Knowledge Representation

Prepared By:
Dipanita Saha
Assistant Professor
Institute of Information Technology(IIT)
Noakhali Science and Technology University
.

What is ontology?

- Ontology is a term used by philosophers to mean the most "essential existence" in the universe.
- For example, if A can be derived from or produced by B, A is NOT essential. Here B can be a single "existence" or a set of "existences".
- some philosophers believe that the universe is made up of various materials or objects, and they are called "materialists".
- Objects defined in a cyber-space are not essential from the point of view of its creator (human), but "the agents" (AI) living in that space may "think" they are.

ONTOLOGICAL ENGINEERING

- Complex domains such as shopping on the Internet or driving a car in traffic require more general and flexible representations.
- how to create these representations, concentrating on general concepts—such as *Events*, *Time*, *Physical Objects*, and *Beliefs*— that occur in many different domains. Representing these abstract concepts is sometimes called **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.
- The general framework of concepts is called an **upper ontology** because of the convention of drawing graphs with the general concepts at the top and the more specific concepts below them, as in Figure.

Basic idea of ontology engineering

- In ontology engineering, knowledge is represented using a graph.
- Each node of the graph represents a “concept”, and the edge between two nodes represents the relation between the two concepts.
- A concept defines a group of existences. The form (shape or outlook), properties, states, etc. of these existences are memorized (capsulized) in the corresponding node.
- The relation between two concepts may include causal relation, inclusion relation, membership relation, etc.
- Using an ontology graph, other people can understand the corresponding knowledge easily, and can improve freely.

