MODAL LOGIC AND AGENCY
PART 2: INTERPRETED SYSTEMS

Alessio Lomuscio

Department of Computer Science
King's College London

alessio@dcs.kcl.ic.ac.uk

Overview

- Modal Logic and Multi Agent Systems.
- Is there a problem with Kripke Semantics?
- Interpreted Systems: A grounded semantics for time and knowledge.
- An example: the bit transmission problem.

Introduction

So far we have seen the following:

- Syntax for mono- and multi-modal modal logic.
- Possible worlds semantics for modal logic.
- Axiomatisations, completeness.
- Epistemic Logic.

The most common methodology to specify aspects of a MAS is by means of syntax, or proof theory. Essentially we define a set of formulas and we see those as a specification for the MAS system, i.e., it specifies the properties that we would like the system to exhibit.
In computing though, an equal and perhaps more important status is given to semantics. We have studied how semantics can be used to reason about the muddy children puzzle. This was done on the semantics of Kripke models.
One observation that has been made is that possible worlds semantics suffer from the problem of being *ungrounded*, that is that no clear correspondence can be drawn between a Kripke model and the state of a computing system that it aims to model. This is seen as a limitation as we lose the power of semantics to describe the computational system.

“Interpreted systems” as one of the answers

Many semantics have been investigated to answer the criticism above. One which we will describe in some detail is *interpreted systems*. Interpreted systems allow us to reason about temporal and epistemic properties of multi-agent systems.
Think of a multi-agent system as a distributed system in which different agents (autonomous processes) participate in performing some task, not necessarily in cooperation. Let us assume from now on that we have a set of agents $A = \{1, \ldots, n\}$; and let us assume they operate in some *environment*. Further assume that each agent i has a set of *possible local states* L_i in which it can be. These states model the instantaneous configuration of an agent in the system, and their content varies depending on what we aim at modelling. It may contain the values of some variables, the facts of a database, etc. The environment’s state can also vary in principle and so we associate a set of states L_e to the environment.
Although we may think the environment as just another agent, it should be seen as conceptually different as it is meant to capture relevant information that is not related to a specific agent, such as messages in transit in a communication channel, etc.

Systems of Global states

Definition 1 *Given a set of agents A, their corresponding sets of local states $L_1, \ldots, L_n$, and a set of local states for the environment L_e, the* set of global states *for the system is defined as $G \subseteq L_e \times L_1 \times \ldots \times L_n$.*

Intuitively a global state $s = (l_e, l_1, \ldots, l_n)$ represents a situation in which agent 1 is in state $l_1 \in L_1$, agent 2 is in state $l_2 \in L_2$, $\ldots$, and the environment is in state $l_e \in L_e$.
This gives us a snapshot of the system, but a system is not static but it changes continuously. To model this we define *runs* over global states.

Definition 2 *Given a set of global states G, a* run *is a function $r : IN \rightarrow G$, where IN is the set of the natural numbers.*

The set IN will intuitively represent the system time. Note that in the definition above we assume deterministic evolution. We will sometimes write $s = r(m)$ to mean that the global state s occurs in run r at tick m.

Definition 3 (Interpreted system) *Given a set of global states S, an interpreted system is defined as $IS = (R, h)$, where R is a nonempty set of runs over S, and $h : P \rightarrow 2^S$ is an interpretation for the atoms.*

So far we have defined a grounded semantics to describe the evolution of local and global states of a Multi-agent system. In the following we define how to interpret a temporal epistemic language on it. This is because the notions of time and knowledge are crucial for modelling multi-agent systems.

