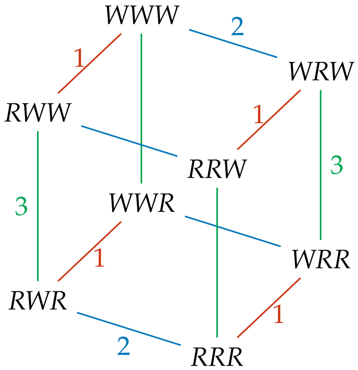
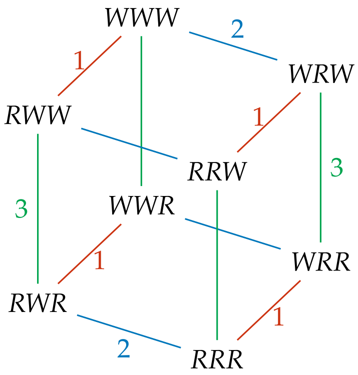
PHIL 222
Philosophical Foundations of Computer Science
Week 10, Thursday

Oct. 31, 2024

Epistemology (2)
Multi-Agent Systems and AI:
Logic of “Knows That”
(cont’d)

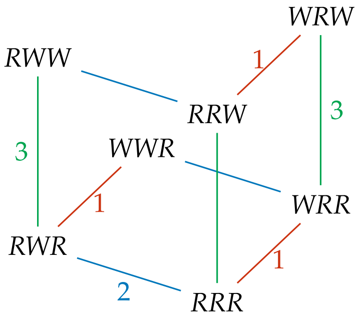


- In *RRR*, every agent knows that at least one hat is red. But . . .



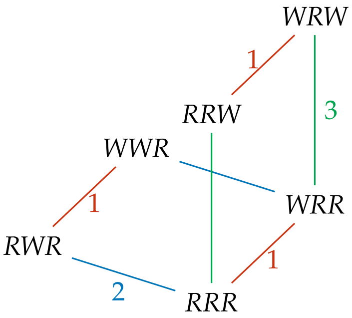
- In RRR, every agent knows that at least one hat is red. But . . .

- The announcement "At least one hat is red" makes it possible for them to jointly infer something new.



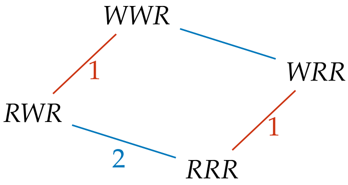
- In *RRR*, every agent knows that at least one hat is red. But . . .

- The announcement "At least one hat is red" makes it possible for them to jointly infer something new.



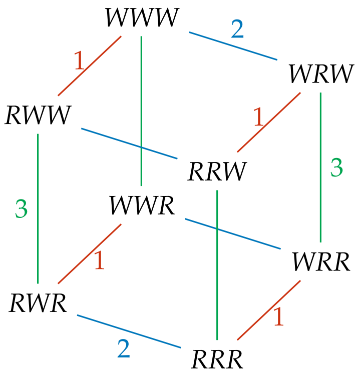
- In *RRR*, every agent knows that at least one hat is red. But . . .

- The announcement “At least one hat is red” makes it possible for them to jointly infer something new.



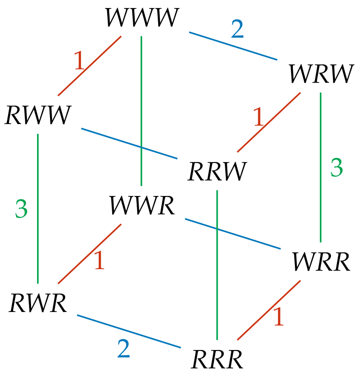
- In RRR, every agent knows that at least one hat is red. But . . .

- The announcement "At least one hat is red" makes it possible for them to jointly infer something new.

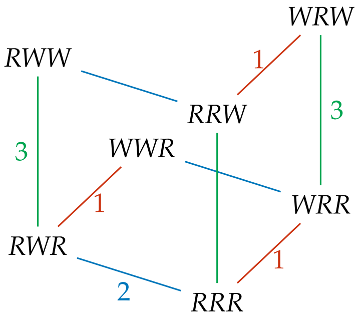


- In *RRR*, every agent knows that at least one hat is red. But . . .