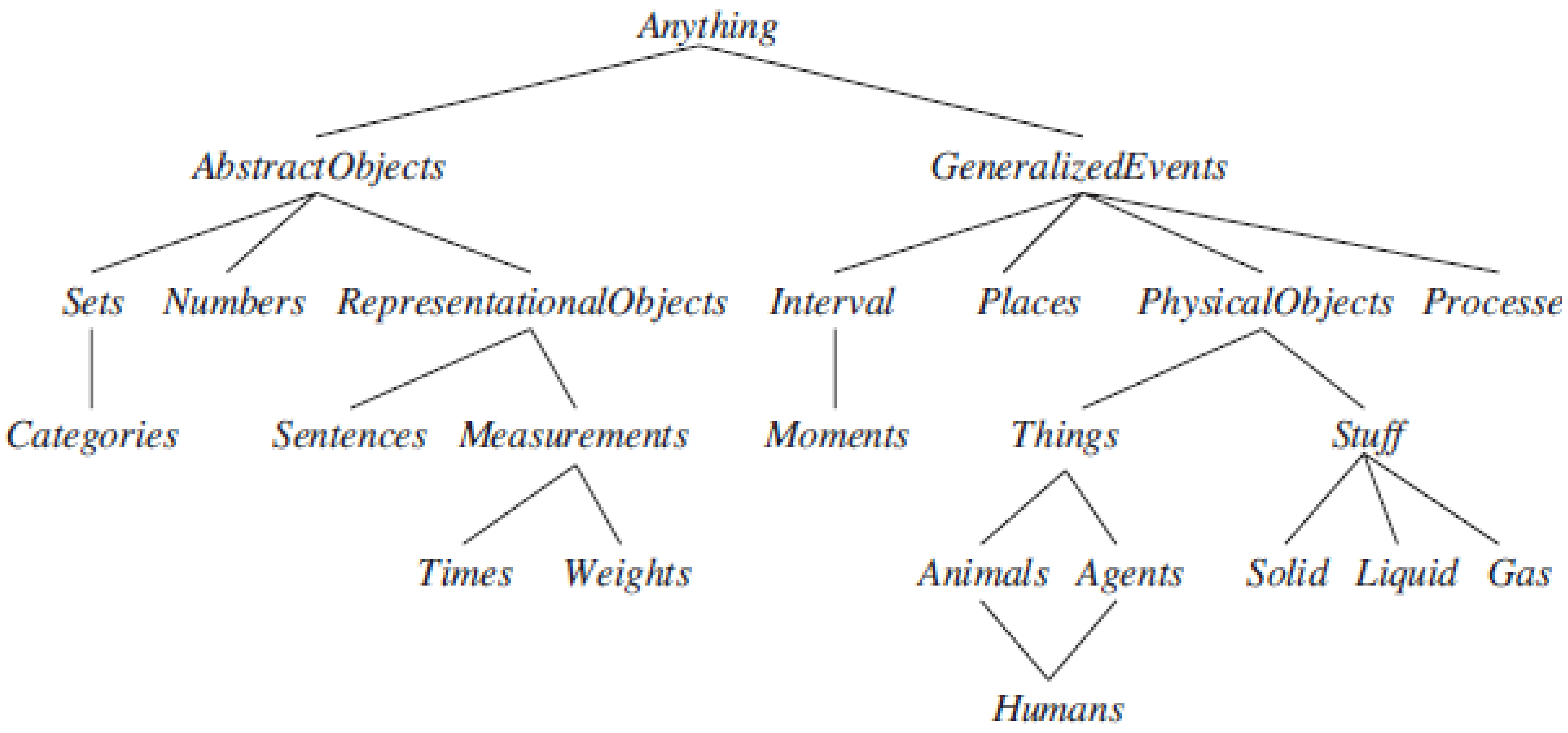

Figure 12.1 The upper ontology of the world, showing the topics to be covered later in the chapter. Each link indicates that the lower concept is a specialization of the upper one. Specializations are not necessarily disjoint; a human is both an animal and an agent, for example. We will see in Section 12.3.3 why physical objects come under generalized events.

▪ 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 would normally 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.
- One infers the presence of certain objects from perceptual input, infers category membership from the perceived properties of the objects, and then uses category information to make predictions about the objects.
- For example, from its green and yellow mottled skin, one-foot diameter, ovoid shape, red flesh, black seeds, and presence in the fruit aisle, one can infer that an object is a watermelon; from this, one infers that it would be useful for fruit salad.

- 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)*, which we will abbreviate as $b \in$ *Basketballs*, to say that b is a member of the category of basketballs.
- We say *Subset(Basketballs, Balls)*, abbreviated as *Basketballs* $\subset$ *Balls*, to say that Basketballs is a **subcategory** of Balls.
- Categories serve to organize and simplify the knowledge base through **inheritance**.
- 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 can infer that every apple is edible.
- Subclass relations organize categories into a **taxonomy**, or **taxonomic hierarchy**.
- Taxonomies have been used explicitly for centuries in technical fields. 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.

- First-order logic makes it easy to state facts about categories, either by relating objects to categories or by quantifying over their members.
- Here are some types of facts, with examples of each:
- An object is a member of a category. $BB9 \in Basketballs$
- A category is a subclass of another category. $Basketballs \subset Balls$
- All members of a category have some properties. $(x \in Basketballs) \Rightarrow Spherical\ (x)$
- Members of a category can be recognized by some properties.

$$Orange(x) \land Round\ (x) \land Diameter(x)=9.5 \land x \in Balls \Rightarrow x \in Basketballs$$

- A category as a whole has some properties. $Dogs \in DomesticatedSpecies$
- we also want to be able to state relations between categories that are not subclasses of each other.
- For example, if we just say that Males and Females are subclasses of Animals, then we have not said that a male cannot be a female.

- We say that two or more categories are **disjoint** if they have no members in common. And even if we know that males and females are disjoint, we will not know that an animal that is not a male must be a female,
- unless we say that males and females constitute an **exhaustive decomposition** of the animals. A disjoint exhaustive decomposition is known as a **partition**. The following examples illustrate these three concepts:
- *Disjoint({Animals,Vegetables})*
- *ExhaustiveDecomposition ({Americans,Canadians, Mexicans},NorthAmericans)*
- *Partition({Males, Females}, Animals)*
- The three predicates are defined as follows:

$Disjoint(s) \Leftrightarrow (\forall c1, c2\ c1 \in s \land c2 \in s \land c1 = c2 \Rightarrow Intersection(c1, c2)=\{\ \})$

- $ExhaustiveDecomposition(s, c) \Leftrightarrow (\forall i\ i \in c \Leftrightarrow \exists c2\ c2 \in s \land i \in c2)$
- $Partition(s, c) \Leftrightarrow Disjoint(s) \land ExhaustiveDecomposition(s, c)$.
- Categories can also be *defined* by providing necessary and sufficient conditions for membership. For example, a bachelor is an unmarried adult male:
- $x \in Bachelors \Leftrightarrow Unmarried(x) \land x \in Adults \land x \in Males$.