Knowledge in Interpreted Systems

Remember earlier in the course we discussed how the modal box could be interpreted as a knowledge operator representing the epistemic state of an agent. In that setting we said that the accessibility relation between Kripke worlds relates worlds which are *epistemically equivalent*, i.e. that wRw' is intended to mean that up to the agent's knowledge w and w' are equivalent (he or she cannot distinguish between the two). Can we relate this idea to the interpreted system setting?
Note that here we are in a multi-agent setting so, intuitively we will have a *family* of modalities (and accessibility relations) not just one. Technically this will not be a problem.
Consider two global states
$s = (l_e, l_1, \ldots, l_n), s' = (l'_e, l'_1, \ldots, l'_n)$. When can we say that an agent considers them to be epistemically equivalent? Well, if its local state is the same in the two global states, then surely it is not able to distinguish between the two. More formally:

Definition 4 (Model associated with an IS) *Given an interpreted system $IS = (R, h)$, the model $M_{IS} = (S, \sim_1, \ldots, \sim_n, h)$ is defined as follows:*

- *S is the set of global states for IS,*
- *For any $i \in A$, and any $s = (l_e, l_1, \ldots, l_n), s' = (l'_e, l'_1, \ldots, l'_n) \in S$ we have that $s \sim_i s'$ if $l_i = l'_i$.*
- *$h : P \to 2^S$ is a valuation for the atoms.*

Note that for any $i \in A$ the accessibility relation $\sim_i$ is an equivalence relation.
Consider a multi-modal language $\mathcal{L}_K$ defined by:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \qquad (i \in A),$$

where the modality K_i represents the knowledge of agent i.
This is an external bird's eye notion, i.e., it is something that we as external observers attribute to agent i. We can formalise the discussion so far in the following:

Definition 5 (Satisfaction) *$(IS, s) \models \varphi$ if $(M_{IS}, s) \models \varphi$, where the indexed modalities are interpreted on the corresponding relations.*

We now have a way of interpreting knowledge operators on interpreted systems. Notice that interpreted systems are *valued structures*, i.e. they have an inbuilt valuation. If we want to provide axiomatisations of classes of interpreted systems we can reason about the underlying unvalued structures: the set R of runs.

Definition 6 *A formula φ is valid on an interpreted system IS if for any $s \in S$ we have that: $(IS, s) \models \varphi$. Given a set of runs R we say that R validates φ if for any valuation h, we have that $(R, h) \models \varphi$. Given a class of sets of runs $\mathcal{R}$, we say that $\mathcal{R}$ validates φ, if for any $R \in \mathcal{R}$ we have that $R \models \varphi$.*

Note the relation of this definition with respect to validity of a formula on a model, on a frame, and on a set of frames.
In the following we will identify classes of runs on interpreted systems that will form complete semantic classes for different

axiomatisations. Before we do this we look at the most general class of interpreted systems.
Consider the logic $S5_n$, obtained as follows.

Definition 7 *L_K is the smallest set of formulas which includes the following (i is an index over $A = \{1, \ldots, n\}$.*

- $Taut \subseteq$ *S5$_n$,*
- $K_i(p \rightarrow q) \rightarrow (K_i p \rightarrow K_i q)$*,*
- $K_i p \rightarrow p$*,*
- $K_i p \rightarrow K_i K_i p$*,*
- $\neg K_i p \rightarrow K_i \neg K_i p$*,*
- *if* $\vdash_{S5_n} \varphi$*, then* $\vdash_{S5_n} K_i \varphi$*,*
- *Modus Ponens,*
- *Uniform substitution.*

We have the following, that we assert without proof.

Theorem 1 *The logic L_K is sound and complete with respect to the most general class of interpreted systems.*

So the epistemic properties of agents in interpreted systems can be modelled by considering n copies of the modal logic S5.
Note that the logic is supposed to provide a bird's eye view of the system and should not be seen as a model of the reasoning capabilities of the agents.
Also note that because of axiom T, we have
$\vdash_{S5_n} K_i p \rightarrow \neg K_j \neg p$. So there are some interaction patterns between the knowledge of the agents. These can be

strengthened by taking logics stronger than S5$_n$. For example if we wanted to model a multi-agent systems in which one agent (say agent i) knew more than everybody else we could consider the logic S5$_n \cup \{K_j p \rightarrow K_i p\}$ for all $j \in A$. A currently active area of research focuses on the study of knowledge properties (in terms of axiomatisations) of particular classes of interpreted systems. The most interesting semantical classes result from the interaction of time and knowledge. We will see some of these later on.

