Dynamic Epistemic Logic

2. The Multi-Agent S5 System

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel and Robert Mattmüller

April 24th, 2019

Language

Basic Epistemic Language

When we want to define the basic epistemic language, we need sets of agent symbols and sets of atomic propositions to talk about. Specifically, we have:

- a finite set A of agent symbols (often: $a, b, a', a'', \dots$)
- a countable set P of atomic propositions (often: $p, q, p', p'', \dots$)

Definition (Basic epistemic language)

Let P be a countable set of atomic propositions and A be a finite set of agent symbols. Then the language $\mathcal{L}_K$ is defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi,$$

where $p \in P$ and $a \in A$.

Basic Epistemic Language

We use some common abbreviations and conventions:

- $(\varphi \vee \psi) = \neg(\neg\varphi \wedge \neg\psi)$
- $(\varphi \rightarrow \varphi) = (\neg\varphi \vee \psi)$
- $(\varphi \leftrightarrow \psi) = (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$
- $\top = p \vee \neg p$ for some $p \in P$
- $\bot = \neg\top$

If there is no risk of confusion, outer parentheses can be omitted.

Basic Epistemic Language

Only interesting addition compared to propositional logic: the knowledge modalities K_a.

- $K_a\varphi$ is read as "agent a knows that φ (is true)".
- Its dual, $\neg K_a \neg\varphi$ is read as "agent a considers φ as possible". Abbreviation: $\hat{K}_a\varphi$.
- For a group of agents $B \subseteq A$, we write $E_B\varphi$ to express that everybody in B knows φ. I. e., $E_B\varphi \equiv \bigwedge_{b\in B} K_b\varphi$.
- Its dual is $\hat{E}_B\varphi = \neg E_B\neg\varphi \equiv \bigvee_{b\in B} \hat{K}_b\varphi$, which can be read as "some agent b in B considers φ as possible".
- Sometimes, when writing *iterated operators*, the following convention comes in handy: if X is a modal operator, then X^n is the n-fold application of X. E. g., $K_a^3\varphi$ means $K_aK_aK_a\varphi$.

Simplified Hanabi

Example (Simplified Hanabi)

In simplified Hanabi, we have four cards ($r1$, $r2$, $g1$, $g2$), two players (a, b), and just one card per player. We write p_c for the fact that player p holds card c. Thus, for instance, a_{r1} is read as "player a has card $r1$". Consider the situation where player a has card $r1$ and player b has card $r2$. In this situation, all of the following formulas are true:

- a_{r1} and b_{r2},
- $K_a b_{r2}$ and $K_b a_{r1}$,
- $K_a\neg a_{r2}$ and $K_b\neg b_{r1}$ (Notice that, to arrive at this conclusion, we need to make use of our background theory that contains assertions such as $\neg(a_{r1} \wedge b_{r1})$),
- $K_a(K_b a_{r1} \vee K_b a_{g1} \vee K_b a_{g2})$.

Semantics

Kripke Models

The semantics of the basic epistemic language is based on a special form of Kripke semantics, where we have

- states (or worlds),
- accessibility relations (or indistinguishability relations) between the worlds, and
- propositional valuations associated with the worlds.

Kripke Models

Example (Kripke models)

Consider two cities, namely Groningen and Liverpool. Assume that:

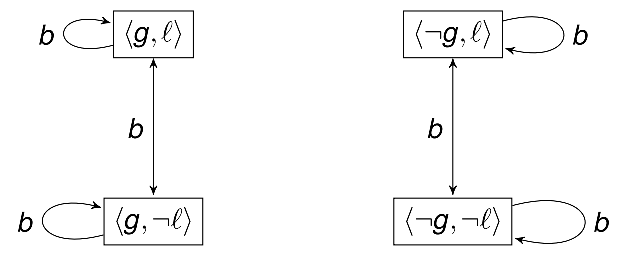
- Person b lives in Groningen.
- Person w lives in Liverpool.
- "The weather in Groningen is sunny" is the atomic proposition g.
- "The weather in Liverpool is sunny" is the atomic proposition ℓ.

States are just possible weather conditions: $\langle g,\ell\rangle$, $\langle\neg g,\ell\rangle$, $\langle g,\neg\ell\rangle$, $\langle\neg g,\neg\ell\rangle$. We want to model what agent b knows. Assume that b is in state $\langle g,\ell\rangle$. He also considers the state $\langle g,\neg\ell\rangle$ possible.

