Reasoning about (Lack of) Knowledge

Christoph Schwering

UNSW Sydney

COMP4418, Week 7

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)
- Proposed Advice Taker (1959)
  - *Programs with Common Sense*

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)
- Proposed Advice Taker (1959)
  - *Programs with Common Sense*
  - Improve program behaviour by making statements to it

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)
- Proposed Advice Taker (1959)
  - *Programs with Common Sense*
  - Improve program behaviour by making statements to it
  - Program draws conclusions from its knowledge

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)
- Proposed Advice Taker (1959)
  - *Programs with Common Sense*
  - Improve program behaviour by making statements to it
  - Program draws conclusions from its knowledge
    - Declarative conclusion: new knowledge
    - Imperative conclusion: take action

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)
- Proposed Advice Taker (1959)
  - *Programs with Common Sense*
  - Improve program behaviour by making statements to it
  - Program draws conclusions from its knowledge
    - Declarative conclusion: new knowledge
    - Imperative conclusion: take action
  - Remains a vision to this date

Motivation

John McCarthy (1927–2011):

- Stanford, MIT, Dartmouth
- Turing Award
- Invented Lisp (1958)
- Invented Garbage Collection (1959)
- Founding Father of AI (with Minsky, Newell, Simon, 1955)
- Proposed Advice Taker (1959)
  - *Programs with Common Sense*
  - Improve program behaviour by making statements to it
  - Program draws conclusions from its knowledge
    - Declarative conclusion: new knowledge
    - Imperative conclusion: take action
  - Remains a vision to this date

Advice Taker motivates (directly or indirectly) a lot of AI research, in particular what we'll be studying for the next three weeks

Motivation

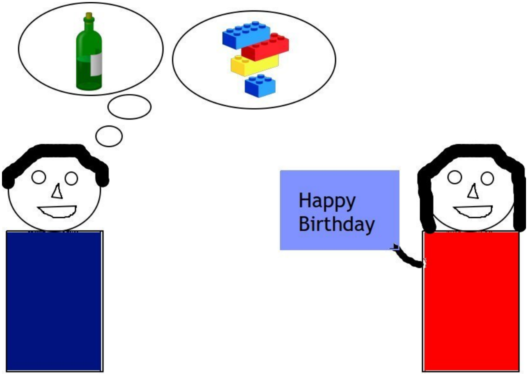
Observation: Non-knowledge is important

Not only what we know is relevant, but also what we *don't* know

Motivation

Observation: Non-knowledge is important

Not only what we know is relevant, but also what we *don't* know



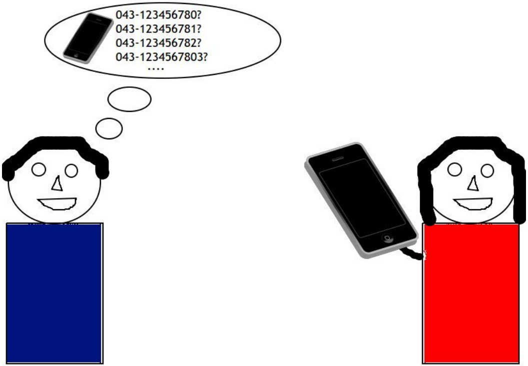
You don't know what's in the gift box.

You'll treat it with great care.

Motivation

Observation: Non-knowledge is important

Not only what we know is relevant, but also what we *don't* know



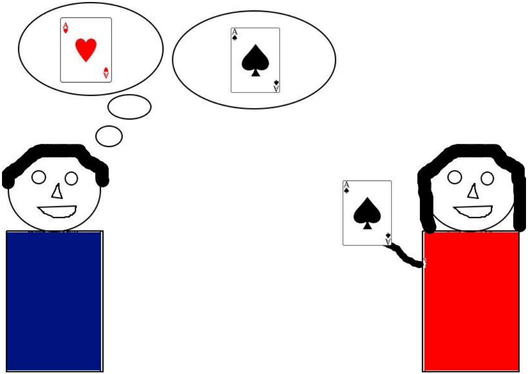
You know Jane has a phone, but you don't know her number.

You'll look it up.

Motivation

Observation: Non-knowledge is important

Not only what we know is relevant, but also what we *don't* know



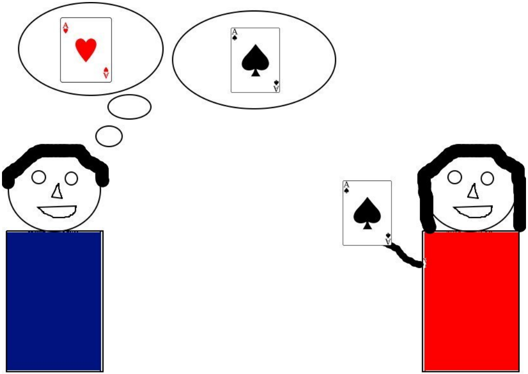
You know Jane is holding ace of spades *or* of hearts, but not which.

You'll need a strategy that wins in either case.

Motivation

Observation: Non-knowledge is important

Not only what we know is relevant, but also what we *don't* know



You know Jane is holding ace of spades *or* of hearts, but not which.

You'll need a strategy that wins in either case.

How can we accurately **capture knowledge and non-knowledge**?

Overview of the Lecture

- **A Logic of Knowledge – The Propositional Fragment**
  - Why not classical logic?
  - Syntax and semantics
  - Omniscience, introspection, only-knowing
  - Representation theorem
- A Logic of Knowledge – The First-Order Case
- Extensions of the Logic of Knowledge

What Is a Knowledge Base?

- A **knowledge base** (KB) is a collection of sentences that describe (a fragment of) the world

What Is a Knowledge Base?

- A **knowledge base** (KB) is a collection of sentences that describe (a fragment of) the world

- KB completely characterises what the agent knows, i.e.,
  - α is known $\Longrightarrow \mathrm{KB} \models \alpha$
  - α is not known $\Longrightarrow \mathrm{KB} \not\models \alpha$

  $\Longrightarrow$ KB is *all* the agent knows

What Is a Knowledge Base?

- A **knowledge base** (KB) is a collection of sentences that describe (a fragment of) the world

- KB completely characterises what the agent knows, i.e.,
  - α is known $\implies \mathrm{KB} \models \alpha$
  - α is not known $\implies \mathrm{KB} \not\models \alpha$

  $\implies$ KB is *all* the agent knows

- Purpose: evaluate queries
  - What is known? What is unknown?
  - Similar to a database, but draws interences

What Is a Knowledge Base?

- A **knowledge base** (KB) is a collection of sentences that describe (a fragment of) the world

- KB completely characterises what the agent knows, i.e.,
  - α is known $\implies \mathrm{KB} \models \alpha$
  - α is not known $\implies \mathrm{KB} \not\models \alpha$

  $\implies$ KB is *all* the agent knows

- Purpose: evaluate queries
  - What is known? What is unknown?
  - Similar to a database, but draws interences

- Usually: what is known $\subsetneq$ what is true
  - Agent's knowledge is incomplete
  - Agent should be aware of that

What Is a Knowledge Base?

