Temporal Logic of Knowledge and its applications in security

Cătălin Dima

LACL, Université Paris 12

8/12/2006

1. **Preliminaries**
   - The muddy children puzzle
   - Logics of knowledge and security
2. The bases
   - Syntax and semantics
   - Knowledge and time
   - Types of temporal knowledge
   - Axiomatics and decidability issues
3. Knowledge and information flow
   - Classical information flow properties

A puzzle game

- n children play together outside,
- None wants to get dirty (Dad punishes!), but would like to see the others dirty! (kids...)
- It happens that, at some moment, k of them get mud on their foreheads
  - ... so each of them cannot see if he's dirty or not!
  - ... and none signals anything to anybody who's dirty!
- Mum comes into the room and says
  At least one of you has mud on his forehead
- Then she asks everybody:
  Does anyone of you know whether you're dirty?
- Assuming that all children are intelligent, perceptive and truthful (!), what happens?

A puzzle game

- *n* children play together outside,
- None wants to get dirty (Dad punishes!), but would like to see the others dirty! (kids...)
- It happens that, at some moment, *k* of them get mud on their foreheads
  - ... so each of them cannot see if he's dirty or not!
  - ... and none signals anything to anybody who's dirty!
- Mum comes into the room and says
  At least one of you has mud on his forehead
- Then she asks everybody:
  Does anyone of you know whether you're dirty?
- Assuming that all children are intelligent, perceptive and truthful (!), what happens?

A puzzle game

- *n* children play together outside,
- None wants to get dirty (Dad punishes!), but would like to see the others dirty! (kids...)
- It happens that, at some moment, *k* of them get mud on their foreheads
  - ... so each of them cannot see if he's dirty or not!
  - ... and none signals anything to anybody who's dirty!
- Mum comes into the room and says
  At least one of you has mud on his forehead
- Then she asks everybody:
  Does anyone of you know whether you're dirty?
- Assuming that all children are intelligent, perceptive and truthful (!), what happens?

A puzzle game

- n children play together outside,
- None wants to get dirty (Dad punishes!), but would like to see the others dirty! (kids...)
- It happens that, at some moment, k of them get mud on their foreheads
  - ... so each of them cannot see if he's dirty or not!
  - ... and none signals anything to anybody who's dirty!
- Mum comes into the room and says
  At least one of you has mud on his forehead
- Then she asks everybody:
  Does anyone of you know whether you're dirty?
- Assuming that all children are intelligent, perceptive and truthful (!), what happens?

Solving the puzzle game

- There is a “formal” proof that
  - the first $k-1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k = 1$ it’s obvious (ain’t it?).
  - For $k = 2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he’s the only one to be dirty, he must have seen mud on someone else’s forehead.
  - So it must be his (a’s) own forehead that was muddy!
  - Generalize the reasoning!

Solving the puzzle game

- There is a "formal" proof that
  - the first $k-1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k = 1$ it's obvious (ain't it?).
  - For $k = 2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he's the only one to be dirty, he must have seen mud on someone else's forehead.
  - So it must be his (a's) own forehead that was muddy!
  - Generalize the reasoning!

Solving the puzzle game

- There is a "formal" proof that
  - the first $k-1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k = 1$ it's obvious (ain't it?).
  - For $k = 2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he's the only one to be dirty, he must have seen mud on someone else's forehead.
  - So it must be his (a's) own forehead that was muddy!
  - Generalize the reasoning!

Solving the puzzle game

- There is a “formal” proof that
  - the first $k-1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k=1$ it's obvious (ain't it?).
  - For $k=2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he's the only one to be dirty, he must have seen mud on someone else's forehead.
  - So it must be his (a's) own forehead that was muddy!
  - Generalize the reasoning!

Solving the puzzle game

- There is a “formal” proof that
  - the first $k-1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k = 1$ it’s obvious (ain’t it?).
  - For $k = 2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he’s the only one to be dirty, he must have seen mud on someone else’s forehead.
  - So it must be his (a’s) own forehead that was muddy!
  - Generalize the reasoning!

Solving the puzzle game

- There is a "formal" proof that
  - the first $k - 1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k = 1$ it's obvious (ain't it?).
  - For $k = 2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he's the only one to be dirty, he must have seen mud on someone else's forehead.
  - So it must be his (a's) own forehead that was muddy!
  - Generalize the reasoning!

Solving the puzzle game

- There is a "formal" proof that
  - the first $k-1$ times Mum asks her question, all will say No, but
  - the k^{th} time she asks her question, exactly those children with muddy foreheads will say Yes, I am dirty!
- Proof: by induction on k:
  - For $k=1$ it's obvious (ain't it?).
  - For $k=2$, the first time everybody says No.
  - ... but then everybody will notice that the two muddy children do not know they are dirty.
  - Hence muddy a concludes that, since muddy b does not deduce that he's the only one to be dirty, he must have seen mud on someone else's forehead.
  - So it must be his (a's) own forehead that was muddy!
  - Generalize the reasoning!

Muddy children and knowledge

- All children do their reasoning provided they know some properties...
- ... and deduce (know) later that the others do not know some other properties.
- Mum's questions serve as synchronization steps.
- Without these, there could be no way for children to achieve their deductions!
- Step $k+1$ also represents the convergence of the system to common knowledge.
    - That is, everybody knows that everybody knows that everybody knows that ...... that $a_1 \ldots a_k$ are dirty

Why studying logics of knowledge?

- Epistemic logics are important in multi-agent systems.
  - Originally developed for AI.
- Security analysis involves at least two agents: the legitimate user(s) and the intruder(s).
- In security protocol analysis, we speak about intruder knowledge!
- Information flow analysis also is concerned with the information an agent gains about security levels to which he is not authorized to access.
  - Information is closely related to knowledge.

What characterizes a logic?

- Its syntax.
- Its semantics.
- Its axiomatic system.
- The possibility to "mechanicise" the deduction = decidability of various decision problems.
- Various interesting extensions.
- Applications in the study of information flow.

Basic knowledge operators

- n agent system – call them $1, 2, \ldots, n$.
- $K_i\phi$: agent i knows formula ϕ.
- Examples:
  1. n children play their muddy forehead game.
  2. p_2 : child i has mud on his forehead.
  3. $K_4 p_2$: child 4 knows that child 2 is muddy.
  4. $K_1(K_4 p_2 \wedge p_1)$:
     - child 1 knows that child 2 knows that 2 is muddy...
     - ... and also knows that he himself is muddy!
- All the other boolean operators: $\wedge, \vee, \neg, \rightarrow \ldots$.
- Temporal operators will be added later!

Basic knowledge operators

- n agent system – call them $1, 2, \ldots, n$.
- $K_i\phi$: agent i knows formula ϕ.
- Examples:
  1. n children play their muddy forehead game.
  2. p_2 : child i has mud on his forehead.
  3. $K_4 p_2$: child 4 knows that child 2 is muddy.
  4. $K_1(K_4 p_2 \wedge p_1)$:
     - child 1 knows that child 2 knows that 2 is muddy...
     - ... and also knows that he himself is muddy!
- All the other boolean operators: $\wedge, \vee, \neg, \rightarrow \ldots$.
- Temporal operators will be added later!

Basic knowledge operators

- n agent system – call them $1, 2, \ldots, n$.
- $K_i\phi$: agent i knows formula ϕ.
- Examples:
  1. n children play their muddy forehead game.
  2. p_2 : child i has mud on his forehead.
  3. $K_4 p_2$: child 4 knows that child 2 is muddy.
  4. $K_1(K_4 p_2 \wedge p_1)$:
     - child 1 knows that child 2 knows that 2 is muddy...
     - ... and also knows that he himself is muddy!
- All the other boolean operators: $\wedge, \vee, \neg, \rightarrow \ldots$.
- Temporal operators will be added later!