Kripke Models

Example (Kripke models (ctd.))

This situation can be graphically captured by the following model $\mathcal{M}_1$:



Kripke Models

Definition (Kripke model)

Given a countable set of atomic propositions P and a finite set of agent names A, a Kripke model is a structure $\mathcal{M} = (S, R_A, V_P)$ where:

- S is a set of states (also called the domain of $\mathcal{M}$, in symbols $\mathcal{D}(\mathcal{M})$),
- R_A is a function yielding, for every $a \in A$, an accessibility relation $R_A(a) = R_a \subseteq S \times S$.
- $V_P : P \to 2^S$ is a *valuation function* that for all $p \in P$ yields the set of worlds $V_P(p) \subseteq S$ where p is true.

Kripke Models

- If A and P are not important or clear from the context, we will often drop them and write $\mathcal{M} = (S, R, V)$.
- If all accessibility relations R_a are equivalence relations (reflexive, symmetric and transitive), then we also use the symbols $\sim$ for R and $\sim_a$ for R_a.
- In that case, $\mathcal{M} = (S, \sim, V)$ is also called an epistemic model.

Kripke Semantics

Formulas are then interpreted over states in models (aka. states, pointed models, epistemic states).

Example

- Assume we have the formula $K_b\ell$.
- This formula is *not* true in state $\langle\neg g,\ell\rangle$, symbolically $\langle\neg g,\ell\rangle \not\models K_b\ell$.
- Reason: In $\langle\neg g,\ell\rangle$, agent b also considers world $\langle\neg g,\neg\ell\rangle$ possible, and in that world, ℓ does not hold.

Kripke Semantics

We can define truth of an epistemic formula in an epistemic state inductively as follows.

Definition

Given a Kripke model $\mathcal{M} = (S,R,V)$ and $s \in S$, the pair $(\mathcal{M},s)$ is called a pointed model. If $\mathcal{M}$ is an epistemic model, then $(\mathcal{M},s)$ is called an epistemic state.

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Kripke Semantics

Definition

A formula φ is true in an epistemic state $(\mathcal{M},s)$, symbolically $\mathcal{M},s \models \varphi$, under the following conditions:

$$
\begin{array}{lll}
\mathcal{M},s \models p & \text{iff} & s \in V(p) \\
\mathcal{M},s \models \varphi \wedge \psi & \text{iff} & \mathcal{M},s \models \varphi \text{ and } \mathcal{M},s \models \psi \\
\mathcal{M},s \models \neg\varphi & \text{iff} & \mathcal{M},s \not\models \varphi \\
\mathcal{M},s \models K_a\varphi & \text{iff} & \mathcal{M},t \models \varphi \text{ for all } t \in S \text{ with } (s,t) \in R_a
\end{array}
$$

This implies, among others, that $\mathcal{M},s \models \hat{K}_a\varphi$ iff $\mathcal{M},t \models \varphi$ for some $t \in S$ with $(s,t) \in R_a$.

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Kripke Semantics

Definition

If $\mathcal{M},s \models \varphi$ for all $s \in \mathcal{D}(\mathcal{M})$, then we say that φ is true in $\mathcal{M}$, symbolically, $\mathcal{M} \models \varphi$.

Definition

If $\mathcal{M} \models \varphi$ for all models $\mathcal{M}$ in a certain class $\mathcal{X}$ of models, then we say that φ is valid in $\mathcal{X}$, symbolically, $\mathcal{X} \models \varphi$.

Example

If φ is valid in the class $\mathcal{K}$ of all Kripke models, then we write $\mathcal{K} \models \varphi$.

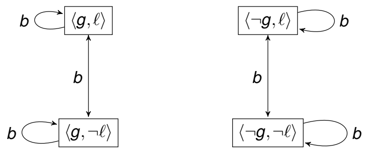
Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Definition

If there exists a pointed model $(\mathcal{M}, s)$ such that φ is true in $(\mathcal{M}, s)$, then we say φ is satisfied in $(\mathcal{M}, s)$. If $\mathcal{M}$ belongs to a class of models $\mathcal{X}$, then φ is satisfiable in $\mathcal{X}$.

Example

Recall model $\mathcal{M}_1$:



- $\mathcal{M}_1, \langle g, \ell\rangle \models K_b g$
- $\mathcal{M}_1, \langle g, \ell\rangle \models \neg K_b \ell$
- $\mathcal{M}_1, \langle g, \ell\rangle \models \neg K_b \neg \ell$
- $\rightsquigarrow \mathcal{M}_1, \langle g, \ell\rangle \models K_b g \land \neg K_b \ell \land \neg K_b \neg \ell$.

