UNIT V: REASONING

TE COMPUTER ENGINEERING

REFERENCE:

"ARTIFICIAL INTELLIGENCE:
A MODERN APPROACH"
STUART J. RUSSELL AND PETER NORVIG

INDEX

- Inference in FOL
- Unification
- Forward and Backword chaining
- Resolution
- Ontological Engineering

UNIVERSAL INSTANTIATION (UI)

- some simple inference rules that can be applied to sentences with quantifiers to obtain sentences without quantifiers. These rules lead naturally to the idea that First-order inference can be done by converting the knowledge base to propositional logic and using propositional inference
- Let us begin with universal quantifiers. Suppose our knowledge base contains the standard folkloric axiom stating that all greedy kings are evil:

$$\forall x\ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

- The rule of **Universal Instantiation** (UI for short) says that we can infer any sentence obtained by substituting a ground term (a term without variables) for the variable.

REDUCTION TO PROPOSITIONAL INFERENCE

- Once we have rules for inferring nonquantified sentences from quantified sentences, it becomes possible to reduce first-order inference to propositional inference
- an existentially quantified sentence can be replaced by one instantiation,
- a universally quantified sentence can be replaced by the set of all possible instantiations.

REDUCTION TO PROPOSITIONAL INFERENCE

- For example, suppose our knowledge base contains just the sentences
  - $\forall x\ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$
  - $King(John)$
  - $Greedy(John)$
  - $Brother(Richard, John)$
- Instantiating the universal sentence in all possible ways, we have:
  - $King(John) \wedge Greedy(John) \Rightarrow Evil(John)$
  - $King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$
- The new KB is propositionalized: propositional symbols are
  - $King(John), King(Richard), Greedy(John), Evil(John)$ etc…

UNIFICATION

- We can get the inference immediately if we can find a substitution θ such that King(x) and Greedy(x) match King(John) and Greedy(y)

$$\theta = \{x/John, y/John\} \text{ works}$$

- Unify(α,β) = θ if αθ = βθ

p	q	θ
Knows(John,x)	Knows(John,Jane)	
Knows(John,x)	Knows(y,OJ)	
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,OJ)	

Standardizing apart eliminates overlap of variables, e.g., Knows(z17,OJ)

UNIFICATION

- We can get the inference immediately if we can find a substitution θ such that King(x) and Greedy(x) match King(John) and Greedy(y)

$$\theta = \{x/John, y/John\} \text{ works}$$

- Unify(α,β) = θ if αθ = βθ

p	q	θ
Knows(John,x)	Knows(John,Jane)	{x/Jane}
Knows(John,x)	Knows(y,OJ)	
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,OJ)	

Standardizing apart eliminates overlap of variables, e.g., Knows(z17,OJ)

UNIFICATION

- We can get the inference immediately if we can find a substitution θ such that King(x) and Greedy(x) match King(John) and Greedy(y)

$$\theta = \{x/John, y/John\} \text{ works}$$

- Unify(α,β) = θ if αθ = βθ

p	q	θ
Knows(John,x)	Knows(John,Jane)	{x/Jane}
Knows(John,x)	Knows(y,OJ)	{x/OJ,y/John}
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,OJ)	

Standardizing apart eliminates overlap of variables, e.g., Knows(z_{17},OJ)

UNIFICATION

- We can get the inference immediately if we can find a substitution θ such that King(x) and Greedy(x) match King(John) and Greedy(y)

$$\theta = \{x/John, y/John\} \text{ works}$$

- Unify(α,β) = θ if αθ = βθ

p	q	θ
Knows(John,x)	Knows(John,Jane)	{x/Jane}
Knows(John,x)	Knows(y,OJ)	{x/OJ,y/John}
Knows(John,x)	Knows(y,Mother(y))	{y/John,x/Mother(John)}
Knows(John,x)	Knows(x,OJ)	

Standardizing apart eliminates overlap of variables, e.g., Knows(z17,OJ)

UNIFICATION

- We can get the inference immediately if we can find a substitution θ such that King(x) and Greedy(x) match King(John) and Greedy(y)

θ = {x/John,y/John} works

- Unify(α,β) = θ if αθ = βθ

p	q	θ
Knows(John,x)	Knows(John,Jane)	{x/Jane}
Knows(John,x)	Knows(y,OJ)	{x/OJ,y/John}
Knows(John,x)	Knows(y,Mother(y))	{y/John,x/Mother(John)}
Knows(John,x)	Knows(x,OJ)	{fail}

Standardizing apart eliminates overlap of variables, e.g., Knows(z17,OJ)

INFERENCE RULES

- patterns of inference that can be applied to derive chains of conclusions that lead to the desired goal. These patterns of inference are called inference rules. The best-known rule is called Modus Ponens and is written as follows:

$$\frac{\alpha \rightarrow \beta,\ \alpha}{\beta}$$

- The notation means that, whenever any sentences of the form $\alpha \rightarrow \beta$, α are given, then the sentence β can be inferred.
- For example, if (WumpusAhead ∧ WumpusAlive) → Shoot means: (WumpusAhead ∧ WumpusAlive) are given,

  then **Shoot** can be inferred.

FORWARD CHAINING

- It is a down-up approach, as it moves from bottom to top.
- It is a process of making a conclusion based on known facts or data, by starting from the initial state and reaches the goal state.
- also called as data-driven as we reach to the goal using available data.
- Forward -chaining approach is commonly used in the expert system, such as CLIPS, business, and production rule systems.