Basic knowledge operators

- n agent system – call them $1, 2, \ldots, n$.
- $K_i\phi$: agent i knows formula ϕ.
- Examples:
  1. n children play their muddy forehead game.
  2. p_2 : child i has mud on his forehead.
  3. $K_4 p_2$: child 4 knows that child 2 is muddy.
  4. $K_1(K_4 p_2 \wedge p_1)$:
     - child 1 knows that child 2 knows that 2 is muddy...
     - ... and also knows that he himself is muddy!
- All the other boolean operators: $\wedge, \vee, \neg, \rightarrow \ldots$.
- Temporal operators will be added later!

Basic knowledge operators

- n agent system – call them $1, 2, \ldots, n$.
- $K_i\phi$: agent i knows formula ϕ.
- Examples:
  1. n children play their muddy forehead game.
  2. p_2 : child i has mud on his forehead.
  3. $K_4 p_2$: child 4 knows that child 2 is muddy.
  4. $K_1(K_4 p_2 \wedge p_1)$:
     - child 1 knows that child 2 knows that 2 is muddy...
     - ... and also knows that he himself is muddy!
- All the other boolean operators: $\wedge, \vee, \neg, \rightarrow \ldots$.
- Temporal operators will be added later!

Basic knowledge operators

- n agent system – call them $1, 2, \dots, n$.
- $K_i\phi$: agent i knows formula ϕ.
- Examples:
  1. n children play their muddy forehead game.
  2. p_2 : child i has mud on his forehead.
  3. $K_4 p_2$: child 4 knows that child 2 is muddy.
  4. $K_1(K_4 p_2 \wedge p_1)$:
     - child 1 knows that child 2 knows that 2 is muddy...
     - ... and also knows that he himself is muddy!
- All the other boolean operators: $\wedge, \vee, \neg, \rightarrow \dots$.
- Temporal operators will be added later!

Semantics

Possible worlds model: Kripke structure for n agents:
$M = (S, \Pi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- S – the set of global states.
  - Sometimes $S = S_1 \times \ldots \times S_n$.
  - S_i = local states for agent i.
- Π – set of primitive propositions (like p_2 : child i is muddy).
- $\pi : S \to 2^\Pi$ – truth value for each primite proposition in each state.
- $\mathcal{K}_i$ – the indistinguishability relation (also called the *possibility* relation).
  - $\mathcal{K}_i(s, s')$ = for agent i, states s and s' cannot be distinguished by prior observation – i.e., according to i's knowledge!
  - Very often $\mathcal{K}_i$ are reflexive, symmetric & transitive – i.e. equivalence relations.

Semantics

Possible worlds model: Kripke structure for n agents:
$M = (S, \Pi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- S – the set of global states.
  - Sometimes $S = S_1 \times \ldots \times S_n$.
  - S_i = local states for agent i.
- Π – set of primitive propositions (like p_2 : child i is muddy).
- $\pi : S \to 2^{\Pi}$ – truth value for each primite proposition in each state.
- $\mathcal{K}_i$ – the indistinguishability relation (also called the *possibility* relation).
  - $\mathcal{K}_i(s, s')$ = for agent i, states s and s' cannot be distinguished by prior observation – i.e., according to i's knowledge!
  - Very often $\mathcal{K}_i$ are reflexive, symmetric & transitive – i.e. equivalence relations.

Semantics

Possible worlds model: Kripke structure for n agents:
$M = (S, \Pi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- S – the set of global states.
  - Sometimes $S = S_1 \times \ldots \times S_n$.
  - S_i = local states for agent i.
- Π – set of primitive propositions (like p_2 : child i is muddy).
- $\pi : S \to 2^{\Pi}$ – truth value for each primite proposition in each state.
- $\mathcal{K}_i$ – the indistinguishibility relation (also called the *possibility* relation).
  - $\mathcal{K}_i(s, s')$ = for agent i, states s and s' cannot be distinguished by prior observation – i.e., according to i's knowledge!
  - Very often $\mathcal{K}_i$ are reflexive, symmetric & transitive – i.e. equivalence relations.

Semantics

Possible worlds model: Kripke structure for n agents:
$M = (S, \Pi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- S – the set of global states.
  - Sometimes $S = S_1 \times \ldots \times S_n$.
  - S_i = local states for agent i.
- Π – set of primitive propositions (like p_2 : child i is muddy).
- $\pi : S \to 2^\Pi$ – truth value for each primite proposition in each state.
- $\mathcal{K}_i$ – the indistinguishability relation (also called the *possibility* relation).
  - $\mathcal{K}_i(s, s')$ = for agent i, states s and s' cannot be distinguished by prior observation – i.e., according to i's knowledge!
  - Very often $\mathcal{K}_i$ are reflexive, symmetric & transitive – i.e. equivalence relations.

Semantics

Possible worlds model: Kripke structure for n agents:
$M = (S, \Pi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- S – the set of global states.
  - Sometimes $S = S_1 \times \ldots \times S_n$.
  - S_i = local states for agent i.
- Π – set of primitive propositions (like p_2 : child i is muddy).
- $\pi : S \to 2^{\Pi}$ – truth value for each primite proposition in each state.
- $\mathcal{K}_i$ – the indistinguishibility relation (also called the *possibility* relation).
  - $\mathcal{K}_i(s, s')$ = for agent i, states s and s' cannot be distinguished by prior observation – i.e., according to i's knowledge!
  - Very often $\mathcal{K}_i$ are reflexive, symmetric & transitive – i.e. equivalence relations.

Semantics (contd.)

- Semantics of formulas: evaluated at each state s:
  - $(M, s) \models \phi$: formula ϕ holds at state s.
- $(M, s) \models p$ iff $p_2 \in \pi(s)$.
- $(M, s) \models \phi_1 \wedge \phi_2$ iff
- $(M, s) \models K_i\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$.
  - ϕ is a formula that is acquired by i.
  - All observations bring i to consider that ϕ must hold.
- Notation: $M \models \phi$ iff $(M, s) \models \phi$ for all $s \in S$.

Semantics (contd.)

- Semantics of formulas: evaluated at each state s:
  - $(M, s) \models \phi$: formula ϕ holds at state s.
- $(M, s) \models p$ iff $p_2 \in \pi(s)$.
- $(M, s) \models \phi_1 \wedge \phi_2$ iff
- $(M, s) \models K_i\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$.
  - ϕ is a formula that is acquired by i.
  - All observations bring i to consider that ϕ must hold.
- Notation: $M \models \phi$ iff $(M, s) \models \phi$ for all $s \in S$.

Semantics (contd.)

- Semantics of formulas: evaluated at each state s:
  - $(M, s) \models \phi$: formula ϕ holds at state s.
- $(M, s) \models p$ iff $p_2 \in \pi(s)$.
- $(M, s) \models \phi_1 \wedge \phi_2$ iff
- $(M, s) \models K_i\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$.
  - ϕ is a formula that is acquired by i.
  - All observations bring i to consider that ϕ must hold.
- Notation: $M \models \phi$ iff $(M, s) \models \phi$ for all $s \in S$.

Semantics (contd.)

- Semantics of formulas: evaluated at each state s:
  - $(M, s) \models \phi$: formula ϕ holds at state s.
- $(M, s) \models p$ iff $p_2 \in \pi(s)$.
- $(M, s) \models \phi_1 \wedge \phi_2$ iff $(M, s) \models \phi_1$ and $(M, s) \models \phi_2$.
- $(M, s) \models K_i\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$.
  - ϕ is a formula that is acquired by i.
  - All observations bring i to consider that ϕ must hold.
- Notation: $M \models \phi$ iff $(M, s) \models \phi$ for all $s \in S$.