- The announcement "At least one hat is red" makes it possible for them to jointly infer something new.



- In *RRR*, every agent knows that at least one hat is red. But . . .
- In *RRR*, it is not the case, e.g., that
  - 3 knows that 2 knows that 1 knows that at least one hat is red.
- The announcement "At least one hat is red" makes it possible for them to jointly infer something new.



- In *RRR*, every agent knows that at least one hat is red. But . . .
- In *RRR*, it is not the case, e.g., that
  - ⊛ 3 knows that 2 knows that 1 knows that at least one hat is red.
- The announcement “At least one hat is red” makes it possible for them to jointly infer something new.
- . . . because it makes sure, e.g., that ⊛ is the case.

We say

- "It is **common knowledge** (among agents $\alpha, \beta, \gamma, \ldots$) that φ"

to mean the big conjunction of

- "α knows that β knows that γ knows that φ", etc.

Public announcement is one way to bring about common knowlege.

We say

- "It is **common knowledge** (among agents α, β, γ, $\ldots$) that φ"

to mean the big conjunction of

- "α knows that β knows that γ knows that φ", etc.

Public announcement is one way to bring about common knowlege.

(The concept of common knowledge is due to several philosophers:

- It played a role in Hume's work (1740).
- The contemporary form of the concept is due to David Lewis (1969), who applied it to the study of the concept of conventions.

The partition model is also due to several philosophers, Saul Kripke and Jaakko Hintikka among others.)

We say

- "It is **common knowledge** (among agents $\alpha, \beta, \gamma, \ldots$) that φ"

to mean the big conjunction of

- "α knows that β knows that γ knows that φ", etc.

Public announcement is one way to bring about common knowlege.

(The concept of common knowledge is due to several philosophers:

- It played a role in Hume's work (1740).
- The contemporary form of the concept is due to David Lewis (1969), who applied it to the study of the concept of conventions.

The partition model is also due to several philosophers, Saul Kripke and Jaakko Hintikka among others.)

A more general point: How information is communicated is essential to what agents can jointly infer.

Epistemology (2)
Multi-Agent Systems and AI:
“Knows That” and Distributed Computing

If computability theory is a theory of what can(not) be computed by a single process, the theory of **distributed computing** is a theory of what can(not) be computed by a collection of processes through communication.

Herlihy et al., *Distributed Computing Through Combinatorial Topology*:

> A *system* is a collection of *processes*, together with a communication environment such as shared read-write memory [...]. A process represents a sequential computing entity, modeled formally as a state machine. Each process executes a finite *protocol*. It starts in an initial state and takes steps until it either *fails*, meaning it halts and takes no additional steps, or it *halts*, usually because it has completed the protocol. Each step typically involves local computation as well as communicating with other processes through the environment provided by the model. Processes are deterministic: Each transition is determined by the process's current state and the state of the environment. [p. 10]

In distributed computing, the analog of a function is called a *task*. An input to a task is distributed: Only part of the input is given to each process. The output from a task is also distributed: Only part of the output is computed by each process. The task specification states which outputs can be produced in response to each input. A *protocol* is a concurrent algorithm to solve a task; initially each process knows its own part of the input, but not the others'. Each process communicates with the others and eventually halts with its own output value. Collectively, the individual output values form the task's output. [p. 12]

The question of what it means for a function to be *computable* is one of the deepest questions addressed by computer science. In sequential systems, computability is understand [*sic*] through the *Church-Turing thesis* [. . .].

In distributed computing, where computations require coordination among multiple participants, computability questions have a different flavor. Here, too, there are many problems that are not computable, but these computability failures reflect the difficulty of making decisions in the face of ambiguity and have little to do with the inherent computational power of individual participants. If the participants could reliably and instantaneously communicate with one another, then each one could learn the complete system state and perform the entire computation by itself. In any realistic model of distributed computing, however, each participant initially knows only part of the global system state, and uncertainties caused by failures and unpredictable timing limit each participant to an incomplete picture. [pp. 11–12]

The puzzle of the hats (a.k.a. the "muddy children puzzle") was a task in which

- each agent's input is the color of the other agents;
- communication is done by "public announcement";
- the (desired) output is "I know my hat color".