- A **knowledge base** (KB) is a collection of sentences that describe (a fragment of) the world

- KB completely characterises what the agent knows, i.e.,
  - α is known $\implies \mathrm{KB} \models \alpha$
  - α is not known $\implies \mathrm{KB} \not\models \alpha$

  $\implies$ KB is *all* the agent knows

- Purpose: evaluate queries
  - What is known? What is unknown?
  - Similar to a database, but draws interences

- Usually: what is known $\subsetneq$ what is true
  - Agent's knowledge is incomplete
  - Agent should be aware of that

- Usually: knowing is more than database lookup
  - $\alpha \in \mathrm{KB} \implies \alpha$ is explicit knowledge (= database lookup)
  - $\mathrm{KB} \models \alpha \implies \alpha$ is implicit knowledge (= logical inference)
  - Usually: explicit knowledge $\subsetneq$ (implicit) knowledge

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r.
2. You don't know that $\neg r$.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r.
2. You don't know that $\neg r$.
3. You know that p or q.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r.
2. You don't know that $\neg r$.
3. You know that p or q.
4. You don't know that p.
5. You don't know that q.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r.
2. You don't know that $\neg r$.
3. You know that p or q.
4. You don't know that p.
5. You don't know that q.
6. You know that p or q, but not which.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r. $\mathrm{KB} \not\models r$
2. You don't know that $\neg r$. $\mathrm{KB} \not\models \neg r$
3. You know that p or q.
4. You don't know that p.
5. You don't know that q.
6. You know that p or q, but not which.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r. $\quad \mathrm{KB} \not\models r$
2. You don't know that $\neg r$. $\quad \mathrm{KB} \not\models \neg r$
3. You know that p or q. $\quad \mathrm{KB} \models (p \vee q)$
4. You don't know that p.
5. You don't know that q.
6. You know that p or q, but not which.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r. $\quad \mathrm{KB} \not\models r$
2. You don't know that $\neg r$. $\quad \mathrm{KB} \not\models \neg r$
3. You know that p or q. $\quad \mathrm{KB} \models (p \vee q)$
4. You don't know that p. $\quad \mathrm{KB} \not\models p$
5. You don't know that q. $\quad \mathrm{KB} \not\models q$
6. You know that p or q, but not which.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r. $\quad \mathrm{KB} \not\models r$
2. You don't know that $\neg r$. $\quad \mathrm{KB} \not\models \neg r$
3. You know that p or q. $\quad \mathrm{KB} \models (p \vee q)$
4. You don't know that p. $\quad \mathrm{KB} \not\models p$
5. You don't know that q. $\quad \mathrm{KB} \not\models q$
6. You know that p or q, but not which. $\quad \mathrm{KB} \models ???$

<u>Problem</u>: Classical logic cannot express 6 directly in one formula

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r) \wedge \neg k\_p \wedge \ldots$

Then:

1. You don't know that r. $\quad \mathrm{KB} \models \neg k\_r$
2. You don't know that $\neg r$. $\quad \mathrm{KB} \models \neg k\_not\_r$
3. You know that p or q. $\quad \mathrm{KB} \models k\_p\_or\_q$
4. You don't know that p. $\quad \mathrm{KB} \models \neg k\_p$
5. You don't know that q. $\quad \mathrm{KB} \models \neg k\_q$
6. You know that p or q, but not which.

$$\mathrm{KB} \models k\_p\_or\_q \wedge \neg k\_p \wedge \neg k\_p$$

Problem: Classical logic cannot express 6 directly in one formula

Idea #1: Compile $(p \vee q \vee c)$ to new atoms $k\_p, k\_p\_or\_q, \ldots$ ✗
Does not scale.

Why Not Classical Logic?

Suppose all you know is $(p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Then:

1. You don't know that r. $\quad \mathrm{KB} \models r = \mathrm{U}$
2. You don't know that $\neg r$. $\quad \mathrm{KB} \models \neg r = \mathrm{U}$
3. You know that p or q. $\quad \mathrm{KB} \models (p \vee q)$
4. You don't know that p. $\quad \mathrm{KB} \models p = \mathrm{U}$
5. You don't know that q. $\quad \mathrm{KB} \models q = \mathrm{U}$
6. You know that p or q, but not which.

$$\mathrm{KB} \models (p \vee q) \wedge p = \mathrm{U} \wedge q = \mathrm{U}$$

<u>Problem</u>: Classical logic cannot express 6 directly in one formula

<u>Idea #2</u>: Three-valued logic $\{0, 1, \mathrm{U}\}$ ✗
How would $\mathrm{U} \vee \mathrm{U}$ behave? Is it known? Unknown?

Why Not Classical Logic?

Suppose all you know is $(p \lor q \lor r) \land (p \lor q \lor \neg r)$

Then:

1. You don't know that r. $\quad \mathbf{O}\mathrm{KB} \models \neg\mathbf{K}r$
2. You don't know that $\neg r$. $\quad \mathbf{O}\mathrm{KB} \models \neg\mathbf{K}\neg r$
3. You know that p or q. $\quad \mathbf{O}\mathrm{KB} \models \mathbf{K}(p \lor q)$
4. You don't know that p. $\quad \mathbf{O}\mathrm{KB} \models \neg\mathbf{K}p$
5. You don't know that q. $\quad \mathbf{O}\mathrm{KB} \models \neg\mathbf{K}q$
6. You know that p or q, but not which.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}(p \lor q) \land \neg\mathbf{K}p \land \neg\mathbf{K}q$$

<u>Problem</u>: Classical logic cannot express 6 directly in one formula

<u>Idea #3</u>: Add unary operators $\mathbf{O}$ and $\mathbf{K}$ to express knowledge ✓

The Language of $\mathcal{OL}_{\text{PL}}$

The language of only-knowing (propositional fragment) $\mathcal{OL}_{\text{PL}}$:

- $p, q, r, \ldots$ — atomic propositions
- $\neg\alpha$ — "not α"
- $(\alpha \vee \beta)$ — "α or β"
- $(\alpha \wedge \beta)$ — "α and β"
- $(\alpha \rightarrow \beta)$ — "α implies β"
- $(\alpha \leftrightarrow \beta)$ — "α is equivalent to β"
- $\mathbf{K}\alpha$ — "α is known"
- $\mathbf{O}\alpha$ — "α is *all* that is known"

The Language of $\mathcal{OL}_{\mathrm{PL}}$

The language of only-knowing (propositional fragment) $\mathcal{OL}_{\mathrm{PL}}$:

- $p, q, r, \ldots$ — atomic propositions
- $\neg\alpha$ — "not α"
- $(\alpha \vee \beta)$ — "α or β"
- $(\alpha \wedge \beta) \overset{\text{def}}{=} \neg(\neg\alpha \vee \neg\beta)$ — "α and β"
- $(\alpha \rightarrow \beta) \overset{\text{def}}{=} (\neg\alpha \vee \beta)$ — "α implies β"
- $(\alpha \leftrightarrow \beta) \overset{\text{def}}{=} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ — "α is equivalent to β"
- $\mathbf{K}\alpha$ — "α is known"
- $\mathbf{O}\alpha$ — "α is *all* that is known"

Recap: Technical Terms (1)

A logical language is a *formal language* over an *alphabet* (here: $\{p, q, r, \ldots, (,), \neg, \vee, \mathbf{K}, \mathbf{O}\}$) and a *grammar* (previous slide), i.e., rules that allow us to phrase sentences in that language.

Recap: Technical Terms (1)

A logical language is a *formal language* over an *alphabet* (here: $\{p, q, r, \ldots, (,), \neg, \vee, \mathbf{K}, \mathbf{O}\}$) and a *grammar* (previous slide), i.e., rules that allow us to phrase sentences in that language.

The sentences carry *no meaning by themselves*. We define a *model theory* to give them a *semantics*, i.e., to define what sort of formal structure interprets a sentence. Such an *interpretation* I satisfies a sentence α, written $I \models \alpha$, or falsifies it, written $I \not\models \alpha$.

Recap: Technical Terms (1)

A logical language is a *formal language* over an *alphabet* (here: $\{p, q, r, \ldots, (,), \neg, \vee, \mathbf{K}, \mathbf{O}\}$) and a *grammar* (previous slide), i.e., rules that allow us to phrase sentences in that language.

The sentences carry *no meaning by themselves*. We define a *model theory* to give them a *semantics*, i.e., to define what sort of formal structure interprets a sentence. Such an *interpretation* I satisfies a sentence α, written $I \models \alpha$, or falsifies it, written $I \not\models \alpha$.

A typical rule of a semantics is

$$I \models (\alpha \vee \beta) \text{ if and only if } I \models \alpha \text{ or } I \models \beta.$$

Note that $\vee$ is a symbol of the logical language, whereas “if and only if” and “or” are natural language expressions. The rule says that the symbol “$\vee$” corresponds to the natural language expression “or”.

Recap: Technical Terms (1)

A logical language is a *formal language* over an *alphabet* (here: $\{p, q, r, \ldots, (,), \neg, \vee, \mathbf{K}, \mathbf{O}\}$) and a *grammar* (previous slide), i.e., rules that allow us to phrase sentences in that language.