Semantics (contd.)

- Semantics of formulas: evaluated at each state s:
  - $(M,s) \models \phi$: formula ϕ holds at state s.
- $(M,s) \models p$ iff $p_2 \in \pi(s)$.
- $(M,s) \models \phi_1 \wedge \phi_2$ iff $(M,s) \models \phi_1$ and $(M,s) \models \phi_2$.
- $(M,s) \models K_i\phi$ iff $(M,s') \models \phi$ for all s' with $\mathcal{K}_i(s,s')$.
  - ϕ is a formula that is acquired by i.
  - All observations bring i to consider that ϕ must hold.
- Notation: $M \models \phi$ iff $(M,s) \models \phi$ for all $s \in S$.

Muddy children – original situation

- Kripke structure $M_{mud} = (S, \Pi, \pi, \mathcal{K}_i)$ for n agents.
- "Local state" for agent i: $S_i = \{0, 1\}$ (muddy or not!).
- $S = S_1 \times \ldots S_n$ – that is, 2^n initial situations.
  - A "global state" is composed of "local states": $s = (s_1, \ldots, s_n)$.
- $\Pi = \{p_1, \ldots, p_n\}$.
  - $(M_{mud}, s) \models p_3$ iff $s_3 = 1$.
- $\mathcal{K}_i(s, s')$ iff $s_j = s'_j$ for all $j \neq i$.
  - "Hypercube" representation of M_{mud}.
- What are the states where $(M_{mud}, s) \models K_1 p_2$?

Muddy children – original situation

- Kripke structure $M_{mud} = (S, \Pi, \pi, \mathcal{K}_i)$ for n agents.
- "Local state" for agent i: $S_i = \{0, 1\}$ (muddy or not!).
- $S = S_1 \times \dots S_n$ – that is, 2^n initial situations.
  - A "global state" is composed of "local states": $s = (s_1, \dots, s_n)$.
- $\Pi = \{p_1, \dots, p_n\}$.
  - $(M_{mud}, s) \models p_3$ iff $s_3 = 1$.
- $\mathcal{K}_i(s, s')$ iff $s_j = s'_j$ for all $j \neq i$.
  - "Hypercube" representation of M_{mud}.
- What are the states where $(M_{mud}, s) \models K_1 p_2$?

Muddy children – original situation

- Kripke structure $M_{mud} = (S, \Pi, \pi, \mathcal{K}_i)$ for n agents.
- "Local state" for agent i: $S_i = \{0, 1\}$ (muddy or not!).
- $S = S_1 \times \ldots S_n$ – that is, 2^n initial situations.
  - A "global state" is composed of "local states": $s = (s_1, \ldots, s_n)$.
- $\Pi = \{p_1, \ldots, p_n\}$.
  - $(M_{mud}, s) \models p_3$ iff $s_3 = 1$.
- $\mathcal{K}_i(s, s')$ iff $s_j = s'_j$ for all $j \neq i$.
  - "Hypercube" representation of M_{mud}.
- What are the states where $(M_{mud}, s) \models K_1 p_2$?

Muddy children – original situation

- Kripke structure $M_{mud} = (S, \Pi, \pi, \mathcal{K}_i)$ for n agents.
- "Local state" for agent i: $S_i = \{0, 1\}$ (muddy or not!).
- $S = S_1 \times \ldots S_n$ – that is, 2^n initial situations.
  - A "global state" is composed of "local states": $s = (s_1, \ldots, s_n)$.
- $\Pi = \{p_1, \ldots, p_n\}$.
  - $(M_{mud}, s) \models p_3$ iff $s_3 = 1$.
- $\mathcal{K}_i(s, s')$ iff $s_j = s'_j$ for all $j \neq i$.
  - "Hypercube" representation of M_{mud}.
- What are the states where $(M_{mud}, s) \models K_1 p_2$?

Other knowledge operators

- i considers ϕ possible – $P_i\phi$ –
  - $(M, s) \models P_i\phi$ iff $(M, s') \models \phi$ for some s' with $\mathcal{K}_i(s, s')$.
- Everybody in the group G knows ϕ – $E_G\phi$ –
  - $(M, s) \models E_G\phi$ iff $(M, s) \models K_i\phi$ for all $i \in G$.
- Distributed knowledge of ϕ within a group : $D_G\phi$
  - $(M, s) \models E_G\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$ $\forall i \in G$.
- Common knowledge of ϕ within a group G : $C_G\phi$
  - $(M, s) \models C_G\phi$ iff $(M, s) \models E_G^k\phi$ for all k.
  - That is, each agent knows that each other agent knows that .... knows that ϕ holds.
  - Stronger than E_G and distributed knowledge!
- What about $P_2p_2, E_{1,2}p_2, E_{1,2}p_3, D_{1,2}p_3, C_{1,2}p_3$ in M_{mud}?

Other knowledge operators

- i considers ϕ possible – $P_i\phi$ –
  - $(M, s) \models P_i\phi$ iff $(M, s') \models \phi$ for some s' with $\mathcal{K}_i(s, s')$.
- Everybody in the group G knows ϕ – $E_G\phi$ –
  - $(M, s) \models E_G\phi$ iff $(M, s) \models K_i\phi$ for all $i \in G$.
- Distributed knowledge of ϕ within a group : $D_G\phi$
  - $(M, s) \models E_G\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$ $\forall i \in G$.
- Common knowledge of ϕ within a group G : $C_G\phi$
  - $(M, s) \models C_G\phi$ iff $(M, s) \models E_G^k\phi$ for all k.
  - That is, each agent knows that each other agent knows that .... knows that ϕ holds.
  - Stronger than E_G and distributed knowledge!
- What about $P_2p_2, E_{1,2}p_2, E_{1,2}p_3, D_{1,2}p_3, C_{1,2}p_3$ in M_{mud}?

Other knowledge operators

- i considers ϕ possible – $P_i\phi$ –
  - $(M, s) \models P_i\phi$ iff $(M, s') \models \phi$ for some s' with $\mathcal{K}_i(s, s')$.
- Everybody in the group G knows ϕ – $E_G\phi$ –
  - $(M, s) \models E_G\phi$ iff $(M, s) \models K_i\phi$ for all $i \in G$.
- Distributed knowledge of ϕ within a group : $D_G\phi$
  - $(M, s) \models E_G\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$ $\forall i \in G$.
- Common knowledge of ϕ within a group G : $C_G\phi$
  - $(M, s) \models C_G\phi$ iff $(M, s) \models E_G^k\phi$ for all k.
  - That is, each agent knows that each other agent knows that .... knows that ϕ holds.
  - Stronger than E_G and distributed knowledge!
- What about $P_2p_2, E_{1,2}p_2, E_{1,2}p_3, D_{1,2}p_3, C_{1,2}p_3$ in M_{mud}?

Other knowledge operators

- i considers ϕ possible – $P_i\phi$ –
  - $(M,s) \models P_i\phi$ iff $(M,s') \models \phi$ for some s' with $\mathcal{K}_i(s,s')$.
- Everybody in the group G knows ϕ – $E_G\phi$ –
  - $(M,s) \models E_G\phi$ iff $(M,s) \models K_i\phi$ for all $i \in G$.
- Distributed knowledge of ϕ within a group : $D_G\phi$
  - $(M,s) \models E_G\phi$ iff $(M,s') \models \phi$ for all s' with $\mathcal{K}_i(s,s')$ $\forall i \in G$.
- Common knowledge of ϕ within a group G : $C_G\phi$
  - $(M,s) \models C_G\phi$ iff $(M,s) \models E_G^k\phi$ for all k.
  - That is, each agent knows that each other agent knows that .... knows that ϕ holds.
  - Stronger than E_G and distributed knowledge!
- What about $P_2p_2, E_{1,2}p_2, E_{1,2}p_3, D_{1,2}p_3, C_{1,2}p_3$ in M_{mud}?

