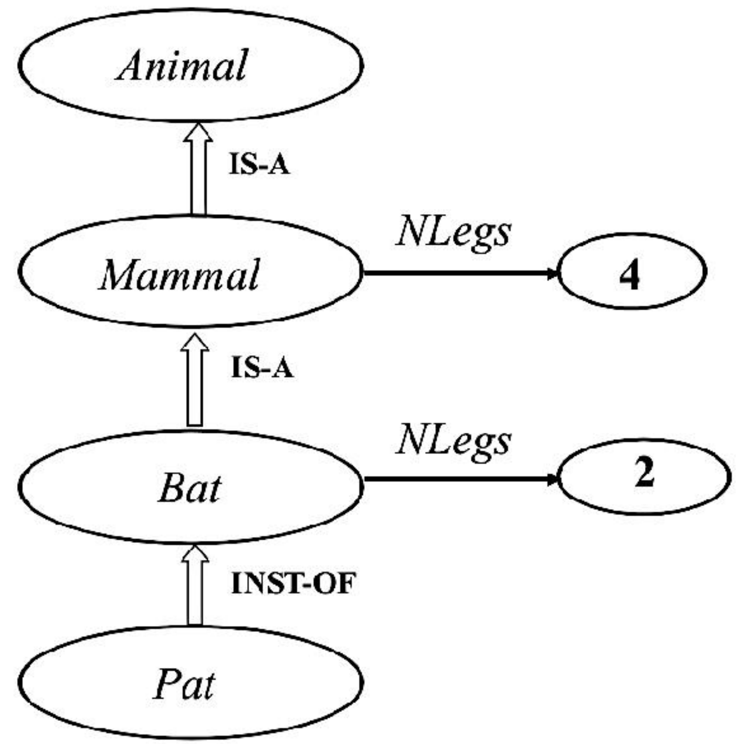
The figure shows the hierarchy of default values example.

- **I*s-a* corresponds to subset ⊆**



Inheritance with Exceptions

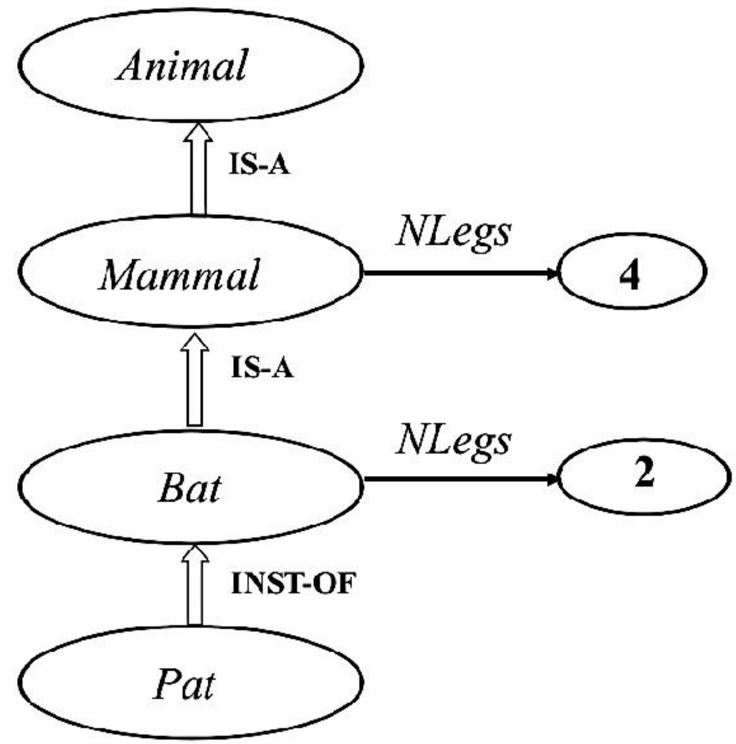
- **How many legs has Pat?**



(Most mammals have four limbs)

Inheritance with Exceptions

- **How many legs has Pat?**
- Just take the most specific information: the first that is found going up the hierarchy
  - ability to represent default values for categories



(Most mammals have four limbs)