The sentences carry *no meaning by themselves*. We define a *model theory* to give them a *semantics*, i.e., to define what sort of formal structure interprets a sentence. Such an *interpretation* I satisfies a sentence α, written $I \models \alpha$, or falsifies it, written $I \not\models \alpha$.

A typical rule of a semantics is

$$I \models (\alpha \vee \beta) \text{ if and only if } I \models \alpha \text{ or } I \models \beta.$$

Note that $\vee$ is a symbol of the logical language, whereas "if and only if" and "or" are natural language expressions. The rule says that the symbol "$\vee$" corresponds to the natural language expression "or".

We will sometimes take the liberty to omit brackets to ease readability. For instance, we write $(p \vee q \vee r)$ instead of $((p \vee q) \vee r)$ or $(p \vee (q \vee r))$, implicitly assuming our semantics of $\vee$ is associative.

Recap: Technical Terms (2)

The form of such an interpretation varies between logics.
Propositional logic uses truth tables, first-order logic usually uses structures with a domain and interpretation function.

- When an interpretation I satisfies a sentence, we write $I \models \alpha$.
- When all interpretations satisfy a sentence α, then α is *valid* and we write $\models \alpha$.
- When all interpretations that satisfy some sentence Σ or set of sentences Σ also satisfy α, we say Σ *entails* α and write $\Sigma \models \alpha$.

Recap: Technical Terms (2)

The form of such an interpretation varies between logics. Propositional logic uses truth tables, first-order logic usually uses structures with a domain and interpretation function.

- When an interpretation I satisfies a sentence, we write $I \models \alpha$.
- When all interpretations satisfy a sentence α, then α is *valid* and we write $\models \alpha$.
- When all interpretations that satisfy some sentence Σ or set of sentences Σ also satisfy α, we say Σ *entails* α and write $\Sigma \models \alpha$.

Different semantics are possible. What justifies a semantics? Typically there is a *proof theory*, and model theory and proof theory should be equivalent ($\models \alpha$ if and only if $\vdash \alpha$). Nevertheless, we will only focus on the semantics in the next weeks.

The Semantics of $\mathcal{OL}_{\mathrm{PL}}$

Definition: semantics of $\mathcal{OL}_{\mathrm{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

The Semantics of $\mathcal{OL}_{\text{PL}}$

Definition: semantics of $\mathcal{OL}_{\text{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

- $w \models P \iff w[P] = 1$
- $w \models \neg\alpha \iff w \not\models \alpha$
- $w \models (\alpha \vee \beta) \iff w \models \alpha$ or $w \models \beta$

The Semantics of $\mathcal{OL}_{\mathrm{PL}}$

Definition: semantics of $\mathcal{OL}_{\mathrm{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

- $w \models P \iff w[P] = 1$
- $w \models \neg\alpha \iff \quad w \not\models \alpha$
- $w \models (\alpha \vee \beta) \iff \quad w \models \alpha$ or $\quad w \models \beta$
- $w \models \mathbf{K}\alpha \iff$???
- $w \models \mathbf{O}\alpha \iff$???

The Semantics of $\mathcal{OL}_{\mathrm{PL}}$

Definition: semantics of $\mathcal{OL}_{\mathrm{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

An **epistemic state** e is a set of worlds.

- $w \models P \iff w[P] = 1$
- $w \models \neg\alpha \iff \quad w \not\models \alpha$
- $w \models (\alpha \vee \beta) \iff \quad w \models \alpha$ or $\quad w \models \beta$
- $w \models \mathbf{K}\alpha \iff$???
- $w \models \mathbf{O}\alpha \iff$???

The Semantics of $\mathcal{OL}_{\text{PL}}$

Definition: semantics of $\mathcal{OL}_{\text{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

An **epistemic state** e is a set of worlds.

- $e, w \models P \iff w[P] = 1$
- $e, w \models \neg\alpha \iff e, w \not\models \alpha$
- $e, w \models (\alpha \vee \beta) \iff e, w \models \alpha$ or $e, w \models \beta$
- $e, w \models \mathbf{K}\alpha \iff$???
- $e, w \models \mathbf{O}\alpha \iff$???

The Semantics of $\mathcal{OL}_{\mathrm{PL}}$

Definition: semantics of $\mathcal{OL}_{\mathrm{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

An **epistemic state** e is a set of worlds.

- $e, w \models P \iff w[P] = 1$
- $e, w \models \neg\alpha \iff e, w \not\models \alpha$
- $e, w \models (\alpha \vee \beta) \iff e, w \models \alpha$ or $e, w \models \beta$
- $e, w \models \mathbf{K}\alpha \iff$ for all worlds w', $w' \in e \Rightarrow e, w' \models \alpha$
- $e, w \models \mathbf{O}\alpha \iff$???

"$\Rightarrow$" stands for natural language expressions "only if".
"$\Leftrightarrow$" and " $\iff$ " stand for natural language expressions "if and only if".

The Semantics of $\mathcal{OL}_{\mathrm{PL}}$

Definition: semantics of $\mathcal{OL}_{\mathrm{PL}}$

A **world** w is a function from the atomic propositions to $\{0, 1\}$.

An **epistemic state** e is a set of worlds.

- $e, w \models P \iff w[P] = 1$
- $e, w \models \neg\alpha \iff e, w \not\models \alpha$
- $e, w \models (\alpha \vee \beta) \iff e, w \models \alpha$ or $e, w \models \beta$
- $e, w \models \mathbf{K}\alpha \iff$ for all worlds w', $w' \in e \Rightarrow e, w' \models \alpha$
- $e, w \models \mathbf{O}\alpha \iff$ for all worlds w', $w' \in e \Leftrightarrow e, w' \models \alpha$

"$\Rightarrow$" stands for natural language expressions "only if".
"$\Leftrightarrow$" and "$\iff$" stand for natural language expressions "if and only if".

Abbreviations

Recall:

- $(\alpha \wedge \beta) \stackrel{\text{def}}{=} \neg(\neg\alpha \vee \neg\beta)$
- $(\alpha \rightarrow \beta) \stackrel{\text{def}}{=} (\neg\alpha \vee \beta)$
- $(\alpha \leftrightarrow \beta) \stackrel{\text{def}}{=} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

$\wedge$ should be "and"
$\rightarrow$ should be "only if"
$\leftrightarrow$ should be "if and only if".

Lemma: abbreviations

- $e, w \models \alpha \wedge \beta \iff e, w \models \alpha$ and $e, w \models \beta$
- $e, w \models \alpha \rightarrow \beta \iff e, w \models \alpha \Rightarrow e, w \models \beta$
- $e, w \models \alpha \leftrightarrow \beta \iff e, w \models \alpha \Leftrightarrow e, w \models \beta$

Proof on paper

Some Lemmas

Definition: objective, subjective

If ϕ mentions no atoms inside $\mathbf{K}$ or $\mathbf{O}$, we say ϕ is **objective**.
If σ mentions atoms only inside $\mathbf{K}$ or $\mathbf{O}$, we say σ is **subjective**.

- $((p \vee q) \wedge p \wedge q)$ is objective
- $\mathbf{K}((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q)$ is subjective

Some Lemmas

Definition: objective, subjective

If ϕ mentions no atoms inside $\mathbf{K}$ or $\mathbf{O}$, we say ϕ is **objective**.
If σ mentions atoms only inside $\mathbf{K}$ or $\mathbf{O}$, we say σ is **subjective**.

- $((p \vee q) \wedge p \wedge q)$ is objective
- $\mathbf{K}((p \vee q) \wedge \neg \mathbf{K} p \wedge \neg \mathbf{K} q)$ is subjective

Lemma: objective, subjective

Let ϕ be objective. Then $e, w \models \phi \iff e', w \models \phi$.
Let σ be subjective. Then $e, w \models \sigma \iff e, w' \models \sigma$.