The puzzle of the hats (a.k.a. the “muddy children puzzle”) was a task in which

- each agent’s input is the color of the other agents;
- communication is done by “public announcement”;
- the (desired) output is “I know my hat color”.

Recall:

- “It is common knowledge (among agents $\alpha, \beta, \gamma, \ldots$) that φ”

is the big conjunction of

- “α knows that β knows that γ knows that φ”, etc.

The puzzle of the hats (a.k.a. the “muddy children puzzle”) was a task in which

- each agent’s input is the color of the other agents;
- communication is done by “public announcement”;
- the (desired) output is “I know my hat color”.

Recall:

- “It is common knowledge (among agents $\alpha, \beta, \gamma, \dots$) that φ”

is the big conjunction of

- “α knows that β knows that γ knows that φ”, etc.

We saw

- public announcement is one way to bring about common knowlege;
- some common knowledge enables the agents to achieve the output.

Here is **the coordinated attack problem**:

> Two army divisions, one commanded by General Alice and one by General Bob, are camped on two hilltops overlooking a valley. The enemy is camped in the valley. If both divisions attack simultaneously, they will win, but if only one division attacks by itself, it will be defeated. As a result, neither general will attack without a guarantee that the other will attack at the same time. In particular, neither general will attack without communication from the other. [Herlihy et al., p. 16]

At the time the divisions are deployed on the hilltops, the generals had not agreed on whether [...] to attack. Now Alice decides to schedule an attack. The generals can communicate only by messengers. [p. 16]

> At the time the divisions are deployed on the hilltops, the generals had not agreed on whether [...] to attack. Now Alice decides to schedule an attack. The generals can communicate only by messengers. [p. 16]

- α's input: she has decided to attack.
- β's input: Does not know whether α has decided so.
- Messengers may reach the other hilltop but may fail.
- The desired output: attack.

SOME CONSTRAINTS AND TRADEOFFS
IN THE DESIGN OF
NETWORK COMMUNICATIONS*

E. A. Akkoyunlu
K. Ekanadham
R. V. Huber†
Department of Computer Science
State University of New York at Stony Brook

A number of properties and features of interprocess communication systems are presented, with emphasis on those necessary or desirable in a network environment. The interactions between these features are examined, and the consequences of their inclusion in a system are explored. Of special interest are the time-out feature which forces all system table entries to "die of old age" after they have remained unused for some period of time, and the insertion property which states that it is always possible to design a process which may be invisibly inserted into the communication path between any two processes. Though not tied to any particular system, the discussion concentrates on distributed systems of sequential processes (no interrupts) with no system buffering.

Key Words and Phrases: interprocess communication, computer networks, ports.

CR Categories: 3.81, 4.32, 4.39

1. Introduction

The design of an interprocess communication mechanism (IPCM) usually starts with a description of the desired behavior of the system and the services to be provided. In selecting the features to be incorporated into the IPCM, the greatest amount of care is required, for these features are interdependent to a great degree, and it is crucial that the design process start with a complete, detailed specification of the system to be designed, with the consequences of each decision fully explored and understood. The temptation of piecemeal design is to be avoided at all costs.

The major aim of the paper is to point out the interdependence of the features to be incorporated results in some horrendous code 'patched' into the system, and much elegance is lost. The resulting system is harder to implement, verify, understand, debug, and maintain. These are the questions which extract a "well, we didn't actually implement it that way," response from system designers.

Unfortunately, with few exceptions [6, 10, 15], there is little guidance to be found in the published literature on this important point - how to arrive at a consistent and elegant design. This paper is a modest attempt to help fill this gap. The paper will address itself to general concepts rather than to the specifics of a particular design, although it was influenced to a considerable degree by the experience gained in the design and implementation of the Stony Brook System [2]. A

General α

General β

General α

General β

β knows that α has decided to attack.

General α → messenger → General β

β knows that α has decided to attack.

General α $\xrightarrow{\text{messenger}}$ General β

β knows that α has decided to attack.

