Class 2

Yoram Moses
moses@ee.technion.ac.il
Meyer 919
Happiness of the Fish

Chuangtse and Hueitse had strolled onto the bridge over the Hao, when the former observed,

“See how the small fish are darting about! That is the happiness of the fish.”

“You are not a fish yourself,” said Hueitse.

“How can you know the happiness of the fish?”

“And you not being I,” retorted Chuangtse,

“how can you know that I do not know?”

Chuangtse, circa 300 B.C.
Possible Worlds:
A Model for Knowledge

When does j know $\phi$?

- Player j has (in any given state of the world) a set “$P_j$” of “worlds” s/he considers possible.
- The actual world must be in $P_j$.
- $K_j\phi$ holds if $\phi$ is true in all worlds of $P_j$.
- More precisely, $K_j\phi$ holds in world $w$ if $\phi$ is true in all worlds of $P_j(w)$.

Knowledge ©
יש מזר לחיים
فقدון בעית נציפור
לרגשת חלומת
מתאימים לכל טויה לדורות אмерיקה. למסעות רישיון חיות אוניב.
לצאת למסעות בבודים בקיניים! מخصوص. יש מזר לכל חלום!
"יש מזר לחיים" הוא פקדון חדש המאיץ את הップית הפ广告服务 קרーション
המסכים המתחיים של זהירות ובושת בקינון חודר ומשlığמד. חכם
משרש השפה ובסוףIGNED עם 5% מעוות, זורר מוחד - 5% מעוות
המסכים שברולט
有用的 הפקודה: ■
הפקודה טובעת בחזרה Kunden- 100 שקלים לשחית
הפקודה זו ש物业服务 150 שקלים לשחית.
יינו בך חזרתי? יש זכר לפקתה הפקדון אופי
בחציו כל שעה מתעך לשעון בשיעור של 5% מסך הפקדון
של שעה, עד 50 ש"ח.
מודיעים שלך - חלם שלך, מידו חזרו.
אלה מהחיתון? יש מזר הזה צışı! צייר לקפץ למסכים לפקדת.
פריסת פלאים פריסת הפקודה בברך.
 sessionId, 2003.2007 또한 ציון
www.bankhapoalim.co.il
Knowledge ©
What are “Possible Worlds”?

Depends on the context:

- **Game of cards**: Cards dealt, cards shown publicly, announcements…
  - A player’s view is consistent with many such “worlds”.
- **Muddy children**: Foreheads, answers so far..
  - A child’s view is consistent with either one or two worlds.
- More generally, “worlds” can contain more detail.
- To capture knowledge w.r.t. possible worlds, we need a language and a model:
The language of Knowledge

Assume we have n players, named 1,...,n.

\( \Phi = \{ p, p', q, q', \ldots \} \) denotes a set of primitive propositions, or basic facts.
- \( p \) could be “it is raining on the Eiffel right now”
- We use Boolean operators \( \land \) (AND) and \( \neg \) (NOT) as well as knowledge operators \( K_1, \ldots, K_n \) for the players.
- Now define inductively:
  - Every \( p \in \Phi \) is a formula
  - If \( \varphi \) and \( \psi \) are formulas then so are
    - \( \neg \varphi \)
    - \( \varphi \land \psi \)
    - \( K_i \varphi \) for \( i = 1, \ldots, n \)

- Later also:
  - \( E_G \varphi \)
  - \( C_G \varphi \)
  - \( D_G \varphi \)
The following four propositions, which appear to the author to be incapable of formal proof, are presented as Fundamental Postulates upon which the entire superstructure of General Systemantics . . . is based . . .

1. EVERYTHING IS A SYSTEM.

2. EVERYTHING IS PART OF A LARGER SYSTEM.

3. THE UNIVERSE IS INFINITELY SYSTEMATIZABLE, BOTH UPWARD (LARGER SYSTEMS) AND DOWNWARD (SMALLER SYSTEMS).

4. ALL SYSTEMS ARE INFINITELY COMPLEX. (The illusion of simplicity comes from focusing attention on one or a few variables.)

John Gall, Systemantics, 1975
Modeling

Your model should be as simple as possible, but not simpler…

A. Einstein

- We need to focus on the essential details and keep it simple
- Try to ensure that the model faithfully captures the intended application
- There is always a gap between the model and reality. We must constantly be aware of this
- There are many ways to cut a cake
Our Basic Model

- A process (agent/player) $i$ is always in a well-defined local state $s_i$.
  - $s_i$ captures the information available to the process.
  - Contents of $s_i$ depend on application and the modeller.

