Knowledge in Multi-Agent Systems

We have introduced interpreted systems $\mathbf{I} = (R, \pi)$ and its semantics $(\mathbf{I}, r, m) \models A$ for all formulas, where R is a set of runs and π is a function on global states which gives the truth of primitive propositions at a point (r,m). Thus, the truth of a primitive proposition q at a point (r,m) depends only on the global state $r(m)$.

This seems like a natural assumption. Quite often, in fact, the truth of a primitive proposition q of interest does not depend on the whole global state, but only on the component of some particular agent.

For example, the truth of a statement "process 2 received from process 1 a message" might depend only on process 2' state. In that case, we expect π to respect the locality of q, i. e. if $s, s' \varepsilon$ $\boldsymbol{G}$ and $s \sim_i s'$, then $\pi(s)(q) = \pi(s)(q)$.

We can also imagine statements that depend on more tha n just one global state. Consider, for example, a statement "at some later point in the run the variable *x* is set to 5". There could be two points (*r, m*) and (*r', m'*) with the same global state, such that this statement is true at (*r, m*) and false at (*r', m'*).

Thus, such a temporal cannot be represented by any formula in our language. This is a consequence of the following proposition.

Proposition. For every formula $A \varepsilon L_n^{CD}$, if $r(m) = r'(m')$, then $(\mathbf{I}, r, m) \models A$ iff $(\mathbf{I}, r', m') \models A$. (1)

Proof. We proceed by the complexity of the formula A .

- If A is a primitive proposition q , (r, m) and (r', m') are equal global states then we have (I, r, m) |= q iff $(\mathbf{I}, r', m') \models q$ since $\pi(r, m)(q) = \pi(r', m')(q)$
- If A is a formula $\neg B$, from the induction hypothesis we have (I, r, m) |= B iff $(\mathbf{I}, r', m') \models B$ and hence A is not true either in (I, r, m) nor in $(\mathbf{I}, r', m')$. Hence (1) holds.
- We proceed similarly if A is the formula B & C .

- If A is a formula K_iB then $r(m) = r'(m')$ implies that every state $s \varepsilon G$ connected with $r(m)$ by an edge in K_i is connected with $r'(m')$ by the same edge. Thus , if $(\mathbf{I}, s) \models B$ for every such s , we have the same for every s connected with $r'(m')$.
Thus, $(\mathbf{I}, r, m) \models K_iB$ iff $(\mathbf{I}, r', m') \models K_iB$ and (1) holds for every i.

- To get (1) for $C\,B$ and $D\,B$ we proceed by induction on E^n .

While we could deal with this problem by allowing the truth of primitive proposition to depend on the point, and not just the global state, the more appropriate way to express such temporal statements is to add modal operators for time into the language.

Definition. (Validity of formulas in interpreted systems)

(i) We say that *a formula* A *is valid in the interpreted sytem* $\mathbf{I} = (R, \pi)$, if $(I, r, m) \models A$ for all points (r, m) in **I**.

(ii) For a class C of interpreted systems, we say that a formula A *is valid* in C , and write $C \models A$ if A is valid in every interpreted system $\mathbf{I} \; \varepsilon \; C$.

We now have a concrete interpretation for knowledge in multi-agent systems. As we already have said, this interpretation of knowledge is an *external* one, ascribed to the agents by someone reasoning about the system. We do not assume that the agents compute their knowledge in any way, or that they can answer questions based on their knowledge.

Note that this notion of knowledge satisfies all the axioms of S5, since $\sim_i$ is an equivalence relation. We have seen already that the Distribution (Kripke) Axion and the Generalization Rule both hold. We have seen that these propertie hold in every Kripke structure no matter how we define the K_i relation in M_l As a consequence, agents know all logical consequences of their knowledge and they know all valid formulas.

Recall that we allow the agents in our system to be processed in a distributive system. It may be strange to view such inanimate agents as possessing knowledge and, in fact as being "logically omniscient".

Nevertheless, the above definition of knowledge is consistent with at least one way it is used in practice. For example, when someone analyzing a distributed protocol says "process 2 does not know that process 3 is faulty at the end of round 5 in run *r* ". This mean s that there is a point at which process 3 is faulty, which is indistinguishable to process 2 from the point (*r,* 5).

There are certain applications, however, for which the externally ascribed knowledge is inappropriate. We shall consider an example involving *knowledge bases* , where it may be more appropriate to consider bases's knowledge as *beliefs*.

The difference consists in the fact that the agent can *believe* some facts that are not true. Hence the Axiom of Truth does not hold for beliefs.

It can be shown that by using a slightly different K_i relations instead of $\sim_i$, we do get a reasonable notion of belief.

Of course, we still have the problem of logical omniscience.

To start with somethig now, we will still use the external notion of knowledge to analyze multi-agent systems.