Modelling time in interpreted systems

We have seen how to import epistemic logic into the semantics of interpreted systems: what we did was to define a family of relations on global states that were then used to interpret the K operators. As they were defined, runs are ideal for interpreting temporal operators. Let us consider the following language $\mathcal{L}_T$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi \mathcal{U} \psi.$$

Here $\bigcirc$ represents the *next-step operator*. Intuitively $\bigcirc\varphi$ is true if at the next step φ is true. $\mathcal{U}$, the *until-operator*, is a binary modal operator. Intuitively $\varphi \mathcal{U} \psi$ codes a situation in which φ is true from the current instant on time up to a certain time, after which ψ is always true. Similarly to what we have done for knowledge, we can now define a (temporal) model associated with a system to give an interpretation to the language $\mathcal{L}_T$. But this would be isomorphic to runs (the relations would code the next step), so we can interpret the language directly on interpreted systems.

Definition 8 *Given an interpreted system IS, satisfaction for the language $\mathcal{L}_T$ is defined as follows (basic boolean connectives are omitted).*
$(IS, r(m)) \models \bigcirc\varphi$ *if* $(IS, r(m+1)) \models \varphi$.
$(IS, r(m)) \models \varphi \mathcal{U} \psi$ *if* $(IS, r(m')) \models \psi$ *for some* $m' \geq m$*, and for all* m'' *such that* $m \leq m'' < m'$ *we have* $(I, r(m'')) \models \varphi$.

Note that the operators $\Diamond$ (at some point in the future), and $\Box$ (always in the future) can be expressed in terms of the above. In fact: $\Diamond\varphi \equiv T\,\mathcal{U}\,\varphi$, and $\Box\varphi \equiv \neg\Diamond\neg\varphi$.
The following logic can be proven to be sound and complete with respect to the language $\mathcal{L}_T$ and the most general class of interpreted systems.

Theorem 2 *The logic L_T defined by*

- *Tautologies $\subseteq L_T$.*
- $\bigcirc p \wedge \bigcirc(p \rightarrow q) \rightarrow \bigcirc q$.
- $\bigcirc\neg p \leftrightarrow \neg \bigcirc p$.
- $\varphi\,\mathcal{U}\,\psi \leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi\,\mathcal{U}\,\psi))$.
- *Uniform substitution/*
- *If $\vdash_{L_T} \varphi$, then $\vdash_{L_T} \bigcirc\varphi$.*
- *If $\vdash_{L_T} \varphi' \rightarrow \neg\psi \wedge \bigcirc\varphi'$, then $\vdash_{L_T} \varphi' \rightarrow \neg(\varphi\,\mathcal{U}\,\psi)$.*

is sound and complete with respect to the most general class of interpreted systems.

As it can be seen above the system is made of two modalities: the mono-modal fragment containing the $\bigcirc$ operator is one in which the box and the diamond collapse, representing linear flows of time. The *interaction axiom* between the modalities intuitively gives a recursive definition of until in terms of next; in particular $\varphi\,\mathcal{U}\,\psi$ is true if and only if either ψ is true now, or φ is true now and at the next step in time $\varphi\,\mathcal{U}\,\psi$. The inference rule gives a way of inferring an until formula from a combination of next formulas.

Knowledge and time

Now that we have complete axiomatisation for the temporal and epistemic fragment, we can take the union of the two languages $\mathcal{L}_K$, and $\mathcal{L}_T$, and study an axiomatisation.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi\,\mathcal{U}\,\psi \mid K_i\varphi \quad (i \in A).$$

Intuitively in this setting we are studying the most general class of runs on global states so we should not expect any interaction between knowledge and time in an axiomatisation.
We report the following result without proof.