α knows that β knows that α has decided to attack.

General α ← messenger — General β

β knows that α has decided to attack.

α knows that β knows that α has decided to attack.

General α ← messenger — General β

β knows that α has decided to attack.

α knows that β knows that α has decided to attack.

β knows that α knows that β knows that α has decided to attack.

$\vdots$

General α messenger $\rightarrow$ General β

β knows that α has decided to attack.

α knows that β knows that α has decided to attack.

β knows that α knows that β knows that α has decided to attack.

$\vdots$

General α → messenger → General β

β knows that α has decided to attack.

α knows that β knows that α has decided to attack.

β knows that α knows that β knows that α has decided to attack.

$\vdots$

Disclaimer: There are so many approaches to this problem.
Even using logic, there are many ways to model it. Let's see one way.

1. When can they attack?

1. When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

1. When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

i. $$\text{Attack} \implies \alpha \text{ has decided}$$

❶ When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

ⓘ $$\text{Attack} \implies \alpha \text{ has decided}$$

ⓘⓘ $$\text{Attack} \implies \alpha \text{ knows that Attack}$$

ⓘⓘⓘ $$\text{Attack} \implies \beta \text{ knows that Attack}$$

1. When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

i. $$\text{Attack} \implies \alpha \text{ has decided}$$

ii. $$\text{Attack} \implies \alpha \text{ knows that Attack}$$

iii. $$\text{Attack} \implies \beta \text{ knows that Attack}$$

Then it follows that, e.g.,

1 When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

i $$\text{Attack} \implies \alpha \text{ has decided}$$

ii $$\text{Attack} \implies \alpha \text{ knows that Attack}$$

iii $$\text{Attack} \implies \beta \text{ knows that Attack}$$

Then it follows that, e.g.,

iv $$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

by i,

1 When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

i
$$\text{Attack} \implies \alpha \text{ has decided}$$

ii
$$\text{Attack} \implies \alpha \text{ knows that Attack}$$

iii
$$\text{Attack} \implies \beta \text{ knows that Attack}$$

Then it follows that, e.g.,

iv
$$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

by i, but then by iii + iv that

v
$$\text{Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

1 When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

i $$\text{Attack} \implies \alpha \text{ has decided}$$

ii $$\text{Attack} \implies \alpha \text{ knows that Attack}$$

iii $$\text{Attack} \implies \beta \text{ knows that Attack}$$

Then it follows that, e.g.,

iv $$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

by i, but then by iii + iv that

v $$\text{Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

Using v similarly to iv and using ii,

$$\text{Attack} \implies \alpha \text{ knows that } \beta \text{ knows that } \alpha \text{ has decided}$$

1 When can they attack?

Let's write "Attack" for the condition under which α and β will attack. It has several necessary conditions:

i $$\text{Attack} \implies \alpha \text{ has decided}$$

ii $$\text{Attack} \implies \alpha \text{ knows that Attack}$$

iii $$\text{Attack} \implies \beta \text{ knows that Attack}$$

Then it follows that, e.g.,

iv $$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

by i, but then by iii + iv that

v $$\text{Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

Using v similarly to iv and using ii,

$$\text{Attack} \implies \alpha \text{ knows that } \beta \text{ knows that } \alpha \text{ has decided}$$

In sum,

$$\text{Attack} \implies \text{it is common knowledge that } \alpha \text{ has decided}$$

2. Can they attack?

2. Can they attack?

Let's recap how the partition model works:

② Can they attack?

Let's recap how the partition model works:

- There are several possibilities, called "states", in each of which atomic sentences may be true or false (like in rows of a truth table).