Example 2. (The bit transmission problem analyzed in informed systems)

Consider the bit-transmission problem again. Now we take Φ to consist of six primitive propositions:

bit = 0, *bit* = 1, *recbit, recack, sendbit* and *sendack* , representing two assertions about *S'*s initial value of bit, assertions saying that *R* has received *S'*s message, *S* received *R*'s acknowledgment, *S* has just sent a message, and *R* has just send a message, respectively.

The appropriate interpreted system is $\mathrm{I}^{bt} = (R^{bt}, \pi^{bt})$ where R^{bt} , consists of the set of runs from Eample 1, and π^{bt} is such that

- $(R^{bt}, r, m) \models bit = k$ if $r_S(m)$ is either k or (k, ack), $i = 1, 2$

- $(R^{bt}, r, m) \models$ *recbit* if $r_R(m)$ is not λ ,
- $(R^{bt}, r, m) \models$ *recack* if $r_S(m) = (0, ack)$ or $r_S(m) = (1, ack)$,
- $(R^{bt}, r, m) \models$ *senbit* if the last tuple in $r_e(m)$ is either (`sendbit`, Λ) or (`sendbit` , `sendack`), and
- $(R^{bt}, r, m) \models$ *sendack* if the last tuple in $r_e(m)$ is either (Λ, `sendack`) or (`sendbit` , `sendack`) .

Note that the truth valueof all these primitive propositions is completely determined by the global state, since we assumed the environment's state records the events taking place in the system.

In fact it is easy to see that *bit* = 0, *bit* = 1, and *recack* are local to *S* and that *recack* is local to *R .*

For the reminder of our dicussion in this example, we need only the primitive propositions *bit* = 0 and *bit* = 1 , the other primitive proposition will be useful later.

Just as the way we choose the model the local states in the system depend on the analysis we plan to carry out, the same applies to the choice of primitive propositions.

Intuitively, after R receives the S's bit, then R *knows* the value of the bit. Indeed it is easy to check that if (r, m) is a point such that $r_R(m) = k$ and k is not λ , i.e. R received S's bit by that point, then $(I, r, m) \models K_R(bit = k)$. This is because at all other points (r', m') , if $(r, m) = (r', m')$, then S must have initial bit k at (r', m') .

Similarly, when S receives R's *ack* message, then S knows that R knows the initial bit. More formally, if $r_S(m) = (k, ack)$, then $(I, r, m) \models K_S K_R(bit = k)$.

It is easy to see, in this setting, if S stops sending messages to R before S knows that R knows the value of the bit

i.e. before either $K_S K_R(bit = 0)$ or $K_S K_R(bit = 1)$ holds, then is is possible that R will never receive the bit.

Although we do not provide a formal proof of thei fact here, this observation already suggests the power of the knowledge-based approach. It allows us to relate actions, such as sending a message or receiving a message, to state of knowledge, and then use the states of knowledge as a guide to what actions should follow. We shall investigate these issues in greater detail later

Incorporating Time

Motivation.

Example 1 has shown that our language is not expressive enough to handle conveniently the full complexity of even simle situations. For example, we might want to make state-ments like “the receiver event-ually knows the sender’s initial bit”. We have already observed that we cannot express such temporal statements in our language.

To be able to make temporal statements, we extend our language by adding *temporal operators*, which are new modal operators for talking about time. From the variety of such operators, we focus here on four of them.

The operators, we use to extend our language are

- □ (“always”), its dual,
- <> (“eventually”),
- O (“next time”) and
- $\underline{\mathrm{U}}$ (“until”)

Intuitively,   *A* is true if *A* is true now and at all later points; <>*A* is true if *A* is true at some point in the future; O*A* is true if *A* is true at the next step; and *A* $\underline{\mathrm{U}}$ *B* is true if *A* is true until *B* is true.

More formally, in interpretated systems, we have

$(\mathrm{I, r, m}) \models \Box A \quad \text{iff} \quad (I, r, m') \models A \text{ for all } m' \underline{\geq} m,$

$(I, r, m) \models \mathord{<}\mathord{>}A \quad \text{iff} \quad (I, r, m') \models A \text{ for some } m' \underline{\geq} m\ ,$

(I, r, m) |= A iff (I, r, m+1) |= A, and

(I, r, m) |= $A \,\underline{\mathrm{U}}\, B$ iff (I, r, m') |= B for some $m' \geq m$ and
. (I, r, m'') |= A for all m'', $m \leq m'' < m'$

Note that our interpretation of *OA* as "*A* holds at the next step" makes sense in discrete time (which is our case). All the other temporal operators make perfect sense even for continuous time.

It is easy to show that

□A <-> ¬<>¬A

<>A <-> ¬□¬A

<>A <-> *true* $\underline{\mathrm{U}}$ A

Thus, we can take O and $\underline{\mathrm{U}}$ as basic operators, and define □ and <> in terms of $\underline{\mathrm{U}}$.