Chap 12 : Reasoning System with Categories

HABIB ULLAH QAMAR

MSCS (SE) MBA-HRM

Introduction

So far we learned about knowledge representation.

How to create these representations, concentrating on general concepts—such as *Events, Time, Physical Objects*, and *Beliefs*— that occur in many different domains. ONTOLOGICAL ENGINEERING is a way to representing *everything* in the world is and overwhelming task.

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. (OOP)

Ontology Engineering

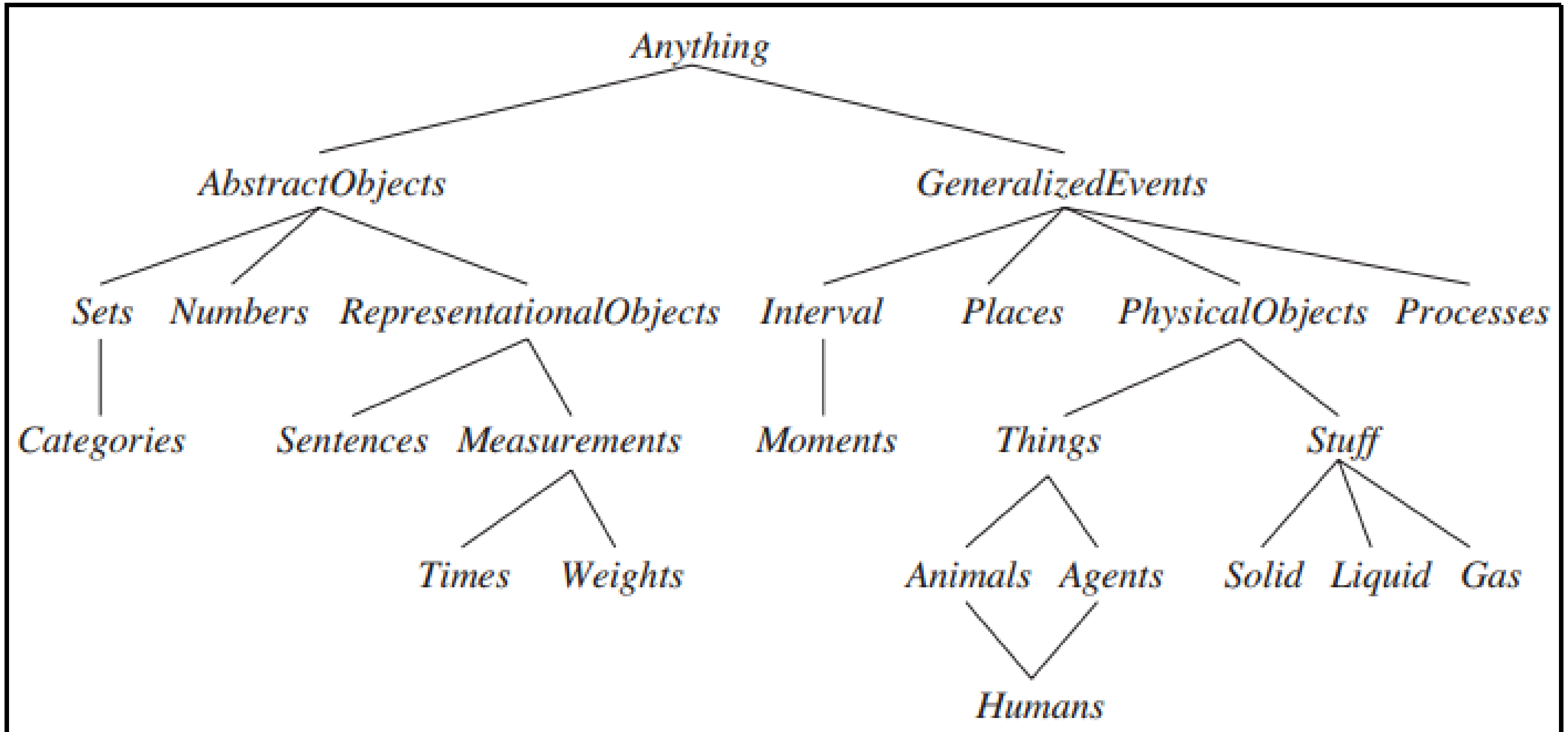


Figure 12.1 The upper ontology of the world, showing the topics to be covered later in the chapter. Each link indicates that the lower concept is a specialization of the upper one. Specializations are not necessarily disjoint; a human is both an animal and an agent, for

Reasoning in categories

Categories are the primary building blocks of large-scale knowledge representation schemes.

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

Description logics provide a formal language for constructing and combining category definitions and efficient algorithms for deciding subset and superset relationships between categories.

Semantic Network

In 1909, Charles S. Peirce proposed a graphical notation of nodes and edges called **existential graphs** that he called "the logic of the future."

Thus began a long-running debate between advocates of "logic" and advocates of "semantic networks."