Some Lemmas

Definition: objective, subjective

If ϕ mentions no atoms inside $\mathbf{K}$ or $\mathbf{O}$, we say ϕ is **objective**.
If σ mentions atoms only inside $\mathbf{K}$ or $\mathbf{O}$, we say σ is **subjective**.

- $((p \vee q) \wedge p \wedge q)$ is objective
- $\mathbf{K}((p \vee q) \wedge \neg \mathbf{K}p \wedge \neg \mathbf{K}q)$ is subjective

Lemma: objective, subjective

Let ϕ be objective. Then $e, w \models \phi \iff e', w \models \phi$.
Let σ be subjective. Then $e, w \models \sigma \iff e, w' \models \sigma$.

When ϕ is objective, "$w \models \phi$" stands for "for every e, $e, w \models \phi$".
When σ is subjective, "$e \models \sigma$" stands for "for every w, $e, w \models \sigma$".

Proof on paper

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \ w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \;\; w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e$
  $\iff \;\; w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \lor q \lor r) \land (p \lor q \lor \neg r)\}$

- $w \in e$

  $\iff \quad w \models (p \lor q \lor r) \land (p \lor q \lor \neg r)$

  $\iff \quad w \models (p \lor q \lor r)$ and $w \models (p \lor q \lor \neg r)$

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \;\; w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e$

$$\begin{aligned}
&\iff && w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r) \\
&\iff && w \models (p \vee q \vee r) \text{ and } w \models (p \vee q \vee \neg r) \\
&\iff && w[p] = 1 \text{ or } w[q] = 1 \text{ or } w[r] = 1\text{, and} \\
& && w[p] = 1 \text{ or } w[q] = 1 \text{ or } w[r] = 0
\end{aligned}$$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \overset{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e$

$$\begin{aligned}
\iff\ & w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r) \\
\iff\ & w \models (p \vee q \vee r) \text{ and } w \models (p \vee q \vee \neg r) \\
\iff\ & w[p] = 1 \text{ or } w[q] = 1 \text{ or } w[r] = 1, \text{ and} \\
& w[p] = 1 \text{ or } w[q] = 1 \text{ or } w[r] = 0 \\
\iff\ & w[p] = 1 \text{ or } w[q] = 1
\end{aligned}$$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \;\; w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$?

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$?
  $\iff$ for all w, $w \in e \Rightarrow w \models (p \vee q)$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \;\; w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$?

$\iff$ for all w, $w \in e \Rightarrow w \models (p \vee q)$
$\iff$ for all w, $w \in e \Rightarrow w \models p$ or $w \models q$

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$?

$\iff$ for all w, $w \in e \Rightarrow w \models (p \vee q)$
$\iff$ for all w, $w \in e \Rightarrow w \models p$ or $w \models q$
$\iff$ for all w, $w \in e \Rightarrow w[p] = 1$ or $w[q] = 1$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$?

$\iff$ for all w, $w \in e \Rightarrow w \models (p \vee q)$
$\iff$ for all w, $w \in e \Rightarrow w \models p$ or $w \models q$
$\iff$ for all w, $w \in e \Rightarrow w[p] = 1$ or $w[q] = 1$
$\iff$ for all w, $\big(w[p] = 1$ or $w[q] = 1\big) \Rightarrow \big(w[p] = 1$ or $w[q] = 1\big)$ ✓

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \;\; w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1 \text{ or } w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$?

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \; w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$?
  $\iff e \not\models \mathbf{K}p$

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$?

  $\iff e \not\models \mathbf{K}p$

  $\iff$ for some w, $w \in e$ and $w \not\models p$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \;\; w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \lor q \lor r) \land (p \lor q \lor \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \lor q)$ ✓
- $e \models \neg\mathbf{K}p$?

  $\iff$ $e \not\models \mathbf{K}p$

  $\iff$ for some w, $w \in e$ and $w \not\models p$

  $\iff$ for some w, $w \in e$ and $w[p] \neq 1$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \; w \in e \Rightarrow e, w' \models \alpha$$

Let $e \overset{\text{def}}{=} \{w \mid w \models (p \lor q \lor r) \land (p \lor q \lor \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \lor q)$ ✓
- $e \models \neg\mathbf{K}p$?

$\iff$ $e \not\models \mathbf{K}p$
$\iff$ for some w, $w \in e$ and $w \not\models p$
$\iff$ for some w, $w \in e$ and $w[p] \neq 1$
$\iff$ for some w, $w[p] = 1$ or $w[q] = 1$, and $w[p] \neq 1$

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$?

$\iff$ $e \not\models \mathbf{K}p$
$\iff$ for some w, $w \in e$ and $w \not\models p$
$\iff$ for some w, $w \in e$ and $w[p] \neq 1$
$\iff$ for some w, $w[p] = 1$ or $w[q] = 1$, and $w[p] \neq 1$
$\iff$ for some w, $w[p] \neq 1$ and $w[q] = 1$ ✓

Examples

$$\boxed{e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \; w \in e \Rightarrow e, w' \models \alpha}$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$ ✓
- $e \models \mathbf{K}(p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q$?

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ \ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \lor q \lor r) \land (p \lor q \lor \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \lor q)$ ✓
- $e \models \neg\mathbf{K}p$ ✓
- $e \models \mathbf{K}(p \lor q) \land \neg\mathbf{K}p \land \neg\mathbf{K}q$?
  $\iff$ $e \models \mathbf{K}(p \lor q)$ and $e \models \neg\mathbf{K}p$ and $e \models \neg\mathbf{K}q$ ✓

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ w \in e \Rightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$ ✓
- $e \models \mathbf{K}(p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q$ ✓
- $e \models \mathbf{O}((p \vee q \vee r) \wedge (p \vee q \vee \neg r))$?

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ w \in e \Rightarrow e, w' \models \alpha$$
$$e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w',\ w \in e \Leftrightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$ ✓
- $e \models \mathbf{K}(p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q$ ✓
- $e \models \mathbf{O}((p \vee q \vee r) \wedge (p \vee q \vee \neg r))$?
  $\iff$ for all w, $w \in e \Leftrightarrow w \models ((p \vee q \vee r) \wedge (p \vee q \vee \neg r))$ ✓

Examples

$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', \ w \in e \Rightarrow e, w' \models \alpha$$
$$e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w', \ w \in e \Leftrightarrow e, w' \models \alpha$$

Let $e \stackrel{\text{def}}{=} \{w \mid w \models (p \vee q \vee r) \wedge (p \vee q \vee \neg r)\}$

- $w \in e \iff w[p] = 1$ or $w[q] = 1$
- $e \models \mathbf{K}(p \vee q)$ ✓
- $e \models \neg\mathbf{K}p$ ✓
- $e \models \mathbf{K}(p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q$ ✓
- $e \models \mathbf{O}((p \vee q \vee r) \wedge (p \vee q \vee \neg r))$ ✓

Logical Omniscience

Logical omniscience means that an agent knows all the consequences of what they know. In particular, they know all valid sentences.

Logical Omniscience

Logical omniscience means that an agent knows all the consequences of what they know. In particular, they know all valid sentences.

Theorem: logical omniscience

If $\models \alpha \to \beta$, then $\models \mathbf{K}\alpha \to \mathbf{K}\beta$.
In particular: If $\models \alpha$, then $\models \mathbf{K}\alpha$.

Logical Omniscience

Logical omniscience means that an agent knows all the consequences of what they know. In particular, they know all valid sentences.

Theorem: logical omniscience

If $\models \alpha \rightarrow \beta$, then $\models \mathbf{K}\alpha \rightarrow \mathbf{K}\beta$.
In particular: If $\models \alpha$, then $\models \mathbf{K}\alpha$.

Logical omniscience is often problematic:

- Philosophical problem: most agents are not omniscient
- Practical problem: omniscience makes reasoning intractable

We will look at methods to avoid these problems next week.

Proof on paper

Only-Knowing

The purpose of only-knowing is to capture a knowledge base.

Knowledge bases are usually objective.

The corresponding epistemic state is then unique:

Theorem: unique-model property

Let ϕ be objective. Then there is a unique e such that $e \models \mathbf{O}\phi$.