FORWARD CHAINING

- Start with atomic sentences in the KB and apply **Modus Ponens** in the **forward** direction, adding new atomic sentences, until no further inferences can be made.
- Definite clauses such as Situation $\Rightarrow$ Response are especially useful for systems that make inferences in response to newly arrived information.
- Many systems can be defined this way, and reasoning with forward chaining can be much more efficient than resolution theorem proving.
- Therefore it is often worthwhile to try to build a knowledge base using only definite clauses so that the cost of resolution can be avoided.

FC: EXAMPLE KNOWLEDGE BASE

- The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy America, has some missiles, and all of its missiles were sold to it by Col. West, who is an American.
- Prove that Col. West is a criminal.

FORWARD CHAINING

- Forward Chaining: Start with atomic sentences in the KB and apply Modus Ponens in the forward direction, adding new atomic sentences, until no further inferences can be made.
  - Start with the known fact
  - Triggers all the rules
  - Keep going until the desired fact is generated
  - Note: a fact is not "new" if it is just a renaming of known fact

FC: EXAMPLE KNOWLEDGE BASE

- …it is a crime for an American to sell weapons to hostile nations

  $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$

- Nono…has some missiles

  $\exists x\ Owns(Nono, x) \wedge Missiles(x)$

  $Owns(Nono, M_1)$ *and* $Missle(M_1)$

- …all of its missiles were sold to it by Col. West

  $\forall x\ Missle(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$

- Missiles are weapons

  $Missle(x) \Rightarrow Weapon(x)$

FC: EXAMPLE KNOWLEDGE BASE

- An enemy of America counts as “hostile”