RWW *WWW* *WRW* *RRW*

2. Can they attack?

Let's recap how the partition model works:

- There are several possibilities, called "states", in each of which atomic sentences may be true or false (like in rows of a truth table).
- Each agent partitions the set of states into several cells (which we indicate with edges connecting states in the same cell). She can distinguish states from different cells (i.e. unconnected states) but not ones from the same cell (i.e. connected states).

$$RWW \overset{1}{\text{———}} WWW \qquad WRW \overset{1}{\text{———}} RRW$$

2. Can they attack?

Let's recap how the partition model works:

- There are several possibilities, called "states", in each of which atomic sentences may be true or false (like in rows of a truth table).
- Each agent partitions the set of states into several cells (which we indicate with edges connecting states in the same cell). She can distinguish states from different cells (i.e. unconnected states) but not ones from the same cell (i.e. connected states).
- "α knows that φ" is true in a state w

  $\iff$ "φ" is true in all the states that α puts in the same cell as w (i.e. all the states connected to w by an α-edge).

$$RWW \overset{1}{\text{———}} WWW \qquad WRW \overset{1}{\text{———}} RRW$$

② Can they attack?

Let's recap how the partition model works:

- There are several possibilities, called "states", in each of which atomic sentences may be true or false (like in rows of a truth table).
- Each agent partitions the set of states into several cells (which we indicate with edges connecting states in the same cell). She can distinguish states from different cells (i.e. unconnected states) but not ones from the same cell (i.e. connected states).
- "α knows that φ" is true in a state w

$\iff$ "φ" is true in all the states that α puts in the same cell as w (i.e. all the states connected to w by an α-edge).

$$RWW \overset{1}{\text{———}} WWW \qquad WRW \overset{1}{\text{———}} RRW$$

"1 knows that 2 has a white hat" is true in WWW,

2. Can they attack?

Let's recap how the partition model works:

- There are several possibilities, called "states", in each of which atomic sentences may be true or false (like in rows of a truth table).
- Each agent partitions the set of states into several cells (which we indicate with edges connecting states in the same cell). She can distinguish states from different cells (i.e. unconnected states) but not ones from the same cell (i.e. connected states).
- "α knows that φ" is true in a state w

  $\iff$ "φ" is true in all the states that α puts in the same cell as w (i.e. all the states connected to w by an α-edge).

$$RWW \overset{1}{\text{———}} WWW \qquad WRW \overset{1}{\text{———}} RRW$$

"1 knows that 2 has a white hat" is true in WWW,
"1 knows that 1 has a white hat" is false in WWW.

2. Can they attack?

2. Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".

❷ Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".

$$\begin{tikzcd} D \arrow[r, dash, "\beta"] & N \end{tikzcd}$$

② Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".

$$D \;\overset{\beta}{\overline{\hspace{6em}}}\; N$$

In the case D, "β knows that D" fails to hold.

❷ Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".

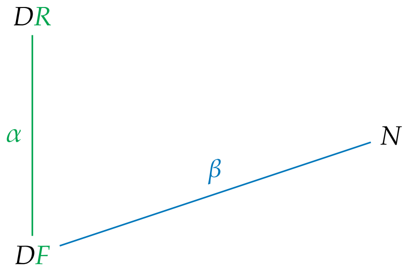
2. Can they attack?

Let's write D for "α has decided to attack",

N for "α has not decided to attack".

R for "the messenger has reached the other side",

F for "the messenger has failed to reach the other side".



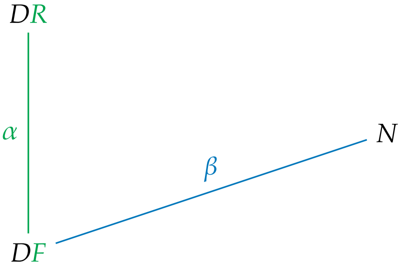
2 Can they attack?

Let's write D for "α has decided to attack",

N for "α has not decided to attack".

R for "the messenger has reached the other side",

F for "the messenger has failed to reach the other side".



In the case DR, "α knows that β knows that D" fails.

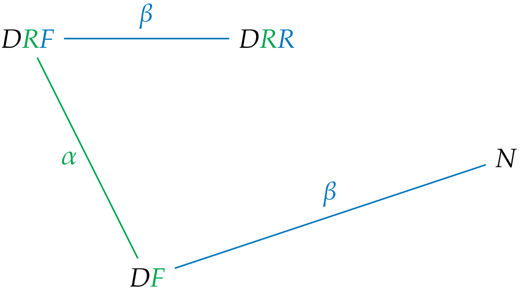
2 Can they attack?

