Common Knowledge

and

Agreement

Motivation. We have already shown that *common knowledge* plays an important role in the muddy children puzzle.

Common kowledge, however, is far more than just a curiosity that ariezes in puzzles.

We shall show that it is a fundamental notion of group knowledge, which is relevant in many applications. In particular, we show that common knowledge is a necessary and sometimes even sufficient condition for reaching agreement and for coordinating actions.

We illustrate the role of common knowledge by examining three well-known problems from the literature known as

- *Coordinated attack,*
- *Agreeing to disagree*,
- *Simultaneous Byzantine agreement*.

A digression.

Recall the basic logic $\mathbf{K}_n$. Its language L_n consists of

- the set $\boldsymbol{\psi}$ of primitive propositions.
- propositional connectives ¬ , & , v , -> , < - >
- modal operators K_i for $i = 1, 2, \ldots , n$

Axioms.

A1. $(K_i \varphi \;\&\; (K_i(\varphi \to \psi)) \to K_i \psi$

A2. $\varphi \to K_i \varphi$ for $i = 1, 2, \ldots , n$

Generalization Rule.

$$\frac{\varphi}{K_i \varphi} \quad \text{for } i = 1, 2, \ldots , n$$

We turn our attention to axiomatizing the modal operators E_G and C_G ., where G is a subset of $\{ 1, 2, \ldots , n\}$.

To this end, we extend the language L_n modal operator C_G (D_G). Let logic K_n^C consists of all the axioms of logic K_n together with following two axioms and inference rule

Axioms;

C1. $E_G \varphi$ <-> $\Lambda_{i \varepsilon G}$ $K_i \varphi$

C2. $C_G \varphi$ -> $E_G (\varphi$ & $C_G \varphi)$

Induction Rule

RC1.

$$\frac{\varphi \ \text{->} \ \mathrm{E}_{\mathrm{G}}(\psi \ \& \ C_G \varphi)}{\varphi \ \text{->} \ C_G \psi}$$

Recall that the operator C_G is infinitary as it is defined by infinite conjunction. This might suggest that we will not be able to characterize it with a finite set of axioms. Somewhat suprisingly, this is possible.

Before we turn to specific examples, let us consider the relationship between common knowledge and agreement.

How we can capture the fact that two players, say Alice and Bob , agree on some statement ψ ?

While we do not attempt to characterize agreement completely, we expect that if Alice and Bob agree on ψ , then each of them knows that they have agreed on ψ .

This is a key property of agreement: in order for there to be agreement, every participant in the agreement must know that there is agreement.

Suppose that *agree*(ψ) is a formula that is true in every state in which the players have agreed on ψ.

Thus, we expect

$$agree(\psi) \;\text{->}\; E_G(agree(\psi))$$

.
to be valid.

The Induction Rule tells us that if this is the case, then

$$agree(\psi) \;\text{->}\; C_G(agree(\psi))$$

is also valid.

Hence agreement implies common knowledge.

Now suppose that Alice and Bob are trying to coordinate their actions, i.e. , they want that Alice performs action **a** precisely when Bob performs action **b** .

Clearly, this involves the agent’s agreement on when to perform the actions; as our analysis shows, this requires common knowledge.

Unlike agreement, which we treat as an intuitive notion, coordination can be defined formally.

We establish a formal connection between *agreement* and *common knowledge* when analysing the problems of Coordinated attack and Simultaneous Byzantine agreement.

Coordinated Attack

As we shall see, the connection between agreement and common knowledge provides us with a sharp tool with which to analyse agreement problems.

(i) We can use it to prove impossibility results, namely, to prove that there are no protocols for solving certain agreement, such as Coordinated Attack or Agreement to Disagree.

(ii) We can use this connection in a positive manner, as a tool for the design of efficient protocols for reaching Simultaneous Byzatine agreement.

Coordinated Attack

Communication plays an important role in facilitating coordination between agents. It is the only means to make possible to an agent arrange to coordinate his actions with the actions of other agents in cases when the coordination was not fixed in advance.