Physical composition

- The idea that one object can be part of another is a familiar one. 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,

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

$$PartOf(x, x) .$$

- Therefore, we can conclude PartOf (Bucharest , Earth).
- Categories of **composite objects** are often characterized by structural relations among parts. For example, a biped has two legs attached to a body:

$$\begin{aligned} Biped(a) \Rightarrow \exists l1, l2, b\ & Leg(l1) \land Leg(l2) \land Body(b) \land \\ & PartOf(l1, a) \land PartOf(l2, a) \land PartOf(b, a) \land \\ & Attached(l1, b) \land Attached(l2, b) \land \\ & l1 = l2 \land [\forall l3\ Leg(l3) \land PartOf(l3, a) \Rightarrow (l3 = l1 \lor l3 = l2)] . \end{aligned}$$

- It is also useful to define composite objects with definite parts but no particular structure.
- For example, we might want to say "The apples in this bag weight two pounds." The temptation would be to ascribe this weight to the *set* of apples in the bag, but this would be a mistake because the set is an abstract mathematical concept that has elements but does not have weight.
- we need a new concept, call **bunch**. For example, if the apples are *Apple1*, *Apple2*, and *Apple3*,
- then *BunchOf ({Apple1,Apple2,Apple3})* denotes the composite object with the three apples as parts (not elements).
- *BunchOf (Apples)* is the composite object consisting of all apples—not to be confused with Apples, the category or set of all apples.
- We can define *BunchOf* in terms of the *PartOf* relation. Obviously, each element of *s* is part of BunchOf *(s)*: $\forall x \; x \in s \Rightarrow PartOf(x, BunchOf(s))$.
- Furthermore, *BunchOf (s) is the smallest object satisfying this condition*.
- In other words, *BunchOf (s)* must be part of any object that has all the elements of s as parts:

$$\forall y \; [\forall x \; x \in s \Rightarrow PartOf(x, y)] \Rightarrow PartOf(BunchOf(s), y) .$$

- These axioms are an example of a general technique called **logical minimization**, which means defining an object as the smallest one satisfying certain conditions.

Measurements

- objects have height, mass, cost, and so on. The values that we assign for these properties are called **measures**.
- We imagine that the universe includes abstract “measure objects,” such as the *length* that is the length of this line segment:
- We can call this length 1.5 inches or 3.81 centimeters.
- Thus, the same length has different names in our language. We represent the length with a **units function** that takes a number as argument.
- 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:

$$Centimeters(2.54 \times d)=Inches(d) .$$

- Measures can be used to describe objects as follows:

$$Diameter(Basketball_{12}) = Inches(9.5) .$$
$$ListPrice(Basketball_{12}) = \$(19) .$$
$$d \in Days \Rightarrow Duration(d) = Hours(24) .$$

Objects: Things and stuff

- The real world can be seen as consisting of primitive objects (e.g., atomic particles) and composite objects built from them.
- There is a significant portion of reality that seems to defy any obvious **individuation**—division into distinct objects.
- We give this portion the generic name **stuff**. For example, suppose I have some butter and an aardvark in front of me.
- I can say there is one aardvark, but there is no obvious number of "butter-objects," because any part of a butter-object is also a butter-object, at least until we get to very small parts indeed.
- This is the major distinction between *stuff* and *things*.
- If we cut an aardvark in half, we do not get two aardvarks (unfortunately).
- We say "an aardvark," but, except in pretentious California restaurants, one cannot say "a butter."
- Linguists distinguish between **count nouns**, such as aardvarks, holes, and theorems, and **mass nouns**, such as butter, water, and energy.