Let's write D for "α has decided to attack",

N for "α has not decided to attack".

R for "the messenger has reached the other side",

F for "the messenger has failed to reach the other side".



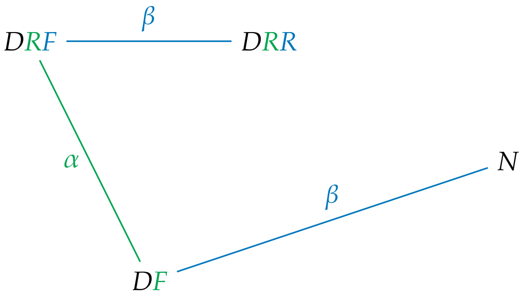
❷ Can they attack?

Let's write D for "α has decided to attack",

N for "α has not decided to attack".

R for "the messenger has reached the other side",

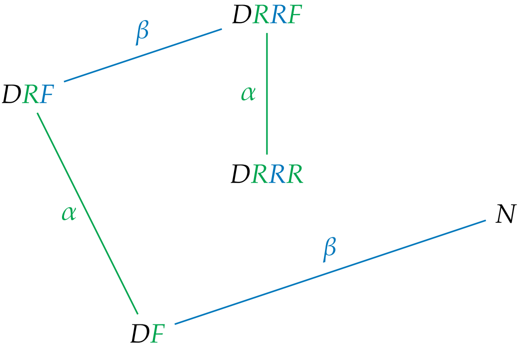
F for "the messenger has failed to reach the other side".



In the case DRR, "β knows that α knows that β knows that D" fails.

❷ Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".



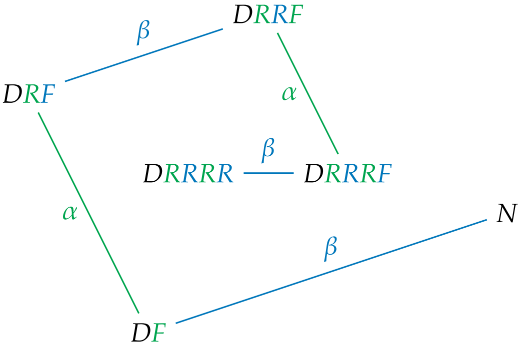
② Can they attack?

Let's write D for "α has decided to attack",

N for "α has not decided to attack".

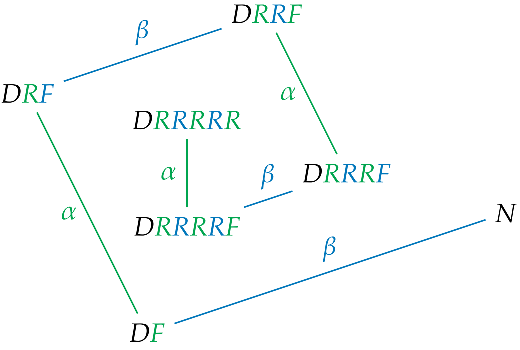
R for "the messenger has reached the other side",

F for "the messenger has failed to reach the other side".



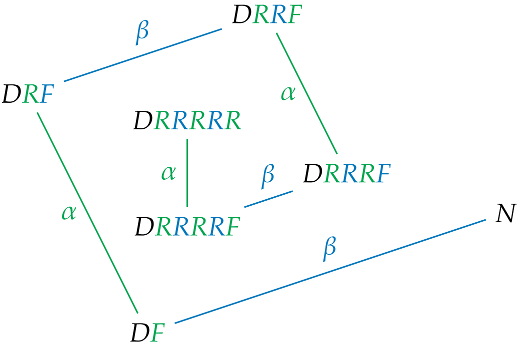
2. Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".



❷ Can they attack?

Let's write D for "α has decided to attack",
N for "α has not decided to attack".
R for "the messenger has reached the other side",
F for "the messenger has failed to reach the other side".



In the case $DRRRRR\cdots$, it is never common knowledge that D.

In sum, the task is:

In sum, the task is:

- Input:

$$D \xrightarrow{\quad\beta\quad} N$$

In sum, the task is:

- Input:

$$D \overset{\beta}{\text{———}} N$$

- (Desired) output:

  a non-connected graph in which some state w is detached from all N-states, so that there is no path from w to an N-state

  (because then, in w, it is common knowledge that D).