It is not surprising that *guaranteed* coordination may require some degree of reliability of the communication medium. Indeed, unreliable communication renders such coordination impossible.

This is particularly well illustrated by the ***coordinated attack problem***, a well-known problem from the distributive systems folklore. The problem can be described informally as follows:

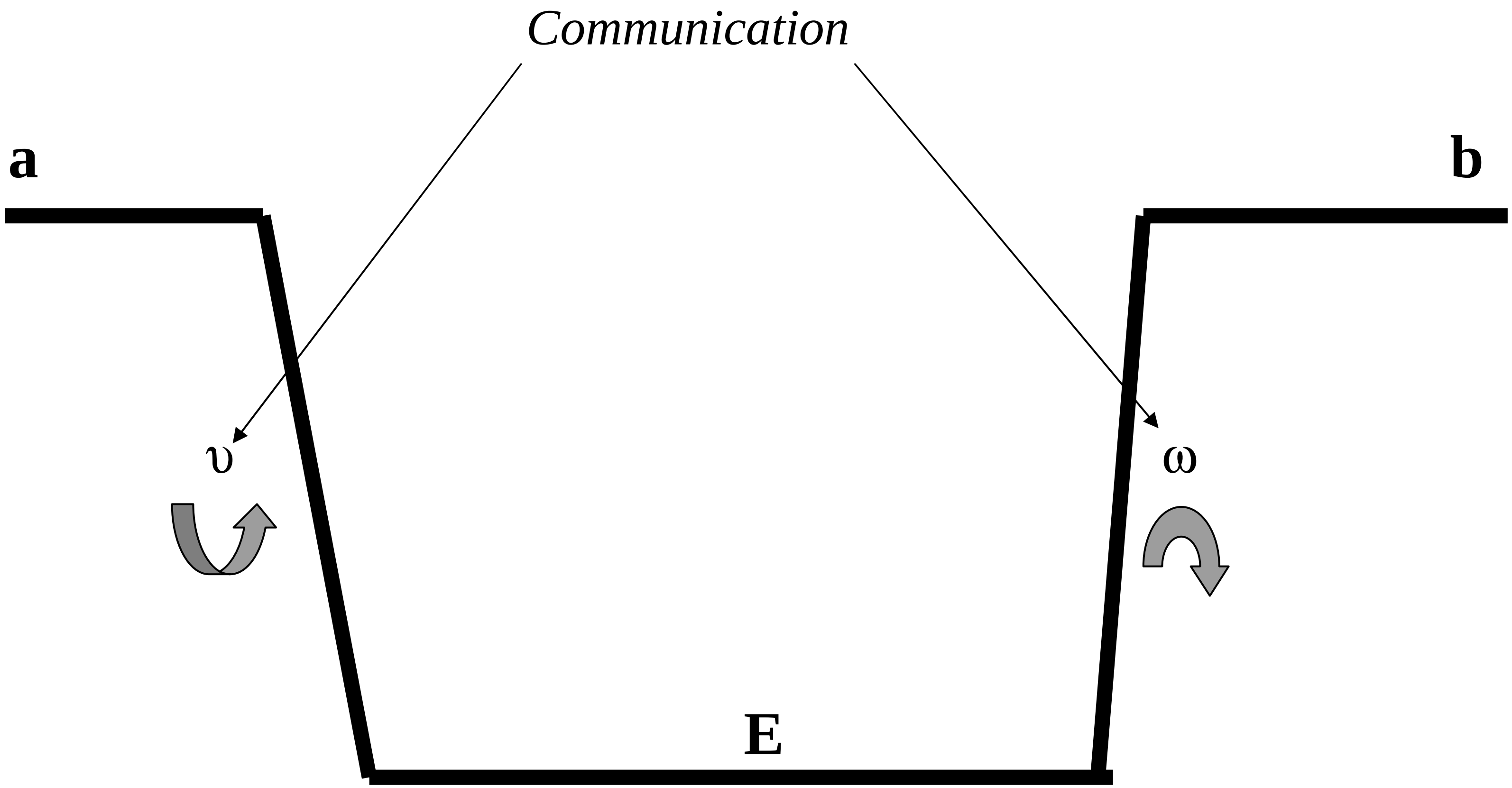
> Two divisions of an army, each commanded by a general, are camped on two hilltops overlooking a valley. In the valley awaits the enemy.
>
> The ballance of military power in this situation tells us that if b oth divisions attack the enemy simultaneously they will win the battle, while if only one division attacks it will be defeated.
>
> As a result, neither general will attack unless he is absolutely sure that the other will attack with him. In particular, a general will not attack if he receives no message.