An entailment problem $\mathbf{O}\phi \models \mathbf{K}\alpha$ thus reduces to model checking:

$$e \models \mathbf{K}\alpha, \text{ where } e = \{w \mid w \models \phi\}?$$

Self-Knowledge

We can nest $\mathbf{K}$ operators to say that we know that we know.

Complete and accurate knowledge about own knowledge:

Theorem: positive and negative introspection

Positive introspection: $\models \mathbf{K}\alpha \rightarrow \mathbf{K}\mathbf{K}\alpha$
Negative introspection: $\models \neg\mathbf{K}\alpha \rightarrow \mathbf{K}\neg\mathbf{K}\alpha$

Why?
$e \models (\neg)\mathbf{K}\alpha \implies e, w \models (\neg)\mathbf{K}\alpha$ for all $w \in e \iff e, w \models \mathbf{K}(\neg)\mathbf{K}\alpha$.

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Theorem

Let KB, ϕ be objective. Then $\mathbf{O}\mathrm{KB} \models \mathbf{K}\phi \iff \mathrm{KB} \models \phi$.

Idea: replace nested $\mathbf{K}\phi$ with TRUE if $\mathrm{KB} \models \phi$, otherwise with FALSE.

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Theorem

Let KB, ϕ be objective. Then $\mathbf{O}\mathrm{KB} \models \mathbf{K}\phi \iff \mathrm{KB} \models \phi$.

Idea: replace nested $\mathbf{K}\phi$ with TRUE if $\mathrm{KB} \models \phi$, otherwise with FALSE.

Ex.: Let $\mathrm{KB} \overset{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)?$$

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Theorem

Let KB, ϕ be objective. Then $\mathbf{O}\mathrm{KB} \models \mathbf{K}\phi \iff \mathrm{KB} \models \phi$.

Idea: replace nested $\mathbf{K}\phi$ with TRUE if $\mathrm{KB} \models \phi$, otherwise with FALSE.

Ex.: Let $\mathrm{KB} \overset{\text{def}}{=} (p \lor q \lor r) \land (p \lor q \lor \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \lor q) \land \neg \underbrace{\mathbf{K}p}_{\mathrm{KB} \models p?\ ✗} \land \neg \underbrace{\mathbf{K}q}_{\mathrm{KB} \models q?\ ✗}\big)?$$

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Theorem

Let KB, ϕ be objective. Then $\mathbf{O}\mathrm{KB} \models \mathbf{K}\phi \iff \mathrm{KB} \models \phi$.

Idea: replace nested $\mathbf{K}\phi$ with TRUE if $\mathrm{KB} \models \phi$, otherwise with FALSE.

Ex.: Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \lor q \lor r) \land (p \lor q \lor \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \lor q) \land \neg\textsf{FALSE} \land \neg\textsf{FALSE}\big)?$$

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Theorem

Let KB, ϕ be objective. Then $\mathbf{O}\mathrm{KB} \models \mathbf{K}\phi \iff \mathrm{KB} \models \phi$.

Idea: replace nested $\mathbf{K}\phi$ with TRUE if $\mathrm{KB} \models \phi$, otherwise with FALSE.

Ex.: Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \lor q \lor r) \land (p \lor q \lor \neg r)$.

$\mathrm{KB} \models \big((p \lor q) \land \neg\text{FALSE} \land \neg\text{FALSE}\big)$? ✓

Representation Theorem

Can we solve $\mathbf{O}\mathrm{KB} \models \alpha$ with ordinary, propositional reasoning?
That is, can we eliminate $\mathbf{K}$ and $\mathbf{O}$?
Then we could use standard reasoning system.

Theorem

Let KB, ϕ be objective. Then $\mathbf{O}\mathrm{KB} \models \mathbf{K}\phi \iff \mathrm{KB} \models \phi$.

Idea: replace nested $\mathbf{K}\phi$ with TRUE if $\mathrm{KB} \models \phi$, otherwise with FALSE.

Ex.: Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \lor q \lor r) \land (p \lor q \lor \neg r)$.

$$\mathrm{KB} \models \big((p \lor q) \land \neg\text{FALSE} \land \neg\text{FALSE}\big)? \quad ✓$$

Next slide formalises this idea.

Sneak preview: It'll become more difficult in the first-order case:
What would you replace $\mathbf{K}Q(x)$ with in $\mathbf{K}\exists x\,(P(x) \land \neg\mathbf{K}Q(x))$? We'll see later.

Representation Theorem (2)

Definition: representation operators

For objective KB and ϕ, let $\text{RES}[\text{KB}, \phi] \stackrel{\text{def}}{=} \begin{cases} \text{TRUE} & \text{if } \text{KB} \models \phi \\ \text{FALSE} & \text{otherwise} \end{cases}$

where TRUE is some tautology (e.g., $p \vee \neg p$) and FALSE is $\neg\text{TRUE}$.

Representation Theorem (2)

Definition: representation operators

For objective KB and ϕ, let $\mathrm{RES}[\mathrm{KB}, \phi] \overset{\text{def}}{=} \begin{cases} \text{TRUE} & \text{if } \mathrm{KB} \models \phi \\ \text{FALSE} & \text{otherwise} \end{cases}$

where TRUE is some tautology (e.g., $p \vee \neg p$) and FALSE is $\neg$TRUE.

- $\|P\|_{\mathrm{KB}} \overset{\text{def}}{=} P$
- $\|\neg\alpha\|_{\mathrm{KB}} \overset{\text{def}}{=} \neg\|\alpha\|_{\mathrm{KB}}$
- $\|(\alpha \vee \beta)\|_{\mathrm{KB}} \overset{\text{def}}{=} (\|\alpha\|_{\mathrm{KB}} \vee \|\beta\|_{\mathrm{KB}})$
- $\|\mathbf{K}\alpha\|_{\mathrm{KB}} \overset{\text{def}}{=} \mathrm{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$

Representation Theorem (2)

Definition: representation operators

For objective KB and ϕ, let $\mathrm{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \begin{cases} \text{TRUE} & \text{if } \mathrm{KB} \models \phi \\ \text{FALSE} & \text{otherwise} \end{cases}$

where TRUE is some tautology (e.g., $p \vee \neg p$) and FALSE is $\neg$TRUE.

- $\|P\|_{\mathrm{KB}} \stackrel{\text{def}}{=} P$
- $\|\neg\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \neg\|\alpha\|_{\mathrm{KB}}$
- $\|(\alpha \vee \beta)\|_{\mathrm{KB}} \stackrel{\text{def}}{=} (\|\alpha\|_{\mathrm{KB}} \vee \|\beta\|_{\mathrm{KB}})$
- $\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathrm{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$

Theorem: representation theorem

$\mathbf{O}\mathrm{KB} \models \alpha \iff \models \|\alpha\|_{\mathrm{KB}}$.

Example

$$\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathsf{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$$

$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \text{“}\mathrm{KB} \models \phi\text{?”}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)$?

Example

$$\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathsf{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$$
$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \text{“}\mathrm{KB} \models \phi\text{?”}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)$$

$$\iff \models \|\mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}$$

Example

$$\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathsf{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$$
$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \text{“}\mathrm{KB} \models \phi\text{?”}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)$$

$$\iff \quad \models \|\mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}$$

$$\iff \quad \models \mathsf{RES}[\mathrm{KB}, \|\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}]$$

Example

$$\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathsf{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$$
$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \text{“}\mathrm{KB} \models \phi?\text{”}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)$

$$\iff \models \|\mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}$$

$$\iff \models \mathsf{RES}[\mathrm{KB}, \|\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}]$$

$$\iff \models \mathsf{RES}[\mathrm{KB}, \big((p \vee q) \wedge \neg \underbrace{\|\mathbf{K}p\|_{\mathrm{KB}}}_{\mathrm{KB}\models p?} \wedge \neg \underbrace{\|\mathbf{K}q\|_{\mathrm{KB}}}_{\mathrm{KB}\models q?}\big)]$$

Example

$$\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathsf{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$$
$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \text{“}\mathrm{KB} \models \phi?\text{”}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)$$

$$\iff \models \|\mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}$$

$$\iff \models \mathsf{RES}[\mathrm{KB}, \|\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}]$$

$$\iff \models \mathsf{RES}[\mathrm{KB}, \big((p \vee q) \wedge \neg \underbrace{\|\mathbf{K}p\|_{\mathrm{KB}}}_{\mathrm{KB}\models p?} \wedge \neg \underbrace{\|\mathbf{K}q\|_{\mathrm{KB}}}_{\mathrm{KB}\models q?}\big)]$$

$$\iff \models \underbrace{\mathsf{RES}[\mathrm{KB}, \big((p \vee q) \wedge \neg\mathsf{FALSE} \wedge \neg\mathsf{FALSE}\big)]}_{\mathrm{KB}\models(p\vee q)\wedge\neg\mathsf{FALSE}\wedge\neg\mathsf{FALSE}?}$$

Example

$$\|\mathbf{K}\,\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathsf{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$$
$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \text{“}\mathrm{KB} \models \phi\text{?”}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} (p \vee q \vee r) \wedge (p \vee q \vee \neg r)$.

