32002: AI (Non-monotonic Logic) **Spring 2014**

Lecture 6: January 09, 2014

Lecturer: K.R. Chowdhary *: Professor of CS (GF)*

6.1 Nonmonotonic Reasoning

Nonmonotonic logic is the study of those ways of inferring additional information from given information that do not satisfy the monotonicity property satisfied by all methods based on classical logic. In Mathematics, if a conclusion is warranted on the basis of certain premises, no additional premises will ever invalidate the conclusion.

In everyday life, however, it seems clear that we, human beings, draw sensible conclusions from what we know and that, on the face of new information, we often have to take back previous conclusions, even when the new information we gathered in no way made us want to take back our previous assumptions. For example, we may hold the assumption that most birds fly, but that penguins are birds that do not fly and, learning that Tweety is a bird, infer that it flies. Learning that Tweety is a penguin, will in no way make us change our mind about the fact that most birds fly and that penguins are birds that do not fly, or about the fact that Tweety is a bird. It should make us abandon our conclusion about its flying capabilities, though. It is most probable that intelligent automated systems will have to do the same kind of (nonmonotonic) inferences.

Some of the systems that perform such nonmonotonic inferences are - negation as failure, circumscription, modal system, default logic, autoepistemic logic, and inheritance systems.

6.2 Predicate Logic

The first Order Predicate Logic(FOPL) offers formal approach to reasoning that has sound theoretical foundations. This aspect is important to mechanize the automated reasoning process where inferences should be correct and *logically sound*.

The statements of FOPL are flexible enough to permit the accurate representation of Natural languages. The words - *sentence* or *well formed formula* will be indicative of predicate statements. Following are some of the translations of English sentences into predicate logic:

- English sentence: Ram is man and Sita is women.

  Predicate form: $man(Ram) \vee woman(Sita)$

- English sentence: Ram is married to Sita.

  Predicate form: $married(Ram, Sita)$

- English sentence: Every person has a mother.

Reorganized form of above: For all x, there exists a y, such that if x is person then x's mother is y.

Predicate form: $\forall x \exists y [person(x) \Rightarrow hasmother(x, y)]$

- English sentence: If x and y are parents of a child z, and x is man, then y is not man.

  $\forall x \forall y [[isparents(x, z) \land isparents(y, z) \land man(x)] \Rightarrow \neg man(y)]$

We note that predicate language comprises constants: {Ram, Sita}, variables {x, y}, operators: $\{\Rightarrow, \land, \lor, \neg\}$, quantifiers: $\{\exists, \forall\}$, and functions/predicates: {married(x,y), person(x)}.

To signal that an expression is universally true, we use the symbol $\forall$, meaning 'for all'. Consider the sentence "any object that has a feathers is a bird."Its predicate is:

$$\forall_x [hasfeathers(x) \Rightarrow isbird(x)].$$

Then certainly,

$$hasfeathers(parrot) \Rightarrow isbird(parrot)$$

is true. Some expressions, although not always True, are True at least for some objects In logic, this is indicted by 'there exists', and the symbol used is $\exists$. For example, $\exists_x [bird(x)]$, when True, this expression means that there is at least one possible object, that when substituted in the position of x, makes the expression inside the brackets True.

Following are some examples of representations of knowledge FOPL.

Example 6.1 *Kinship Relations.*

$mother(priti, namrata)$ *(That is, preeti's mother is namrata)*

$mother(bharat, namrata)$

$father(priti, rajan)$

$father(bharat, rajan)$

$\forall x \forall y \forall z \ father(x, y) \lor mother(x, z) \Rightarrow areparent(y, z),$

$\forall x \forall y \forall z \ mother(x, y) \land parent(z, y) \Rightarrow aresibling(x, z)$

In above, the predicate father(x, y) means x's father is y, and areparents(x, y) *is a function.*

References

[GFL09] D. George F. Ludger, "Artificial Intelligence - Structures and Strategies for Complex Problem Solving," *5th Edition, Pearson Education, India*, 2009, Chapter 2.

[PATT01] Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems," *PHI India*, 2001, Chapter 4.

[KRC12] Chowdhary, K.R., "Fundamentals of Discrete Mathematical Structures, 2nd Edition" *PHI India*, 2012, Chapter 6, 7.