Knowledge and an introduction to it's application in distributed systems

Rahul V

September 5, 2012

Abstract

Knowledge and the concept of knowledge has been a topic of debate since ages and there exists no definition acceptable to everyone.Plato, once famously described knowledge as “justified true belief”. C. I. Lewis was the first scientist to represent knowledge in terms of formal logic.Epistemic modal logic is a subfield of modal logic that is concerned with representation of knowledge.In specific, *common knowledge* has been studied with interest in distributed computing because of its use in reasoning about some fundamental distributed systems problems like consensus, clocks etc. This essay provides an introduction to formal representation of knowledge and it's application in interpreting and proving a number of results in distributed computing.In specific, we shall look into how different types of clocks can be described in terms of formal epistemic knowledge. The problem of acheiving common knowledge in distributed systems and possible other types of knowledge with relaxed restrictions are discussed in this essay.

Contents

0.1 Knowledge

The concept of attempting to model usually falls into the meta physcial realms. And indeed, one of the starting blocks of modelling knowledge is the possible worlds model. According to theorists,there are a number of worlds existing parallely. Our's is only one of the worlds. Hence, if we consider a proposition $\phi =$ Dr. Sadagopan is IIITB's director, it is true in our world,which we also call as the actual world. Such propositions are called as true propositions. Similarly, there are notions of false propositions, which are propositions that are not true in our world. However, a proposition $\phi =$ Rahul is the director of IIITB could be true in some other world. There are also some propositions which are impossible like message being received before it was sent and some propositions which are true irrespective of the world like 2+2=4. This theory is also called as Possible world theory.Possible world theory[1]

0.1.1 Hierarchy of knowledge states

Knowledge representation for a group of agents is different from propositional logic because of the additional complexities which are introduced because of the one agent knowing how much the other agent knows. When we say, an individual has some knowledge of ϕ, it simply means he knows it. However, when we say, a group G has knowledge ϕ, a number of possibilities exist. Let us say, an agent A_i knows ϕ. It could either mean that A_i can deduce ϕ from whatever he already knows or the proposition is stored directly in the agent A_i. We denote the agent A_i's knowledge by κ_i

However, if a group G of n agents has to know ϕ, it could result in a number of possibilities:

1. Distributed knowledge D_G of ϕ
   Distributed knowledge is commonly known as the wisdom of the crowds or community knowledge. Suppose each agent A_i has knowledge ϕ_i. However,each of the ϕ_i alone may not be able to solve the problem. The entire knowledge of all the agents has to be combined to solve the problem.Suppose, I and my friend want to find if there is class today(ω). Let us assume there are no classes on Saturdays on Friday.My friend A_j knows that $\phi_j =$ There is no class today. I,(A_i) know that phi_i= Today is a Saturday. Hence, individually, we would have never been able to know if there was class today(ω). However,combining each other's knowledge we are able to arrive at the right conclusion.

2. Someone in G knows ϕ
   We can say that someone in G knows ϕ if there is any agent A_i which knows ϕ. Formally, it can be represented as, $S_G = \bigvee_{i \in G} K_i\phi$. Continuing with the same classroom anology, if one of our friends knew that there is no class today, it would be an example of someone in G knew ϕ.

3. Everyone in G knows ϕ
   In this model, all the agents in G know that proposition ϕ is true. Formally, $E_G\phi = \wedge_{i \in G} K_i\phi$.

[1] http://en.wikipedia.org/wiki/Possible_world

4. $E_G^k\phi$ knowledge in G
   $E_G^1\phi = E_G\phi$.
   $E_G^k\phi = E_G(E_G^{k-1})\phi$.
   In this level of knowledge hierarchy E_G^k, each agent knows that,every agent has knowledge about,every agent has knowledge about,....k times.