Unfortunately, the debate obscured the fact that semantics networks—at least those with well-defined semantics—*are* a form of logic.

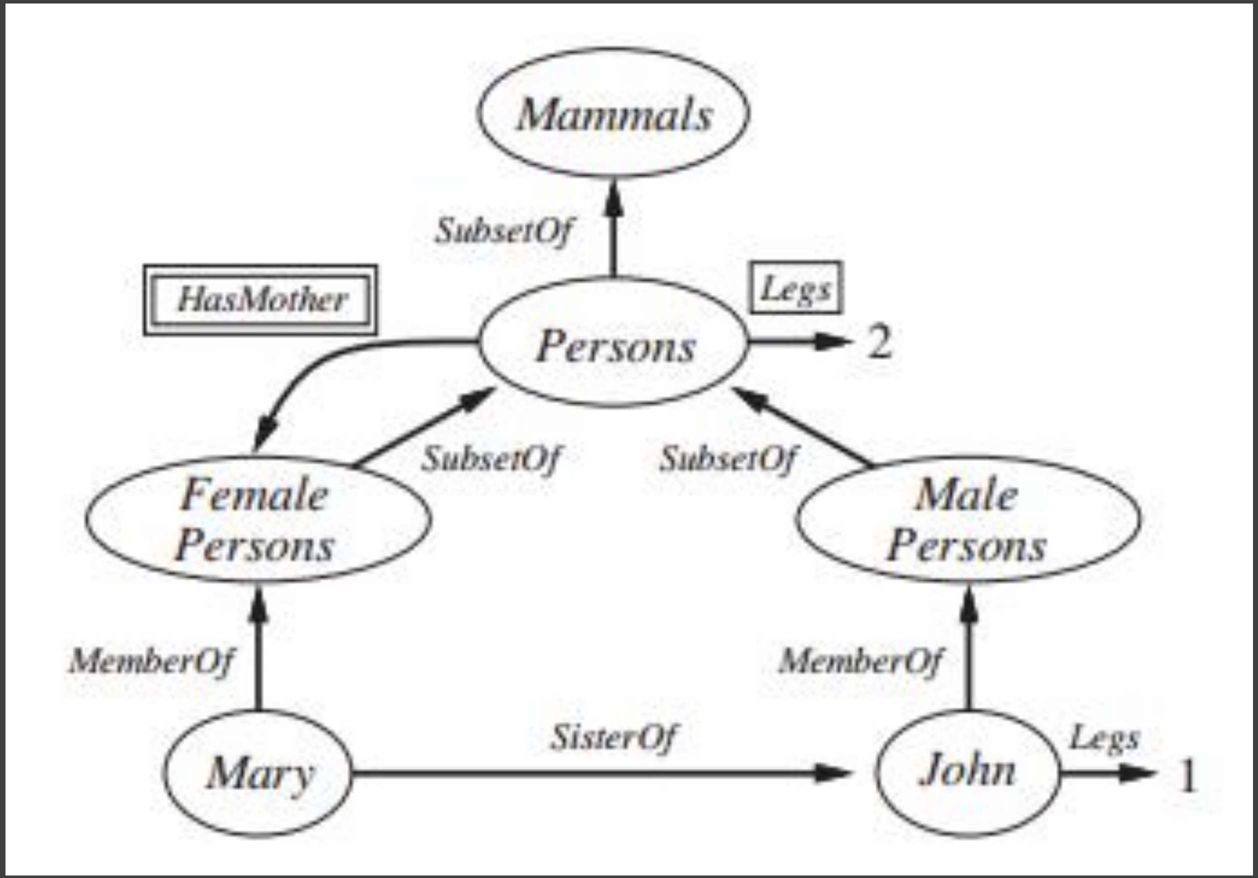
Semantic Network

There are many ***<u>variants</u>*** 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 ***<u>ovals or boxes</u>***, and connects them with labeled links.

See exmaple

Semantic Network



Semantic Network

Semantic networks are graphs, with nodes representing objects and arcs representing relationships between objects.

Various types of relationships may be defined using semantic networks. The two most common types of relationships are

–IS-A (Inheritance relation)

–HAS (Ownership relation)