Example (Higher-order knowledge)

$\mathcal{M}_1, \langle g, \ell\rangle \models K_b(K_b g \land \neg K_b \ell)$.

To see this, we have to verify that:

- $\mathcal{M}_1, \langle g, \ell\rangle \models K_b g \land \neg K_b \ell$.
- $\mathcal{M}_1, \langle g, \neg\ell\rangle \models K_b g \land \neg K_b \ell$.

In both cases, agent b considers the same states as possible, namely $\langle g, \ell\rangle$ and $\langle g, \neg\ell\rangle$.

- $K_b g$ is true because in all accessible states, g is true.
- $\neg K_b \ell$ is true because there is an accessible state, namely $\langle g, \neg\ell\rangle$, where ℓ is not true.

Example

$\mathcal{M}_1 \models (K_b g \lor K_b \neg g) \land (\neg K_b \ell \land \neg K_b \neg l)$.
Easy to see that both clauses are true and thus the whole formula is true.

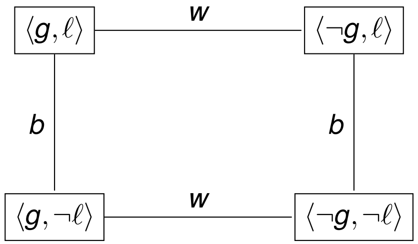
Convention

From now on: Visualizations of epistemic models use undirected edges and leave out reflexive and transitive edges.

Kripke Semantics

Example

Model $\mathcal{M}_2$:



Kripke Semantics

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Example

- $\mathcal{M}_2, \langle g, \ell \rangle \models (K_b g \vee K_b \neg g) \wedge (K_w \ell \vee K_w \neg \ell)$
  (agent b knows whether g, and w knows whether ℓ).
- $\mathcal{M}_2, \langle g, \ell \rangle \models \neg K_w g \wedge \neg K_w \neg g \wedge K_w (K_b g \vee K_b \neg g)$
  (although agent b is ignorant about g, he knows that agent w actually knows whether g holds).

Question: Can we also come up with a model that describes ignorance about what the other knows?

Answer: Yes, but to do that we need to introduce more worlds. Note that there can be distinct states with identical valuations!

Kripke Semantics

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Example

Another agent h (from Otago, NZ) calls w on the phone. w tells h that ℓ is true. Then h tells w that he will call b afterwards, but he does not say whether he will tell b about ℓ. So, w does not know whether b knows that ℓ is true.

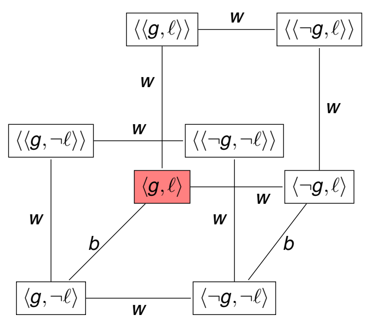
Remark: The construction of the corresponding epistemic model basically means starting with the original model and updating it with a particular action, namely h calling b.

Kripke Semantics

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Example

Model $\mathcal{M}_2$:



$\mathcal{M}_3, \langle g, \ell \rangle \models \ell \wedge \neg K_b \ell \wedge K_b (\neg K_w K_b \ell \wedge \neg K_w \neg K_b \ell)$

Kripke Semantics
Logical Omniscience

Proposition

Let φ and ψ be formulas of $\mathcal{L}_K$ and let K_a be an epistemic operator for some $a \in A$. Let $\mathcal{K}$ be the set of all Kripke models and $\mathcal{S}5$ be the set of all epistemic models. Then the following hold:

- *(LO1)* $\mathcal{K} \models K_a\varphi \wedge K_a(\varphi \rightarrow \psi) \rightarrow K_a\psi$
- *(LO2)* $\mathcal{K} \models \varphi$ *implies* $\mathcal{K} \models K_a\varphi$
- *(LO3)* $\mathcal{K} \models \varphi \rightarrow \psi$ *implies* $\mathcal{K} \models K_a\varphi \rightarrow K_a\psi$
- *(LO4)* $\mathcal{K} \models \varphi \leftrightarrow \psi$ *implies* $\mathcal{K} \models K_a\varphi \leftrightarrow K_a\psi$
- *(LO5)* $\mathcal{K} \models (K_a\varphi \wedge K_a\psi) \rightarrow K_a(\varphi \wedge \psi)$
- *(LO6)* $\mathcal{K} \models K_a\varphi \rightarrow K_a(\varphi \vee \psi)$
- *(LO7)* $\mathcal{S}5 \models \neg(K_a\varphi \wedge K_a\neg\varphi)$