The commanding general of the first division wishes to coordinate a simultaneous attack (at some time the next day).

The commanding generals can communicate only by means of messangers. Normally, it takes a messanger one hour to get from one ecampment to the other.

However it is possible that he will get lost bin the dark or, worse yet, be captured by enemy. Fortunately, assume that on this particular night, everything goes smoothly.

How long it will take to coordinate an attack?



Coordinated attack

Suppose that a message sent by General a reaches General b with a message saying *“attack at dawn”*. Should General b attack?

Although the message was in fact delivered, General a has no way of knowing that it was delivered. Hence a must therefore consider it possible that b did not received the message (in which case b would definitely not attack).

Hence a will not attack given his current state of knowledge.

Knowing this, b cannot attack based solely on receiving a’s message. Of course, b can try to to improve matters sending the messanger back to a with an acknowledgment. Imagine that the messanger is again successful and delivers the acknowledgment.

When a receives the acknowledgment, can he attack ? General a here is in a similar position to the one b was in when he received the original message.

This time b does not know, that the acknowledgment was delivered. Since b knows that without receiving the acknowledgment a will not attack.General b cannot attack as long as he considers it possible that a did not received the acknowledgment.

Hence, a cannot attack before he ensures that b knows the acknowledgment has been delivered. At this point, a might try to improve matters sending the messanger back to b with an acknowledment of the acknowledgement … etc.

Unfortunately, similar reasoning shows that this again will not suffice.

It is not difficult to check, that this time the problem is

$$\neg\, K_a K_b K_a(\text{b } \textit{received} \text{ a's initial message})$$

I will follow from our later results that no number of successful deliveries of ack nowledments to acknowledments can allow the Generals to attack.

Comment. Note that the problem is not caused by what actually happens, but by the uncerteinty regarding what might have happend.

In the scenario we just considered, communication proceed as smoothly as we could hope - all the acknowledgments sent are received - and still coordination is not attained.

More formally, let

delivered represents the fact that at least one message was delivered.

(i) when b gets a's initial message,

$K_b(delivered)$

holds.

(ii) when a gets b's acknowledgment,

$K_aK_b(delivered)$

holds.

(iii) when b gets a's acknowledgment,

$K_bK_aK_b(delivered)$

holds etcetc.

Comment. However, if all the messages that are sent are received, common knowledge of *delivered* never holds. We are about to prove the result.

We shall show that common knowledge is a prerequisite for coordination, in particular, the type of coordination required in the coordinated attack problem. Thus, the coordinated attack is not possible in systems with unreliable communication.

Our first step in proving these results is to define a class of contexts in which it makes sense to talk about agents' knowledge regarding message delivery.

To make our results as general as possible, we want to assume as little as possible about these contexts.

- we do not assume anything about the internal actions of agents,

- we do not assume anything about the environment's states and actions, beyond assuming that

- message delivery event can take place,

- the environment records the events taking place in the system.

Formally, we call an interpreted context (γ , π) a *message-delivery context* if it satisfies the following assumptions:

- The environment and/or some of the agents have actions that we recognize as message-delivery actions,
- γ is a recording context,
- the language includes the proposition *delivered* .

Comment. Intuitively, the message-delivery actions results in messages being delivered to agents.

The environment's state includes the sequence of joint actions that have been performed so far, and the transformation fumction τ updates states appropriately.