Other knowledge operators

- i considers ϕ possible – $P_i\phi$ –
  - $(M, s) \models P_i\phi$ iff $(M, s') \models \phi$ for some s' with $\mathcal{K}_i(s, s')$.
- Everybody in the group G knows ϕ – $E_G\phi$ –
  - $(M, s) \models E_G\phi$ iff $(M, s) \models K_i\phi$ for all $i \in G$.
- Distributed knowledge of ϕ within a group : $D_G\phi$
  - $(M, s) \models E_G\phi$ iff $(M, s') \models \phi$ for all s' with $\mathcal{K}_i(s, s')$ $\forall i \in G$.
- Common knowledge of ϕ within a group G : $C_G\phi$
  - $(M, s) \models C_G\phi$ iff $(M, s) \models E_G^k\phi$ for all k.
  - That is, each agent knows that each other agent knows that .... knows that ϕ holds.
  - Stronger than E_G and distributed knowledge!
- What about $P_2p_2, E_{1,2}p_2, E_{1,2}p_3, D_{1,2}p_3, C_{1,2}p_3$ in M_{mud}?

Evolving knowledge

- Consider again the muddy children Kripke structure M_{mud}.
- What happens when Mum speaks the first time?
- Answer: state $(0, 0, ..., 0)$ disappears!
  - After Mum's announcement, it is common knowledge that someone has mud on his forehead!
- What happens when Mum speaks the second time?
- All states with only one 1 dissapear!
  - After Mum's announcement, it is common knowledge that at least two children are dirty!
- And so on...
- But this is not exactly captured by our system model!

Evolving knowledge

- Consider again the muddy children Kripke structure M_{mud}.
- What happens when Mum speaks the first time?
- Answer: state $(0, 0, ..., 0)$ disappears!
  - After Mum's announcement, it is common knowledge that someone has mud on his forehead!
- What happens when Mum speaks the second time?
- All states with only one 1 dissapear!
  - After Mum's announcement, it is common knowledge that at least two children are dirty!
- And so on...
- But this is not exactly captured by our system model!

Evolving knowledge

- Consider again the muddy children Kripke structure M_{mud}.
- What happens when Mum speaks the first time?
- Answer: state $(0, 0, ..., 0)$ disappears!
  - After Mum's announcement, it is common knowledge that someone has mud on his forehead!
- What happens when Mum speaks the second time?
- All states with only one 1 dissapear!
  - After Mum's announcement, it is common knowledge that at least two children are dirty!
- And so on...
- But this is not exactly captured by our system model!

Evolving knowledge

- Consider again the muddy children Kripke structure M_{mud}.
- What happens when Mum speaks the first time?
- Answer: state $(0, 0, ..., 0)$ disappears!
  - After Mum's announcement, it is common knowledge that someone has mud on his forehead!
- What happens when Mum speaks the second time?
- All states with only one 1 dissapear!
  - After Mum's announcement, it is common knowledge that at least two children are dirty!
- And so on...
- But this is not exactly captured by our system model!

Evolving knowledge

- Consider again the muddy children Kripke structure M_{mud}.
- What happens when Mum speaks the first time?
- Answer: state $(0, 0, ..., 0)$ disappears!
  - After Mum's announcement, it is common knowledge that someone has mud on his forehead!
- What happens when Mum speaks the second time?
- All states with only one 1 dissapear!
  - After Mum's announcement, it is common knowledge that at least two children are dirty!
- And so on...
- But this is not exactly captured by our system model!

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\newmoon\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\newmoon\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Incorporating temporal operators

Future temporal operators:

- $\bigcirc\phi$ – next time, ϕ holds.
- $\Box\phi$ – ϕ holds forever, from now on.
- $\phi\mathcal{U}\psi$ – ϕ holds in every time point until ψ holds.
- $\Diamond\phi$ – there exists a point in the future where ϕ will hold.

And past temporal operators:

- $\bullet\phi$ – last time, ϕ held.
- $\blacksquare\phi$ – always before, ϕ held.
- $\blacklozenge\phi$ – ϕ held sometime in the past.
- $\phi\mathcal{S}\psi$ – ϕ held in every time point since ψ held.

Other operators can be added (e.g. fixpoints).

Temporal semantics

- Transition system for n agents $\mathcal{T} = (S, \succ)$:
  - $\succ \subseteq S \times S$ – temporal evolution of the system.
  - Runs in $\mathcal{T}$ = infinite sequences of states in S.
- Temporal interpreted system over $\mathcal{T}$: $\mathcal{I} = (Q, \Pi, \pi)$:
  - $Q = \text{Runs}(\mathcal{T}) \times \mathbb{N}$ – points.
  - $\pi : Q \to 2^{\Pi}$ – interpretation of propositional symbols.
- Semantics of temporal formulas: $(\mathcal{I}, r, n) \models \phi$.
  - $(r, n) \in Q$.

Temporal semantics

- Transition system for n agents $\mathcal{T} = (S, \succ)$:
  - $\succ \subseteq S \times S$ – temporal evolution of the system.
  - Runs in $\mathcal{T}$ = infinite sequences of states in S.
- Temporal interpreted system over $\mathcal{T}$: $\mathcal{I} = (Q, \Pi, \pi)$:
  - $Q = \text{Runs}(\mathcal{T}) \times \mathbb{N}$ – points.
  - $\pi : Q \to 2^{\Pi}$ – interpretation of propositional symbols.
- Semantics of temporal formulas: $(\mathcal{I}, r, n) \models \phi$.
  - $(r, n) \in Q$.

Temporal semantics

- Transition system for n agents $\mathcal{T} = (S, \succ)$:
  - $\succ \subseteq S \times S$ – temporal evolution of the system.
  - Runs in $\mathcal{T}$ = infinite sequences of states in S.
- Temporal interpreted system over $\mathcal{T}$: $\mathcal{I} = (Q, \Pi, \pi)$:
  - $Q = \text{Runs}(\mathcal{T}) \times \mathbb{N}$ – points.
  - $\pi : Q \to 2^{\Pi}$ – interpretation of propositional symbols.
- Semantics of temporal formulas: $(\mathcal{I}, r, n) \models \phi$.
  - $(r, n) \in Q$.

Temporal semantics (contd.)

- $(\mathcal{I}, r, n) \models \bigcirc\phi$ iff $(\mathcal{I}, r, n+1) \models \phi$.
- $(\mathcal{I}, r, n) \models \Box\phi$ iff $(\mathcal{I}, r, m) \models \phi$ for all $m \geq n$.
- $(\mathcal{I}, r, n) \models \Diamond\phi$ iff $(\mathcal{I}, r, m) \models \phi$ for some $m \geq n$.
- $(\mathcal{I}, r, n) \models \phi\mathcal{U}\psi$ iff $(\mathcal{I}, r, m) \models \psi$ for some $m \geq n$ and $(\mathcal{I}, r, p) \models \phi$ for all $n \leq p < m$.
- $(\mathcal{I}, r, n) \models \bullet\phi$ iff $(\mathcal{I}, r, n-1) \models \phi$ ($n > 0$!).
- $(\mathcal{I}, r, n) \models \blacksquare\phi$ iff $(\mathcal{I}, r, m) \models \phi$ for all $m \leq n$.
- $(\mathcal{I}, r, n) \models \blacklozenge\phi$ iff $(\mathcal{I}, r, n+1) \models \phi$ for some $m \leq n$.
- $(\mathcal{I}, r, n) \models \phi\mathcal{U}\psi$ iff $(\mathcal{I}, r, m) \models \psi$ for some $m \leq n$ and $(\mathcal{I}, r, p) \models \phi$ for all $m < p \leq m$.