Theorem 3 *The logic $L_K \cup L_T$ is sound and complete with respect to the most general class of interpreted systems.*

Interaction between time and knowledge

Having both time and knowledge as part of the logical language opens the possibility of very expressive models. For instance, by nesting temporal formulas into knowledge ones, (as in $K\Box\varphi$) we can model *knowledge about a changing world*, whereas by nesting epistemic formulas into temporal ones (as in $\Box K\varphi$) we can talk about the *temporal evolution of knowledge* (for example in knowledge bases). There is a multitude of complex interactions that can be studied here — we only focus on three simple cases which have applications both in AI and in distributed computing:

- Synchronicity,
- Perfect recall.

Synchronicity is an assumption commonly made while studying distributed systems. A system is synchronous if all agents in the system share a common clock. All systems in which agents or processes proceed in turns (for example by waiting for all the agents to have performed an action) are synchronous. In the framework of interpreted systems we can code that all agents know what time it is by imposing:

$$r(m) \sim_i r'(m') \rightarrow m = m',$$

for all agents $i \in A$ and all points $r(m), r'(m') \in S$. Intuitively this is saying that points with different times are not considered to be possible.
Let us ask now what a complete axiomatisation for the class of synchronous interpreted systems is. Clearly, the logic $L_T \cup L_K$ is sound with respect to synchronous systems, the question is what axioms we need to add to it to make it complete.
Surprisingly, it can be shown that no axiom is needed:

Theorem 4 *The logic $L_T \cup L_K$ is sound and complete with respect to the class of synchronous interpreted systems*

This is another situation in which the syntax of the modal language is not powerful enough to capture what is expressible semantically (like in the case of irreflexive frames for possible worlds semantics).
Let us now add *perfect recall* to synchronicity; in this case we will find that something does need to be added to the logic above. Intuitively an agent's knowledge may grow over time. The assumption of perfect recall imposes that agents remember all the information they have been exposed to, or equivalently

that they make perfect use of the information they acquired. Although this is an unrealistic assumption in practical applications, this assumption is used often in many areas. For example in cryptography, one is not interested in whether or not an agent has decoded the message, but in whether or not in principle it has enough information to decode the message given the information it already has, (i.e., in our terminology it knows it).

So, let us assume that a local state encodes all the information that an agent has been exposed to. We can then model perfect recall by imposing that: if $r(m) \sim_i r'(m')$, then i's local state sequence is the same in $l_i(r(m))$, and $l_i(r'(m'))$, where $l_i : S \to L_i$ is the function returning the local state of agent i from a global state.

Let us reflect on what we need to add in order to provide a complete axiomatisation. Intuitively we need to express the fact that local states persist in time. This may lead us to think that the formula

$$K_i\varphi \to \Box K_i\varphi,$$

should be added to $L_K \cup L_T$ to provide a complete axiomatisation (note that the formula above is not provable in the logic $L_K \cup L_T$). This intuition is right, but it is not quite what we require. Indeed if φ is coding a truth which is contingent to a particular instant of time (such as the value of x now is 5) then surely it should not be known forever in the future. Still the axiom should hold for φ being "stable" formulas, i.e., formulas that once true remain true forever.

To get a complete axiomatisation for synchronous agents with perfect recall we need the observation that these agents are able to keep track of the time.

Theorem 5 *The logic* $L_T \cup L_K \cup \{K_i \bigcirc \varphi \rightarrow \bigcirc K_i \varphi\}$ *is sound and complete with respect to the class of synchronous interpreted systems with perfect recall.*

As a last remark, given that synchronicity does not contribute to an axiom in the basic case, one might think that for the case of perfect recall and no synchronicity one might get the same logic above. This suspicion can be proven to be unfounded, as the axiom one needs to add to get completeness for perfect recall agents is considerably more complicated and not reported here.

