

Reasoning about knowledge in multi-agent systems

Joe Halpern
Cornell University

Collaborators:
Ron Fagin, Yoram Moses, and Moshe Vardi

Reasoning about knowledge and beliefs is crucial in areas such as:

- AI (knowledge bases, planning systems, etc.)
- distributed systems
- cryptography
- economics
- linguistics
- ...

The focus of this talk is on distributed systems (i.e., a group of processors, people, robots, ...).

Parallel vs. Distributed Computing

Many similar concerns (multiprocessing, concurrency, synchronization, interprocess communication). But ...

- Parallel computing emphasizes *performance*
- Distributed computing is largely concerned with *uncertainty*:
 - Whether or not messages arrive
 - In what order messages arrive
 - Which processors are faulty

A good way to analyze this uncertainty is in terms of *knowledge*, and how this knowledge changes over time (due to communication)

The muddy children puzzle

We can prove by induction on k that if k children have muddy foreheads, they say “yes” on the k^{th} question.

It appears as if the father didn't tell the children anything they didn't already know. Yet without the father's statement, they could not have deduced anything.

So what was the role of the father's statement?

When reasoning about the knowledge of a group of agents, states of “group knowledge” become relevant:

- distributed knowledge
- everyone knows
- everyone knows that everyone knows
- everyone knows that everyone knows that everyone knows . . .
- common knowledge

We have a hierarchy:

$$Cp \Rightarrow \dots \Rightarrow E^3p \Rightarrow E^2p \Rightarrow Ep \Rightarrow K_ip \Rightarrow Dp$$

Communication moves you up the hierarchy.

Deadlock detection algorithms convert a situation where the group has distributed knowledge of the deadlock to one where everyone knows about it (and so can take appropriate action).

Communication conventions must be common knowledge.

Agreement requires common knowledge

The father gives the children *common knowledge* of the fact that at least one child has a muddy forehead.

The “classical” model

The *possible worlds* model (over 40 years old!):

Besides the actual state of affairs, an agent considers a number of other states of affairs to be possible.

An agent *knows* a fact p if p is true in all the states of affairs, or worlds, that he thinks possible.

Knowledge in multi-agent systems

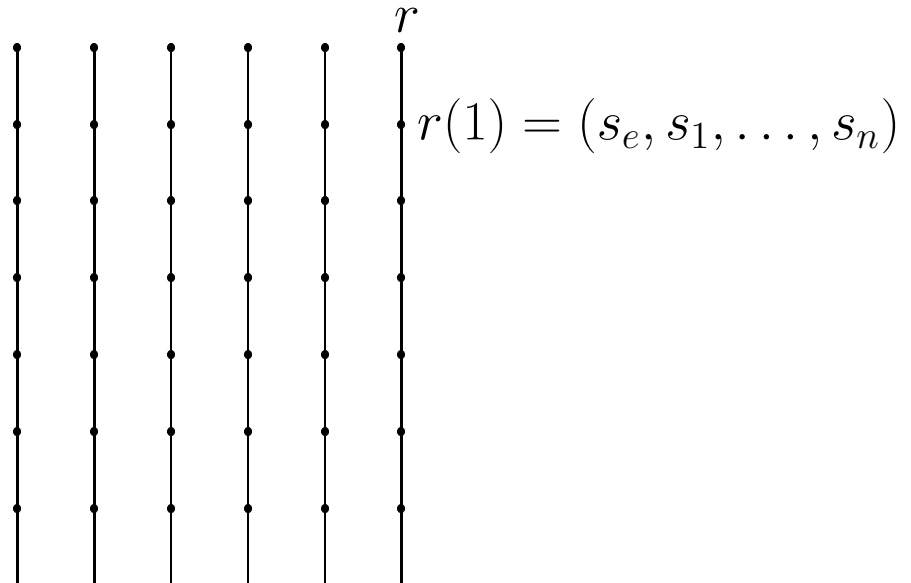
A multi-agent system consists of a collection of communicating agents (people, robots, processes).

Each agent has a local state (depending on the initial state, messages received, etc.). The *global state* of the system is a tuple consisting of each agent's (local) state.

A *run* of the system is a complete description of the system over time: a function from times to global states.

A protocol is described by a set of *runs*: each one describes what one possible execution of the protocol. At time m in run r , the system is in some global state.

A system:



An agent *knows* a fact p at some point (r, m) if p is true at all the points (r', m') it considers possible; i.e., at all the points (r', m') where it is in the same local state.

This is an *external* definition of knowledge. Agents cannot answer questions based on their knowledge. Nevertheless, it is useful for analyzing distributed systems.

