Muddy Children, Other Logic Puzzles, and Temporal Dynamic Epistemic Logic

Muddy Children

Three brilliant children go to the park to play. When their father comes to find them, he sees that two of them have mud on their foreheads. He then says, "At least one of you has mud on your forehead", and then asks, "Do you know if you have mud on your forehead?" The children simultaneously respond, "No".

The father repeats his question, "Do you know if you have mud on your forehead?" and this time the two children with muddy foreheads simultaneously answer, "Yes, I have!" while the remaining child answers, "No".

Claim

If there are n children, k of whom are muddy, then the k children will announce that they are muddy after the father repeats his question k times.

All we care about n is that $n \geq k$.

Induction

Base case: Suppose 1 child was muddy. After the father announces, "At least one of you has mud on your forehead":

- What does the muddy child know? That he is the muddy one
- What do the clean children consider possible? Either 1 or 2 muddy children total.

When the father asks, "Do you know if you have mud on your forehead?", the muddy child will respond, "Yes, I am!".

Just another step to build intuition: Suppose 2 children were muddy. After the father announces, "At least one of you has mud on your forehead":

- What do the muddy children consider possible? Either 1 or 2 muddy children total
- What do the clean children consider possible? Either 2 or 3 muddy children total.

When the father asks, "Do you know if you have mud on your forehead?", everyone answers, "No". As a result, everyone knows that only 1 muddy child is not a possibility. Hence, the muddy children now know that they are muddy.

Inductive hypothesis: Assume that it is common knowledge that at least k children are muddy if the father asked $k-1$ times if the children knew if they were muddy, and each time they *all* answered, "No"

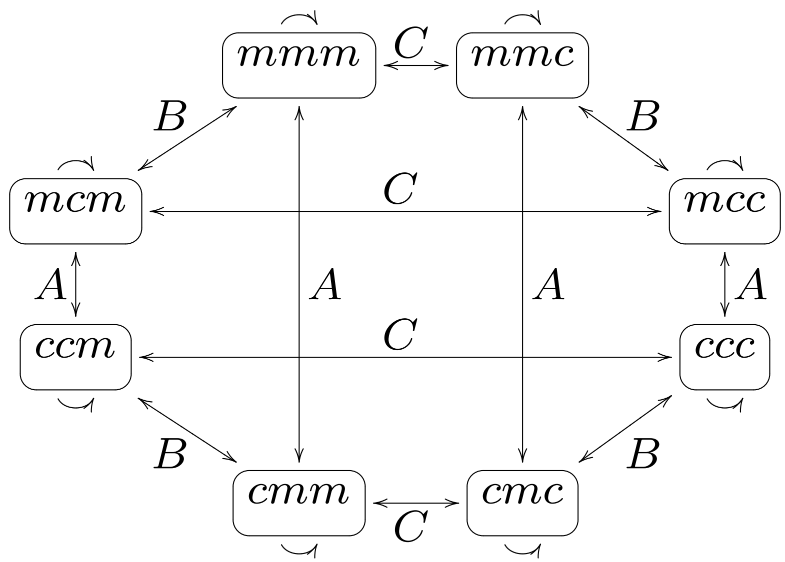
If *more* than $k+1$ children are muddy, then they all would have known all along that *at least* $k+1$ are muddy. Consider two cases:

• k children are muddy: Each muddy child sees only $k-1$ muddy children, but knows that at least k are muddy. He then answer, "Yes, I am!" to their fathers k^{th} question.

• $k+1$ children are muddy: Their father asks if the know if they are muddy, and they *all* answer, "No", which is not the expected response if there had been exactly k muddy children. Thus they all know there are at least $k+1$ muddy children.

