



Knowledge Representation and Reasoning (KRR-2)

Dr. Nasima Begum Associate Professor Dept. of CSE, UAP

Knowledge Representation

- Propositional Logic
- Predicate Logic (chapter 8)
- Semantic Networks
- Frames
- Fuzzy Logic



■ Basic Concepts

- Logical expressions are built out of components:
 - ➤ Objects (Constants)
 - ➤ Variables
 - > Functions
 - > The above three are called **Terms**
- Predicates
- Connectives
- Quantifiers

Objects:

- Symbols that denote specific things or individuals:
 - **➢JOHN**
 - > MARY
 - **BASKETBALL**
 - **►**TRIANGLE
 - **▶** Quantifiers

□ Variables:

- Unspecific references to objects
 - $\triangleright x$
 - > y
 - > z

Functions:

- An argument to a function is either an object or a variable
 - Starting with a lowercase letter
- The value of a function is either an object or a variable

$$exp(0) = 1$$

$$next-day(THURSDAY) = FRIDAY$$

$$brother(JOHN) = JIM$$

- ☐ Predicates (or functions returning True/False):
 - Functions which denote attributes of objects or relationships between individuals
 - Starting with a uppercase letter

Loves(JOHN, MARY)

Man(SOCRATES)

Sunny(THURDAY)

□ Logical Connectives:

- ➤AND (^)
- **>**OR ([∨])
- ➤ NOT (~)
- \triangleright IMPLIES (\rightarrow)

Quantifiers:

- Logical operators which assert the scope of a predicate
 - ∀ For All (universal quantifier)
 - ∃ There Exists (existential quantifier)

- In predicate logic the basic unit is a predicate/argument structure called an atomic sentence:
 - LIKES (azad, chocolate)
 - TALL (habib)
- Arguments can be any of:
 - constant symbol, such as 'azad'
 - variable symbol, such as x
 - function expression, e.g., FATHER_OF (hasan)
- So we can have:
 - LIKES (X, chocolate)
 - FRIENDS (FATHER_OF (rita), FATHER_OF (choiti))

Syntax of Predicate Logic

- These atomic sentences can be combined using logic connectives
 - LIKES (rita, hasan) ∧ TALL (hasan)
 - BASKET_BALL_PLAYER (hasan) ⇒ TALL (hasan)
- Sentences can also be formed using quantifiers
 - $\forall x$ LOVELY (x) Everything is lovely.
 - $\exists x \text{ LOVELY } (x)$ Something is lovely.
 - $\forall x$ IN $(x, \text{garden}) \Rightarrow \text{LOVELY } (x)$ Everything in the garden is lovely.

Predicate Logic: Examples

- All employees earning TK. 30,000 or more per year, pay taxes.
- $\forall x ((E(x) \land GE (I(x), 30000)) \rightarrow T(x)$
- Some employees are sick today
- $\exists y ((E(y) \rightarrow S(y))$
- No employee earns more than the president
- $\forall x \forall y ((E(x) \land P(y)) \rightarrow \neg GE((I(x), I(y)))$

Examples of Predicate Logic

- Can have several quantifiers, e.g.,
 - $\forall x \exists y (LOVES (x, y))$
 - $\forall x \text{ (HANDSOME } (x) \Rightarrow \exists y \text{ (LOVES } (y, x)))$
- So we can represent things like:
 - All men are mortal.
 - No one likes hartal.
 - Everyone taking AI will pass their exams.
 - Every race has a winner.
 - Sajjad likes everyone who is tall.
 - Rita doesn't like anyone who prefers arguments.
 - There is something small and slimy on the table.



Examples of Predicate Logic

- Tony, Mike, and John are members of the Alpine Club.
- Every member of the Alpine Club who is not a skier is a mountain climber.
- Mountain climbers do not like rain, and anyone who does not like snow is not a skier.
- Mike dislikes whatever Tony likes, and likes whatever Tony dislikes.
- Tony likes rain and snow.
- Is there a member of the Alpine Club who is a mountain climber but not a skier?

- Individuals: Constants (0-ary Functions)
 - tony, mike, john
 - rain, snow
- Types: Unary Predicates
 - -AC(x): x belongs to Alpine Club.
 - -S(x): x is a skier.
 - -C(x): x is a mountain climber.
- Relationships: Binary Predicates
 - -L(x; y): x likes y.