In sum, the task is:

- Input:

$$D \xrightarrow{\quad\beta\quad} N$$

- (Desired) output:

  > a non-connected graph in which some state w is detached from all N-states, so that there is no path from w to an N-state

  (because then, in w, it is common knowledge that D).

This is not solvable in the given communication environment, since

- applying "send a messenger" to a connected graph always results in another connected graph.

In sum, the task is:

- Input:

$$D \overset{\beta}{\text{———}} N$$

- (Desired) output:

  a non-connected graph in which some state w is detached from all N-states, so that there is no path from w to an N-state

  (because then, in w, it is common knowledge that D).

This is not solvable in the given communication environment, since

- applying "send a messenger" to a connected graph always results in another connected graph.

$$DRR \overset{\beta}{\text{———}} DRF \overset{\alpha}{\text{———}} DF \overset{\beta}{\text{———}} N$$

In sum, the task is:

- Input:

$$D \overset{\beta}{\text{———}} N$$

- (Desired) output:

  a non-connected graph in which some state w is detached from all N-states, so that there is no path from w to an N-state

  (because then, in w, it is common knowledge that D).

This is not solvable in the given communication environment, since

- applying "send a messenger" to a connected graph always results in another connected graph.

$$DRR \overset{\beta}{\text{———}} DRF \overset{\alpha}{\text{———}} DF \overset{\beta}{\text{———}} N$$

$$DRRR \overset{\alpha}{\text{———}} DRRF \overset{\beta}{\text{———}} DRF \overset{\alpha}{\text{———}} DF \overset{\beta}{\text{———}} N$$

Recall this remark by Herlihy et al.:

> Here, too, there are many problems that are not computable, but these computability failures reflect the difficulty of making decisions in the face of ambiguity and have little to do with the inherent computational power of individual participants. [p. 11]

Recall this remark by Herlihy et al.:

> Here, too, there are many problems that are not computable, but these computability failures reflect the difficulty of making decisions in the face of ambiguity and have little to do with the inherent computational power of individual participants. [p. 11]

- Limited views of the state and limited ability in communication (unreliable, not-instantaneous, asynchronous, etc.) give rise to uncertainty, which is the source of uncomputability studied here.

Recall this remark by Herlihy et al.:

> Here, too, there are many problems that are not computable, but these computability failures reflect the difficulty of making decisions in the face of ambiguity and have little to do with the inherent computational power of individual participants. [p. 11]

- Limited views of the state and limited ability in communication (unreliable, not-instantaneous, asynchronous, etc.) give rise to uncertainty, which is the source of uncomputability studied here.

Recall this remark by Herlihy et al.:

> Here, too, there are many problems that are not computable, but these computability failures reflect the difficulty of making decisions in the face of ambiguity and have little to do with the inherent computational power of individual participants. [p. 11]

- Limited views of the state and limited ability in communication (unreliable, not-instantaneous, asynchronous, etc.) give rise to uncertainty, which is the source of uncomputability studied here.
- In fact, the inference rules we used and the partition model assume that agents have a *lot* of computational power, perhaps even beyond any Turing machine — which philosophers recognize as "**the problem of logical omniscience**" with the logic of "knows that".

Epistemology (2)
Multi-Agent Systems and AI:
The “Problem of Logical Omniscience”

This is the second “problem of something” we cover in epistemology.

This is the second “problem of something” we cover in epistemology.

We saw the problem of induction.

- “Induction is important, but here is a problem: it may lack foundation / justification.”
- So, a solution may try to defend induction.

This is the second "problem of something" we cover in epistemology.

We saw the problem of induction.

- "Induction is important, but here is a problem: it may lack foundation / justification."
- So, a solution may try to defend induction.

We are going to see the problem of logical omniscience, but . . .

- "Logical omniscience is impossible / absurd, but here is a problem: the logic of 'knows that' may end up with it."
- So, a solution may try to avert / explain away logical omniscience.

This is the second "problem of something" we cover in epistemology.

We saw the problem of induction.