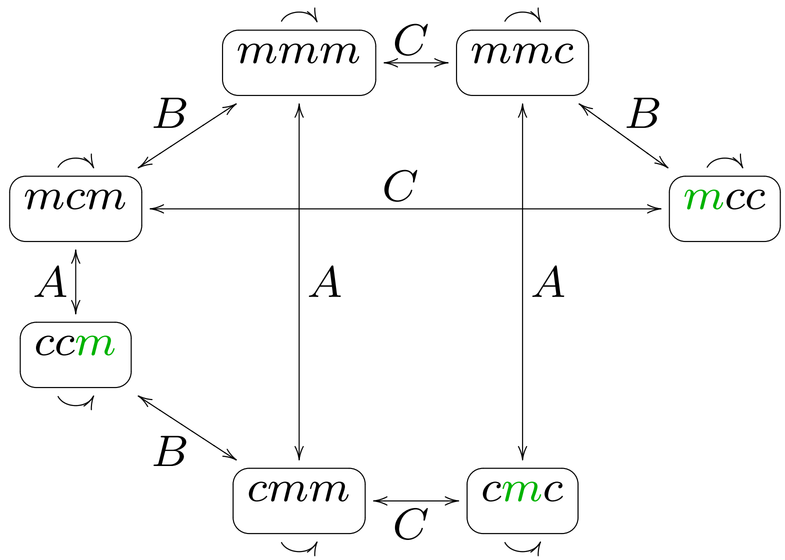
Back to initial example: $n = 3, k = 2$

An arrow labeled A (B, C resp.) linking two states indicates that A (B, C resp.) cannot distinguish between the states (reflexive arrows indicate that every agent considers the actual state possible). Initial situation:



Note that at every state, each agent cannot distinguish between two states

Right after the father announces, “At least one of you has mud on your forehead”:



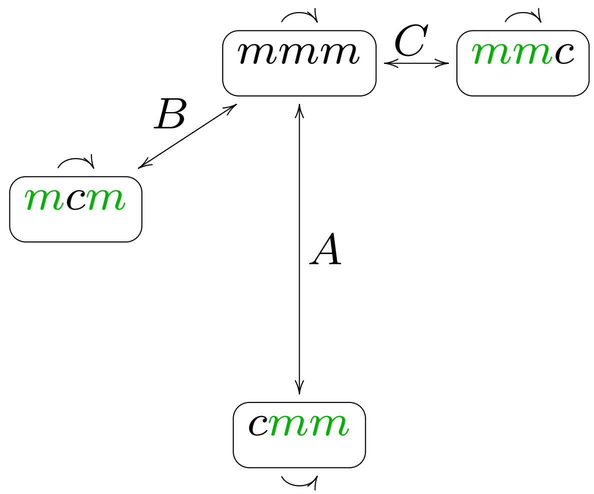
Note that

at mcc, A is certain mcc is the correct state,

at cmc, B is certain cmc is the correct state,

at ccm, C is certain ccm is the correct state.

Right after it is revealed that no one knows whether he is muddy:



Note that

at mmc, A and B are certain mmc is the correct state,

at mcm, A and C are certain mcm is the correct state,

at cmm, B and C are certain cmm is the correct state.

State Model (AKA Epistemic Model)

Fix a set Φ of "properties" or "atomic propositions" and a set $\mathcal{A}$ of agents.
$\mathbf{S} = (S, \xrightarrow{\mathcal{A}}, \| \cdot \|)$, where

1. S is a set
2. $\xrightarrow{A}$ is an "epistemic relation" over S, that is $s \xrightarrow{A} t$ means A considers t possible at state s.
3. $\| \cdot \|$ is function from Φ to $\mathcal{P}(S)$.

Often $\xrightarrow{A}$ is an equivalence relation and is called an "indistinguishability relation".

The language of Epistemic Logic

Fix a set Φ of "atomic propositions" and a set $\mathcal{A}$ of "agents". Epistemic logic sentences (AKA formulas) are generated by the rule

$$\varphi ::= p \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \Box_A \varphi_1 \mid \Diamond_A \varphi_1$$

Reading:

$\neg$ as "not"

$\wedge$ as "and"

$\vee$ as "or"

$\rightarrow$ as "only if"; ($\leftarrow$ as "if"; $\leftrightarrow$ as "if and only if")

$\Box_A$ as "A believes that ..."

$\Diamond_A$ as "A considers it possible that ..."