The coordinated attack problem

Each time the messenger makes it, the level of knowledge rises.

Let $m =$ “General R sent a message saying ‘attack at dawn’ ”

First $K_G m$, then $K_R K_G m$, $K_G K_R K_G m$, ...

Proposition: (Halpern-Moses) m will never become common knowledge using a k -round handshake protocol.

Theorem: m will never become common knowledge in any run of any protocol. In fact, common knowledge is not attainable in any system where communication is not guaranteed.

But what about coordinated attack?

Agreement implies common knowledge.

Corollary: Any protocol that guarantees that if one of the generals attacks, then the other does so at the same time, is a protocol where necessarily neither general attacks.

(N.B. We need to also assume that in the absence of messages, neither general will attack.)

We have shown that common knowledge is not attainable if communication is not guaranteed.

We can easily show that common knowledge is also not attainable if communication is guaranteed, but there is no upper bound on message delivery time.

What if there is an upper bound on message delivery time, but the actual message delivery time is uncertain?

Suppose we have an upper bound of ϵ , but messages might take anywhere from 0 to ϵ to arrive:

- At time $t_R + \epsilon$, have $K_R K_D m$
- At time $t_D + \epsilon$, have $K_D K_R K_D m$
- At time $t_R + 2\epsilon$, have $K_R K_D K_R K_D m$
- ...

$Cm?$ – Never!

The situation is very different if there is a global clock and the messages are timestamped:

If R2 says “ m ; the time is 5 P.M.”, this message becomes common knowledge at $5 + \epsilon$.

Theorem: Common knowledge requires synchronized clocks.

Corollary: In any system where message delivery time is uncertain and clocks are not initially synchronized, common knowledge is not attainable.

Conclusion?

Although common knowledge is a desirable and consistent state of knowledge, it is not attainable in practical systems. A paradox?

Two possible resolutions

1. It's all a modeling problem:
 - Whether or not we attain common knowledge depends in part on the granularity at which we model time
2. We don't really need common knowledge
 - Weaker variants often suffice

The granularity of time

- If we model time at a finer grain, the muddy children don't gain common knowledge either.
- Is it “safe” to model time in a coarse way?
 - It depends on the specifications of the problem
 - In the case of the muddy children, it is (they still give the right answer)

Consider the following “eager protocol”:

- R2 sends m to D2 (at t_R) and claims “ Cm ” at t_R .
- D2 claims Cm upon receiving m

R2 and D2 are lying. Nevertheless, this essentially amounts to viewing communication as happening in rounds, with receiving and sending occurring in the same rounds. *Under reasonable assumptions this is safe.*

This may help to explain how people can act as if they have common knowledge

Weaker (attainable) variants of common knowledge

- Epsilon common knowledge:

$$C^\epsilon p \equiv \bigcirc^\epsilon E p \wedge \bigcirc^\epsilon E \bigcirc^\epsilon E p \wedge \dots$$

(Fixed point of $C^\epsilon p \equiv \bigcirc^\epsilon E C^\epsilon p$)

- attainable when there is a bound of ϵ on message delivery time.

- Eventual common knowledge:

$$C^\diamond p \equiv \diamond E C^\diamond p$$

- Appropriate when there is no bound on message delivery time.

- Time stamped common knowledge

$$C^{TS} p \equiv E^{TS} C^{TS} p$$

- For systems with clocks.

Can also have probabilistic variants and combinations.

Using eventual common knowledge

Theorem: Eventual common knowledge is not attainable in any system where communication is not guaranteed.

Back to coordinated attack ...

Corollary: Any protocol that guarantees that if one of the generals attacks, then eventually the other one will is necessarily a protocol where necessarily neither general attacks.

Final conclusions and other work

- It is occasionally useful to leap to conclusions
- Various relaxations of common knowledge are attainable and useful
- Reasoning about knowledge is a useful way to understand, reason, and design protocols. (Note: this technique has now been successfully used to analyze a number of protocols)
- This may also be a useful tool for specifying and synthesizing protocols.
- *Knowledge-based programs* are a useful high-level tool for describing and thinking about protocols; actions depend explicitly on knowledge
- This notion of knowledge assumes perfect reasoners; lots of work on getting resource-bounded notions of knowledge.
- The model can easily be extended to incorporate time and probabilities.

Shameless advertising

- R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi, *Reasoning About Knowledge*, MIT Press, 1994
- Talk on more recent work, “From Statistics to Beliefs”, in Mathematical Sciences 6627 at 4 PM