- For example, we might recognize a lump of butter as the one left on the table the night before; we might pick it up, weigh it, sell it, or whatever. In these senses, it is an object just like the aardvark. We also define the category *Butter* . Informally, its elements will be all those things of which one might say "It's butter," including *Butter* . part of a butter-object is also a butter-object: $b \in Butter \wedge PartOf(p, b) \Rightarrow p \in Butter$.
- We can now say that butter melts at around 30 degrees centigrade: $b \in Butter \Rightarrow MeltingPoint(b, Centigrade(30))$.
- We could go on to say that butter is yellow, is less dense than water, is soft at room temperature, has a high fat content, and so on. On the other hand, butter has no particular size, shape, or weight. We can define more specialized categories of butter such as UnsaltedButter , which is also a kind of *stuff*.
- If we cut a pound of butter in half, we do not, alas, get two pounds of butter.
- What is actually going on is this: some properties are **intrinsic**: they belong to the very substance of the object, rather than to the object as a whole. When you cut an instance of *stuff* in half, the two pieces retain the intrinsic properties—things like density, boiling point, flavor, color, ownership, and so on.
- On the other hand, their **extrinsic** properties—weight, length, shape, and so on—are not retained under subdivision.
- A category of objects that includes in its definition only *intrinsic* properties is then a substance, or mass noun; a class that includes *any* extrinsic properties in its definition is a count noun.

EVENTS

- Consider a continuous action, such as filling a bathtub.
- Situation calculus can say that the tub is empty before the action and full when the action is done, but it can't talk about what happens *during* the action.
- It also can't describe two actions happening at the same time—such as brushing one's teeth while waiting for the tub to fill.
- To handle such cases we introduce an alternative formalism known as **event calculus**, which is based on points of time rather than on situations.
- Event calculus reifies fluents and events.
- The fluent *At(Shankar , Berkeley)* is an object that refers to the fact of Shankar being in Berkeley, but does not by itself say anything about whether it is true.
- To assert that a fluent is actually true at some point in time we use the predicate *T*, as in *T(At(Shankar , Berkeley), t)*.
- Events are described as instances of event categories.
- The event E1 of Shankar flying from San Francisco to Washington, D.C. is described as
  - $E1 \in Flyings \wedge Flyer(E1, Shankar) \wedge Origin(E1, SF) \wedge Destination(E1, DC)$.

- we can define an alternative three-argument version of the category of flying events and say $E1 \in Flyings(Shankar, SF, DC)$.

Processes

- **discrete events** they have a definite structure.
- Shankar's trip has a beginning, middle, and end. If interrupted halfway, the event would be something different—it would not be a trip from San Francisco to Washington, but instead a trip from San Francisco to somewhere over Kansas.
- The category of events denoted by Flyings has a different quality. If we take a small interval of Shankar's flight, say, the third 20-minute segment, that event is still a member of Flyings.
- Categories of events with this property are called **process** categories or **liquid event** categories. Any process *e* that happens over an interval also happens over any subinterval:
- $(e \in Processes) \land Happens(e, (t1, t4)) \land (t1 < t2 < t3 < t4) \Rightarrow Happens(e, (t2, t3))$.
- The distinction between liquid and nonliquid events is exactly analogous to the difference between substances, or *stuff*, and individual objects, or *things*.
- In fact, some have called liquid events **temporal substances**, whereas substances like butter are **spatial substances**.

- **Time intervals**

- We will consider two kinds of time intervals: moments and extended intervals.
- The distinction is that only moments have zero duration:
  - *Partition({Moments,ExtendedIntervals}, Intervals)*
  - $i \in Moments \Leftrightarrow Duration(i)=Seconds(0)$.
- Next we invent a time scale and associate points on that scale with moments, giving us absolute times.
- The functions *Begin* and *End* pick out the earliest and latest moments in an interval, and the function Time delivers the point on the time scale for a moment.
- The function *Duration* gives the difference between the end time and the start time.
- $Interval(i) \Rightarrow Duration(i)=(Time(End(i)) - Time(Begin(i)))$.
- $Time(Begin(AD1900))=Seconds(0)$.
- $Time(Begin(AD2001))=Seconds(3187324800)$.
- $Time(End(AD2001))=Seconds(3218860800)$.
- $Duration(AD2001)=Seconds(31536000)$.
- To make these numbers easier to read, we also introduce a function *Date*, which takes six arguments (hours, minutes, seconds, day, month, and year) and returns a time point:
- $Time(Begin(AD2001))=Date(0, 0, 0, 1, Jan, 2001)$
- $Date(0, 20, 21, 24, 1, 1995)=Seconds(3000000000)$.