  $Enemy(\ x,\ America\ ) \Rightarrow Hostile(x)$

- Col. West who is an American

  $American(\ Col.West\ )$

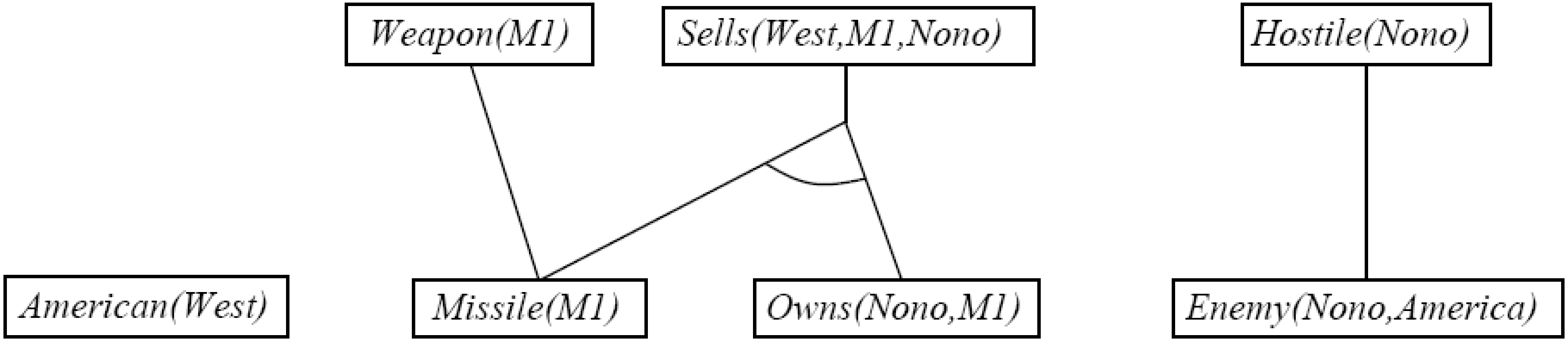
- The country Nono, an enemy of America

  $Enemy(Nono,\ America)$

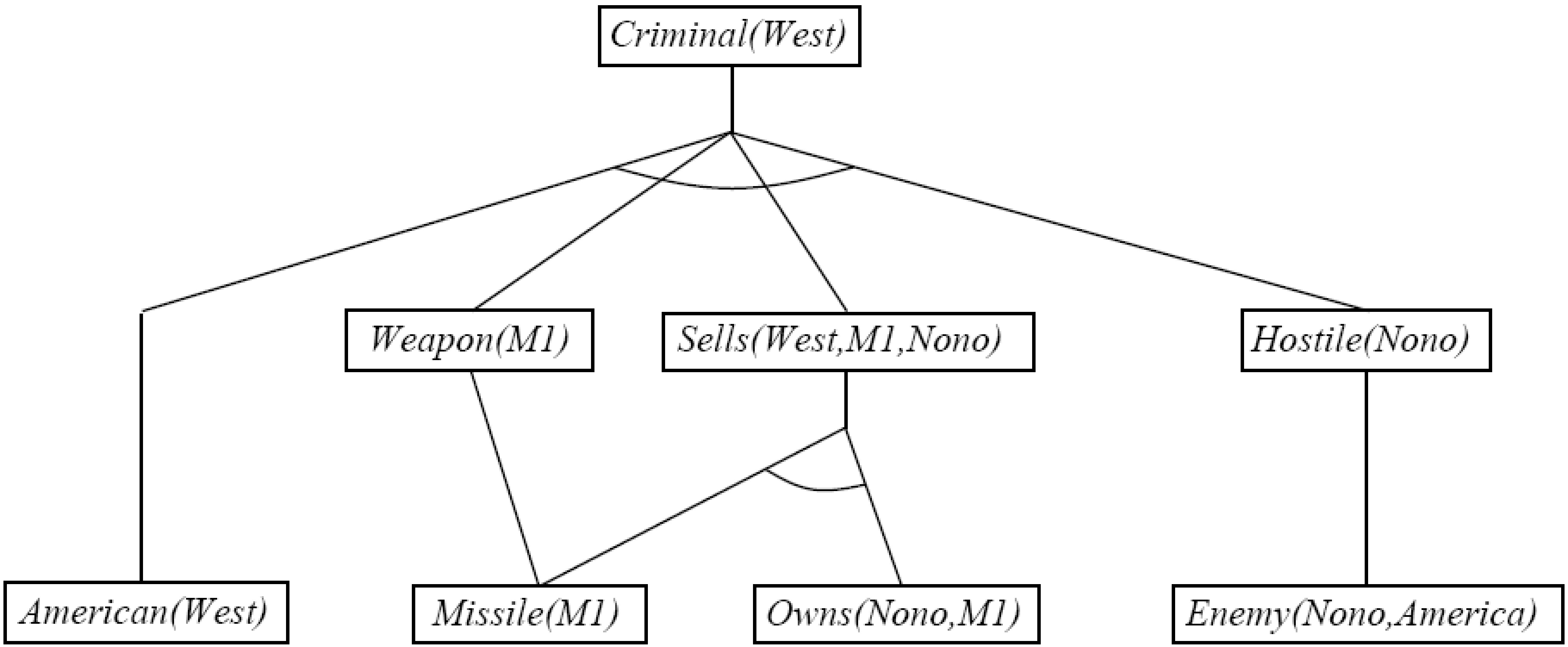
FC: EXAMPLE KNOWLEDGE BASE

$American(West)$ $Missile(M1)$ $Owns(Nono,M1)$ $Enemy(Nono,America)$

FC: EXAMPLE KNOWLEDGE BASE



FC: EXAMPLE KNOWLEDGE BASE



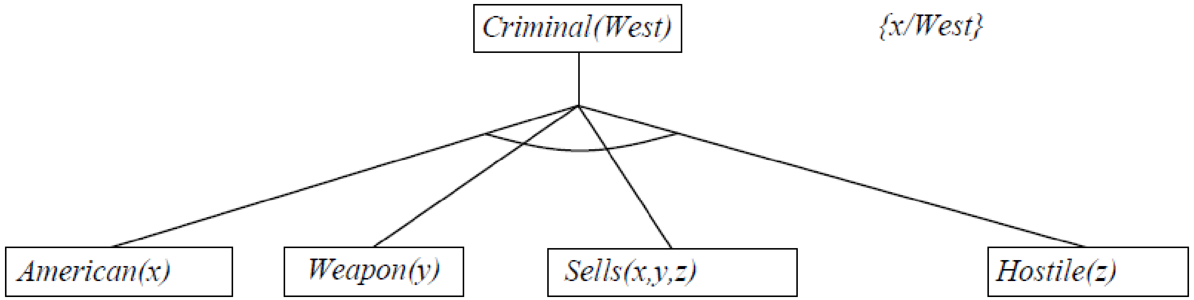
BACKWARD CHAINING

- Backward-chaining : It is known as a top-down approach.
- is based on modus ponens inference rule.
- the goal is broken into sub-goal or sub-goals to prove the facts true.
- It is called a goal-driven approach, as a list of goals decides which rules are selected and used.
- used in game theory, automated theorem proving tools etc..
- mostly used a **depth-first search** strategy for proof.