- The global state captures the state of the system frozen at an instant.
  - Contains the local states of all processes
  - May contain more – aspects of the state that are not local to/visible by any process.
  - Formally, $g=(s_e,s_1,...,s_n)$ where $1,...,n$ are the processes and $e$ in $s_e$ stands for the “environment.”
  - The set of global states is denoted by $G$
To capture the evolution of a system we define:

- **A run** is a function \( r: \text{Time} \rightarrow G \)
  - Time will usually range over the natural numbers.
  - Time is external to the processes.
  - It need not model constant intervals.
  - Processes may or may not be aware of the time, or of the passage of time and its rate. This will depend on the information available to a given process in a given application.
- \( r(0) \) – the initial state in the run \( r \).
- A given system can allow many histories and hence it can have many runs.
- We define a **System** to be a set \( R \) of runs.
Systems

- There is a huge variety of systems.
- In some cases, an initial state determines a single run.
  - E.g., muddy children.
- If there is any source of nondeterminism – the same local state can appear in many runs.
- A system containing all runs of a protocol or all possible histories of a device is a natural tool for the analysis of the protocol or device.
- Many properties can be stated in terms of the runs of the system.
Incorporating Knowledge

- We denote by $r_i(m)$ the local state of $i$ in $r(m)$
- We start from a set of primitive propositions $\Phi$ (usually not stated explicitly) for stating the relevant basic facts
- An interpretation $\pi : G \times \Phi \rightarrow \{T,F\}$ assigns truth values to propositions at global states
- An interpreted system is a pair $I=(R,\pi)$
  - Sometimes $\pi$ is obvious, and we use $R$
|= for Interpreted systems I=(R,π)

We define satisfaction directly for interpreted systems:

\[(I,r,m) |= p \quad (\text{for } p \in \Phi) \quad \text{if} \quad \pi(r(m))(p)=T\]

\[(I,r,m) |= \varphi \land \psi \quad \text{if both } (I,r,m) |= \varphi \text{ and } (I,r,m) |= \psi\]

\[(I,r,m) |= \neg \varphi \quad \text{if it is not the case that } (I,r,m) |= \varphi\]

and

\[(I,r,m) |= K_i \varphi \quad \text{if } \quad (I,r',m') |= \varphi \quad \text{holds for every } (r',m') \text{ in } I \text{ such that } r_i(m) = r'_i(m')\]
Bit Transmission Example

- Sender S has bit $b \in \{0, 1\}$
- The channel from S to R is not reliable
- S sends the bit until it receives an ack message
- R sends an ack repeatedly after bit is received

To model this situation and “protocol” as a system:
1. Decide the structure of local and global states
2. Describe the structure of runs
3. Define the set of runs
   Many ways to cut the cake – we choose one
Modeling Bit Transmission

Local states
- of S: \{0, 1, (0;ack), (1;ack)\}
- of R: \{\lambda, 0, 1\}
- of e: a sequence of joint actions performed in each round. “Actions” are from \{(sendbit,\bot), (\bot,sendack), (sendbit,sendack)\}

Note: (\bot, \bot) is also a possible joint action, but it does not occur in the scenario described above
Modeling Bit Transmission

**Runs:**
- \( r(0) \) an initial global state from \{((\langle\rangle), 0, \lambda), ((\langle\rangle), 1, \lambda)\}
- \( r(m+1) = (s'_e, s'_S, s'_R) \) obtained from \( r(m) = (s_e, s_S, s_R) \) by one of the rules –
  - \( s_R = \lambda: \ s'_e = s_e \cdot (sendbit, \bot), s'_S = s_S, s'_R = s_S \) or \( s'_R = \lambda \)
  - \( s_R = s_S = k \in \{0, 1\}: \ s'_e = s_e \cdot (sendbit, sendack), s'_R = s_R, \) and \( s'_S = k \) or \( s'_S = (k, \text{ack}) \)
  - \( s_S = (k, \text{ack}): \ s'_e = s_e \cdot (\bot, \text{sendack}), s'_S = s_S, s'_R = s_R \)