Semantic Network : IS-A

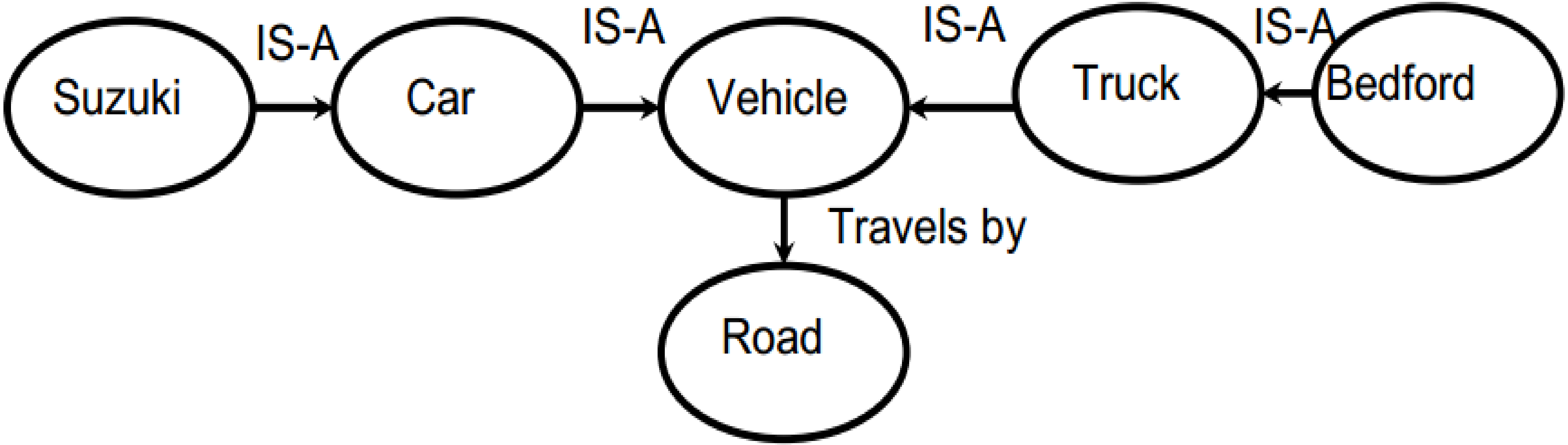


Figure: Vehicle Semantic Network

Semantic Network : Network Operations

To infer new information from semantic networks, we can ask questions from nodes

– Ask node vehicle: ‘How do you travel?’

 – This node looks at arc and replies: road

– Ask node Suzuki: ‘How do you travel?’

 – This node does not have a link to travel therefore it asks other nodes linked by the IS-A link

 – Asks node Car (because of IS-A relationship)

 – Asks node Vehicle (IS-A relationship)

 – Node Vehicle Replies: road

Semantic Network : Problems

Semantic networks are ***computationally expensive*** at run-time as we need to traverse the network to answer some question. In the worst case, we may need to traverse the entire network and then discover that the requested info does not exist.

They try to model human associative memory (store information using associations), but in the human brain the number of neurons and links are in the order of 10^{15}. It is not practical to build such a large semantic network, hence this scheme is not feasible for this type of problems.

Semantic Network : Problems

- Semantic networks are logically inadequate as they do not have any equivalent quantifiers, e.g., for all, for some, none.

Description Logics

The syntax of first-order logic is designed to make it easy to say things about objects.

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.

Description Logics : Syntax

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

Knowledge Representations in Description Logics

In DLs we make a clear distinction between Intensional knowledge and extensional knowledge.

- A KB consists of two components: a TBox and an Abox.

– TBox: intensional knowledge in the form of concepts (terms), their properties and their relations.

– Abox: extensional (assertional) knowledge.

- TELL and ASK interface

TBox

In the TBox one defines concepts of the application domain, their properties and their relations:

Example:

$$\texttt{Woman} \equiv \texttt{Person} \sqcap \texttt{Female}$$

$$\texttt{Male} \sqsubseteq \neg\texttt{Female}$$

ABox

In the ABox one makes assertions about the individuals in the application domain: **membership in classes** and **role filling.**

Example:

$$(\texttt{Person} \sqcap \texttt{Female})(\texttt{ANNA}), \quad \texttt{child}(\texttt{ANNA}, \texttt{JOHN})$$

Open and closed worlds

The **Closed World** Assumption (CWA) is the assumption that what is not known to be true must be false. This is also called ***circumscription***.

The **Open World** Assumption (OWA) is the opposite. In other words, it is the assumption that what is not known to be true is simply unknown. **This is also called default logic**

Default logic

This is a very common from of **non-monotonic reasoning**. Here *We want to draw conclusions based on what is **most likely to be true.*** We have already seen examples of this and possible ways to represent this knowledge.

We will discuss two approaches to do this:

- Non-Monotonic logic (might be wrong when adding more info…. **This is also called belief revision**.)
- Default logic (monotonic) always true even adding more knowledge

Truth maintenance System (TMS)

A TMS deals with uncertainty by permitting new knowledge to replace old knowledge which is believed to be outdated or erroneous.

- Enforcing logical relations among beliefs.
- Generating explanations for conclusions.
- Finding solutions to search problems
- supporting default reasoning.
- Identifying causes for failure and recover from inconsistencies.

Types of Truth maintenance System (JTMS)

Justification-Based Truth Maintenance System (ATMS) JTMS : It is a simple TMS where one can examine the consequences of the current set of assumptions. The meaning of sentences is not known.

: It allows to maintain and reason with a number of simultaneous, possibly incompatible, current sets of assumption. Otherwise it is similar to JTMS, i.e. it does not recognize the meaning of sentences.

Thanks……

HABIB ULLAH QAMAR

MSCS SE (MBA HRM)