5. Common knowledge
   In the next section, we will dwell more upon common knowledge. Let us briefly introduce common knowledge here. The proposition ϕ is said to be common knowledge across a group G if each agent has the knowledge that,
   $C_G\phi$=$E_G^1(\phi) \wedge E_G^2(\phi) \wedge E_G^3(\phi) \wedge E_G^k(\phi)$.
   where, k > 0. k is usually infinite. However, the notion of infinity can be approximated to some number for which the agent cannot keep track.

All these knowledge levels form a hierarchy which can be represented in the form:
$\phi \subset D_G\phi \subset S_G\phi \subset E_G\phi \subset C_G\phi$.
Thus, common knowledge has the strictest requirements.

0.1.2 Properties of knowledge

There are a number of properties of knowledge which are defined formally. Some of these axioms sound intutive.These properties hold for all the formulas described by modal logic.

- Knowledge axiom
  $K_i\psi \Rightarrow \psi$
  It is always assumed that whatever knowledge I have is true. Note that it is important to differentiate knowledge from beliefs which may be false.

- Positive Introspection
  $K_i\psi \Rightarrow K_iK_i\psi$
  Every agent knows what it knows something.

- Negative Introspection
  $\neg K_i\psi \Rightarrow K_i\neg K_i\psi$
  Every agent knows what it does not know. I am not sure how this axiom holds, since it appears to be a contradiction to the open world problem, where we cannot make statements about what we do not know.

- Knowledge generalization
  A valid formula or a fact ψ is true for all formulas in all the worlds. Hence, it is assumed that all the process know this fact $k_i\psi$.

- Distribution Rule
  Each agent knows the logical consequences of its logical knowledge.
  $K_i \wedge K_i(\phi \Rightarrow \psi) \Rightarrow K_i\phi$

0.2 Common Knowledge

With a general idea of the formal methods for defining and representing knowledge, it is important to study common knowledge for it's various applications in distributed systems and computing in general. Common knowledge is used for reasoning about a number of problems in distributed computing like consensus, clocks etc. In a group G of n agents, if all the agents know ψ and all the agents know that all the other agents know ψ, and all the agents know that all the other agents know that... till infinity, then such knowledge is called common knowledge. The concept of common knowledge and knowledge, in general can be better understood through a puzzle famously known as the muddy children problem. We describe a basic variant of the problem.

0.2.1 The muddy children example

There are a number of variants of the muddy children problem.Muddy children problem[2] Several children are playing together outside. After playing they come inside, and their mother says to them, at least one of you has mud on your head. Each child can see the mud on others but cannot see his or her own forehead. She then asks the following question over and over:

can you tell for sure whether or not you have mud on your head and atleast one of you have mud in your forehead?

Assuming that all of the children are intelligent, honest, and answer simultaneously, what will happen? In this assignment, we will analyze this puzzle. To get a feeling for what is being asked, we now figure out what happens if there are two children. First, suppose that exactly one is muddy. When the mother asks the question, the muddy child sees no mud on the other child, and then can conclude that he has mud on his forehead. The other child cannot tell whether or not she has mud on her forehead. Now, suppose that both children have mud on their forehead. When the mother asks the question, neither can determine if they have mud on their foreheads since they see they other the child with mud. So, neither can answer yes to the question. Now, when the mother asks the question the second time, both children realize that they must have mud on their head; if either didn't have mud on their head, then the other child would have seen this and would have been able to answer yes the first time the mother asked the question. So, both answer yes to the second time the question is asked. Now this is possible only if their mother had told them mud is there on the forehead of atleast one child. Else, the children would not be able to figure out that they have mud on their face. It is not only important for every kid to know that someone has mud in his face.It is also important that every child knows that every other child also knows it and so on.

0.2.2 Describing Common Knowledge formally

0.3 Agreement in asynchronous systems

The agreement problem can be described as " Two problems need to communicate among themselves to decide on a binary value. Can we design a protocol to ensure this?". Reaching a consensus can be regarded as attaining common knowledge.