Kripke Semantics
Accessibility Relation Properties

Definition (Relation properties)

A relation R is called

- reflexive if for all s, we have $(s,s) \in R$,
- symmetric if for all s,t, $(s,t) \in R$ implies $(t,s) \in R$,
- transitive if for all s,t,u, $(s,t) \in R$ and $(t,u) \in R$ implies $(s,u) \in R$,
- serial if for all s there is t such that $(s,t) \in R$,
- Euclidean if for all s,t,u, $(s,t) \in R$ and $(s,u) \in R$ implies $(t,u) \in R$, and
- an equivalence relation if it is reflexive, transitive, and symmetric (or: reflexive, transitive, and Euclidean).

Kripke Semantics
Model Classes

Definition

Kripke models are classified according to the properties of the accessibility relation R_a as follows:

Relation property	**Name**
No restriction	$\mathcal{K}$
Serial	$\mathcal{KD}$
Reflexive	$\mathcal{T}$
Transitive	$\mathcal{K}4$
Reflexive and transitive	$\mathcal{S}4$
Transitive and Euclidean	$\mathcal{K}45$
Serial, transitive and Euclidean	$\mathcal{KD}45$
Serial, transitive, Euclidean and reflexive	$\mathcal{S}5$

Bisimulations

Definition (Bisimulation)

Let two models $\mathcal{M} = (S,R,V)$ and $\mathcal{M}' = (S',R',V')$ be given. A non-empty relation $\mathcal{B} \subseteq S \times S'$ is a bisimulation iff for all $s \in S$ and $s' \in S'$ with $(s,s') \in \mathcal{B}$:

- (atoms) $s \in V(p)$ iff $s' \in V'(p)$ for all $p \in P$,
- (forth) for all $a \in A$ and all $t \in S$, if $(s,t) \in R_a$, then there is a $t' \in S'$ such that $(s',t') \in R'_a$ and $(t,t') \in \mathcal{B}$, and
- (back) for all $a \in A$ and all $t' \in S'$, if $(s',t') \in R'_a$, then there is a $t \in S$ such that $(s,t) \in R_a$ and $(t,t') \in \mathcal{B}$.

We write $(\mathcal{M},s) \leftrightarrows (\mathcal{M}',s')$ iff there is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ linking s and s', and we then say that $(\mathcal{M},s)$ and $(\mathcal{M}',s')$ are bisimilar.

The epistemic language $\mathcal{L}_K$ cannot distinguish between bisimilar models.

We write $(\mathcal{M},s) \equiv_{\mathcal{L}_K} (\mathcal{M}',s')$ if and only if $(\mathcal{M},s) \models \varphi$ iff $(\mathcal{M}',s') \models \varphi$ for all formulas $\varphi \in \mathcal{L}_K$.

Theorem (Bisimulation)

For all pointed models $(\mathcal{M},s)$ and $(\mathcal{M}',s')$, if $(\mathcal{M},s) \leftrightarrows (\mathcal{M}',s')$, then $(\mathcal{M},s) \equiv_{\mathcal{L}_K} (\mathcal{M}',s')$.

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Proof.

By structural induction on φ. Suppose that $(\mathcal{M},s) \leftrightarrows (\mathcal{M}',s')$.

- Base case: For atomic formulas $\varphi = p \in P$, by atoms, it must be the case that $\mathcal{M},s \models p$ iff $\mathcal{M}',s' \models p$ for all $p \in P$.
- Inductive cases: Given formula φ, assume that the claim is already proven for all strict subformulas φ' of φ.
  - Negation: Suppose that $\mathcal{M},s \models \neg\varphi'$. By definition, this holds iff $\mathcal{M},s \not\models \varphi'$. By induction hypothesis, this is equivalent to $\mathcal{M}',s' \not\models \varphi'$, which in turn is equivalent to $\mathcal{M}',s' \models \neg\varphi'$.

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Proof (ctd.)