Language for Muddy Children

Let $\mathcal{A} = \{A, B, C\}$.
Let $\Phi = \{(Am), (Ac), (Bm), (Bc), (Cm), (Cc)\}$, where we read (Am) as "A is muddy", and (Ac) as "A is clean", etc.

Then for example $\Box_C(Am) \wedge \Box_B \neg(Am) \wedge \neg\Box_A(Am)$ can be read as
"C believes that A is muddy, B believes that A is not muddy, and A does not believe he is muddy (A may be uncertain)".

Semantics

In a state model $\mathcal{S}$, define relation $\models$ between S and all formulas as follows:

$s \models p$ iff $s \in \|p\|$.
$s \models \neg\varphi$ iff $s \not\models \varphi$
$s \models \varphi_1 \wedge \varphi_2$ iff $s \models \varphi_1$ and $s \models \varphi_2$.
$s \models \Box_A \varphi$ iff $t \models \varphi$ whenever $s \xrightarrow{A} t$.
$s \models \Diamond_A \varphi$ iff $t \models \varphi$ for some t for which $s \xrightarrow{A} t$.

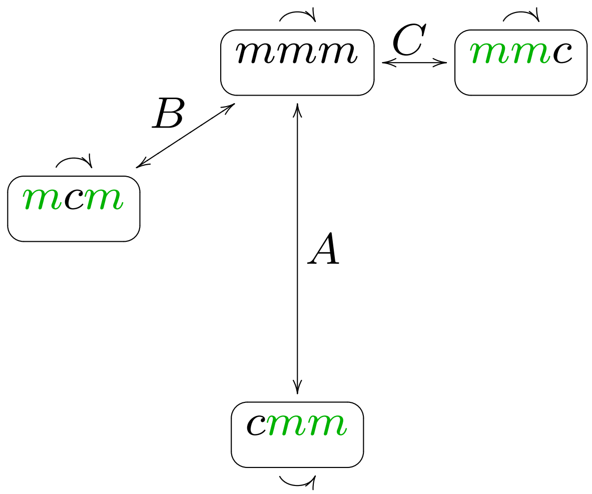
Duality

We could define $\Diamond_A\varphi$ to be $\neg\Box_A\neg\varphi$
read "A does not believe that φ is not the case".

There is similar duality between $\exists$ and $\forall$:
$\exists x\varphi(x)$ is equivalent to $\neg\forall x\neg\varphi(x)$.

$\Box$ and $\Diamond$ can be treated as local $\forall$ and $\exists$ respectively.

Examples Concerning Semantics



$$(cmm) \models \neg\Box_A(Am) \wedge \Box_B(Bm) \wedge \Box_C(Cm)$$
$$(cmm) \models \neg\Box_A(Ac) \wedge \Box_B\neg(Bc) \wedge \Box_C\neg(Cc)$$
$$(cmm) \models \Diamond_A\Diamond_B(Bc)$$

Dynamics

Add to the language formulas of the form:

- $[\psi!]\varphi$,

  read as "φ results from any truthful announcement of ψ"

  or "If ψ can truthfully be announced, then φ will be case after the announcement".

Example: $[\neg\Box_A(Am) \wedge \neg\Box_B(Bm) \wedge \neg\Box_C(Cm)!]\Box_B(Bm)$, read as "After it is announced that each agent does not know he is muddy, B knows that he is muddy".

Such as language is called *Public Announcement Logic*, which is a kind of *Dynamic Epistemic Logic*.

Semantics with Dynamics

Define relation $\models$ between model/state pairs $(\mathcal{S}, s)$ (with s in $\mathcal{S}$) and formulas as follows:

$(\mathcal{S}, s) \models [\psi!]\varphi$ iff $(\mathcal{S}', s) \models \varphi$ whenever $(\mathcal{S}, s) \models \psi$ and $\mathcal{S}'$ is the state model obtained from $\mathcal{S}$ by removing any state for which ψ is not true.

Duality for Actions

Define $\langle\psi!\rangle\varphi \equiv \neg[\psi!]\neg\varphi$

"It is not the case that if ψ can truthfully be announced, then announcing it will result in $\neg\varphi$."
or
"ψ can be truthfully announced, and announcing ψ will result in φ"

Sum and Product

A says to S and P: "I have chosen two integers x, y such that $1 < x < y$ and $x + y \leq 100$. In a moment, I will inform S only of $S = x + y$ and P only of $p = xy$." After the announcements are made, the following conversation takes place.