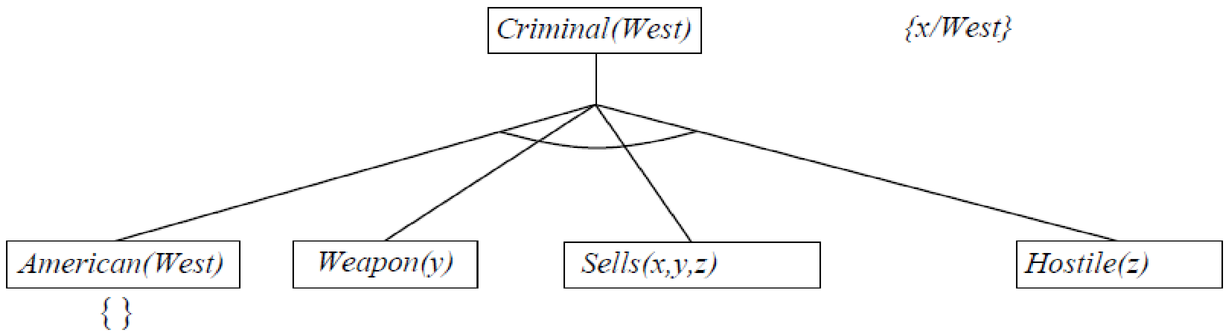
BACKWARD CHAINING

$\boxed{Criminal(West)}$

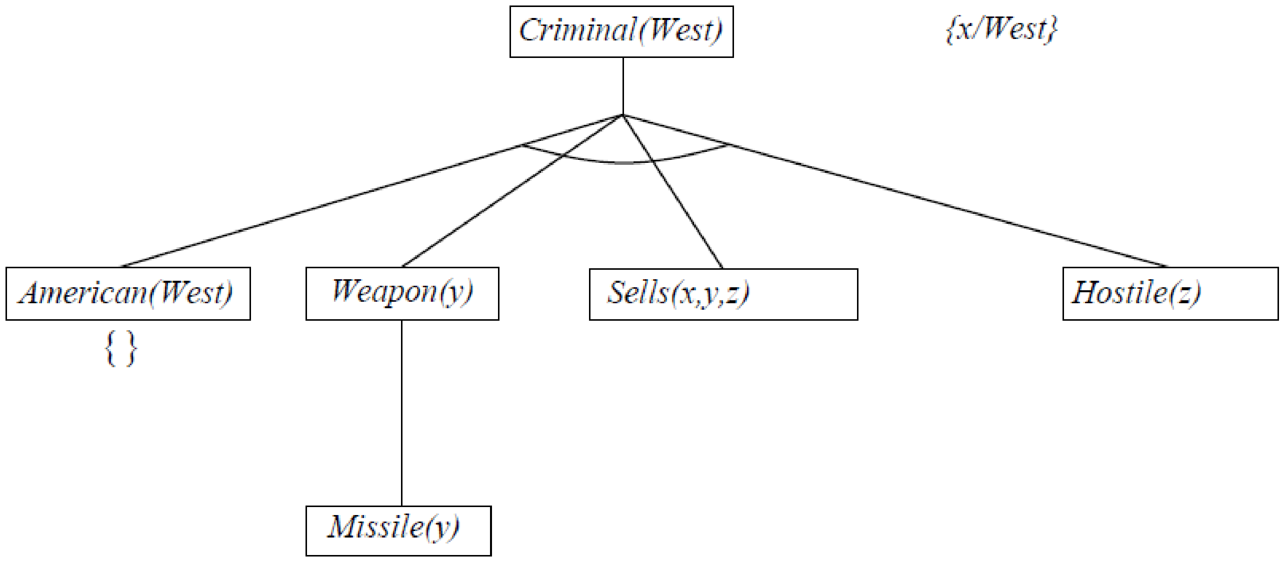
BACKWARD CHAINING



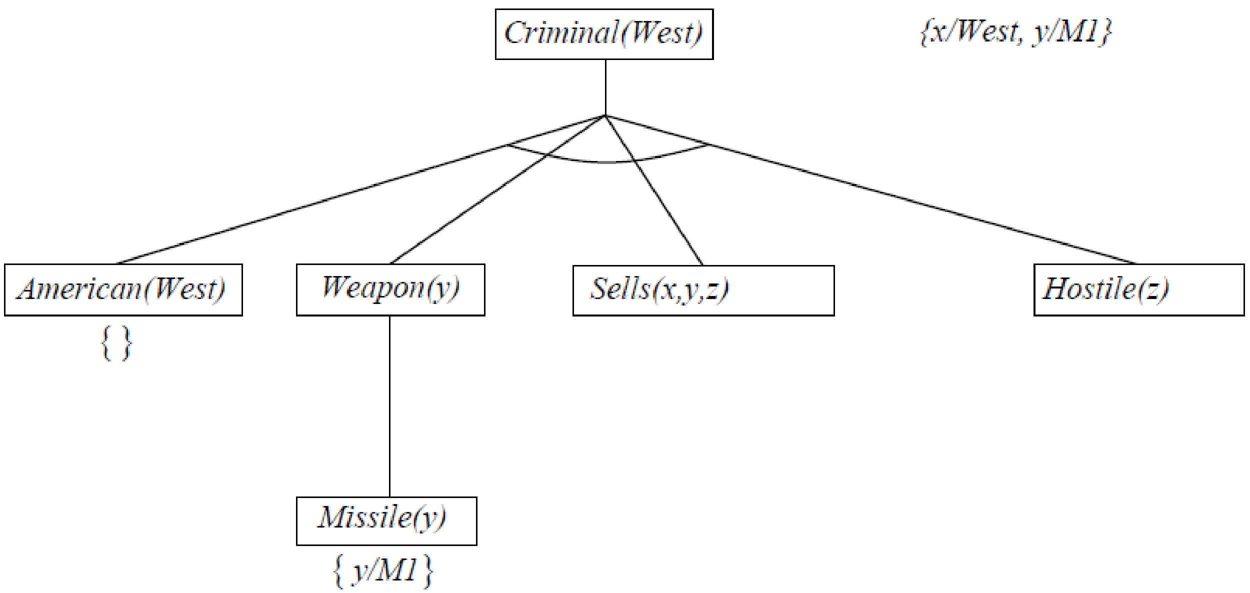
BACKWARD CHAINING



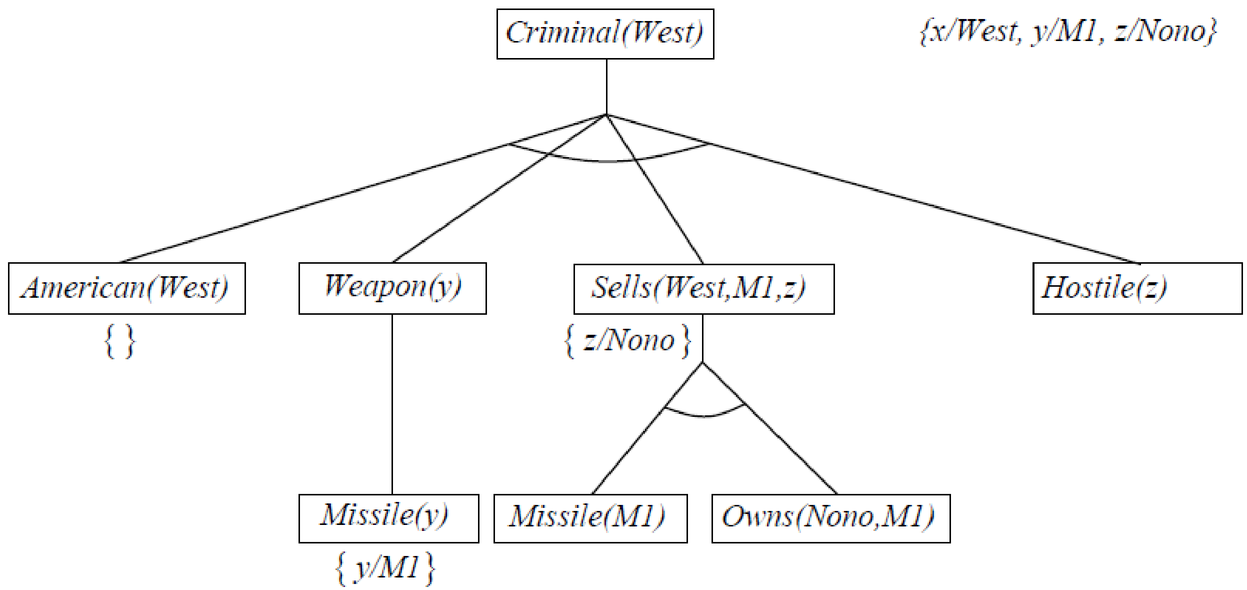
BACKWARD CHAINING



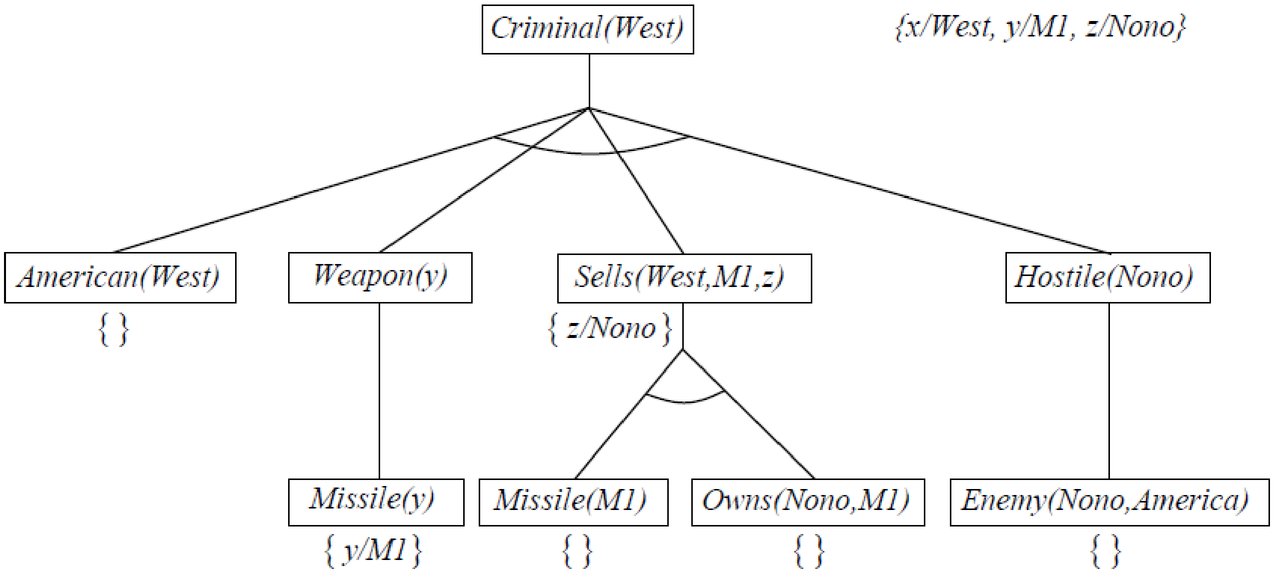
BACKWARD CHAINING



BACKWARD CHAINING



BACKWARD CHAINING



RESOLUTION

- Resolution is a valid inference rule producing a new clause implied by two clauses containing complementary literals
- A literal is an atomic symbol or its negation, i.e., P, ~P
- Amazingly, this is the only interference rule you need to build a sound and complete theorem prover
- The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 60s

RESOLUTION

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.
- To use resolution, put KB into conjunctive normal form (CNF),
- A sentence expressed as a conjunction of disjunctions of literals is said to be in conjunctive normal form.
- Conversion to CNF by using the following equivalences,

$$A \leftrightarrow B = (A \to B) \wedge (B \to A)$$

$$A \to B = \neg A \vee B$$

$$\neg(A \wedge B) = \neg A \vee \neg B$$

$$\neg(A \vee B) = \neg A \wedge \neg B$$

$$\neg\neg A = A$$

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