**System:** \( R^{bt} \) consists of all runs meeting the above
Interpreted System $I^{bt}$

**Primitive Propositions**

$\Phi = \{\text{bit}=0, \text{bit}=1, \text{recbit}, \text{recack}, \text{sentbit}, \text{sentack}\}$

$I^{bt} = (R^{bt}, \pi^{bt})$ where $\pi^{bt}: G \times \Phi \to \{T,F\}$ is given by

1. $\pi^{bt}(r(m), \text{bit}=k) = T$ iff $r_S(m) \in \{k, (k;\text{ack})\}$
2. $\pi^{bt}(r(m), \text{recbit}) = T$ iff $r_R(m) \in \{0,1\}$
3. $\pi^{bt}(r(m), \text{recack}) = T$ iff $r_S(m) \in \{(0;\text{ack}), (1;\text{ack})\}$
4. $\pi^{bt}(r(m), \text{sentbit}) = T$ iff last element in $r_e(m)$ is one of
   $\{(\text{sendbit}, \bot), (\text{sendbit}, \text{sendack})\}$
5. $\pi^{bt}(r(m), \text{sentack}) = T$ iff last element in $r_e(m)$ is one of
   $\{\bot, \text{sendack})$, $(\text{sendbit, sendack})\}$
Knowledge in $I^{bt}$

For every $r \in R^{bt}$:

$$(I^{bt}, r, 0) \models \neg K_R(bit=0) \land \neg K_R(bit=1)$$

$$(I^{bt}, r, 0) \models K_S(bit=0) \lor K_S(bit=1)$$

$I^{bt} \models bit=k \implies K_S(bit=k)$

$I^{bt} \models (bit=k \land recbit) \implies K_R(bit=k)$

$I^{bt} \models (bit=k \land recack) \implies K_SK_R(bit=k)$

$I^{bt} \models \neg K_RK_SK_R(bit=k)$
Adding Time

We add modal (temporal) operators for time:

- \( O\varphi \) “Next \( \varphi \)”, \( \Diamond \varphi \) “Eventually \( \varphi \)”, \( \Box \varphi \) “Forever \( \varphi \)”, and \( \varphi U \psi \) “\( \varphi \) Until \( \psi \)”. 

- \((I,r,m) \models O\varphi\) if \((I,r,m+1) \models \varphi\)
- \((I,r,m) \models \Diamond \varphi\) if \((I,r,m') \models \varphi\) for some \(m' \geq m\)
- \((I,r,m) \models \Box \varphi\) if \((I,r,m') \models \varphi\) for all \(m' \geq m\)
- \((I,r,m) \models \varphi \psi\) if \((I,r,m') \models \psi\) for some \(m' \geq m\) and \((I,r,m'') \models \varphi\) for all \(m' > m'' \geq m\)
Knowledge and Time in $I^{bt}$

For every $r \in R^{bt}$ and $k \in \{0, 1\}$:

$$(I^{bt}, r, 0) \models \text{bit=}k \Rightarrow \neg K_S \Diamond K_R(\text{bit=}k)$$

$$(I^{bt}, r, 0) \models K_S(\text{bit=}0) \lor K_S(\text{bit=}1)$$

$I^{bt} \models \text{bit=}k \Rightarrow \Box \text{bit=}k$ \hspace{2cm} (stable fact)

$I^{bt} \models \text{bit=}k \Rightarrow \Box K_S(\text{bit=}k)$

$I^{bt} \models (\text{bit=}k \land \text{recbit}) \Rightarrow \Box K_R(\text{bit=}k)$

$I^{bt} \models (\text{bit=}k \land \text{recack}) \Rightarrow \Box K_S K_R(\text{bit=}k)$

The goal of Bit Transmission:

$\Diamond K_S K_R(\text{bit=}0) \lor \Diamond K_S K_R(\text{bit=}1)$
Synchronous Systems

Some systems have a “global clock”

\( R \) is called \textit{synchronous} if \( r_i(m) = r_i'(m') \) implies \( m = m' \)

The system for Muddy children is \textit{synchronous}.

The system \( R^{bt} \) for bit transmission is \textit{not synchronous}.
Perfect Recall

$r_i(m+1)$ need not encode the information in $r_i(m)$, so processes may

“forget”. Not forgetting is called perfect recall and defined by:

The local state sequence of process $i$ at point $(r,m)$ is the sequence of its
local states from $(r,0)$ to $(r,m)$, without consecutive repetitions.