An example: the bit transmission problem

Suppose to have two agents S (sender), R (receiver), communicating over a faulty communication channel with the task of transmitting one bit of information (from S to R). Assume that any message sent is either lost or transmitted correctly and immediately (or more precisely within the next round).
One of the possible *protocols* governing the communication from S to R is for S to continue sending an information bit until it gets an acknowledgement from R, and then stop. Because of the fault in the line R sends his ack repeatedly.
Observe that even if the protocol is commonly known, while S knows that if he receives the ack, R must have received the bit, R will never be in the position to know whether or not S received its ack. Also note that imposing the ack a level further by insisting that S issues an ack-ack would simply push the problem down one level.
How do we model this scenario by means of interpreted systems? We have many options depending on how much information we want to include in the states. One simple way of doing it is to take:

$$L_S = \{0, 1, (0, ack), (1, ack)\}, L_R = \{\lambda, 0, 1\},$$

where the states in L_S code the information bit being sent and whether or not an ack has been received, and the states in L_R code the received bit (λ being nil). We can use the environment to record the communication taking place in the channel. So we can take L_e to be sequences of elements in

$$E = \{(sendbit, \Lambda), (\Lambda, sendack), (sendbit, sendack)\},$$

where the pairs represent the action of respectively agent S, and agent R (Λ representing the null action).
Let us now turn to describing the runs of the system. Clearly only some global states can be initial in our model: nothing must have been transmitted, nor received. Formally, an initial state can only be of the form:

$$(., k, \lambda) \text{ where } k \in \{0, 1\}.$$

But, there are also constraints that need to be imposed for the evolution of the system from a state to another. For example, once R has started sending acks, it will never stop doing so. We write the constraints as follows. Given a state $s = r(m) = (l_e, l_s, l_r)$, the next state $s' = r(m+1) = (l'_e, l'_s, l'_r)$ must satisfy the following:

- If $l_r = \lambda$, then $l'_s = l_s$, $l'_e = l_e(sendbit, \Lambda)$, and either $l'_r = \lambda$, or $l'_r = l_s$.
- If $l_s = l_r = k$, then $l'_r = k$, and $l'_e = l_e(sendbit, sendack)$, and either $l'_s = k$, or $l'_s = (k, ack)$.
- If $l_s = (k, ack)$, then $l'_e = l_e(\Lambda, sendack)$, $l'_s = l_s$, and $l'_r = l_r$.

To study this problem, we can consider the class of all the interpreted systems satisfying the constraints above, i.e. we can consider all the allowed initial points and construct all the possible runs for the system.

Let us now interpret epistemic temporal formulas on the resulting runs. We have 2 agents, and the environment, and all we need to model can be captured by the set of atomic propositions:
$P = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$, representing in turn that the bit of information has value 0, the bit of information has value 1, the bit has been received by R, the ack has been received by S, the bit has been sent by S, the ack has been sent by R. These are evaluated as follows.

- $r(m) \models (bit = k)$ if $l_S(r(m)) = k$, or $l_S(r(m)) = (k, ack)$,
- $r(m) \models recbit$ if $l_R(r(m)) \neq \lambda$,
- $r(m) \models recack$ if $l_S(r(m)) = (0, ack)$, or $l_S = (1, ack)$,
- $r(m) \models sendbit$ if $l_e(r(m)) = \ldots.(sendbit, \Lambda)$, or $l_e(r(m)) = \ldots.(sendbit, sendack)$,
- $r(m) \models sendack$ if $l_e(r(m)) = \ldots.(sendbit, sendack)$, or $l_e(r(m)) = \ldots.(\Lambda, sendack)$.

Note that the interpretation of the atoms depends only by the global state (as we assumed).
It is now easy to check that if s is a global point in which $l_R(s) = k$ with $k \neq \Lambda$, then $r(m) \models K_R(bit = k)$. In a similar way, it can be checked that when S receives the ack, S knows that R knows the bit: if $l_S(s) = (k, ack)$, then $r(m) \models K_S K_R = 0 \vee K_S K_R = 1$.
In this way we can verify properties of the functioning protocol of the MAS. In particular we have:

$$\models reckack \rightarrow K_S(K_R(bit = 0) \vee K_R(bit = 1))$$

Exercise: write and check temporal formulas which are true at and initial state. For example, is it true that at any initial state the formula: $\Diamond(K_R(bit = 0) \vee K_R(bit = 1))$ holds?