There does not exist a protocol for two processes to reach common knowledge about a binary value in an asynchronous message passing system with unreliable

[2] `http://sierra.nmsu.edu/morandi/coursematerials/MuddyChildren.html`

knowledge.
Informally consider two processes, P_i and P_j and P_i sends a message to P_j. Now, P_i does not know if P_j has received the message or no.If P_j sends back an ACK, it will not know if P_i has received it back. This protocol can never converge to a value in finite number of iterations.

0.4 Other variants of common knowledge

Since the Two general problem and the above section proves that it is impossible to achieve common knowledge in asynchronous environments.Hence, in such environments, the strict definition of common knowledge could be relaxed in different directions to form notions of other relaxed defitions or variants of common knowledge. Some of the other types of knowledge reprsentations are:

- Epsilon common knowledge
  In this type, all the processs agree on a certain value after a point of time.In this system model, we assume that a sent message is received within ϵ time interval. Hence, when a message is sent, within ϵ time interval, the message becomes common knowledge. Thus E^{ϵ} represents that a peice of knowledge would become common knowledge within ϵ time units.

- Eventual common knowledge

  Let us consider systems where communication is asynchronous and there is no bound in message delivery time.When a process sends a broadcast message sent(m), all the processes eventually will receive the message m. Thus, every process knows that a message m would be received by all processes and it will become common knowledge. This is known as eventual common knowledge as there is no fixed as when will the message m become common knowledge.

- Concurrent Common knowledge

  From Lamport's idea of using casuality to create clocks in distributed system, the idea of concurrent knowledge is to define the notion of common knowledge in asynchronous systems where global clocks are not possible. Representing a real time global state in asynchronous systems can be simulated by considering the set of all possible local states in the system, which can be obtained by a consistent cut. We define everyone concurrently knows ψ if each process knows ψ and knows that every other process also knows that all other process knows ψ and so on.
  Thus, concurrent common knowledge is implemented by using logical locks like vector clocks.matrix clocks etc..

0.5 Clocks and knowledge

In the logical clocks that we consider, our clocks are functions which keep increasing monotonically.In such systems, the global state of a system is represented by the set of values of local clocks of all the systems. These clocks are updated by piggybacking timing information along with the clocks. These updates represent transfer of knowledge between the processes.

Vector clocks are clocks where each process P_i stores an array $A_i[1...n]$ to represent P_i's knowledge about the local states of all the other processes. When a message m_j is received by process P_j from process P_i, m_j also updates process P_i's vector clock about the latest clock values of all other processs P_k.

Matrix clocks go one step further and at each process, a matrix clock is represented by $M_i[1...n][1...n]$. Each entry $M_i[j,k]$ in matrix clock represents process P_i's knowledge about process what process P_j knows about the clock value of process P_k.Thus, while vector clocks can be thought of as imparting knowledge to a process (i.e. when $A_i[j] = x$, at process P_i then we can be sure that every process P_j has executed atleast x events.)Vector clocks represent one more step of knowledge. Let us represent them formally to understand the differences.

A vector clock $A_i[1...n]$ can be represented in modal logic by $K_i K_j \psi_j$ where ψ_j is the local component of process P_j's clock. Similarly, the matrix clock $M_i[j][k]$ can be represented by $K_i K_j \psi_k$, where ψ_k represents the local component of process P_k. Thus, a vector clock represents $E^0\psi$ where ψ is global state of the system while matrix clock for process P_k represents $K_j(E^1(\psi))$.

0.6 Conclusion

Thus, this essay has made an attempt to describe some of the formalims involved in modelling knowledge in systems involving multiple agents. It also provides a brief overview on the different types of knowledge and the importance of common knowledge in solving many of the problems in distributed sytems. Finally, we have tried to reason out the working of vector and matrix clocks in terms of knowledge. I hope that thinking in terms of knowledge, will give a better persepective while designing distributed systems in the future.

Bibliography

[1] Joseph Y. Halpern, Yoram Moses, *Knowledge and Common knowledge in Distributed systems.*. CoRR

[2] Ajay D Kshemakalyani,Mukesh Singhal *Distributed Computing* Cambridge