We intend *delivered* delivered to be true if at least one message-delivery action has been performed. Because the environment's state includes the sequence of joint actions performed, it is easy to define π to enforce this.

We can use a context to characterize a class of systems.

Definition. A *message-delivery system* is a system of the form

$$\mathbf{I}^{rep}(P, \gamma, \pi),$$

where (γ, π) is a message-delivery context and P is a protocol that can be run in the context γ .

Comment. In a message-delivery context, we can talk about message-delivery and what the agents know about it.

What can we say about the agents' knowledge of *delivered* in a message delivery system ?

The formula *delivered* is necessarily false at the beginning of a run (since no messages have been delivered by time 0 . It immediately follows that *delivered* cannot be a common knowledge at time 0.

Recall that in an asynchronous message-passing system common knowledge cannot be gained or lost.

Thus, in an a.m.p. system the agents never attain common knowledge of *delivered* .

In fact, as we now show, *delivered* can never become common knowledge even in a synchronous systems, as long as message is “sufficiently unreliable”.

What should it mean for message delivery to be “sufficently ureliable” ?

Intuitively, we take this to mean that there may be unbounded message delivery, so that it can také arbitrarily long for a message to arrive.

As a consequence, the only way an agent (other than recipient) can find about successful message delivery is through the receipt of other messages.

In particular, if R has unbounded message delivery, i receives a message at a point (r, ℓ) in R, and no agent receives a message from i in run r between times ℓ and m, then all other agents will consider it possible at time m that i has not yet received the message since they have no reason to believe otherwise.

We formalize the notion of unbouded message-delivery as a richness condition on the set of runs.

Definition. (Unbounded message delivery)

(i) Let R be a system such that, for every appropriately chosen π , the interpreted system $\boldsymbol{I} = (R\ , \gamma)$ is a message-delivery system. Given a run r in R , we write $d(r, m) = k$ if exactly k messages have been delivered in the first m rounds of r . Clearly, we always have $d(r, 0) = 0$.

(ii) We say that such a system R *displays umd* (*umd* stands for *unbounded message delivery*) if for all points (r, m) in R with d(r, m) > 0, there exists an agent i and a run r' in R such that

(1) for all agents j different from i and times m' up to m we have $r'_j (m') = r_j (m')$ and

(2) $d(r', m) < d(r, m)$.

Comment. Intuitively, we can think of i as the last agent to receive a message in r at or before round m , and r' as a run that is like r except that i does not receive this last message by round m .

Clause (1) ensures that no other agent can tell by round m that i has not received this message.

Clause (2) tells us that $d(r', m) < d(r, m)$ because the last message to i in r is not delivered in r'.

Comment. A number of systems of interest displays *umd.*

For example, it is easy to see that every a.m.p. system displays *umd,* as does every a.r.m.p. system.

In fact, we can make a stronger statement.

Definition. (Contexts displying *umd*)

We say that *a context* γ *displays umd*, if all system described by γ display *umd*, i.e. if $\mathbf{R}^{rep}(P, \gamma)$ displays *umd* for every protocol P that can be run in context γ.

Example. It is easy to see that the context γ^{amp} characterizing a.m.p. Systems displays *umd*, as do the contexts that arise by replacing the condition *True* by *Rel* or *Fair*.

Finally the context implicitely characterized by the coordinating attack story also displys *umd*.

The *umd* condition is just what we need to show that common knowledge of message delivery is not attainable.

Theorem 1.

Let $I = (\mathbf{R}, \pi)$ *be a message-delivery system such that* **R** *, displays umd, and let* G *be a set of two or more agents. Then*

$$I \models C_G(\textit{delivered}).$$

Comment. Note that the form of the above theorem is somewhat weaker than Theorem 4.5.4.

Unlike a.m.p. systems , it is not necessarily the case in a system satisfying *umd* that *no* common knowledge can be gained.

In a synchronous system satisfying *umd* , at two o'clock it is always common knowledge that the time is two o'clock.