- Inductive cases: ...
  - Conjunction: Suppose that $\mathcal{M},s \models \varphi_1 \land \varphi_2$. By definition, this holds iff $\mathcal{M},s \models \varphi_1$ and $\mathcal{M},s \models \varphi_2$. By two applications of the induction hypothesis, this is equivalent to $\mathcal{M}',s' \models \varphi_1$ and $\mathcal{M}',s' \models \varphi_2$, which in turn is equivalent to $\mathcal{M}',s' \models \varphi_1 \land \varphi_2$.

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Proof (ctd.)

- Inductive cases: ...
  - Individual epistemic operators: Suppose that $\mathcal{M},s \models K_a\varphi'$. Take an arbitrary t' such that $(s',t') \in R'_a$. By back, there is a state $t \in S$ such that $(s,t) \in R_a$ and $(t,t') \in \mathcal{B}$. With $(t,t') \in \mathcal{B}$ and by induction hypothesis, we get $\mathcal{M},t \models \varphi'$ iff $\mathcal{M}',t' \models \varphi'$. Since $\mathcal{M},s \models K_a\varphi'$ and $(s,t) \in R_a$, also $\mathcal{M},t \models \varphi'$ must hold. Therefore, $\mathcal{M}',t' \models \varphi'$. Since t' was chosen arbitrarily from the states indistinguishable from s', it must be the case that $\mathcal{M}',t' \models \varphi'$ for all t' such that $(s',t') \in R'_a$. Therefore, by the semantics of knowledge operators, $\mathcal{M}',s' \models K_a\varphi'$.
  The opposite direction is similar, but the forth condition is used.

□

Bisimulations

Remarks:

- $(\mathcal{M}, s) \leftrightarrows (\mathcal{M}', s')$ implies $(\mathcal{M}, s) \equiv_{\mathcal{L}_K} (\mathcal{M}', s')$, but the converse does not hold.
- The proof applies to all classes of models, not only epistemic models.

Axiomatization

Axiomatization

Logic = set of formulas

Possible ways of characterizing a logic and reasoning in it:

- Semantic derivation of valid formulas via Kripke models
- Syntatic derivation of valid formulas via axioms and inference rules

Axiomatization

Axioms and inference rules of minimal modal logic **K**:

- (*Prop*) all instantiations of propositional tautologies
- (*K*) $K_a(\varphi \rightarrow \psi) \rightarrow (K_a\varphi \rightarrow K_a\psi)$
  (Distribution of K_a over $\rightarrow$)
- (*MP*) From φ and $\varphi \rightarrow \psi$, infer ψ
  (Modus ponens)
- (*Nec*) From φ, infer $K_a\varphi$
  (Necessitation of K_a)

Axiomatization

Definition (Derivation)

Let **X** be an arbitrary axiomatisation with axioms $Ax_1, \ldots, Ax_n$ and rules $Ru_1, \ldots, Ru_k$, where each rule Ru_j, $1 \leq j \leq k$, is of the form "From $\varphi_1, \ldots, \varphi_{j_{ar}}$, infer φ_j". We call j_{ar} the arity of the rule. Then a derivation of a formula φ within **X** is a finite sequence $\varphi_1, \ldots, \varphi_m$ of formulas such that:

1. $\varphi_m = \varphi$ and
2. every φ_i in the sequence is:
   1. either an instance of one of the axioms $Ax_1, \ldots, Ax_n$,
   2. or else the result of the application of one of the rules $Ru_1, \ldots, Ru_k$ to j_{ar} formulas in the sequence that appear before φ_i.

Axiomatization

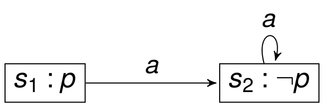
If there is a derivation for φ in **X**, the we write $\vdash_{\mathbf{X}} \varphi$, or, if the system **X** is clear from the context, just $\vdash \varphi$.

We then say that φ is a theorem of **X**.

Axiomatization

Logic **K** describes only (arbitrary) Kripke models, including models where R_a does not necessarily reflect knowledge.

Consider, e. g., model $\mathcal{M}$ below:



- $(\mathcal{M}, s_1) \models p$, but
- $(\mathcal{M}, s_1) \models K_a \neg p$.

$\rightsquigarrow$ this violates that knowledge should imply truth.

Axiomatization

We would like a logic where something like $\neg(p \wedge K_a \neg p)$ is a theorem.

Semantically, we solved this by requiring epistemic models to have reflexive accessibility relations (among other requirements).

Syntatically, we can add axiom $K_a \varphi \rightarrow \varphi$.