LOGICAL EQUIVALENCE

- Two sentences are logically equivalent iff true in same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

EXAMPLE

- For example, we will convert $(A \rightarrow B) \rightarrow C$ to CNF:

  $(A \rightarrow B) \rightarrow C$

  $\neg(A \rightarrow B) \lor C$

  $\neg(\neg A \lor B) \lor C$

  $(A \land \neg B) \lor C$

  $(A \lor C) \land (\neg B \lor C)$

- $A \leftrightarrow (B \land C)$

  $(A \rightarrow (B \land C)) \land ((B \land C) \rightarrow A)$

  $(\neg A \lor (B \land C)) \land (\neg(B \land C) \lor A)$

  $(\neg A \lor (B \land C)) \land (\neg B \lor \neg C \lor A)$

  $(\neg A \lor B) \land (\neg A \lor C) \land (\neg B \lor \neg C \lor A)$

RESOLUTION EXAMPLES (PROPOSITIONAL)

- Example

  $A \vee (B \vee C)$

  $(\neg A)$

  $(B \vee C)$

RESOLUTION IN FOL

- The resolution rule for first-order clauses is simply a lifted version of the propositional resolution.
  - Two clauses, which are assumed to be standardized apart so that they share no variables, can be resolved if they contain complementary literals.
  - Propositional literals are complementary if one is the negation of the other; first-order literals are complementary if one unifies with the negation of the other. Thus we have