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)$$

$$\iff \quad \models \|\mathbf{K}\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}$$

$$\iff \quad \models \mathsf{RES}[\mathrm{KB}, \|\big((p \vee q) \wedge \neg\mathbf{K}p \wedge \neg\mathbf{K}q\big)\|_{\mathrm{KB}}]$$

$$\iff \quad \models \mathsf{RES}[\mathrm{KB}, \big((p \vee q) \wedge \neg \underbrace{\|\mathbf{K}p\|_{\mathrm{KB}}}_{\mathrm{KB}\models p?} \wedge \neg \underbrace{\|\mathbf{K}q\|_{\mathrm{KB}}}_{\mathrm{KB}\models q?}\big)]$$

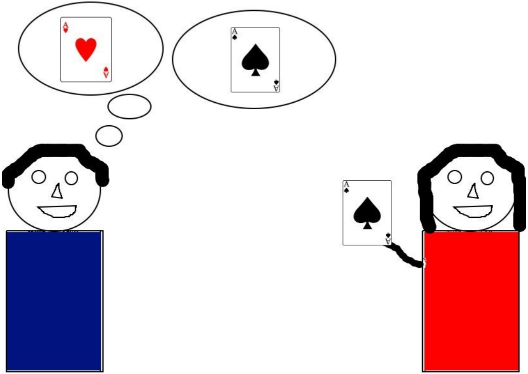
$$\iff \quad \models \underbrace{\mathsf{RES}[\mathrm{KB}, \big((p \vee q) \wedge \neg\mathsf{FALSE} \wedge \neg\mathsf{FALSE}\big)]}_{\mathrm{KB}\models(p\vee q)\wedge\neg\mathsf{FALSE}\wedge\neg\mathsf{FALSE}?}$$

$$\iff \quad \models \mathsf{TRUE} \quad \checkmark$$

Overview of the Lecture

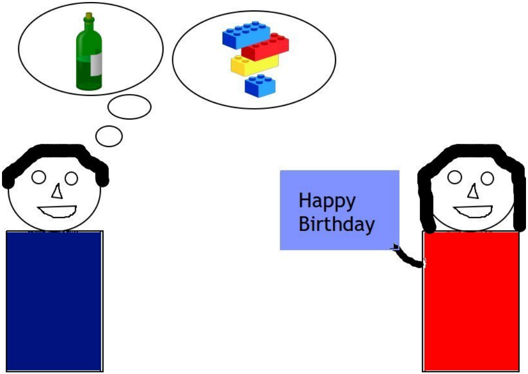
- A Logic of Knowledge – The Propositional Fragment
- **A Logic of Knowledge – The First-Order Case**
  - Why first-order logic?
  - Syntax and semantics
  - Knowing that vs knowing what
  - Representation theorem
- Extensions of the Logic of Knowledge

Why Is $\mathcal{OL}_{\mathrm{PL}}$ Not Enough?



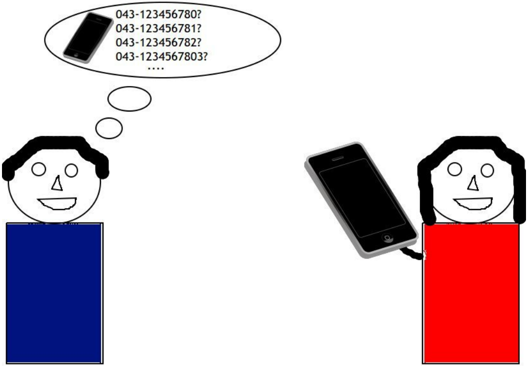
$$\mathbf{K}\big((\spadesuit \vee \heartsuit) \wedge \neg\mathbf{K}\spadesuit \wedge \neg\mathbf{K}\heartsuit\big)$$

Why Is $\mathcal{OL}_{\mathrm{PL}}$ Not Enough?



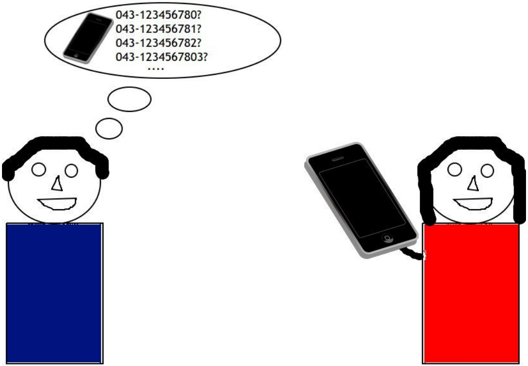
$$\mathbf{K}\exists x\,\big(\mathrm{InBox}(x) \land \neg\mathbf{K}\,\mathrm{InBox}(x)\big)$$

Why Is $\mathcal{OL}_{\text{PL}}$ Not Enough?



$$\mathbf{K}\exists x\left(\text{numberOf}(\text{Jane}) = x \land \neg\mathbf{K}\,\text{numberOf}(\text{Jane}) = x\right)$$

Why Is $\mathcal{OL}_{\mathrm{PL}}$ Not Enough?



$$\mathbf{K}\exists x\left(\mathrm{numberOf}(\mathrm{Jane}) = x \land \neg\mathbf{K}\,\mathrm{numberOf}(\mathrm{Jane}) = x\right)$$

"all" or "some" $\Longrightarrow$ first-order quantification

The Language of $\mathcal{OL}$

Terms:

- $x, x', x_1, x_2, \ldots$ — first-order variables
- $^\#1, ^\#2, ^\#3, \ldots$ — standard names
- $f(t_1, \ldots, t_j)$ — functions

Formulas:

- $P(t_1, \ldots, t_j)$ — atomic formulas
- $t_1 = t_2$ — equality expressions
- $\exists x\, \alpha$ — "for some x, α"
- $\forall x\, \alpha$ — "for all x, α"
- $\neg\alpha \quad (\alpha \vee \beta) \quad (\alpha \wedge \beta) \quad (\alpha \rightarrow \beta) \quad (\alpha \leftrightarrow \beta) \quad \mathbf{K}\alpha \quad \mathbf{O}\alpha$

The Language of $\mathcal{OL}$

Terms:

- $x, x', x_1, x_2, \ldots$ first-order variables
- ${}^{\#}1, {}^{\#}2, {}^{\#}3, \ldots$ standard names
- $f(t_1, \ldots, t_j)$ functions

Formulas:

- $P(t_1, \ldots, t_j)$ atomic formulas
- $t_1 = t_2$ equality expressions
- $\exists x\, \alpha$ "for some x, α"
- $\forall x\, \alpha \stackrel{\text{def}}{=} \neg \exists x \neg \alpha$ "for all x, α"
- $\neg\alpha \quad (\alpha \vee \beta) \quad (\alpha \wedge \beta) \quad (\alpha \rightarrow \beta) \quad (\alpha \leftrightarrow \beta) \quad \mathbf{K}\alpha \quad \mathbf{O}\alpha$

Why Standard Names?