1. P says: "I do not know the pair".

2. S says: "I knew you didn't".

3. P says: "I now know the pair".

4. S says: "I now know too".

Adding Previous-Time Operator

Include formulas of the form:

- $Y_{\Diamond}\varphi$ read as "Previously φ".

Such a language is called *Temporal Public Announcement Logic*.

Language for Sum and Product

$\mathcal{A} = \{P, S\}$
$\Phi = \{(x = n), (y = m) : 1 < n, m < 100\}$
Let Pknows $\equiv \bigvee\{\Box_P((x = n) \wedge (y = m)) : 1 < n < m < 100\}$.
and Sknows $\equiv \bigvee\{\Box_S((x = n) \wedge (y = m)) : 1 < n < m < 100\}$.

1. $\neg P$knows

2. $Y_\Diamond \Box_S \neg P$knows

3. Pknows

4. Sknows

Model for Sum and Product

Construct a model $(S, \xrightarrow{S}, \xrightarrow{P}, \|\cdot\|)$ such that

$S = \{(n, m) : 1 < n < m < 100\}$

$(n, m) \xrightarrow{S} (n', m')$ iff $n + m = n' + m'$.
$(n, m) \xrightarrow{P} (n', m')$ iff $nm = n'm'$.

$\|(x = n)\| = \{(a, b) : a = n\}$
$\|(y = m)\| = \{(a, b) : b = m\}$.

Formula for Sum and Product Conversation

Define $\top = p \vee \neg p$ (this formula is always true)

Finding a solution to Sum and Product amounts to determining if there is a state in the Sum and Product Model for which the following is true.

$\langle \neg P\text{knows!}\rangle \langle Y_{\Diamond} \Box_S \neg P\text{knows!}\rangle \langle P\text{knows!}\rangle \langle S\text{knows!}\rangle \top$

Model Checker Programs

There is a unique solution to Sum and Product. It can be solved by a computer program.

DEMO (Dynamic Epistemic MOdelling) is an existing program, written in Haskell, that can check whether a formula in Dynamic Epistemic Logic is true at a certain state.

DEMO has been used to solve an equivalent version of Sum and Product that does not use the past tense.

Avoiding Semantics

A *proof system* is a set of axioms and rules that can be used to generate (or derive) a set of formulas called *theorems*.

A proof system may be an adequate way of determining what formulas are always true (valid) in the semantics if the following hold:

- Soundness: Every theorem generated by the proof system is valid in the semantics

- Completeness: Every formula that is valid in the semantics is a theorem generated by the proof system.

Surprise Exam Puzzle

Can a teacher truthfully announce the following: "You will have an exam during class next week, but the exam will be a surprise in that you will not know the exam will be given that day until it is actually given out."?

Assume the last day of class is Friday. Can the exam be given out on Friday? What about Thursday? Wednesday?

Related Papers

1. J. van Benthem. *What One May Come to Know*. ms. Universiteit van Amsterdam, 2007.

2. J. van Benthem, E. Pacuit. The Tree of Knowledge in Action: Towards a Common Perspective. Proceedings of *Advances in Modal Logic*, 2006.

3. H.P. van Ditmarsch, J. Ruan, L.C. Verbrugge. Sum and Product in Dynamic Epistemic Logic. *Journal of Logic and Computation*, to appear 2007.

4. D. Gerbrandy. *Bisimulations on Planet Kripke*. Dissertation, ILLC, 1998.

5. B. Kooi: Probabilistic Dynamic Epistemic Logic. *Journal of Logic Language and Information*, 12(4): 381-408, 2003.

6. J. Sack. Temporal Languages for Epistemic Programs. To appear in the *Journal of Logic Language and Information*. 2007

7. A. Yap. Product Update and Looking Backward. ms. 2005, Stanford University. Also at www.illc.uva.nl/lgc/papers/bms-temporal.pdf