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\textsc{Subst}(\theta, \ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)}$$

RESOLUTION IN FOL

- Contd.…

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\textsc{Subst}(\theta, \ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)}$$

- where Unify(l_i , $\neg m_j$) = θ.
- The two clauses are assumed to be standardized apart so that they share no variables.
- For example: $\dfrac{\neg Rich(x) \vee Unhappy(x) \quad Rich(Ken)}{Unhappy(Ken)}$
- with θ = {x/Ken}
- Apply resolution steps to CNF(KB ∧ ¬ α); complete for FOL

RESOLUTION IN FOL

- Original sentence: Everyone who loves all animals is loved by someone:

  $\forall x\ [\forall\ y\ Animal(y)\ \rightarrow Loves(x, y)] \rightarrow [\exists\ y\ Loves(y, x)]$

- The steps are as follows:
- Eliminate implications:

  $\forall x\ [\neg\ \forall y\ \neg Animal(y)\ \vee Loves(x, y)] \rightarrow [\exists\ y\ Loves(y, x)]$

- Move $\neg$ inwards:
  - $\neg\ \forall x\ \mathbf{p}$ becomes $\exists x\ \neg \mathbf{p}$

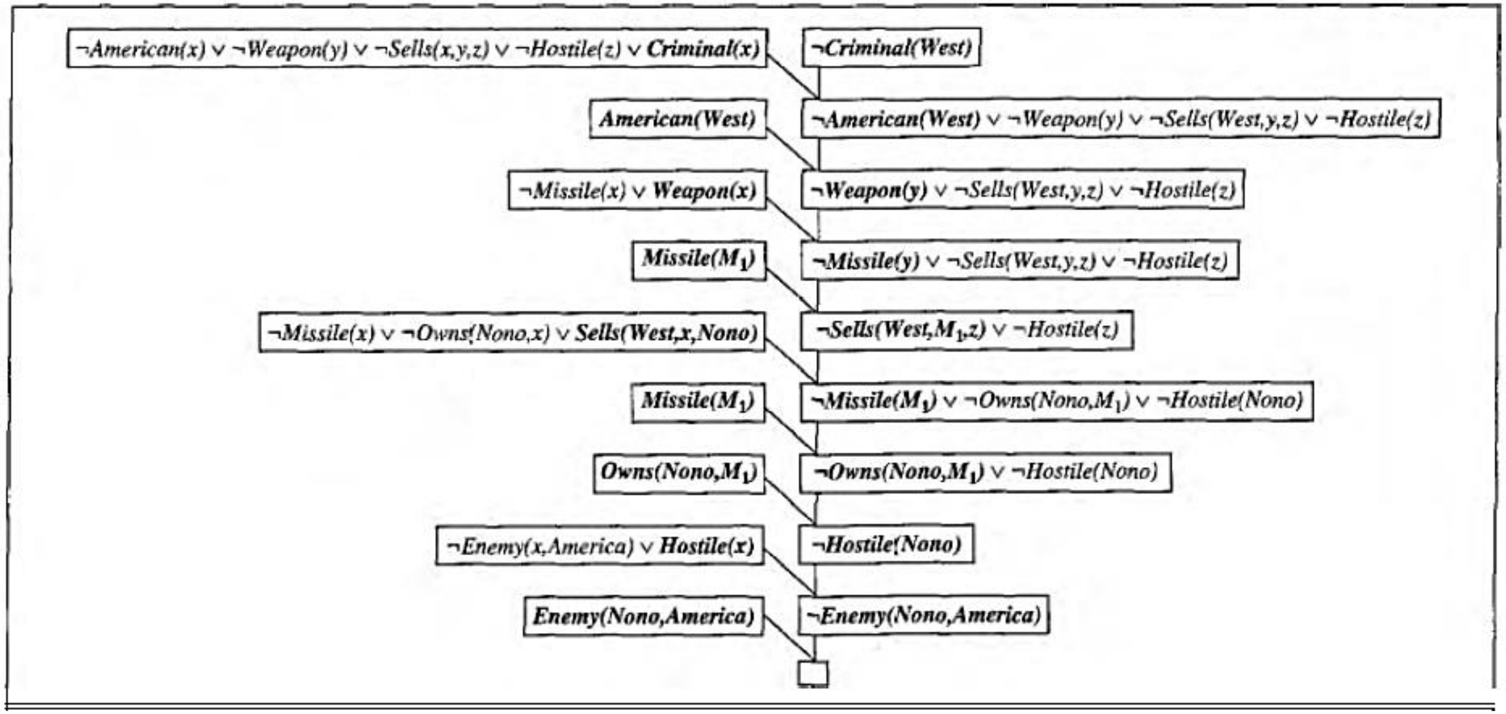
•

•

•

RESOLUTION IN FOL

Example: Everyone who loves all animals is loved by someone



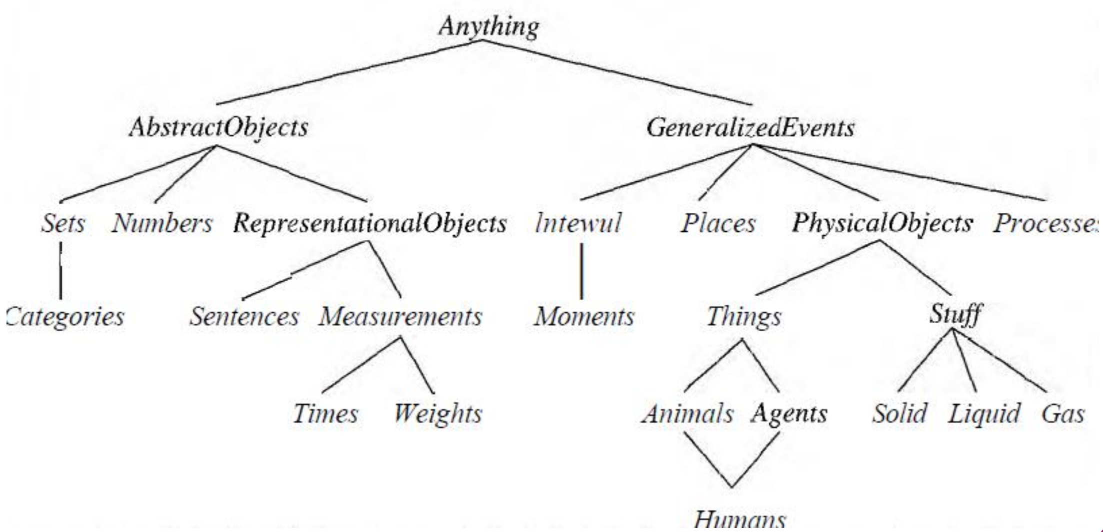
KNOWLEDGE REPRESENTATION

ONTOLOGICAL ENGINEERING

- For general and flexible representations we will see, how to create representations, concentrating on general concepts-such as Actions, Time, Physical Objects, and Beliefs-that occur in many different domains.
- Representing these abstract concepts is sometimes called **ontological engineering**.
- representing everything in the world is daunting. We won't actually write a complete description of everything but we will leave placeholders where new knowledge for any domain can fit in.

ONTOLOGICAL ENGINEERING

- For example, we will define what it means to be a physical object, and the details of different types of objects-robots, televisions, books, or whatever-can be filled in later.
- The general framework of concepts is called an **upper ontology**.



ONTOLOGICAL ENGINEERING

- Certain aspects of the real world are hard to capture in FOL. The principal difficulty is that almost all generalizations have exceptions, or hold only to a degree.
  - For example, although "tomatoes are red" is a useful rule, some tomatoes are green, yellow, or orange.
  - The ability to handle exceptions and uncertainty is extremely important, but is orthogonal to the task of understanding the general ontology.

OE- CATEGORIES AND OBJECTS

- The organization of objects into categories is a vital part of knowledge representation.
- Although interaction with the world takes place at the level of individual objects, much reasoning takes place at the level of categories.
  - For example, a shopper might have the goal of buying a basketball, rather than a particular basketball such as BB9.
  - Categories also serve to make predictions about objects once they are classified

OE- CATEGORIES AND OBJECTS