- Consider in classical logic:
  $\text{fatherOf}(\text{Sally}) = \text{bestFriend}(\text{Jane}) \wedge$
  $\text{fatherOf}(\text{Sally}) = \text{bossOf}(\text{John})$
  - Who is father of Sally?
  - "Jane's best friend" is not a good answer
  - "John's boss" is not a good answer
  - Classical logic offers no way of identifying him
  - Reason: interpretations $\langle D, \Phi \rangle$ have different domains

- Standard names correspond to an implicit infinite domain

- Standard names allow to *identify* individuals in formulas:
  $\text{fatherOf}(\text{Sally}) = \text{Frank}$

The Semantics of $\mathcal{OL}$ (1)

Definition: semantics of $\mathcal{OL}$ (1)

$f(\vec{n})$ or $P(\vec{n})$ are **primitive** iff all n_i are standard names.
A term or a formula is **ground** iff it mentions no variable.

The Semantics of $\mathcal{OL}$ (1)

Definition: semantics of $\mathcal{OL}$ (1)

$f(\vec{n})$ or $P(\vec{n})$ are **primitive** iff all n_i are standard names.
A term or a formula is **ground** iff it mentions no variable.

A **world** w is a function that maps

- primitive functions $f(\vec{n})$ to standard names
- primitive atomic formulas $P(\vec{n})$ to $\{0, 1\}$

The Semantics of $\mathcal{OL}$ (1)

Definition: semantics of $\mathcal{OL}$ (1)

$f(\vec{n})$ or $P(\vec{n})$ are **primitive** iff all n_i are standard names.
A term or a formula is **ground** iff it mentions no variable.

A **world** w is a function that maps

- primitive functions $f(\vec{n})$ to standard names
- primitive atomic formulas $P(\vec{n})$ to $\{0, 1\}$

The **denotation** of a ground term w.r.t. w is defined as

- $w(n) \stackrel{\text{def}}{=} n$ for every standard name n
- $w(f(t_1, \ldots, t_j)) \stackrel{\text{def}}{=} w[f(w(t_1), \ldots, w(t_j))]$

The Semantics of $\mathcal{OL}$ (1)

Definition: semantics of $\mathcal{OL}$ (1)

$f(\vec{n})$ or $P(\vec{n})$ are **primitive** iff all n_i are standard names.
A term or a formula is **ground** iff it mentions no variable.

A **world** w is a function that maps

- primitive functions $f(\vec{n})$ to standard names
- primitive atomic formulas $P(\vec{n})$ to $\{0, 1\}$

The **denotation** of a ground term w.r.t. w is defined as

- $w(n) \stackrel{\text{def}}{=} n$ for every standard name n
- $w(f(t_1, \ldots, t_j)) \stackrel{\text{def}}{=} w[f(w(t_1), \ldots, w(t_j))]$

E.g., if Frank is Mia's father and Mia is Jane's mother:

$$\begin{aligned} & w(\text{fatherOf}(\text{motherOf}(\text{Jane}))) \\ = {} & w[\text{fatherOf}(w[\text{motherOf}(\text{Jane})])] \\ = {} & w[\text{fatherOf}(\text{Mia})] \\ = {} & \text{Frank}. \end{aligned}$$

The Semantics of $\mathcal{OL}$ (2)

Definition: semantics of $\mathcal{OL}$

An **epistemic state** e is a set of worlds.

- $e, w \models P(t_1, \ldots, t_j) \iff w[P(w(t_1), \ldots, w(t_j))] = 1$
- $e, w \models t_1 = t_2 \iff w(t_1) = w(t_2)$

The Semantics of $\mathcal{OL}$ (2)

Definition: semantics of $\mathcal{OL}$

An **epistemic state** e is a set of worlds.

- $e, w \models P(t_1, \ldots, t_j) \iff w[P(w(t_1), \ldots, w(t_j))] = 1$
- $e, w \models t_1 = t_2 \iff w(t_1) = w(t_2)$
- $e, w \models \neg\alpha \iff e, w \not\models \alpha$
- $e, w \models (\alpha \vee \beta) \iff e, w \models \alpha$ or $e, w \models \beta$

The Semantics of $\mathcal{OL}$ (2)

Definition: semantics of $\mathcal{OL}$

An **epistemic state** e is a set of worlds.

- $e, w \models P(t_1, \ldots, t_j) \iff w[P(w(t_1), \ldots, w(t_j))] = 1$
- $e, w \models t_1 = t_2 \iff w(t_1) = w(t_2)$
- $e, w \models \neg\alpha \iff e, w \not\models \alpha$
- $e, w \models (\alpha \vee \beta) \iff e, w \models \alpha$ or $e, w \models \beta$
- $e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n$ for some standard name n

The Semantics of $\mathcal{OL}$ (2)

Definition: semantics of $\mathcal{OL}$

An **epistemic state** e is a set of worlds.

- $e, w \models P(t_1, \ldots, t_j) \iff w[P(w(t_1), \ldots, w(t_j))] = 1$
- $e, w \models t_1 = t_2 \iff w(t_1) = w(t_2)$
- $e, w \models \neg\alpha \iff e, w \not\models \alpha$
- $e, w \models (\alpha \vee \beta) \iff e, w \models \alpha$ or $e, w \models \beta$
- $e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n$ for some standard name n
- $e, w \models \mathbf{K}\alpha \iff$ for all worlds w', $w' \in e \Rightarrow e, w' \models \alpha$
- $e, w \models \mathbf{O}\alpha \iff$ for all worlds w', $w' \in e \Leftrightarrow e, w' \models \alpha$

Knowing That vs Knowing What

- $\mathbf{K}\exists x\,\mathrm{Secret}(x)$ I know *that* some x is a secret
- $\exists x\,\mathbf{K}\,\mathrm{Secret}(x)$ I know *which* x is a secret

- $\mathbf{K}\exists x\,\mathrm{fatherOf}(\mathrm{Sally}) = x$ I know *that* Sally has a father
- $\exists x\,\mathbf{K}\,\mathrm{fatherOf}(\mathrm{Sally}) = x$ I know *who* Sally's father is

- $\mathbf{K}\exists x\,\alpha$ = *de dicto* knowledge
- $\exists x\,\mathbf{K}\,\alpha$ = *de re* knowledge

Theorem: quantifying-in

$\models \forall x\,\mathbf{K}\,\alpha \leftrightarrow \mathbf{K}\forall x\,\alpha$
$\models \exists x\,\mathbf{K}\,\alpha \rightarrow \mathbf{K}\exists x\,\alpha$
$\not\models \mathbf{K}\exists x\,\alpha \rightarrow \exists x\,\mathbf{K}\,\alpha$

Some Properties Inherited From $\mathcal{OL}_{\text{PL}}$

Definition: subjective, objective

If ϕ mentions no fun/pred inside $\mathbf{K}$ or $\mathbf{O}$, we say ϕ is **objective**.
If σ mentions fun/pred only inside $\mathbf{K}$ or $\mathbf{O}$, we say σ is **subjective**.

Theorem: logical omniscience

If $\models \alpha \to \beta$, then $\models \mathbf{K}\alpha \to \mathbf{K}\beta$.
If $\models \alpha$, then $\models \mathbf{K}\alpha$.

Theorem: unique-model property

Let ϕ be objective. Then there is a unique e such that $e \models \mathbf{O}\phi$.