Axiomatization

Axioms and inference rules of **S5**:

- All axioms and rules of **K**
- (*T*) $K_a\varphi \rightarrow \varphi$
  (Truth)
- (*4*) $K_a\varphi \rightarrow K_aK_a\varphi$
  (Positive introspection)
- (*5*) $\neg K_a\varphi \rightarrow K_a\neg K_a\varphi$
  (Negative introspection)

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Axiomatization

Example

Proof of $\vdash_{\mathbf{S5}} K_aK_bp \rightarrow K_ap$:

1. $K_bp \rightarrow p$
   (axiom *T*)
2. $K_a(K_bp \rightarrow p)$
   (Necessitation of K_a, 1)
3. $K_a(K_bp \rightarrow p) \rightarrow (K_aK_bp \rightarrow K_ap)$
   (axiom *K* with $\varphi = K_bp$ and $\psi = p$)
4. $K_aK_bp \rightarrow K_ap$
   (Modus ponens, 2+3)

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Axiomatization

Theorem

Axiom system **K** *is sound and complete w.r.t. the class* $\mathcal{K}$ *of all Kripke models, i. e., for every formula* φ *in* $\mathcal{L}_K$*, we have* $\vdash_{\mathbf{K}} \varphi$ *iff* $\mathcal{K} \models \varphi$. □

Theorem

Axiom system **S5** *is sound and complete w.r.t. the class* $\mathcal{S}5$ *of all epistemic models, i. e., for every formula* φ *in* $\mathcal{L}_K$*, we have* $\vdash_{\mathbf{S5}} \varphi$ *iff* $\mathcal{S}5 \models \varphi$. □

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Common knowledge

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Common Knowledge

Recall "everybody knows": $E_B\varphi \equiv \bigwedge_{b\in B} K_b\varphi$.

- E_B satisifes axiom T, but not (positive or negative) introspection.
- I. e., $E_B\varphi \rightarrow E_B E_B\varphi$ is not valid.
- E. g., if agents a and b are both (separately) told that p is true, $E_{ab}p$ is true but not $E_{ab}E_{ab}p$.
- So, how to model that everybody knows that everybody knows that ... that p?
- $\rightsquigarrow$ the common knowledge operator:
  For $B \subseteq A$, $C_B\varphi \equiv \varphi \wedge E_B\varphi \wedge E_B^2\varphi \wedge E_B^3\varphi \wedge \dots$,
  where $E_B^n\varphi = \underbrace{E_B E_B \dots E_B}_{n \text{ times}}\varphi$.

Common Knowledge

Notational conventions:

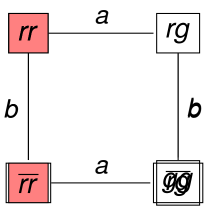
- Instead of $C_{\{a,b\}}$ or $E_{\{a,b\}}$, we often write C_{ab} and E_{ab}, respectively, etc.
- Instead of C_A or E_A, we usually write C and E, respectively, if A is the set of all agents.

Common Knowledge

Example (Common knowledge in card games)

Agents a and b are dealt one card each, both (independently) either red or green. They only see their own card. The actual card deal is rr. Now a tells b that she has a red card. Next, b leaves the room, giving a the chance to secretly look at b's card. She doesn't have to, but she does look.

Model $\mathcal{M}_1^{rg}$:



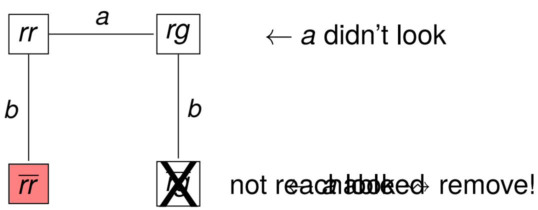
$\mathcal{M}_2^{rg}, rr \models C_{ab}red(a)$

Common Knowledge

Example (Common knowledge in card games, ctd.)

... She doesn't have to, but she does look. Now, a tells b that she looked at his card.

Model $\mathcal{M}_3^{rg}$:



Common Knowledge

By language $\mathcal{L}_{KC}$, we refer to the language defined like $\mathcal{L}_K$, but with the additional common knowledge modality C.

Definition (Epistemic language with common knowledge)

Let P be a countable set of atomic propositions and A be a finite set of agent symbols. Then the language $\mathcal{L}_{KC}$ is defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid C_B\varphi,$$

where $p \in P$, $a \in A$, and $B \subseteq A$.

Common Knowledge

Semantics of common knowledge modality: as before, using (epistemic) Kripke models.