Basic Facts:

- Tony, Mike, and John belong to the Alpine Club:

```
AC(tony); AC(mike); AC(john)
```

– Tony likes rain and snow:

```
L(tony, rain); L(tony, snow)
```

Complex Facts:

Every member of the Alpine Club who is not a skier is a mountain climber.

$$\forall x [AC(x) \land \neg S(x) \Rightarrow C(x)]$$

Mountain climbers do not like rain, and anyone who does not like snow is not a skier

$$\forall x [C(x) \Rightarrow ^{\sim} L(x, rain)] \land \forall x [^{\sim} L(x, snow) \Rightarrow ^{\sim} S(x)]$$

Mike dislikes whatever Tony likes, and likes whatever Tony dislikes.

$$\forall x [L (tony, x) \Rightarrow ^{\sim} L (mike, x)] ^{\wedge} \forall x [^{\sim} L (tony, x) \Rightarrow L (mike, x)]$$

– Is there a member of the Alpine Club who is a mountain climber but not a skier?

(There are some member of the AC who is mountain climber but not a skier)

$$\exists x [AC(x) \land C(x) \land \neg S(x)]$$

- Whoever can read is literate.

$$\forall x [read (x) \Rightarrow lit (x)]$$

- Dolphins are not literate.

$$\forall x [dolp (x) \Rightarrow ^{\sim} lit (x)]$$

- Flipper is an intelligent dolphin.

- Who is intelligent but cannot read?

(Whoever that is intelligent but cannot read is the answer)

$$\forall x [(intell (x) ^ \sim read (x)) \Rightarrow answer (x)]$$

Semantics of Predicate Logic

- There is a precise meaning to expressions in predicate logic.
- Like in propositional logic, it is all about determining whether something is true or false.
- $\forall x \ P(x)$ means that P(x) must be true for every object x in the domain of interest.
- $\exists xP(x)$ means that P(x) must be true for at least one object x in the domain of interest.
- So if we have a domain of interest consisting of just two people, Hasan and Belal, and we know that TALL(hasan) and TALL(belal) are true, we can say that ∀x TALL(x) is true.

Problem of FOPL

- It is difficult to represent the following using predicate logic:
 - **≻**Time
 - ➤ Beliefs and
 - **≻**Uncertainty

Knowledge Representation

- Propositional Logic
- Predicate Logic
- Semantic Networks (chapter-12)
- Frames
- Fuzzy Logic



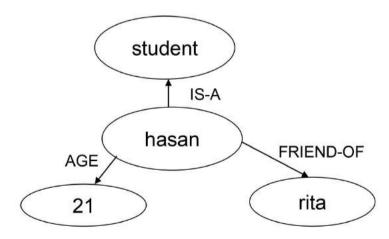
Semantic Network

- Semantic net is a knowledge representation technique which is used to represent knowledge by some nodes and arcs where:
 - i) Nodes → represent concepts/objects and
 - ii) Arcs → represent relationship

- An long existing notion: there are different pieces of knowledge of world, and they are all linked together through certain semantics.
- Knowledge is expressed as a collection of concepts, represented by nodes (shown as boxes in the diagram), connected together by relationships, represented by arcs (shown as arrows in the diagram).
- Certain arcs particularly IS-A arcs allow inheritance of properties.

Basic Components:

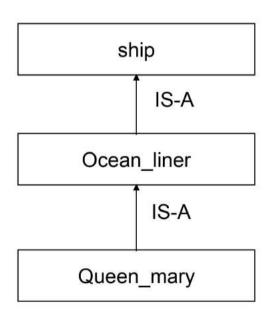
- Nodes
 - -Represent concepts
- Arcs
 - -Represent relations
- Labels for nodes and arcs



- Common arcs used for representing hierarchies include IS-A and has-part.
 - Example:

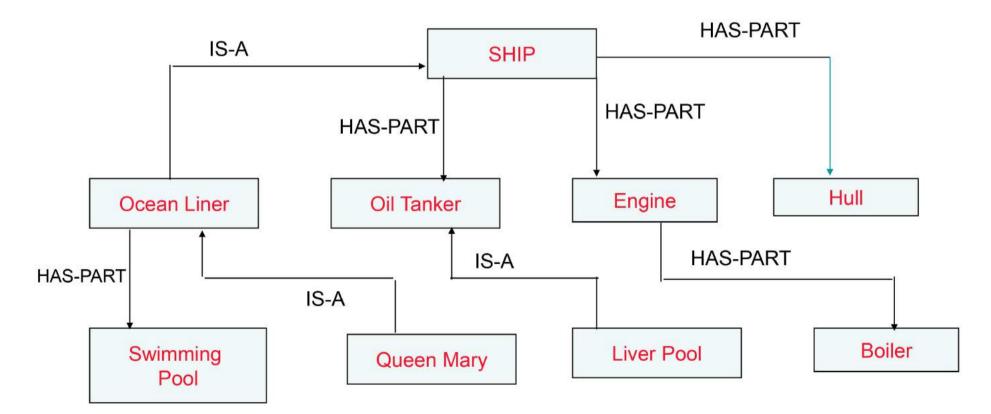
The Queen Mary is an ocean liner.

