Slide 1

COMP3152/9152
Lecture 8
Common Knowledge and Agreement
Ron van der Meyden

Slide 2

Agreement Implies Common Knowledge

Let the proposition $\mathtt{agree_G}(\psi)$ express that "group G agrees on ψ".

We expect that in any interpreted system $\mathcal{I}$ where this is interpreted appropriately, we have

$$\mathcal{I} \models \mathtt{agree_G}(\psi) \Rightarrow E_G \mathtt{agree_G}(\psi)$$

It follows that

$$\mathcal{I} \models \mathtt{agree_G}(\psi) \Rightarrow C_G \mathtt{agree_G}(\psi)$$

Slide 3

Unattainability of Common Knowledge

We have seen there can be no changes to Common Knowledge in Asynchronous Message Passing Systems. The result can be made more general....

Say that the interpreted context (γ, π) is a *message delivery context* if

1. Some of the environment/agent actions are designated as message-delivery actions,
2. γ is a recording context (the environment state includes the sequence of all joint actions that have been performed),
3. There is a proposition `delivered` that is true if at least one message-delivery action has been performed.

A *message-delivery system* is a system of the form $\mathcal{I}^{rep}(P, \gamma, \pi)$ where (γ, π) is a message-delivery context.

Slide 4

Unbounded message delivery

Suppose that $\mathcal{I} = (\mathcal{R}, \pi)$ is a message-delivery system.

Write $d(r, m) = k$ if exactly k message-delivery actions have occurred in the first m rounds of the run r.

Say that $\mathcal{R}$ *displays unbounded message delivery (umd)* if for all points (r, m) of $\mathcal{R}$ with $d(r, m) > 0$, there exists an agent i and a run $r' \in \mathcal{R}$ such that

1. for all agents $j \neq i$ and times $m' \leq m$ we have $r'_j(m') = r_j(m')$ and
2. $d(r', m) < d(r, m)$.

A context γ displays umd if for all joint protocols P, the system $\mathcal{R}^{rep}(P, \gamma)$ displays umd.

Slide 5

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

$$\mathcal{I} \models \neg C_G(\texttt{delivered})$$

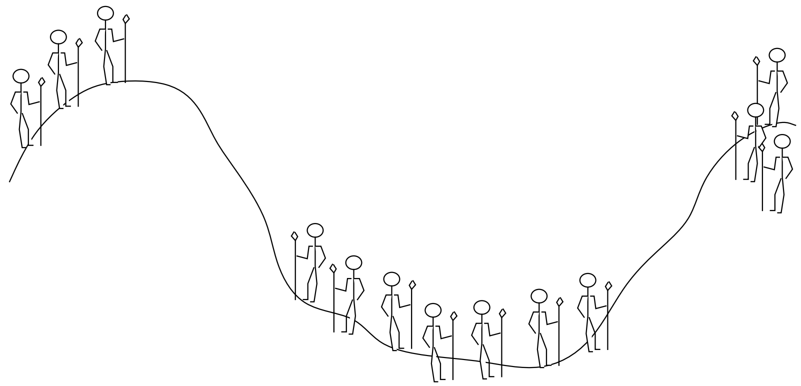
Note: common knowledge *can* be gained and lost in umd systems. E.g. in a synchronous system displaying umd we have

$$(\mathcal{I}, r, n{-}1) \models \neg C_G(time = n) \wedge \bigcirc C_G(time = n) \wedge \bigcirc \bigcirc \neg C_G(time = n)$$

Slide 6

Byzantine Generals

Slide 7



Slide 8

An interpreted context (γ, π) is ca-compatible if it is a message delivery context with two agents A, B and for $i = A, B$,

1. there is an action $attack_i$
2. agent i's state records whether $attack_i$ has been performed
3. there is a proposition $\texttt{attacked}_i$, true just if $attack_i$ has been performed in the past

Define $\texttt{attacking}$ by $\neg\texttt{attacked}_i \wedge \bigcirc\, \texttt{attacked}_i$.

Define $\texttt{attack}$ by $\texttt{attacking}_A \wedge \texttt{attacking}_B$.