- "Induction is important, but here is a problem: it may lack foundation / justification."
- So, a solution may try to defend induction.

We are going to see the problem of logical omniscience, but . . .

- "Logical omniscience is impossible / absurd, but here is a problem: the logic of 'knows that' may end up with it."
- So, a solution may try to avert / explain away logical omniscience.

Anyway, what is logical omniscience?

Recall that we applied inferences like the following.

From the following principle,

ⓘ $$\text{Attack} \implies \alpha \text{ has decided}$$

Recall that we applied inferences like the following.

From the following principle,

i $$\text{Attack} \implies \alpha \text{ has decided}$$

we derived

$$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

Recall that we applied inferences like the following.

From the following principle,

$$\text{Attack} \implies \alpha \text{ has decided}$$

we derived

$$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

In this derivation we used the following inference rule:

- If $\varphi \Rightarrow \psi$ is provable, then so is

$$\beta \text{ knows that } \varphi \implies \beta \text{ knows that } \psi.$$

Recall that we applied inferences like the following.

From the following principle,

$$\text{Attack} \implies \alpha \text{ has decided}$$

we derived

$$\beta \text{ knows that Attack} \implies \beta \text{ knows that } \alpha \text{ has decided}$$

In this derivation we used the following inference rule:

- If $\varphi \Rightarrow \psi$ is provable, then so is

$$\beta \text{ knows that } \varphi \implies \beta \text{ knows that } \psi.$$

We sort of justified this rule by saying:

"If *we* can infer that ψ follows from φ, why can't β? She should be able to do the same inference, right?"

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

You: What is the tallest building in the world?

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

You: What is the tallest building in the world?

Sili: The Burj Khalifa.

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

You: What is the tallest building in the world?

Sili: The Burj Khalifa.

You: How tall is the Burj Khalifa?

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

You: What is the tallest building in the world?

Sili: The Burj Khalifa.

You: How tall is the Burj Khalifa?

Sili: 830 m.

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

You: What is the tallest building in the world?

Sili: The Burj Khalifa.

You: How tall is the Burj Khalifa?

Sili: 830 m.

You: How tall is the tallest building in the world?

Indeed, an agent with no inference power at all would be quite silly.

E.g., suppose Sili's knowledge base contains the following data:

- The Burj Khalifa is a building with a height of 830 m.
- The Burj Khalifa is the tallest building in the world.

You: What is the tallest building in the world?

Sili: The Burj Khalifa.

You: How tall is the Burj Khalifa?

Sili: 830 m.

You: How tall is the tallest building in the world?

Sili: I don't know.

But how much inference power can / should we assume a human / AI agent to have?

But how much inference power can / should we assume a human / AI agent to have?

You: How tall is the Burj Khalifa?

But how much inference power can / should we assume a human / AI agent to have?

You: How tall is the Burj Khalifa?

Sili: 830 m.

But how much inference power can / should we assume a human / AI agent to have?

You: How tall is the Burj Khalifa?

Sili: 830 m.

You: What is the height of the Burj Khalifa times the biggest known prime number's next prime number?

But how much inference power can / should we assume a human / AI agent to have?

You: How tall is the Burj Khalifa?

Sili: 830 m.

You: What is the height of the Burj Khalifa times the biggest known prime number's next prime number?

Sili: I don't know.

But how much inference power can / should we assume a human / AI agent to have?

You: How tall is the Burj Khalifa?

Sili: 830 m.

You: What is the height of the Burj Khalifa times the biggest known prime number's next prime number?

Sili: I don't know.

Well, here it is *you* who are silly, isn't it? It seems silly to assume an agent to have super-power in inference.

But how much inference power can / should we assume a human / AI agent to have?

You: How tall is the Burj Khalifa?

Sili: 830 m.

You: What is the height of the Burj Khalifa times the biggest known prime number's next prime number?

Sili: I don't know.

Well, here it is *you* who are silly, isn't it? It seems silly to assume an agent to have super-power in inference.

The **problem of logical omniscience** is that, when we use the logic of "knows that" (either its inference rules or model), it is very hard not to end up assuming agents to have super-power in inference.