CS371 Lecture: Introduction to Knowledge Representation Revised 1/4/06

I. Introduction

A. Classically, two key areas of concern in AI have been knowledge representation and search.

1. A key step in computer problem solving is finding a good way to represent the problem data in the computer.

   a. This is true of computer programming in general - hence the emphasis on data structures in a CS curriculum.

   b. It is particularly true in AI, because the problems we are dealing with tend to be less structured.

      i. Where a good representation is available, the solution to the problem is often fairly easy.

      ii. Many important AI problems are currently “hung” on the problem of representation - e.g. representing commonsense knowledge is a major stumbling block for natural language systems.

2. Search is concerned with efficiently finding the “right” solution to a problem, given a large space of alternatives.

B. We will devote about two weeks to search later in the course. For now, we focus on knowledge representation.

II. The Knowledge Representation Problem

A. Knowledge representation is a key challenge in AI for several reasons.

1. Most computer problems involve data that has a fairly natural structure to it - e.g. the instance varables of a class or the columns of a relational database table. But AI problems are often require using knowledge that is not easily structured.

   Example: consider how you would represent everything you know about me.

   Do you know some things about me that you don’t know about other faculty members?

   ASK

   Do you know some things about other faculty members that you don’t know about me?

   ASK

2. Representing exceptional cases

   Example: In an isa hierarchy representing information about living creatures, we may choose to record "birds fly". This would allow us to infer from "tweety is a bird" that "tweety flies". But how do we handle the fact that certain kinds of birds (e.g. penguins) do not fly?

3. Representing knowledge that is incomplete, uncertain, or "fuzzy".

   Example: a medical diagnostic system may need to make use of a person's height. Sometimes we may know a person's exact height, while at other times we may know nothing at all about a person's height. Sometimes, we we may know just an approximation to it - e.g. that the person is "tall". (But what do we mean by "tall"? A person whose height is 6'5" is clearly tall, and one whose height is 4'11" is clearly not tall. Is a person whose height is 6'0" tall?)

   Our knowledge representation scheme has to be able to handle all these cases without "choking".

4. Representing knowledge that changes over time - the "Frame Problem".

   Example: Suppose we have a robot that can stack and unstack children's blocks. We wish to write a program that can plan the steps needed to accomplish a given task - e.g. Given the following configuration:

```
    -----
    | a |
    -----          -----
    | b |          | c |
 ---------------------------
```

   The initial configuration can be described by:

   on(a,b)
   on(b,table)
   on(c,table)
   clear(a)
   clear(c)

   Now suppose the robot puts block a on top of block c. In this case, some of our initial description has ceased to be true, while other parts remain true, and one new fact has become true. Which is each?

   ASK

5. Representing knowledge that agents possess about other agents - e.g.

   a. A particular statement may be true, but some agent may not know that it is true.

   b. Or, a particular statement may be false, but some agent may believe that it is true.

   c. In either case, reasoning about the agent's behavior requires the ability to represent the agent's knowledge and beliefs, which may differ in some ways from the true facts.

B. A good knowledge representation scheme should have the following qualities (here using terminology from Rich and Knight's intro AI book)

   1. It should be able to represent all of the knowledge needed for the kind(s) of problem it is used for. (Rich/Knight calls this representational adequacy.)

   2. It should facilitate inferring new knowlege from existing knowledge. (Rich/Knight calls this inferential adequacy). Ideally, the inference procedure would have the following properties:

a. Soundness - something can be inferred only if it is truly entailed by the knowledge base.

b. Completeness - anything that is entailed by the knowledge base can be inferred.

c. Decidability - if a statement is true, it should be possible to infer its truth in finite time; if it is false, it should be possible to infer its falsehood in finite time.

Unfortunately, it can be proven that if the kind of knowledge we need to represent is sufficiently rich, then an inference procedure that has all these properties cannot exist! At the very least, though, we usually insist on soundness. (This is related to Goedel's Theorem in mathematics and the halting problem in CS)

3. It should facilitate efficient means of accessing the specific knowledge that is relevant to a particular problem. (Rich/Knight calls this inferential efficiency).

4. It should support addition of new knowledge to the database as the program is running. (Rich/Knight calls this acquisitional adequacy.)

5. It should allow knowledge to be readily converted between its internal form and a form readable by humans (e.g. some English-like notation.) (This was not explicitly mentioned by Rich/Knight - might be called explanational adequacy)

C. Because knowledge representation is so important, much work has been done on it, and many schemes have been developed. These schemes typically take the form of a knowledge representation language and associated tools for maintaining a knowledge base. These tools can be highly-sophisticated.

D. For now, we will focus our attention on one major (and very important) representation scheme: formal logic - the predicate calculus.

1. In and of itself, the predicate calculus is very flexible - almost too flexible. But other systems may build on it by imposing some additional structure, as we shall see later.

2. It is the formal basis of PROLOG.

3. There are some issues (e.g. representing knowledge that is incomplete, uncertain, fuzzy, changing over time, or representing the knowledge and beliefs of particular agents) where predicate calculus itself is not adequate. There are, however, formal logics that do address some of these issues, as we shall see.