MENTAL EVENTS AND MENTAL OBJECTS

- The agents we have constructed so far have beliefs and can deduce new beliefs.
- Knowledge about one's own knowledge and reasoning processes is useful for controlling inference.
- For example, suppose Alice asks "what is the square root of 1764" and Bob replies "I don't know." If Alice insists "think harder," Bob should realize that with some more thought, this question can in fact be answered.
- On the other hand, if the question were "Is your mother sitting down right now?" then Bob should realize that thinking harder is unlikely to help.
- Knowledge about the knowledge of other agents is also important; Bob should realize that his mother knows whether she is sitting or not, and that asking her would be a way to find out.
- What we need is a model of the mental objects that are in someone's head and of the mental processes that manipulate those mental objects.
- We do not have to be able to predict how many milliseconds it will take for a particular agent to make a deduction.
- We will be happy just to be able to conclude that mother knows whether or not she is sitting

- We begin with the **propositional attitudes** that an agent can have toward mental objects: attitudes such as Believes, Knows, Wants, Intends, and Informs.
- The difficulty is that these attitudes do not behave like “normal” predicates.
- For example, suppose we try to assert that Lois knows that Superman can fly:
  - *Knows(Lois, CanFly(Superman))* .
- One minor issue with this is that we normally think of *CanFly(Superman)* as a sentence, but here it appears as a term.
- That issue can be patched up just be reifying *CanFly(Superman);* making it a fluent.
- A more serious problem is that, if it is true that Superman is Clark Kent, then we must conclude that Lois knows that Clark can fly:

$$(\text{Superman} = \text{Clark}) \land \text{Knows}(\text{Lois}, \text{CanFly}(\text{Superman}))$$

$$\models \text{Knows}(\text{Lois}, \text{CanFly}(\text{Clark})) .$$

- if our agent knows that $2 + 2 = 4$ and $4 < 5$, then we want our agent to know that $2 + 2 < 5$.
- This property is called **referential transparency**—it doesn’t matter what term a logic uses to refer to an object, what matters is the object that the term names.

- Regular logic is concerned with a single modality, the modality of truth, allowing us to express "*P* is true."
- Modal logic includes special modal operators that take sentences as arguments.
- For example, "*A* knows *P*" is represented with the notation $\mathbf{K}_A P$ where **K** is the modal operator for knowledge.
- It takes two arguments, an agent and a sentence.
- The **syntax** of modal logic is the same as first-order logic, except that sentences can also be formed with modal operators.
- The **semantics** of modal logic is more complicated.
- In first-order logic a **model** contains a set of objects and an interpretation that maps each name to the appropriate object, relation, or function.
- Therefore, we will need a more complicated model, one that consists of a collection of **possible worlds** rather than just one true world. The worlds are connected in a graph by **accessibility relations**, one relation for each modal operator.
- We say that world *w1* is accessible from world *w0* with respect to the modal operator ***KA*** if everything in *w1* is consistent with what *A* knows in *w0*, and we write this as *Acc(**K**A,w0,w1)*.
- As an example, in the real world, Bucharest is the capital of Romania, but for an agent that did not know that, other possible worlds are accessible, including ones where the capital of Romania is Sibiu or Sofia.