Temporal semantics (contd.)

- $(\mathcal{I}, r, n) \models \bigcirc\phi$ iff $(\mathcal{I}, r, n+1) \models \phi$.
- $(\mathcal{I}, r, n) \models \Box\phi$ iff $(\mathcal{I}, r, m) \models \phi$ for all $m \geq n$.
- $(\mathcal{I}, r, n) \models \Diamond\phi$ iff $(\mathcal{I}, r, m) \models \phi$ for some $m \geq n$.
- $(\mathcal{I}, r, n) \models \phi\mathcal{U}\psi$ iff $(\mathcal{I}, r, m) \models \psi$ for some $m \geq n$ and $(\mathcal{I}, r, p) \models \phi$ for all $n \leq p < m$.
- $(\mathcal{I}, r, n) \models \bullet\phi$ iff $(\mathcal{I}, r, n-1) \models \phi$ ($n > 0$!).
- $(\mathcal{I}, r, n) \models \blacksquare\phi$ iff $(\mathcal{I}, r, m) \models \phi$ for all $m \leq n$.
- $(\mathcal{I}, r, n) \models \blacklozenge\phi$ iff $(\mathcal{I}, r, n+1) \models \phi$ for some $m \leq n$.
- $(\mathcal{I}, r, n) \models \phi\mathcal{U}\psi$ iff $(\mathcal{I}, r, m) \models \psi$ for some $m \leq n$ and $(\mathcal{I}, r, p) \models \phi$ for all $m < p \leq m$.

Temporal and knowledge semantics

- Temporal interpreted system $\mathcal{I} = (Q, \Pi, \pi)$ over a transition system $\mathcal{T}$.
- Kripke structure over $\mathcal{T}$: $M_{\mathcal{T}} = (\mathcal{I}, \mathcal{K}_1, \ldots \mathcal{K}_n)$.
  - $\mathcal{K}_i \subseteq Q \times Q$.
- Semantics : unchanged from what we've seen!
- Example formulas: $K_1 \Box p_1 \wedge p_2 \mathcal{U} C_{2,3} p_3$.

Temporal and knowledge semantics

- Temporal interpreted system $\mathcal{I} = (Q, \Pi, \pi)$ over a transition system $\mathcal{T}$.
- Kripke structure over $\mathcal{T}$: $M_{\mathcal{T}} = (\mathcal{I}, \mathcal{K}_1, \ldots \mathcal{K}_n)$.
  - $\mathcal{K}_i \subseteq Q \times Q$.
- Semantics : unchanged from what we've seen!
- Example formulas: $K_1 \Box p_1 \wedge p_2 \mathcal{U} C_{2,3} p_3$.

Muddy children example

- Transition system: $\mathcal{T} = (S, \succ)$ with $\succ = \{(s, s) \mid s \in S\}$.
  - Local states are unchanged during the run!
- Run – identified with the (unique) state occurring in it!
  - Hence points = pairs (state, timepoint).
- Interpretation: $\pi(s, n) = \{p_i \mid s_i = 1\}$.
- Possibility relations:

$$\mathcal{K}_i((s,k),(s'.k)) \text{ iff } s = s' \text{ or } \mathrm{supp}(s), \mathrm{supp}(s') \geq k$$
$$\text{and } s_j = s'_j \ \forall j \neq i$$

  - $\mathrm{supp}(s) = \{i \mid s_i = 1\}$.
- Draw it!

Muddy children example

- Transition system: $\mathcal{T} = (S, \succ)$ with $\succ = \{(s, s) \mid s \in S\}$.
  - Local states are unchanged during the run!
- Run – identified with the (unique) state occurring in it!
  - Hence points = pairs (state, timepoint).
- Interpretation: $\pi(s, n) = \{p_i \mid s_i = 1\}$.
- Possibility relations:

$$\mathcal{K}_i\big((s,k),(s'.k)\big) \text{ iff } s = s' \text{ or } \mathrm{supp}(s), \mathrm{supp}(s') \geq k$$
$$\text{and } s_j = s'_j \ \forall j \neq i$$

  - $\mathrm{supp}(s) = \{i \mid s_i = 1\}$.
- Draw it!

Muddy children example

- Transition system: $\mathcal{T} = (S, \succ)$ with $\succ = \{(s, s) \mid s \in S\}$.
  - Local states are unchanged during the run!
- Run – identified with the (unique) state occurring in it!
  - Hence points = pairs (state, timepoint).
- Interpretation: $\pi(s, n) = \{p_i \mid s_i = 1\}$.
- Possibility relations:

$$\mathcal{K}_i\big((s,k),(s'.k)\big) \text{ iff } s = s' \text{ or } \mathrm{supp}(s), \mathrm{supp}(s') \geq k \\ \text{and } s_j = s'_j \; \forall j \neq i$$

  - $\mathrm{supp}(s) = \{i \mid s_i = 1\}$.
- Draw it!

Temporal knowledge properties of the muddy children

- $(s, 1) \models C(p_1 \vee \ldots \vee p_n)$ iff .
- In general, $(s, k) \models C$
- If $(s, k) \models P_i p_i$ then $(s, k+1) \models C(P_i p_i \wedge P_i \neg p_i)$.
- If $\mathrm{supp}(s) = k$ then for each i with $s_i = 1$ we have $(s, k) \models K_i p_i$.

Temporal knowledge properties of the muddy children

- $(s, 1) \models C(p_1 \vee \ldots \vee p_n)$ iff $s \neq (0, \ldots, 0)$.
- In general, $(s, k) \models C$
- If $(s, k) \models P_i p_i$ then $(s, k+1) \models C(P_i p_i \wedge P_i \neg p_i)$.
- If $\mathrm{supp}(s) = k$ then for each i with $s_i = 1$ we have $(s, k) \models K_i p_i$.

Temporal knowledge properties of the muddy children

- $(s, 1) \models C(p_1 \vee \ldots \vee p_n)$ iff $s \neq (0, \ldots, 0)$.
- In general, $(s, k) \models C \bigvee_{|S| \geq k} \bigwedge_{j \in S} p_i$
- If $(s, k) \models P_i p_i$ then $(s, k+1) \models C(P_i p_i \wedge P_i \neg p_i)$.
- If $\text{supp}(s) = k$ then for each i with $s_i = 1$ we have $(s, k) \models K_i p_i$.

Temporal knowledge properties of the muddy children

- $(s,1) \models C(p_1 \vee \ldots \vee p_n)$ iff $s \neq (0,\ldots,0)$.
- In general, $(s,k) \models C \bigvee_{|S|\geq k} \bigwedge_{j\in S} p_i$
- If $(s,k) \models P_i p_i$ then $(s,k+1) \models C(P_i p_i \wedge P_i \neg p_i)$.
- If $\mathrm{supp}(s) = k$ then for each i with $s_i = 1$ we have $(s,k) \models K_i p_i$.