- There are two choices for representing categories in first-order logic: **predicates** and **objects**.
- That is, we can use the predicate Basketball(b), or we can reify the category as an object, Basketballs.
- We could then say Member(b, Basketballs) to say that b is a member of the category of basketballs.
- We say Subset(Basketballs, Balls) to say that Basketballs is a subcategory, or subset, of Balls.
- We can think of it as a more complex object that just happens to have the Member and Subset relations defined for it.

OE- CATEGORIES AND OBJECTS

- Categories serve to organize and simplify the knowledge base through **inheritance**.
- **Example:**
  - If we say that all instances of the category Food are edible, and if we assert that Fruit is a subclass of Food and Apples is a subclass of Fruit, then we know that every apple is edible.
  - We say that the individual apples inherit the property of edibility, in this case from their membership in the Food category.

OE- CATEGORIES AND OBJECTS

- Subclass relations organize categories into a taxonomy, or taxonomic hierarchy. Taxonomies have been used explicitly for centuries in technical fields.
  - For example, systematic biology aims to provide a taxonomy of all living and extinct species;
  - library science has developed a taxonomy of all fields of knowledge, encoded as the Dewey Decimal system; and
  - tax authorities and other government departments have developed extensive taxonomies of occupations and commercial products.
  - Taxonomies are also an important aspect of general commonsense knowledge.

OE- CATEGORIES AND OBJECTS

- First-order logic makes it easy to state facts about categories, either by relating objects to categories or by quantifying over their members:
  - An object is a member of a category. For example:
    $BB_9 \in Basketballs$
  - A category is a subclass of another category. For example:
    $Basketballs \subset Balls$
  - All members of a category have some properties. For example:
    $x \in Basketballs \Rightarrow Round(x)$
  - Members of a category can be recognized by some properties. For example:
    $Orange(x)\ A\ Round(x) A\ Diameter(x) = 9.5''\ A\ x \in Balls \Rightarrow x \in Basketballs$
  - A category as a whole has some properties. For example:
    $Dogs \in DomesticatedSpecies$

OE- PHYSICAL COMPOSITION

- The idea that one object can be part of another is a familiar one. One's nose is part of one's head, Romania is part of Europe.
- We use the general ***PartOf*** relation to say that one thing is part of another. Objects can be grouped into part of hierarchies, reminiscent of the Subset hierarchy:

$$PartOf(Bucharest, Romania)$$
$$PartOf(Romania, EasternEurope)$$
$$PartOf(EasternEurope, Europe)$$
$$PartOf(Europe, Earth).$$

The *PartOf* relation is transitive and reflexive; that is,

$$PartOf(x, y) \wedge PartOf(y, z) \Rightarrow PartOf(x, z).$$
$$PartOf(x, x).$$

Therefore, we can conclude $PartOf(Bucharest, Earth)$.