Every ocean liner is a ship



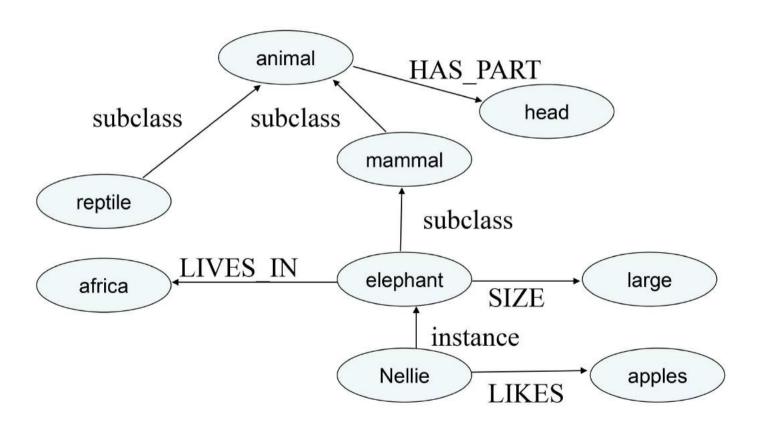
Semantic Net ...

Common arcs used for representing hierarchies include is-a and has-part.



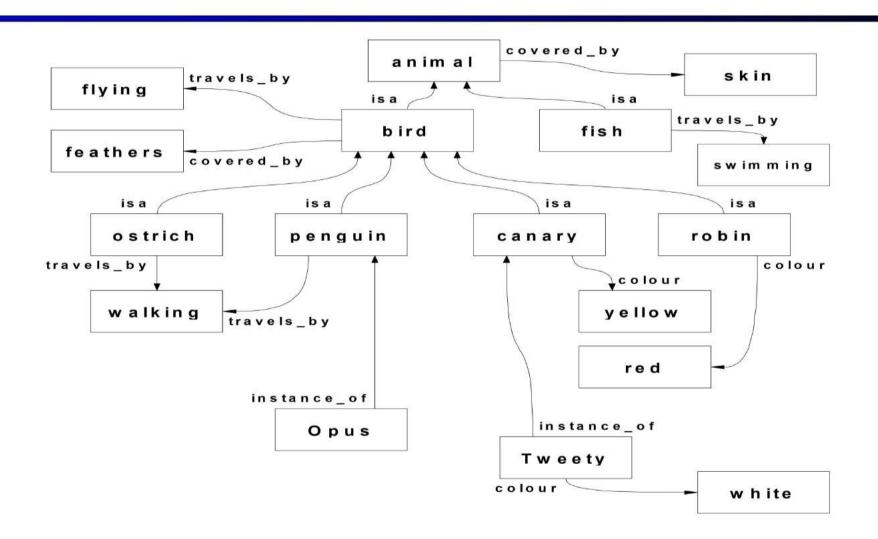
Semantic Networks

Knowledge is represented as a network or a graph



An long existing notion:

There are different pieces of knowledge of world, and they are all linked together through certain semantics.



- Developments of the semantic nets idea:
 - psychological research into whether human memory really was organised in this way.
 - used in the knowledge bases in certain expert systems: e.g. PROSPECTOR.
 - special-purpose languages have been written to express knowledge in semantic nets.

Organization of Knowledge

- By traversing network we can find:
 - That Nellie has a head (by inheritance)
 - That certain concepts related in certain ways (e.g., apples and elephants).
- BUT: Meaning of semantic networks was not always well defined.
 - Are all Elephants big, or just typical elephants?
 - Do all Elephants live in the "same" Africa?
 - Do all animals have the same head?
- For machine processing these things must be defined.

Limitations/Problems

Lack of Semantics

- No formal semantic of the relations
 - E.g. Does "IS-A" mean subclass, member, etc?
- Possible multiple interpretations
- Restricted expressiveness
 - E.g. can not distinguish between instance and class

Advantages:

Easy to follow hierarchy, easy to trace association, flexible

Disadvantages:

- Meaning attached to nodes might be ambiguous
- Exception handling is difficult
- Difficult to program

Semantic Nets ...

Problems with semantic nets

- logical inadequacy vagueness about what types and tokens really mean.
- heuristic inadequacy finding a specific piece of information could be chronically inefficient.
- trying to establish negation is likely to lead to a combinatorial explosion.
- "spreading activation" search is very inefficient, because it is not knowledge-guided.

Acknowledgement

- AIMA = Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norving (3rd edition)
- UC Berkeley (Some slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley)
- U of toronto
- Other online resources

Thank You