$Lss(\langle s_i, s_i, s_i', t_i, t_i, t_i, s_i \rangle) = \langle s_i, s_i', t_i, s_i \rangle$

Process $i$ has perfect recall in $\mathbb{R}$ if $r_i(m) = r_i'(m')$ implies that $i$’s $Lss$ is the
same at both points.
Perfect Recall cont’d…

Let \( I = (R, \pi) \) where \( i \) has perfect recall in \( R \).

Does \( I \models K_i \varphi \Rightarrow \Box K_i \varphi \)?

\( \varphi \) is stable in \( I \) if \( I \models \varphi \Rightarrow \Box \varphi \)

If \( i \) has perfect recall in \( I \) and \( \varphi \) is stable then
\[ I \models K_i \varphi \Rightarrow \Box K_i \varphi \]

All processes have perfect recall in the muddy children puzzle and in \( R^{bt} \).
Protocols, Actions & Contexts
Review

- A system is a set $\mathcal{R}$ of RUNS
- Where do the runs come from?
- We are interested runs generated by behaviors
- A behavior is determined by a protocol:
  - $P_i : L_i \to Act_i$ in the deterministic case and
  - $P_i : L_i \to 2^{Act_i}$ more generally
- We think of a system generated by an automaton:
  - A graph with nodes for states and edges determined by actions. In our case:
    - Nodes represent global states
    - Actions are joint (global) actions – with a component for every agent and one for the environment.
      - Often simpler actions suffice. Joint actions matter when we care to model dependent actions that influence one another.
Contexts

- A run:
  - Starts in an initial state
  - In every state, a joint action is performed
  - Next state is determined by the state and joint-action.

- A context is a tuple $\gamma = (G_0, P_e, \tau, \Psi)$ where
  - $G_0$ is a set of initial states
  - $P_e$ is a protocol for the environment
  - $\tau$ is a transition function, mapping a joint action and a global state and to a new global state: $\tau(a, g) = g'$ and
  - $\Psi$ is an “admissibility condition” handling conditions on eventual behaviors: Useful for specifying that “communication is reliable”
    - We write $\text{TRUE}$ for $\Psi$ or omit it if all runs are admissible
    - $\Psi$ may be omitted, and additional components can be added to a context if needed.
Protocol + Context = System
The run \( r \) is a run of \( P = (P_1, \ldots, P_n) \) in \( \gamma \) if:

- \( r(0) \in G_0 \),
- \( r(m+1) = \tau(a_e, a_1, \ldots, a_n)(r(m)) \), where each \( a_j \in P_j(r_j(m)) \), for all \( m \geq 0 \)
- \( r \in \Psi \)

A set of protocols \( P = (P_1, \ldots, P_n) \)
and a context \( \gamma = (G_0, P_e, \tau, \Psi) \)
determine a system

\[ R(P, \gamma) = \{ r: r \text{ is a run of } P \text{ in } \gamma \} \]
Interpreted Contexts:

$$P + (\gamma, \pi) = I(P, \gamma, \pi):$$

The set $\Phi$ is interpreted over the global states $G$, or over points $(r, m)$ with $r$ a run of $\gamma$

Adding $\pi$ to $\gamma$ we obtain $(\gamma, \pi)$: an interpreted context

Now $P + (\gamma, \pi) = I(P, \gamma, \pi)$, where $I(P, \gamma, \pi) = (R(P, \gamma), \pi)$.

For a given context $\gamma$ and interpretation $\pi$, every protocol $P$ defines a unique interpreted system!
Coordinated Attack

Two divisions are located on two sides of a valley.

They communicate via messengers:
Messengers are not guaranteed to deliver

Required is a protocol for coordinating an attack:
- Division A attacks iff B attacks
- If no messenger delivers nobody attacks
- An attack is possible: In at least one run, they attack
Coordinated Attack

At dawn
Preview of next week

We formalize CA
Define common knowledge \( C \)
We prove

**Theorem**: If \( P \) solves CA and \( I = I(P, \gamma, \pi) \) Then

\[ I \models \text{attacking} \Rightarrow C \text{attacking} \]

and

\[ I \models \text{attacking} \Rightarrow C \text{some-msg-delivered} \]

but

\[ I \models \neg C \text{some-msg-delivered} \]