Theorem. *Suppose* I *is an interpreted a.m.p. system,* r *is a run in* I*, and* G *is a group of at least two processes. Then for all formulas* φ *and times* $m >\geq 0$*, we have*

$$(I, r, m) \models C_G\, \varphi \quad \text{iff} \quad (I, r, 0) \models C_G\, \varphi$$

(This theorem has not been yet included in A4.)

Comment.

Theorem 1 essentially implies that *communication* in such systems cannot make formulas common knowledge.

A formula that is common knowledge at some point must also be common knowledge at a point where no messages have been delivered.

Of course, Theorem 1 is not strictly weaker than the theorem cited below, because Theorem 1 applies in a much wider class of contexts than the Theorem 4.5.4. does.

As an example of an application of Theorem 1, we now use it to prove the impossibility of coordinated attack.

To be able to discuss a coordinated attack by the generals, we define a corresponding class of contexts.

Definition. (*ca-compatible* contexts)

(i) An interpreted context (γ, π) is *ca-compatible* if it is a message-delivery context in which two of the agents are the generals A and B, and for each i in $\{A, B\}$, one of General i's actions is denoted $\mathbf{attack}_i$.

(ii) Moreover, we require that there be propositions $attacked_i$, for i in $\{A, B\}$. We take $attacked_i$ to be true at a point if $\mathbf{attack}_i$ was performed at some point in the past, i.e., if $\mathbf{attack}_i$ is recorded in the environment's state. (Recall that (γ, π) is a recording context.)

(iii) We take $attacking_i$ to be an abbreviation for

$$\neg\, attacked_i \;\&\; \mathrm{O}attacked_i$$

Thus $attacking_i$ is true if General i's next action is to attack.

(iv) We take *attack* to be an abbreviation for

$$attacking_A \ \& \ attacking_B$$

So *attack* is true if both generals are about to attack.

Comment.

Notice that the definition of *ca-compatible* contexts makes no assumptions whatsoever about the form of general's local states.

Definition. (Specifications)

Let the specification σ^{ca} consists of all ca-compatible interpreted systems I such that

1. $I \models attacking_A$ <-> $attacking_B$
2. $I \models \neg\, delivered$ -> $\neg\, attack$
3. $(I, r, m) \models attack$ for at least one point (r, m) of I

Comment.

The first condition says that General A attacks at (r, m) iff General B attacks at (r, m) .

The second says that no attack is carried out if no messages are delivered.

The third prevents the trivial solution to the problem where no generals attack.

Notice that the first two conditions are run-based while the third is not.

Definition. (Ca-compatible protocols)

We say that P is a *protocol for coordinated attack* in a ca-compatible interpreted context (γ, π) if P satisfies σ^{ca} in (γ, π).

We can now make precise our earlier claim that common knowledge is a prerequisitefor coordinated attack. We start by showing that when when the generals attack, it must be common knowledge that they are attacking.

To simplify notation, we simply write E and C leaving out the subscripts $\{A, B\}$.

We focus here on the case in which the protocols the generals follow are *deterministic.*

Proposition 1. *Let* (γ, π) *be a ca-compatible interpreted context and let P be a deterministic protocol. If* $\mathbf{I}^{rep}(P, \gamma, \pi)$ *satisfies* σ^{ca} *, then*

$$I \models attack \rightarrow C(attack)$$

Comment. Proposition 1. draws a formal connection between an action (here attacking) and a state of knowledge (here common knowledge of attack).

We stress that the generals need not be doing any reasoning for this result to hold and even need not be aware of the notion common knowledge.

Nevertheless when they attack they *must* have common knowledge of the fact they are attacking.

Because the successful delivery of at least one message is a prerequisite of an attack, we have the following:

Corollary 1. *Let* (γ, π) *be a ca-compatible interpreted context and let* P *be a deterministic protocol. If* $\mathbf{I}^{rep}(P, \gamma, \pi)$ *satisfies* σ^{ca} *, then*

$$I \models attack \ \ {-}{>} \ \ C\,(delivered)$$

Comment. Corollary 1. and Theorem 1. together imply that the generals in the coordinated attack problem are never able to to attack.