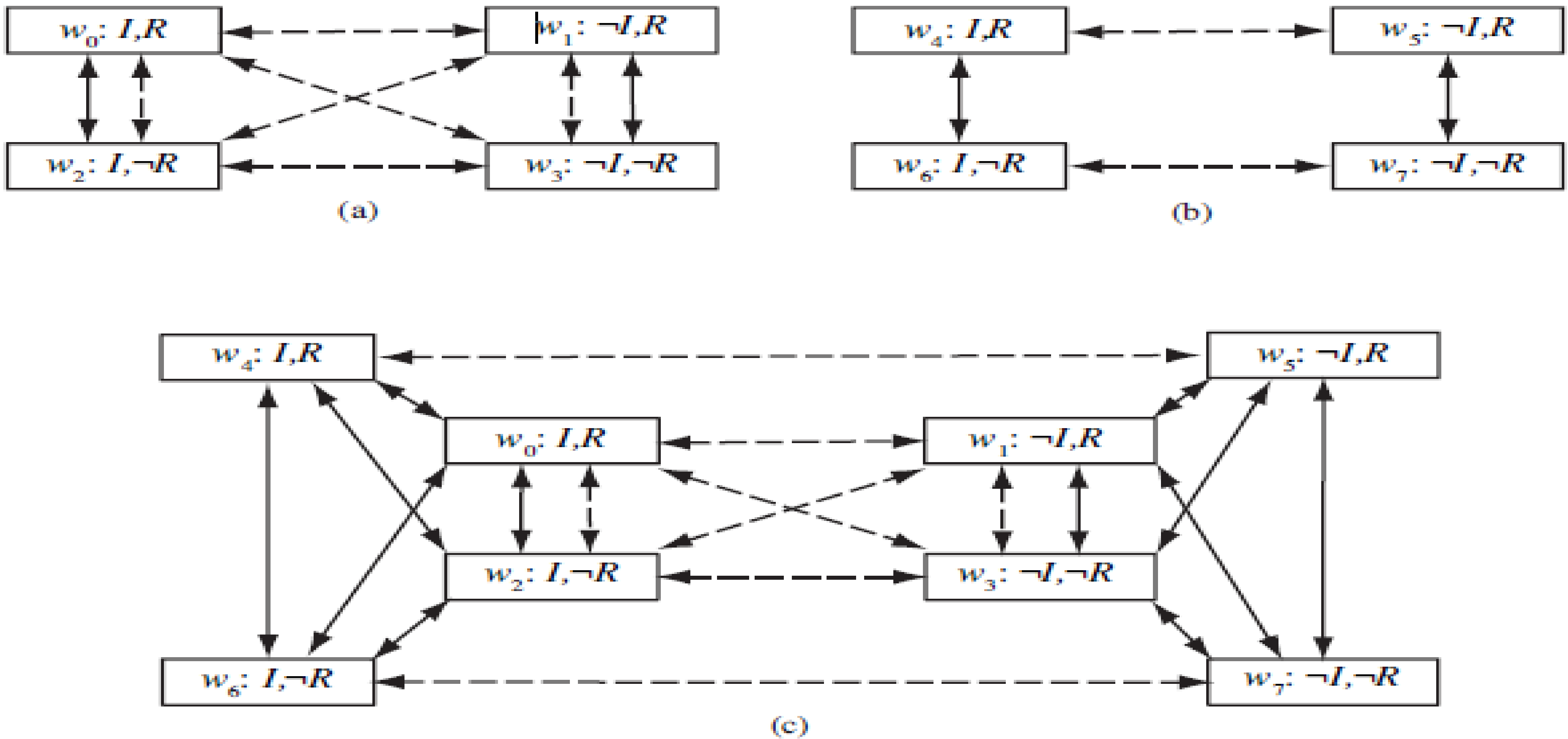

Figure 12.4 Possible worlds with accessibility relations $\mathbf{K}_{Superman}$ (solid arrows) and $\mathbf{K}_{Lois}$ (dotted arrows). The proposition R means "the weather report for tomorrow is rain" and I means "Superman's secret identity is Clark Kent." All worlds are accessible to themselves; the arrows from a world to itself are not shown.

- Figure shows some possible worlds for this domain, with accessibility relations for Lois and Superman.
- In the TOP-LEFT diagram, it is common knowledge that Superman knows his own identity, and neither he nor Lois has seen the weather report.
- So in *w0* the worlds *w0* and *w2* are accessible to Superman; maybe rain is predicted, maybe not.
- For Lois all four worlds are accessible from each other; she doesn't know anything about the report or if Clark is Superman.
- But she does know that Superman knows whether he is Clark, because in every world that is accessible to Lois, either Superman knows I, or he knows ¬I. Lois does not know which is the case, but either way she knows Superman knows.
- In the TOP-RIGHT diagram it is common knowledge that Lois has seen the weather report. So in *w4* she knows rain is predicted and in w6 she knows rain is not predicted. Superman does not know the report, but he knows that Lois knows, because in every world that is accessible to him, either she knows R or she knows ¬R.
- In the BOTTOM diagram we represent the scenario where it is common knowledge that Superman knows his identity, and Lois might or might not have seen the weather report.
- We represent this by combining the two top scenarios, and adding arrows to show that Superman does not know which scenario actually holds. Lois does know, so we don't need to add any arrows for her.
- In w0 Superman still knows I but not R, and now he does not know whether Lois knows R.
- From what Superman knows, he might be in w0 or w2, in which case Lois does not know whether R is true, or he could be in w4, in which case she knows R, or w6, in which case she knows ¬R.