Definition (Accessibility relations for E_B and C_B)

Let $\mathcal{M} = (S, R, V)$ be a Kripke model with agents A and $B \subseteq A$.

- Then $R_{E_B} = \bigcup_{b \in B} R_b$.
- The transitive closure of a relation R is the smallest relation R^+ such that:
  - $R \subseteq R^+$, and
  - for all x, y, z, if $(x, y) \in R^+$ and $(y, z) \in R^+$ then also $(x, z) \in R^+$.

  If, additionally, $(x, x) \in R^+$ for all x, then R^+ is the reflexive-transitive closure of R, symbolically R^*.
- Then, define $R_{C_B} = R^*_{E_B}$. (Sometimes also $\sim_{C_B}$.)

Common Knowledge

Definition

The truth of an $\mathcal{L}_{KC}$ formula φ in an epistemic state $(\mathcal{M}, s)$, symbolically $\mathcal{M}, s \models \varphi$, is defined as for $\mathcal{L}_K$, with an additional clause for common knowledge C_B, $B \subseteq A$:

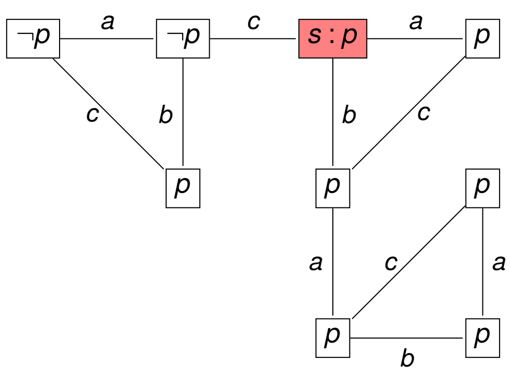
$$\mathcal{M}, s \models C_B\varphi \quad \text{iff} \quad \mathcal{M}, t \models \varphi \text{ for all } t \in S \text{ with } (s, t) \in R_{C_B}.$$

Common Knowledge

Example

$\mathcal{M}, s \models C_{ab}p$

$\mathcal{M}, s \not\models C_{abc}p$



Common Knowledge

Additional axioms and inference rules for common knowledge:

- $C_B(\varphi \rightarrow \psi) \rightarrow (C_B\varphi \rightarrow C_B\psi)$
  (Distribution of C_B over $\rightarrow$)
- $C_B\varphi \rightarrow (\varphi \wedge E_B C_B\varphi)$
  (Mix)
- $C_B(\varphi \rightarrow E_B\varphi) \rightarrow (\varphi \rightarrow C_B\varphi)$
  (Induction of common knowledge)
- From φ, infer $C_B\varphi$
  (Necessitation of C_B)

Theorem

Together with **S5** *axioms and rules, the above axiomatization is sound and complete with respect to epistemic models with common knowledge.* □

Model Checking

Model Checking

Question 1 (local model checking): Given model $\mathcal{M}$, state s of $\mathcal{M}$, and formula φ. How to test (algorithmically) whether $\mathcal{M}, s \models \varphi$?

Possible answer (Q1): Determine whether $\mathcal{M}, s \models \varphi$ by iteratively unraveling definition of $\models$ relation. For efficiency, cache intermediate results.
This works even if $\mathcal{M}$ is only given implicitly.

Model Checking

Question 2 (global model checking): Given model $\mathcal{M}$ and formula φ. How to determine (algorithmically) the set of all states s of $\mathcal{M}$ such that $\mathcal{M}, s \models \varphi$?

Possible answer (Q2): For all subformulas ψ of φ, determine the sets of states where ψ is true, inductively from small to large subformulas. Details below.

Model Checking

Definition (Subformula)

Let φ be an $\mathcal{L}_{KC}$ formula. Then the set of subformulas of φ, $subf(\varphi)$, is inductively defined as follows:

$$\begin{aligned} subf(p) &= \{p\} \text{ for } p \in P \\ subf(\neg\varphi) &= \{\neg\varphi\} \cup subf(\varphi) \\ subf(\varphi \wedge \psi) &= \{\varphi \wedge \psi\} \cup subf(\varphi) \cup subf(\psi) \\ subf(K_a\varphi) &= \{K_a\varphi\} \cup subf(\varphi) \\ subf(C_B\varphi) &= \{C_B\varphi\} \cup subf(\varphi) \end{aligned}$$

If $\psi \in subf(\varphi) \setminus \{\varphi\}$, then ψ is called a proper subformula of φ.