Synchronicity

- Agents have access to a shared clock.
  - For the muddy children, it is Mum's announcements that play the role of a clock.
  - The system is synchronous.
- Synchronous Kripke structure over a transition system $\mathcal{T}$: $M = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$:
  - If $\mathcal{K}_i((r,n),(r',n'))$ then $n = n'$.
  - The points that i considers possible at (r,n) are those whose clock is n too.

Synchronicity

- Agents have access to a shared clock.
    - For the muddy children, it is Mum's announcements that play the role of a clock.
    - The system is synchronous.
- Synchronous Kripke structure over a transition system $\mathcal{T}$: $M = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$:
    - If $\mathcal{K}_i\big((r,n),(r',n')\big)$ then $n = n'$.
    - The points that i considers possible at (r,n) are those whose clock is n too.

Perfect recall

- With the general definition of $\mathcal{K}_i$, agent i's knowledge may vary during system evolution.
- We would like it to be only cumulative
  - What i learned at a point (r, n) has to be “preserved” at later points (r, n') $(n' \geq n)$.
- Kripke structure with perfect recall: $M = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$:
  - Local state sequence at (r, n): sequence of s_i, without repetitions.
  - E.g. if i's local states at instants $0 \ldots 4$ are $(s_i, s_i, s'_i, s'_i, s_i)$, then $\mathrm{lss}(r, 4) = (s_i, s'_i, s_i)$.
  - Perfect recall: equivalent points only if local state sequence is the same:

$$\text{If } \mathcal{K}_i((r, n), (r', n')) \text{ then } \mathrm{lss}(r, n) = \mathrm{lss}(r', n')$$

Perfect recall

- With the general definition of $\mathcal{K}_i$, agent i's knowledge may vary during system evolution.
- We would like it to be only cumulative
  - What i learned at a point (r, n) has to be "preserved" at later points (r, n') $(n' \geq n)$.
- Kripke structure with perfect recall: $M = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$:
  - Local state sequence at (r, n): sequence of s_i, without repetitions.
  - E.g. if i's local states at instants $0 \ldots 4$ are $(s_i, s_i, s_i', s_i', s_i)$, then $\mathrm{lss}(r, 4) = (s_i, s_i', s_i)$.
  - Perfect recall: equivalent points only if local state sequence is the same:

$$\text{If } \mathcal{K}_i((r, n), (r', n')) \text{ then } \mathrm{lss}(r, n) = \mathrm{lss}(r', n')$$

Perfect recall

- With the general definition of $\mathcal{K}_i$, agent i's knowledge may vary during system evolution.
- We would like it to be only cumulative
  - What i learned at a point (r, n) has to be "preserved" at later points (r, n') $(n' \geq n)$.
- Kripke structure with perfect recall: $M = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$:
  - Local state sequence at (r, n): sequence of s_i, without repetitions.
  - E.g. if i's local states at instants $0 \ldots 4$ are $(s_i, s_i, s_i', s_i', s_i)$, then $\mathrm{lss}(r, 4) = (s_i, s_i', s_i)$.
  - Perfect recall: equivalent points only if local state sequence is the same:

$$\text{If } \mathcal{K}_i\big((r, n), (r', n')\big) \text{ then } \mathrm{lss}(r, n) = \mathrm{lss}(r', n')$$

Synchrony & perfect recall

- Perfect recall does not mean $K_i\phi \rightarrow \Box K_i\phi$!
- Example: muddy children with $\phi = P_i p_i \wedge P_i \neg p_i$.
- Dual notion: no learning:
  - Speaks about future local state sequence.

Synchrony & perfect recall

- Perfect recall does not mean $K_i\phi \rightarrow \Box K_i\phi$!
- Example: muddy children with $\phi = P_i p_i \wedge P_i \neg p_i$.
- Dual notion: no learning:
  - Speaks about future local state sequence.

Axioms for knowledge without time

Pr Axioms and rules for the propositional operators.

K. Distribution axiom: $(K_i\phi \wedge K_i(\phi \to \psi) \to K_i\psi$

T. Knowledge axiom: $K_i\phi \to \phi$

4. Positive introspection axiom: $K_i\phi \to K_iK_i\phi$

5. Negative introspection axiom: $\neg K_i \to K_i\neg K_i\phi$

Axioms for knowledge without time

Pr Axioms and rules for the propositional operators.

K. Distribution axiom: $(K_i\phi \wedge K_i(\phi \rightarrow \psi) \rightarrow K_i\psi$

T. Knowledge axiom: $K_i\phi \rightarrow \phi$

4. Positive introspection axiom: $K_i\phi \rightarrow K_iK_i\phi$

5. Negative introspection axiom: $\neg K_i \rightarrow K_i\neg K_i\phi$

Correctness and completeness

- Knowledge generalization rule:

$$\text{If } M \models \phi \text{ then } M \models K_i\phi$$

- The whole = system $S5_n$.

Theorem

For any structure M in which each possibility relation $\mathcal{K}_i$ is an equivalence, and all agents i, the above axioms and rule hold.

Theorem

$S5_n$ is a sound and complete axiomatization of the logic of knowledge in which $\mathcal{K}_i$ are all equivalence relations.

Correctness and completeness

- Knowledge generalization rule:

$$\text{If } M \models \phi \text{ then } M \models K_i\phi$$

- The whole = system $S5_n$.

Theorem

For any structure M in which each possibility relation $\mathcal{K}_i$ is an equivalence, and all agents i, the above axioms and rule hold.

Theorem

$S5_n$ is a sound and complete axiomatization of the logic of knowledge in which $\mathcal{K}_i$ are all equivalence relations.

Correctness and completeness

- Knowledge generalization rule:

$$\text{If } M \models \phi \text{ then } M \models K_i\phi$$

- The whole = system $S5_n$.

Theorem

For any structure M in which each possibility relation $\mathcal{K}_i$ is an equivalence, and all agents i, the above axioms and rule hold.

Theorem

$S5_n$ is a sound and complete axiomatization of the logic of knowledge in which $\mathcal{K}_i$ are all equivalence relations.

Common knowledge and distributed knowledge

1. Defining axiom for “everibody knows”: $E_G\phi \to \bigwedge_{i\in G} K_i\phi$
2. Fixpoint axiom for common knowledge:
   $C_G\phi \leftrightarrow E_G(\phi \wedge C_G\phi)$
3. Induction rule for common knowledge:
   If $M \models E_G(\phi \wedge C_G\phi)$ then $M \models C_G\phi$
4. Subgroup axioms: $E_G\phi \to E_H\phi$ for all $H \subseteq G$.
5. Similarly for C_G and D_G.
6. System $S5_n^C$ – correct and complete.

Axiomatizing time

- $\Box$ and $\Diamond$ can be expressed in terms of $\mathcal{U}$
  - How?
- Axioms for $\bigcirc$ and $\mathcal{U}$:
  - Distributivity: $\bigcirc\phi \wedge \bigcirc(\phi \rightarrow \psi) \rightarrow \bigcirc\psi$.
  - Linear time: $\neg \bigcirc \phi \leftrightarrow \bigcirc\neg\phi$.
  - Fixpoint axiom for until: $\phi\mathcal{U}\psi \leftrightarrow \psi \vee (\phi \wedge \bigcirc(\phi\mathcal{U}\psi))$.
  - Next time rule: from ϕ infer $\Box\phi$.
  - Until inference rule: from $\phi' \rightarrow \neg\psi \wedge \bigcirc\phi'$ infer $\phi' \rightarrow \neg(\phi\mathcal{U}\psi)$.