- The English sentence "Bond knows that someone is a spy" is ambiguous. The first reading is that there is a particular someone who Bond knows is a spy; we can write this as
  - $\exists x\, \mathbf{K}_{Bond} Spy(x)$,
- which in modal logic means that there is an *x* that, in all accessible worlds, Bond knows to be a spy. The second reading is that Bond just knows that there is at least one spy:
  - $\mathbf{K}_{bond} \exists x\, Spy(x)$.
- The modal logic interpretation is that in each accessible world there is an *x* that is a spy, but it need not be the same *x* in each world. Now that we have a modal operator for knowledge, we can write axioms for it.
- First, we can say that agents are able to draw deductions; if an agent knows P and knows that P implies Q, then the agent knows Q:
  - $(\mathbf{K}_a P \wedge \mathbf{K}_a(P \Rightarrow Q)) \Rightarrow \mathbf{K}_a Q.$
- From this we can establish that $\mathbf{K}A(P \vee \neg P)$ is a tautology; every agent knows every proposition P is either true or false.
- On the other hand, $(\mathbf{K}AP) \vee (\mathbf{K}A\neg P)$ is not a tautology; in general, there will be lots of propositions that an agent does not know to be true and does not know to be false.
- knowledge is justified true belief. That is, if it is true, if you believe it, and if you have an unassailably good reason, then you know it.

- That means that if you know something, it must be true, and we have the axiom:
  - $\mathbf{K}aP \Rightarrow P$.
- Furthermore, logical agents should be able to introspect on their own knowledge. If they know something, then they know $\mathbf{K}_a P \Rightarrow \mathbf{K}_a(\mathbf{K}_a P)$

Semantic networks

- Semantic network is a special case of ontology graph.
- In a semantic network, the knowledge is represented by a directed graph.
- In each node, a concept is defined by–Concept (object) name;–Attributes; and–Attribute values.
- An edge defines the relation between two concepts.
- 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 links. For example,

- 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.
- For example, we know that persons have female persons as mothers, so can we draw a *HasMother* link from Persons to *FemalePersons*?
- The answer is no, because *HasMother* is a relation between a person and his or her mother, and categories do not have mothers

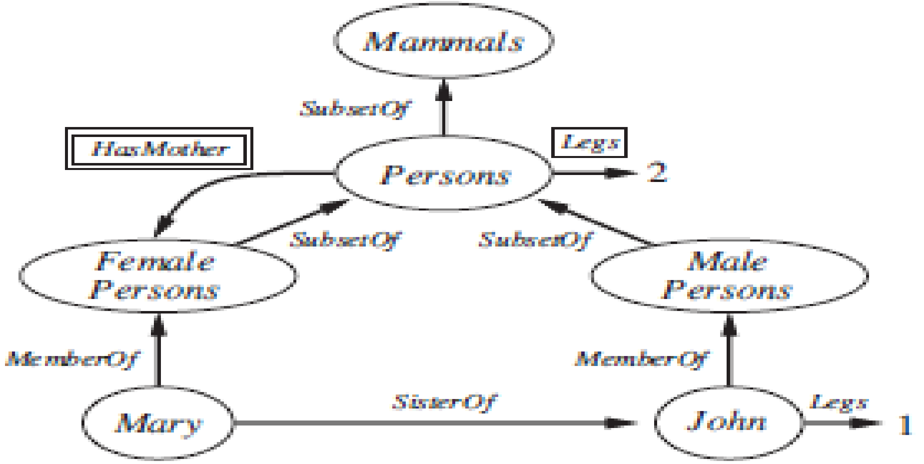

Figure 12.5 A semantic network with four objects (John, Mary, 1, and 2) and four categories. Relations are denoted by labeled links.