More generally, there is no deterministic protocol for a coordinated attack ina system that displays *umd* .

Corollary 2. *Let* (γ, π) *be a ca-compatible interpreted context such that* γ *displays umd , then there is no deterministic protocol* P *that satisfies* σ^{ca} *in* (γ, π) .

Comment. We assume that our contexts display *umd*. Corollary 2 says that there is no deterministic protocol for the coordinated attack in such a context !

It might be undestandable that the coordinated attack is not attainable in some runs of a protocol (i.e. where the messanger gets lost etc.).

Corollary 2 makes a far stronger claim: it says that an attack is never attainable in *any* run of *any* deterministic protocol for coordinated attack.

Thus, even if every message is delivered, coordinated attack is not possible as long as there is a possibility that messages will not be delivered.

Case Studies

Case 1. (Simultaneity) The fact, that according to Corollary 2. coordinated attack implies common knowledge depends on our requirement tha coordinated attack must be simultaneous and assumption that the generals are usi ng deterministic protocols.

In practice, simultaneity might be too strong a requirements. A protocol that guarantees that the generals attack within a short time seems to be more realistic.

In a system where generals attack within a short time of each other, attacking does not necessarily imply common knowledge of the attack.

Nevertheless, as we shall show later, similar arguments show that even such weaker forms of coordination are unattainable if communication is unreliable.

Case 2. (Deterministic Protocols) While the assumption that the generals are following deterministic protocol is quite reasonable in practice, it is instructive to find out where it is used it in the proof of Proposition 2.

The fact that P is a deterministic makes the event $attacking_i$ depend on i's local state for i in $\{A, B\}$. As a result, we have that

$$\mathrm{I} \models attacking_A \rightarrow K_A(attacking_A)$$

To see how this may fail for nondeterministic protocols, consider the following protocol P : General A simply sends a message saying *attack* ; after that he nondeterministically chooses in each round whether or not to attack. After receiving the message, General B nondeterministically chooses in each round whether or not to attack.

Suppose, we are given a ca-compatible context (γ, π) , where

$$\gamma = (P_e, G_0, \tau, \psi)$$

If ψ does not put any constraints on the set of acceptable runs, then it is clear that $I^{rep}(P, \gamma, \pi)$ will not satisfy σ^{ca} . There will be many runs where on general attacks and the other not.

Case 3. (*umd*) It is possible to choose $\boldsymbol{\psi}$ in such a way that
. $I^{rep} (P, \gamma, \pi)$ satisfies σ^{ca} and $\mathbf{R}^{rep} (P, \gamma)$ displays *umd*
then the formula
. *attack* -> *C* (*attack*) is not valid in $I = \mathbf{I}^{rep} (P, \gamma, \pi)$

In fact, even $attacking_A$ -> $K_A(attacking_A)$
is not valid there; because of the nondeterminism, at a point where General *A* is about to attack he does not know he is about to attack.

The reason that *I* still manages to satisfy σ^{ca} is that $\boldsymbol{\psi}$ here „magically“ rejects all runs where the generals do not coordinate.

It is possible to show by putting reasonable constraints on $\boldsymbol{\psi}$, we can ensure that this does not happen and that the analogue of Proposition 1 holds even for nondeterministic protocols.

Case 4. ($attacked_i$ depending on *i*‘s local state) There is another way in which we could have proved impossibility of coordinated attack with respect to nondeterministic protocols.

Notice that in the definition of ca-compatible context we did not assume that a general records the fact that he has just attacked in his local state.

By assuming that the generals do keep track of the fact they have attacked, we would make $attacked_i$ be a formula that depends on *i*‘s local state. This would make it possible to prove analogues of Proposition 1 and Corollary 1 for arbitrary protocols and contexts, using . .
. $attack = (attacked_A \;\&\; attacked_B)$ instead of $attack$

As a result, we could again use Theorem 1 to obtain an analogue of Corollary 2. This would prove the impossibility of solving coordinated attack when *umd* holds in this type of ca-compatible contexts, even for nondeterministic protocols.