Axiomatizing time

- $\square$ and $\diamond$ can be expressed in terms of $\mathcal{U}$
  - How?
- Axioms for $\bigcirc$ and $\mathcal{U}$:
  - Distributivity: $\bigcirc\phi \wedge \bigcirc(\phi \to \psi) \to \bigcirc\psi$.
  - Linear time: $\neg\bigcirc\phi \leftrightarrow \bigcirc\neg\phi$.
  - Fixpoint axiom for until: $\phi\mathcal{U}\psi \leftrightarrow \psi \vee (\phi \wedge \bigcirc(\phi\mathcal{U}\psi))$.
  - Next time rule: from ϕ infer $\square\phi$.
  - Until inference rule: from $\phi' \to \neg\psi \wedge \bigcirc\phi'$ infer $\phi' \to \neg(\phi\mathcal{U}\psi)$.

Combining time and knowledge axiomatically

- General systems: no additional axioms!
  - Knowledge and time are independent in general!
- Perfect recall: they do interact

$$(KT1) \qquad K_i \Box \phi \rightarrow \Box K_i \phi$$

  - Formulas known to be always true must always be known to be true (!)
- Synchrony & perfect recall: stronger interaction

$$(KT2) \qquad K_i \bigcirc \phi \rightarrow \bigcirc K_i \phi$$

Theorem

$S5_n^U + KT2$ is a sound and complete axiomatization for synchrony and perfect recall.

Combining time and knowledge axiomatically

- General systems: no additional axioms!
  - Knowledge and time are independent in general!
- Perfect recall: they do interact

$$(KT1) \qquad K_i \Box \phi \to \Box K_i \phi$$

  - Formulas known to be always true must always be known to be true (!)
- Synchrony & perfect recall: stronger interaction

$$(KT2) \qquad K_i \bigcirc \phi \to \bigcirc K_i \phi$$

Theorem

$S5_n^U + KT2$ is a sound and complete axiomatization for synchrony and perfect recall.

Satisfiability – pure knowledge case

Theorem

The satisfiability problem for $S5_n$ is PSPACE-complete – and thus, the validity problem for $S5_n$ is co-PSPACE-complete.
The satisfiability problem for $S5_n^C$ is EXPTIME-complete – and thus the validity problem for $S5_n^C$ is co-EXPTIME-complete.

Based on theorems on the existence of *finite models*.

Model checking

- Basic case – no common knowledge, no time:

Theorem

There is an algorithm that, given a Kripke structure M, a state s and a formula ϕ, determines in time $O(|M| \times |\phi|)$, whether $(M, s) \models \phi$.

- Common knowledge, no until:

Theorem

The model checking problem for synchronous perfect recall systems and the temporal logic with common knowledge but without until is PSPACE-complete.

Model checking

- Until, no common knowledge:

Theorem

The model checking problem for synchronous perfect recall systems and the temporal logic of knowledge with until but without common knowledge is decidable in nonelementary time.

- Full (future) temporal logic and knowledge operators:

Theorem

The model checking problem for synchronous perfect recall systems and the temporal logic of knowledge with until and common knowledge is undecidable.

Noninterference and its "derivatives"

- Noninterference (Goguen & Meseguer, 1982):

  One group of users [...] is noninterfering with another group of users if what[ever] the first group of users does [...] has no effect on what the second group of users can see.

- Variants
  - Separability (McLean, 1994),
  - Generalized noninterference (McCullough, 1987),
  - Nondeducibility on strategies (Wittbold & Johnson, 1990),
  - Forward correctability, the Perfect Security Property, etc., etc.

Noninterference and its "derivatives"

- Noninterference (Goguen & Meseguer, 1982):

  One group of users [...] is noninterfering with another group of users if what[ever] the first group of users does [...] has no effect on what the second group of users can see.

- Variants
  - Separability (McLean, 1994),
  - Generalized noninterference (McCullough, 1987),
  - Nondeducibility on strategies (Wittbold & Johnson, 1990),
  - Forward correctability, the Perfect Security Property, etc., etc.

Synchronous trace model

- HI – high-level inputs, HO – high-level outputs, $H = HI \cup HO$, $HI \cap HO = \emptyset$.
- LI – low-level inputs, LO – low-level outputs, $L = LI \cup LO$, $LI \cap LO = \emptyset$.
- System states $Q = LI \times HI \times LO \times HO$.
- Traces = infinite sequences of states in Q – denoted $\mathrm{Tr}(Q)$.
- HI-projection of a trace ρ = $\rho\big|_{HI}$ = sequence of HI-actions in ρ.
  - $\rho\big|_{HO}, \rho\big|_{LI}, \rho\big|_{LO}, \rho\big|_{H}, \rho\big|_{L}$ defined similarly.
- k-length prefix of a trace $\rho[1..k]$ = sequence of k initial states.

Synchronous trace model

- HI – high-level inputs, HO – high-level outputs, $H = HI \cup HO$, $HI \cap HO = \emptyset$.
- LI – low-level inputs, LO – low-level outputs, $L = LI \cup LO$, $LI \cap LO = \emptyset$.
- System states $Q = LI \times HI \times LO \times HO$.
- Traces = infinite sequences of states in Q – denoted $\mathrm{Tr}(Q)$.
- HI-projection of a trace ρ = $\rho\big|_{HI}$ = sequence of HI-actions in ρ.
  - $\rho\big|_{HO}, \rho\big|_{LI}, \rho\big|_{LO}, \rho\big|_{H}, \rho\big|_{L}$ defined similarly.
- k-length prefix of a trace $\rho[1..k]$ = sequence of k initial states.

Synchronous trace model

- HI – high-level inputs, HO – high-level outputs, $H = HI \cup HO$, $HI \cap HO = \emptyset$.
- LI – low-level inputs, LO – low-level outputs, $L = LI \cup LO$, $LI \cap LO = \emptyset$.
- System states $Q = LI \times HI \times LO \times HO$.
- Traces = infinite sequences of states in Q – denoted $\mathrm{Tr}(Q)$.
- HI-projection of a trace ρ = $\rho|_{HI}$ = sequence of HI-actions in ρ.
  - $\rho|_{HO}, \rho|_{LI}, \rho|_{LO}, \rho|_{H}, \rho|_{L}$ defined similarly.
- k-length prefix of a trace $\rho[1..k]$ = sequence of k initial states.

Synchronous trace model

- HI – high-level inputs, HO – high-level outputs, $H = HI \cup HO$, $HI \cap HO = \emptyset$.
- LI – low-level inputs, LO – low-level outputs, $L = LI \cup LO$, $LI \cap LO = \emptyset$.
- System states $Q = LI \times HI \times LO \times HO$.
- Traces = infinite sequences of states in Q – denoted $\mathrm{Tr}(Q)$.
- HI-projection of a trace ρ = $\rho\big|_{HI}$ = sequence of HI-actions in ρ.
  - $\rho\big|_{HO}, \rho\big|_{LI}, \rho\big|_{LO}, \rho\big|_{H}, \rho\big|_{L}$ defined similarly.
- k-length prefix of a trace $\rho[1..k]$ = sequence of k initial states.

Synchronous trace model

- $\mathrm{Tr}(Q)$ = a transition system, with traces $\simeq$ runs.
- We may further define $\mathcal{K}_H((\rho, m), (\rho', m))$ iff $\rho\big|_H = \rho'\big|_H$.
- Similarly for $\mathcal{K}_L$.
  - Synchronous with perfect recall!

Synchronous trace model

- Tr(Q) = a transition system, with traces $\simeq$ runs.
- We may further define $\mathcal{K}_H((\rho, m), (\rho', m))$ iff $\rho|_H = \rho'|_H$.
- Similarly for $\mathcal{K}_L$.
  - Synchronous with perfect recall!