Model Checking

Definition

Let a be an agent and $S' \subseteq S$. Then the strong preimage of S' with respect to R_a is the set of states

$$preim_a(S') = \{s \in S \mid s' \in S' \text{ for all } s' \in S \text{ with } (s,s') \in R_a\}.$$

For $B \subseteq A$, we write

$$preim_B(S') = \bigcap_{b \in B} preim_b(S').$$

Notation:

When the model $\mathcal{M}$ and domain S are clear from the context, for a given formula φ, we write $[\![\varphi]\!]$ for the set of states where φ is true, i. e., for $\{s \in S \mid \mathcal{M}, s \models \varphi\}$.

Model Checking

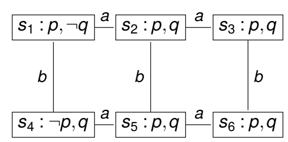
Algorithm

Let $\mathcal{M} = \langle S, R, V \rangle$ be an (epistemic) Kripke model and $\varphi \in \mathcal{L}_{KC}$ a formula. Let $\varphi_1, \dots, \varphi_n$ be the subformulas of φ ordered from small to large ($\varphi_n = \varphi$). For $i = 1, \dots, n$, do:

- **switch** φ_i **do**
  - **case** $p \in P$
    - $[\![\varphi_i]\!] := V(p)$
  - **case** $\neg\varphi'$
    - $[\![\varphi_i]\!] := S \setminus [\![\varphi']\!]$
  - **case** $\varphi' \wedge \varphi''$
    - $[\![\varphi_i]\!] := [\![\varphi']\!] \cap [\![\varphi'']\!]$
  - **case** $K_a\varphi'$
    - $[\![\varphi_i]\!] := preim_a([\![\varphi']\!])$
  - **case** $C_B\varphi'$
    - $S' := [\![\varphi']\!]$
    - **while** not $fixpt(S')$ **do**
      - $S' := S' \cap preim_B(S')$
    - **end while**
    - $[\![\varphi_i]\!] := S'$

Model Checking

Example ($[\![\neg K_b(K_a p \wedge q)]\!] = ?$)



$$\begin{aligned} [\![p]\!] &= \{s_1, s_2, s_3, s_5, s_6\} \\ [\![q]\!] &= \{s_2, s_3, s_4, s_5, s_6\} \\ [\![K_a p]\!] &= \{s_1, s_2, s_3\} \\ [\![K_a p \wedge q]\!] &= \{s_2, s_3\} \\ [\![K_b(K_a p \wedge q)]\!] &= \emptyset \\ [\![\neg K_b(K_a p \wedge q)]\!] &= \{s_1, s_2, s_3, s_4, s_5, s_6\} \end{aligned}$$

Model Checking

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Example ($[\![C_{ab}p]\!]$ =?)

$s_1 : p \overset{a}{—} s_2 : p \overset{a}{—} s_3 : p \overset{a}{—} s_4 : p \overset{b}{—} s_5 : p \overset{c}{—} s_6 : p \overset{a}{—} s_7 : p \overset{b}{—} s_8 : \neg p$

$$[\![p]\!] = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$$
$$S' := [\![p]\!]$$
$$S' := S' \cap preim_{ab}(S') = \{s_1, s_2, s_3, s_4, s_5, s_6\}$$
$$S' := S' \cap preim_{ab}(S') = \{s_1, s_2, s_3, s_4, s_5\}$$
$$S' := S' \cap preim_{ab}(S') = \{s_1, s_2, s_3, s_4, s_5\} \quad (\textit{fixpoint!})$$
$$[\![C_{ab}p]\!] = \{s_1, s_2, s_3, s_4, s_5\}$$

Summary

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

Summary

Language
Semantics
Axioms
Common knowledge
Model Checking
Summary

- Basic epistemic language $\mathcal{L}_K$: like propositional logic, plus knowledge modalities
- Kripke semantics: possible worlds, accessibility relations, propositional valuations
- $\mathcal{S}5$ (knowledge): accessibility relations are equivalence relations
- $\mathcal{L}_K$ formulas cannot distinguish between bisimilar models.
- Several axioms have 1-to-1 correspondence to properties of accessibility relations.
- Sound and complete axiomatizations of $\mathcal{K}$ and $\mathcal{S}5$
- Common knowledge = transitive closure of general knowledge (“everybody knows”)
- Algorithmic aspect of epistemic logic (so far): model checking