Slide 9

An interpreted system satisfies the ca-specification if it is a ca-compatible system such that

1. $\mathcal{I} \models \texttt{attacking}_A \iff \texttt{attacking}_B$
2. $\mathcal{I} \models \neg\texttt{delivered} \Rightarrow \neg\texttt{attack}$
3. $(\mathcal{I}, r, m) \models \texttt{attack}$ for at least one point (r, m) of $\mathcal{I}$.

P is a *protocol for coordinated attack* in a ca-compatible interpreted context (γ, π) if $\mathcal{I}^{rep}(P, \gamma, \pi)$ satisfies the ca-specification.

Slide 10

Write $\texttt{attacked}$ for $\texttt{attacked}_A \wedge \texttt{attacked}_B$.

Proposition: Let (γ, π) be a ca-compatible interpreted context and let P be a (deterministic or nondeterministic) protocol. If $\mathcal{I} = \mathcal{I}^{rep}(P, \gamma, \pi)$ satisfies the ca-specification, then

$$\mathcal{I} \models \texttt{attacked} \Rightarrow C_{\{A,B\}}\texttt{attacked}$$

Slide 11

Proposition: Let (γ, π) be a ca-compatible interpreted context and let P is be a (deterministic or nondeterministic) protocol. If $\mathcal{I} = \mathcal{I}^{rep}(P, \gamma, \pi)$ satisfies the ca-specification, then

$$\mathcal{I} \models \texttt{attacked} \Rightarrow C_{\{A,B\}}\texttt{delivered}$$

Slide 12

Corollary: If (γ, π) is a ca-compatible interpreted context such that γ displays umd, then there is no (deterministic or non-deterministic) protocol P such that $\mathcal{I}^{rep}(P, \gamma, \pi)$ satisfies the ca-specification.

lide 13

Agreeing to Disagree

A stock market trade is a simultaneous action in which one agent performs a *buy* action and the other performs a *sell action*. For the trade to occur, it seems the buyer and seller must have common knowledge that they are trading, i.e., that

1. the buyer will perform *buy*, and
2. the seller will perform *sell*.

$$C_{\{1,2\}}(act_1(buy) \wedge act_2(sell))$$

On the other hand, if they are both perfectly rational, then they must follow the same rule for making their decision.

Surprise: This is a contradiction!

lide 14

(γ, π) is an *interpreted context for agreement* if

1. the players actions are all taken from the same set ACT
2. γ is a recording context,
3. for each action $a \in ACT$, there is a proposition $perf_i(a)$, true at a global state just when i has performed a

Write $act_i(a)$ for $\neg perf_i(a) \wedge \bigcirc perf_i(a)$.

Slide 15

Decision Functions

A decision function expresses how an agent makes decisions based on its information in a system $\mathcal{I}$

$$D : \mathcal{P}(Points(\mathcal{I})) \rightarrow ACT$$

D is *union consistent* if for every action a and disjoint sets $T_1, \ldots T_n$ of points, if $D(T_j) = a$ for all j, then $D(T_1 \cup \ldots \cup T_n) = a$

Slide 16

Example: The Risk Averse Decision Function

Suppose that each player receives a payoff $payoff(s, a) \in \mathbf{N}$ when it performs action a in state s.

(The payoff does not depend on what other players do)

The worst case payoff at a set S of points is

$$payoff(S, a) = min_{(r,m) \in S}\ (payoff(r(m), a)$$

Assume ACT is finite and for all sets S, and distinct actions a, b

$$payoff(S, a) \neq payoff(S, b)$$

The *risk averse decision function* chooses the action with the greatest worst case payoff, i.e., $D(S)$ is the unique action a such that $payoff(S, a) > payoff(S, b)$ for all actions $b \neq a$.

Exercise: this decision function is union consistent

lide 17

If $\mathcal{I} = (\mathcal{R}, \pi)$ is an interpreted system and l is a local state of agent i, write

$$IS_i(l, \mathcal{R}) = \{(r, m) \mid r \in \mathcal{R},\ m \in \mathbf{N},\ r_i(m) = l\}$$

for agent i's information set when in state l.

