**Nonmonotonic Reasoning.**

**Traditional systems based on predicate logic are monotonic . Here number of statements known to be true increases with time. New statements are added and new theorems are proved, but the previously known statements never become invalid.**

**In monotonic systems there is no need to check for in consistencies between new statements and the old knowledge. When a proof is made , the basis of the proof need not be remembered, since the old statements never disappear. But monotonic systems are not good in real problem domains where the information is incomplete, situations change and new assumptions are generated while solving new problems.**

**Non monotonic reasoning is based on default reasoning or “most probabilistic choice”. S is assumed to be true as long as there is no evidence to the contrary. For example when we visit a friend’s home, we buy biscuits for the children . because we believe that most children like biscuits. In this case we do not have information to the contrary. A computational description of default reasoning must relate the lack of information on X to conclude on Y.**

**Default reasoning ( or most probabilistic choice) is defined as follows:**

**Definition 1 : If X is not known, then conclude Y.**

**Definition 2 : If X can not be proved, then conclude Y.**

**Definition 3: If X can not be proved in some allocated amount of time then conclude Y.**

**It is to be noted that the above reasoning process lies outside the realm of logic. It conclude on Y if X can not be proved, but never bothers to find whether X can be proved or not. Hence the default reasoning systems can not be characterized formally. Even if one succeeds in gaining complete information at the moment, the validity of the information may not be for ever, since it is a changing world. What appears to be true now, may be so at a later time ( in a non monotonic system).**

**One way to solve the problem of a changing world to delete statements when they are no longer accurate, and replace them by more accurate statements. This leads to a non monotonic system in which statements can be deleted as well as added to the knowledge base. When a statement is deleted, other related statements may also have to be deleted. Non monotonic reasoning systems may be necessary due to any of the following reasons.**

**- The presence of incomplete inforamtion requires default reasoning.**

**- A changing world must be described by a changing database.**

**- Generating a complete solution to a problem may require temporary**

**assumptions about partial solutions.**

**Non monotonic system are harder to deal with than monotonic systems. This is because when a statement is deleted as “ no more valid”, other related statements have to be backtracked and they should be either deleted or new proofs have to be found for them. This is called dependency directed backtracking (DDB). In order to propagate the current changes into the database, the statements on which a particular proof depends, should also be stored along with the proof. Thus non – monotonic systems require more storage space as well as more processing time than monotonic systems.**

**Example: non-monotonic reasoning**

**Your are a student, it's 8am , you are in bed.**

**You slip out of your dreams and think: Today is Sunday. No classes today. l don't have to get up. You go back to sleep.**

**You wake up again. It's 9:30amnow and it is slowly coming to your mind: Today is Tuesday. What an unpleasant surprise.**

**P1 = today-is-Tuesday P2 = today-is-Sunday**

**P3 = have-class-at-10am P4 = no-classes**

**P5 = have-to-get-up P6 = can-stay-in-bed**