- For this reason, we have used a special notation—the double-boxed link. This link asserts that

$$\forall x \quad x \in Persons \;\Rightarrow\; [\forall y \;\; HasMother(x, y) \;\Rightarrow\; y \in FemalePersons]\,.$$

- We might also want to assert that persons have two legs—that is,

$$\forall x \quad x \in Persons \;\Rightarrow\; Legs(x, 2)$$

- Inheritance becomes complicated when an object can belong to more than one category or when a category can be a subset of more than one other category; this is called **multiple inheritance**.
- the inheritance algorithm might find two or more conflicting values answering the query.
- For this reason, multiple inheritance is banned in some **object-oriented programming** (OOP) languages, such as Java, that use inheritance in a class hierarchy. It is usually allowed in semantic networks,

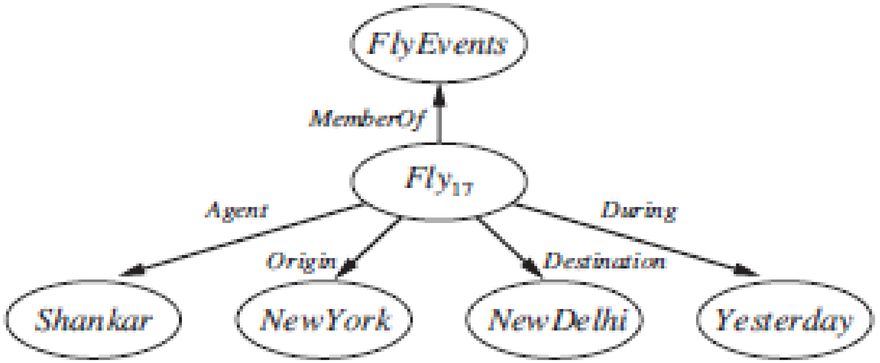

Figure 12.6 A fragment of a semantic network showing the representation of the logical assertion $Fly(Shankar, NewYork, NewDelhi, Yesterday)$.

Description logics

- **Description logics** are notations that are designed to make it easier to describe definitions and properties of categories.
- Description logic systems evolved from semantic networks in response to pressure to formalize what the networks mean while retaining the emphasis on taxonomic structure as an organizing principle.
- The principal inference tasks for description logics are **subsumption** (checking if one category is a subset of another by comparing their definitions) and **classification** (checking whether an object belongs to a category).
- Some systems also include **consistency** of a category definition—whether the membership criteria are logically satisfiable.

$$
\begin{array}{rcl}
\mathit{Concept} & \rightarrow & \mathbf{Thing} \mid \mathit{ConceptName} \\
& \mid & \mathbf{And}(\mathit{Concept}, \ldots) \\
& \mid & \mathbf{All}(\mathit{RoleName}, \mathit{Concept}) \\
& \mid & \mathbf{AtLeast}(\mathit{Integer}, \mathit{RoleName}) \\
& \mid & \mathbf{AtMost}(\mathit{Integer}, \mathit{RoleName}) \\
& \mid & \mathbf{Fills}(\mathit{RoleName}, \mathit{IndividualName}, \ldots) \\
& \mid & \mathbf{SameAs}(\mathit{Path}, \mathit{Path}) \\
& \mid & \mathbf{OneOf}(\mathit{IndividualName}, \ldots) \\
\mathit{Path} & \rightarrow & [\mathit{RoleName}, \ldots]
\end{array}
$$

Figure 12.7 The syntax of descriptions in a subset of the CLASSIC language.

- The CLASSIC language is a typical description logic. The syntax of CLASSIC descriptions is shown in Figure.
- For example, to say that bachelors are unmarried adult males we would write
  - *Bachelor = And(Unmarried, Adult ,Male) .*
- The equivalent in first-order logic would be
  - $Bachelor\ (x) \Leftrightarrow Unmarried(x) \wedge Adult(x) \wedge Male(x)$.
- Any description in CLASSIC can be translated into an equivalent first-order sentence, but some descriptions are more straightforward in CLASSIC.
- For example, to describe the set of men with at least three sons who are all unemployed and married to doctors, and at most two daughters who are all professors in physics or math departments, we would use

And(Man, AtLeast(3, Son), AtMost(2, Daughter),

All(Son, And(Unemployed,Married, All(Spouse, Doctor))),

All(Daughter , And(Professor , Fills(Department , Physics,Math)))) .