OE- MEASUREMENT

- In both scientific and commonsense theories of the world, objects have height, mass, cost, MEASURES and so on. The values that we assign for these properties are called measures. Ordinary quantitative measures are quite easy to represent.
- If the line segment is called L1, we can write:

$$Length(L1) = Inches(1.5) = Centimeters(3.81)$$

Conversion between units is done by equating multiples of one unit to another: $Centimeters(2.54 \times d) = Inches(d)$

$$\text{Diarmeter}(\text{Basketball}_{12}) = \text{Inches}(9.5)$$

$$\text{ListPrice}(\text{Baskettball}_{12}) = \$(19)$$

MENTAL EVENTS & MENTAL OBJECTS

- The agents we have constructed so far have beliefs and can deduce new beliefs. For single-agent domains, knowledge about one's own knowledge and reasoning processes is useful for controlling inference.
  - For example, if one knows that one does not know anything about Romanian geography, then one need not expend enormous computational effort trying to calculate the shortest path from Arad to Bucharest.
  - One can also reason about one's own knowledge in order to construct plans that will change it-for example by buying a map of Romania.
  - In multiagent domains, it becomes important for an agent to reason about the mental states of the other agents. For example, a Romanian police officer might well know the best way to get to Bucharest, so the agent might ask for help.

MENTAL EVENTS & MENTAL OBJECTS

- We need a model of the mental objects that are in someone's head (or something's knowledge base) and of the mental processes that manipulate those mental objects.
- The model should be faithful, but it does not have to be detailed.
  - We do not have to be able to predict how many milliseconds it will take for a particular agent to make a deduction, nor do we have to predict what neurons will fire when an animal is faced with a particular visual stimulus.
- We will be happy to conclude that the Romanian police officer will tell us how to get to Bucharest if he or she knows the way and believes we are lost.

A FORMAL THEORY OF BELIEFS

- We begin with the relationships between agents and "mental objects"-relationships such as Believes, Knows, and Wants (**propositional attitudes)**
- Suppose that Lois believes something-that is,
  - Believes(Lois, x). What kind of thing is x? Clearly, x cannot be a logical sentence. If Flies(Superman) is a logical sentence, we can't say Believes(Lois, Flies(Superman)), because only terms (not sentences) can be arguments of predicates.
  - But if Flies is a function, then Flies(Superman) is a candidate for being a mental object, and Believes can be a relation between an agent and a propositional fluent. Turning a proposition into an object is called **reification.**

REASONING SYSTEMS FOR CATEGORIES

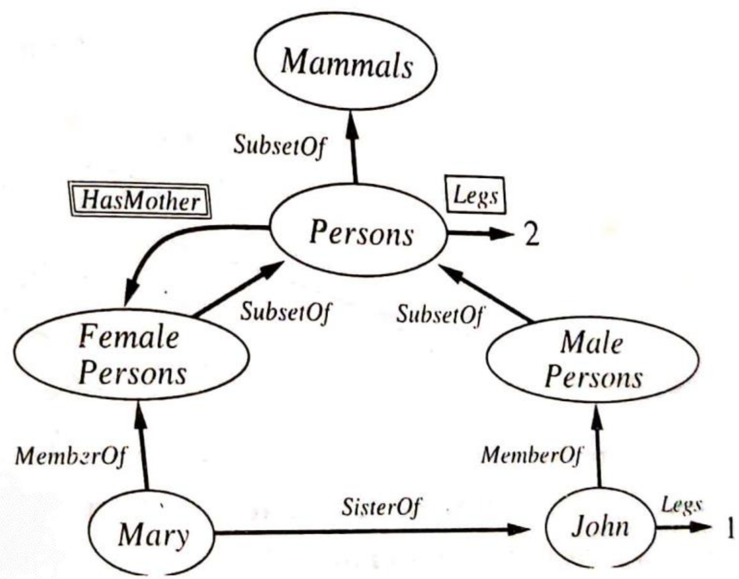
- We have seen that categories are the primary building blocks of any large-scale knowledge representation scheme.
- There are two closely related families of systems: **semantic networks** provide graphical aids for visualizing a knowledge base and efficient algorithms for inferring properties of an object on the basis of its category membership; and
- **description logics** provide a formal language for constructing and combining category definitions and efficient algorithms for deciding subset and superset relationships between categories.

SEMANTIC NETWORKS

- In 1909, Charles Peirce proposed a graphical notation of nodes and arcs called existential graphs that he called "the logic of the future.“
- The notation that semantic networks provide for certain kinds of sentences is often more convenient, but if we strip away the "human interface" issues, the underlying concepts-objects, relations, quantification, and so on-are the same.
- There are many variants of semantic networks, but all are capable of representing individual objects, categories of objects, and relations among objects.
- A typical graphical notation displays object or category names in ovals or boxes, and connects them with labeled arcs.

SEMANTIC NETWORKS

- Figure has a MemberOf link between Mary and FemalePersons, corresponding to the logical assertion Mary ∈ FemalePersons; similarly, the SisterOf link between Mary and John corresponds to the assertion SisterOf (Mary, John).
- We can connect categories using SubsetOf links, and so on. It is such fun drawing bubbles and arrows that one can get carried away.



REASONING WITH DEFAULT INFORMATION

- Default reasoning is concerned with making inferences in cases where the information at hand is incomplete. In such cases it is necessary to make plausible assumptions, which in default reasoning are based on default rules.
- the set of inferences that we can make does not grow monotonically with what we know, and could in fact become smaller when we add more facts.
- This form of reasoning is called non-monotonic reasoning, because the set of inferred sentences does not grow monotonically with the set of known facts.