A protocol P_i for agent i is *compatible* with D in $\mathcal{R}$ if

$$P_i(l) = D(IS_i(l, \mathcal{R}))$$

for all $l \in L_i$.

A joint protocol P *implements* the decision function D in context γ if for all agents i, P_i is compatible with D in $\mathcal{R}^{rep}(P, \gamma)$.

(cf definition for knowledge-based programs)

lide 18

Theorem: Suppose that $\mathcal{I} = \mathcal{I}^{rep}(P, \gamma, \pi)$ where P is a joint protocol and (γ, π) is an interpreted context for agreement. If P implements a union-consistent decision function in context γ and a and b are distinct actions, then

$$\mathcal{I} \models \neg C_{\{1,2\}}(act_1(a) \wedge act_2(b))$$

Slide 19

So, in the stock market, the following cannot all be true:

1. The buyer and seller use the same rules for deciding their actions
2. Just before the trade happens $(act_1(buy) \wedge act_2(sell))$, they have common knowledge that they are about to trade.

Slide 20

Simultaneous Byzantine Agreement

Suppose there are n generals, t of them are traitors, the rest are loyal. But initially, nobody knows who the traitors are. There are no broadcast actions, only message passing. Every general has a preference about whether to attack.

Can we design a protocol so that

1. At some point, all the loyal generals either attack, or they all retreat.
2. If all the generals prefer to atack, then the agreement is to attack.

Even though the traitors may misbehave (e.g., tell one general they want to attack, and another that they want to retreat.)

Motivation: fault-tolerant protocols

Slide 21

(γ, π) is a ba-compatible interpreted context if

1. Each agent i has an action $\texttt{decide}_i(y)$ for $y \in 0, 1$
2. The environments actions include actions $(a_{e1}, \ldots, a_{en})$, where the a_{ei} are tuples that describe
   (a) which messages sent by process j are delivered to process i in that round
   (b) whether or not i fails in that round ($\texttt{fail}_i$), and the nature of the failure
   (c) γ is a recording context
3. Process i's initial state is a tuple of the form $(x_i, \ldots)$, where x_i is i's preference for the decision.
4. The environment's initial state also contains x_i

Slide 22

5. There is a proposition $\mathit{decided}_i(y)$ for $y \in \{0, 1\}$ that is true if i tried to perform $\texttt{decide}_i(y)$ at some previous round
6. There is a proposition $\exists y$ for $y \in \{0, 1\}$ that is true if some process i has $x_i = y$.

Slide 23

Notation:

$\mathit{deciding}_i(y)$ for $\neg \mathit{decided}_i(y) \land \bigcirc\, \mathit{decided}_i(y)$

At a point (r, m), let $\mathcal{N}(r, m)$ be the set of nonfaulty agents (for which the enviornment has not yet performed $\texttt{fail}_i$).

$(\mathcal{I}, r, m) \models \mathit{deciding}_{\mathcal{N}}(y)$ if $(\mathcal{I}, r, m) \models \mathit{deciding}_i(y)$ for all $i \in \mathcal{N}(r, m)$

Slide 24

Specification for SBA

A system $\mathcal{I}$ satisfies the SBA specification if for every run r:

1. *Decision:* Every process that is nonfaulty in r performs exactly one $\texttt{decide}_i(y)$ action in r
2. *Agreement:* If i is nonfaulty at (r, m) and is about to decide y at (r, m) and j is nonfaulty at (r, m') and is about to decide y' at (r, m) then $y = y'$.
3. *Validity:* If all the processes have the same initial preference x then all the nonfaulty processes decide x
4. *Simultaneity:* the nonfaulty processes decide simultaneously, i.e., if i and j are nonfaulty at (r, m) and i is about to decide at (r, m), then so is j.

lide 25

Failure Modes

The following are possible failure modes:

1. *Crash Failures:* A faulty process follows its protocol up to the time when it fails, after which it sends no messages.
2. *Omission Failures:* A faulty process follows its protocol, but in any round the set of messages it sends or receives is a subset of what it should be.
3. *Byzantine Failures:* faulty processes may deviate from the protocol in any way: send a subset of messages, send false messages, collude with other faulty processes to deceive the non-faulty processes, etc