Information flow properties in trace systems

- Separability:

$$\forall \rho, \rho' \in \mathrm{Tr}(Q)\ \exists \rho'' \in \mathrm{Tr}(Q), \rho''\big|_H = \rho\big|_H, \rho''\big|_L = \rho'\big|_L$$

- Generalized Noninterference:

$$\forall \rho, \rho' \in \mathrm{Tr}(Q)\ \exists \rho'' \in \mathrm{Tr}(Q), \rho''\big|_{HI} = \rho\big|_{HI}, \rho''\big|_L = \rho'\big|_L$$

Information flow in the TLK framework

Kripke structure over an interpreted system $M_{\mathcal{I}} = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- Agent i maintains total secrecy w.r.t. agent j in $M_{\mathcal{I}}$ if

$$\forall (r,n),(r',n') \in Q, \mathcal{K}_i(r,n) \cap \mathcal{K}_j(r',n') \neq \emptyset$$

- Here $\mathcal{K}_i(r,n) = \big\{(r'',n'') \mid \mathcal{K}_i((r,n),(r'',n''))\big\}$.
- Synchronous total secrecy: synchronous system & total secrecy.

Theorem

Suppose that the (Kripke structure corresponding to the) trace system $\mathrm{Tr}(Q)$ is limit closed. Then $\mathrm{Tr}(Q)$ satisfies separability iff H maintains total secrecy w.r.t. L.

Information flow in the TLK framework

Kripke structure over an interpreted system $M_{\mathcal{I}} = (\mathcal{I}, \mathcal{K}_1, \ldots, \mathcal{K}_n)$.

- Agent i maintains total secrecy w.r.t. agent j in $M_{\mathcal{I}}$ if

$$\forall (r,n),(r',n') \in Q, \mathcal{K}_i(r,n) \cap \mathcal{K}_j(r',n') \neq \emptyset$$

- Here $\mathcal{K}_i(r,n) = \{(r'',n'') \mid \mathcal{K}_i((r,n),(r'',n''))\}$.
- Synchronous total secrecy: synchronous system & total secrecy.

Theorem

Suppose that the (Kripke structure corresponding to the) trace system $\mathrm{Tr}(Q)$ *is limit closed. Then* $\mathrm{Tr}(Q)$ *satisfies separability iff H maintains total secrecy w.r.t. L.*

Generalized noninterference in TLK

- j-information function = $f : Q \to X$ (X any set!) such that
$$f(r,m) = f(r',m') \Rightarrow \mathcal{K}_j((r,m),(r',m'))$$
  - Synchronous form: $f(r,m) = f(r',m) \Rightarrow \mathcal{K}_j((r,m),(r',m))$.
  - Example, in trace systems: $f : Q \to HI^*$, $f(\rho, m) = \rho[1..m]\big|_{HI}$.
- Given f a j-information function, H maintains total f-secrecy if
$$\forall (r,m) \in Q, \forall v \in X, \mathcal{K}_i(r,m) \cap f^{-1}(v) \neq \emptyset$$

Theorem

Suppose that the (Kripke structure corresponding to the) trace system Tr(Q) *is limit closed. Then* Tr(Q) *satisfies generalized noninterference iff H maintains total f-secrecy w.r.t. L.*

Generalized noninterference in TLK

- *j*-information function = $f : Q \to X$ (X any set!) such that
$$f(r,m) = f(r',m') \Rightarrow \mathcal{K}_j((r,m),(r',m'))$$
  - Synchronous form: $f(r,m) = f(r',m) \Rightarrow \mathcal{K}_j((r,m),(r',m))$.
  - Example, in trace systems: $f : Q \to HI^*$, $f(\rho,m) = \rho[1..m]\big|_{HI}$.
- Given f a j-information function, H maintains total f-secrecy if
$$\forall (r,m) \in Q, \forall v \in X, \mathcal{K}_i(r,m) \cap f^{-1}(v) \neq \emptyset$$

Theorem

Suppose that the (Kripke structure corresponding to the) trace system $\mathrm{Tr}(Q)$ *is limit closed. Then* $\mathrm{Tr}(Q)$ *satisfies generalized noninterference iff H maintains total f-secrecy w.r.t. L.*

Generalized noninterference in TLK

- j-information function = $f : Q \to X$ (X any set!) such that

$$f(r,m) = f(r',m') \Rightarrow \mathcal{K}_j((r,m),(r',m'))$$

  - Synchronous form: $f(r,m) = f(r',m) \Rightarrow \mathcal{K}_j((r,m),(r',m))$.
  - Example, in trace systems: $f : Q \to HI^*$, $f(\rho,m) = \rho[1..m]\big|_{HI}$.
- Given f a j-information function, H maintains total f-secrecy if

$$\forall (r,m) \in Q, \forall v \in X, \mathcal{K}_i(r,m) \cap f^{-1}(v) \neq \emptyset$$

Theorem

Suppose that the (Kripke structure corresponding to the) trace system $\mathrm{Tr}(Q)$ *is limit closed. Then* $\mathrm{Tr}(Q)$ *satisfies generalized noninterference iff H maintains total f-secrecy w.r.t. L.*

Information flow in syntactic form

- Formula ϕ is *i*-local in system M if

$$\forall (r,m),(r',m') \text{ with } \mathcal{K}_i((r,m),(r',m')),\\ (\mathcal{I},r,m) \models \phi \text{ iff } (\mathcal{I},r',m') \models \phi$$

- Syntactic characterization: $\mathcal{I} \models K_i\phi \vee K_i\neg\phi$.

Theorem

Suppose M is a synchronous system. Then agent i maintains total secrecy w.r.t. agent j in system M iff for every i-local formula ϕ, $\mathcal{I} \models P_j \blacklozenge \Diamond \phi$.

- More constraints on formulas ϕ for GNI and NDS.

Information flow in syntactic form

- Formula ϕ is *i*-local in system M if

$$\forall (r,m),(r',m') \text{ with } \mathcal{K}_i((r,m),(r',m')),$$
$$(\mathcal{I},r,m) \models \phi \text{ iff } (\mathcal{I},r',m') \models \phi$$

- Syntactic characterization: $\mathcal{I} \models K_i\phi \vee K_i\neg\phi$.

Theorem

Suppose M is a synchronous system. Then agent i maintains total secrecy w.r.t. agent j in system M iff for every i-local formula ϕ, $\mathcal{I} \models P_j\blacklozenge\Diamond\phi$.

- More constraints on formulas ϕ for GNI and NDS.

Information flow in syntactic form

- Formula ϕ is *i*-local in system M if

$$\forall (r,m),(r',m') \text{ with } \mathcal{K}_i((r,m),(r',m')),$$
$$(\mathcal{I},r,m) \models \phi \text{ iff } (\mathcal{I},r',m') \models \phi$$

- Syntactic characterization: $\mathcal{I} \models K_i\phi \vee K_i\neg\phi$.

Theorem

Suppose M is a synchronous system. Then agent i maintains total secrecy w.r.t. agent j in system M iff for every i-local formula ϕ, $\mathcal{I} \models P_j \blacklozenge \Diamond \phi$.

- More constraints on formulas ϕ for GNI and NDS.

Sujets de stage de M2 recherche

Model checking des propriétés de sécurité:

- Formalisation des propriétés de fuite d'information dans des langages de programmation et/ou protocoles de sécurité.
- Comparaison d'outils de model checking des logiques épistemiques: MCMAS, MCK, LYS, par rapport leur expressivité en relation avec l'analyse de propriétés de sécurité.
- Synthse d'algorithmes de model checking pour NDS (Wittbold & Johnson).
- Analyse de propriétés de fuite d'information par abstraction.

Deux sujets possibles.