Theorem: positive and negative introspection

Positive introspection: $\models \mathbf{K}\alpha \to \mathbf{K}\mathbf{K}\alpha$
Negative introspection: $\models \neg\mathbf{K}\alpha \to \mathbf{K}\neg\mathbf{K}\alpha$

Example

$$e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n \text{ for some standard name } n$$
$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', w \in e \Rightarrow e, w' \models \alpha$$
$$e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w', w \in e \Leftrightarrow e, w' \models \alpha$$

Let KB $\stackrel{\text{def}}{=} \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

Example

$$\begin{array}{l} e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n \text{ for some standard name } n \\ e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', w \in e \Rightarrow e, w' \models \alpha \\ e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w', w \in e \Leftrightarrow e, w' \models \alpha \end{array}$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} \exists x\, (x \neq \#1 \wedge P(x))$

- $e \models \mathbf{O}\mathrm{KB}$

Example

$$e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n \text{ for some standard name } n$$
$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\ w \in e \Rightarrow e, w' \models \alpha$$
$$e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w',\ w \in e \Leftrightarrow e, w' \models \alpha$$

Let KB $\stackrel{\text{def}}{=} \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

- $e \models \mathbf{O}\mathrm{KB}$

$\iff \quad w \in e \Leftrightarrow w \models \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

$\iff \quad w \in e \Leftrightarrow w[P(n)] = 1$ for some $n \in \{{}^{\#}2, {}^{\#}3, \ldots\}$

Example

$$e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n \text{ for some standard name } n$$
$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w', w \in e \Rightarrow e, w' \models \alpha$$
$$e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w', w \in e \Leftrightarrow e, w' \models \alpha$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

- $e \models \mathbf{O}\mathrm{KB}$

  $\iff w \in e \Leftrightarrow w \models \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

  $\iff w \in e \Leftrightarrow w[P(n)] = 1$ for some $n \in \{{}^{\#}2, {}^{\#}3, \ldots\}$

- $e \models \mathbf{K}\exists x\, (P(x) \wedge \neg\mathbf{K}P(x))$

Example

$$
\boxed{\begin{array}{l}
e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n \text{ for some standard name } n \\
e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\, w \in e \Rightarrow e, w' \models \alpha \\
e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w',\, w \in e \Leftrightarrow e, w' \models \alpha
\end{array}}
$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} \exists x\,(x \neq {}^{\#}1 \wedge P(x))$

- $e \models \mathbf{O}\mathrm{KB}$

  $\iff \quad w \in e \Leftrightarrow w \models \exists x\,(x \neq {}^{\#}1 \wedge P(x))$

  $\iff \quad w \in e \Leftrightarrow w[P(n)] = 1$ for some $n \in \{{}^{\#}2, {}^{\#}3, \ldots\}$

- $e \models \mathbf{K}\exists x\,(P(x) \wedge \neg\mathbf{K}P(x))$

  $\iff \quad$ for all w, $\;w \in e \Rightarrow$ for some n, $\;e, w \models P(n) \wedge \neg\mathbf{K}P(n)$

Example

$$e, w \models \exists x\, \alpha \iff e, w \models \alpha^x_n \text{ for some standard name } n$$
$$e, w \models \mathbf{K}\alpha \iff \text{for all worlds } w',\, w \in e \Rightarrow e, w' \models \alpha$$
$$e, w \models \mathbf{O}\alpha \iff \text{for all worlds } w',\, w \in e \Leftrightarrow e, w' \models \alpha$$

Let $\mathrm{KB} \stackrel{\text{def}}{=} \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

- $e \models \mathbf{O}\mathrm{KB}$

  $\iff \quad w \in e \Leftrightarrow w \models \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

  $\iff \quad w \in e \Leftrightarrow w[P(n)] = 1$ for some $n \in \{{}^{\#}2, {}^{\#}3, \ldots\}$

- $e \models \mathbf{K}\exists x\, (P(x) \wedge \neg \mathbf{K}P(x))$

  $\iff \quad$ for all w, $\; w \in e \Rightarrow$ for some n, $\; e, w \models P(n) \wedge \neg \mathbf{K}P(n)$

  $\iff \quad$ for all w, $\; w \in e \Rightarrow$ for some n, $\; e, w \models P(n)$ and $e, w \models \neg \mathbf{K}P(n)$

Example

$$
\begin{aligned}
e, w \models \exists x\, \alpha &\iff e, w \models \alpha^x_n \text{ for some standard name } n \\
e, w \models \mathbf{K}\alpha &\iff \text{for all worlds } w',\ w \in e \Rightarrow e, w' \models \alpha \\
e, w \models \mathbf{O}\alpha &\iff \text{for all worlds } w',\ w \in e \Leftrightarrow e, w' \models \alpha
\end{aligned}
$$

Let KB $\stackrel{\text{def}}{=} \exists x\, (x \neq {}^{\#}1 \wedge P(x))$

- $e \models \mathbf{O}\text{KB}$

$$
\begin{aligned}
&\iff w \in e \Leftrightarrow w \models \exists x\, (x \neq {}^{\#}1 \wedge P(x)) \\
&\iff w \in e \Leftrightarrow w[P(n)] = 1 \text{ for some } n \in \{{}^{\#}2, {}^{\#}3, \ldots\}
\end{aligned}
$$

- $e \models \mathbf{K}\exists x\, (P(x) \wedge \neg\mathbf{K}P(x))$

$$
\begin{aligned}
&\iff \text{for all } w,\ \ w \in e \Rightarrow \text{for some } n,\ \ e, w \models P(n) \wedge \neg\mathbf{K}P(n) \\
&\iff \text{for all } w,\ \ w \in e \Rightarrow \text{for some } n,\ \ e, w \models P(n) \text{ and } e, w \models \neg\mathbf{K}P(n) \\
&\iff \text{for all } w,\ \ w \in e \Rightarrow \text{for some } n,\ \ e, w \models P(n) \text{ and} \\
&\qquad\qquad \text{for some } w',\ \ w' \in e \text{ and } e, w' \not\models P(n) \\
&\iff \text{for all } w,\ \ w \in e \Rightarrow \text{for some } n,\ \ w[P(n)] = 1 \text{ and} \\
&\qquad\qquad \text{for some } w',\ \ w' \in e \text{ and } w'[P(n)] \neq 1
\end{aligned}
$$

Comparison with Tarski Semantics

- Traditional FOL semantics
  - Interpretation $\langle D, \Phi \rangle$ plus variable mapping μ
  - $\langle D, \Phi \rangle, \mu \models P(t_1, \ldots, t_j) \iff \langle d_1, \ldots, d_j \rangle \in \Phi(P)$
    where $d_i = \langle D, \Phi \rangle, \mu \| t_i \|$
  - $\langle D, \Phi \rangle, \mu \models \exists x\, \alpha \iff \langle D, \Phi \rangle, \mu^x_d \models \alpha$ for some $d \in D$
  - Purpose: reason about mathematics
  - Disadvantage: cumbersome to work with

- Our semantics
  - World maps primitive functions to names, predicates to $\{0, 1\}$
  - $w \models P(t_1, \ldots, t_j) \iff w[P(n_1, \ldots, n_j)] = 1$ where $n_i = w(t_i)$
  - $w \models \exists x\, \alpha \iff w \models \alpha^x_n$ for some standard name n
  - Purpose: reason about knowledge
  - Disadvantage: domain is always countably infinite
    - $\forall x\, (x = t_1 \vee \ldots \vee x = t_j)$ asserts finite domain in classical FOL
    - $\forall x\, (x = t_1 \vee \ldots \vee x = t_j)$ is unsatisfiable in $\mathcal{OL}$
    - but can be simulated with predicate: $\forall x\, (P(x) \leftrightarrow (x = t_1 \vee \ldots \vee x = t_j))$
    - classical FOL cannot distinguish countably infinite from uncountably infinite domains anyway

Representation Theorem (1)

$$\mathbf{O}\text{KB} \models \exists x\, \mathbf{K} P(x)?$$

How can we represent the known instances of an objective formula?

- $\text{KB} \stackrel{\text{def}}{=} (P(\#1) \land P(\#2))$ — $\#1, \#2$ are known P-instances
- $\text{KB} \stackrel{\text{def}}{=} (P(\#1) \lor P(\#2))$ — no known P-instances
- $\text{KB} \stackrel{\text{def}}{=} \forall x P(x)$ — all names are known P-instances
- $\text{KB} \stackrel{\text{def}}{=} \forall x\, (x \neq \#1 \to P(x))$ — $\#2, \#3, \ldots$ are known P-instances
- $\text{KB} \stackrel{\text{def}}{=} (Q(\#1) \land \forall x\, (Q(x) \to P(x))$ — $\#1$ is known P-instance

Representation Theorem (1)

$$\mathbf{O}\mathrm{KB} \models \exists x\, \mathbf{K} P(x)?$$

How can we represent the known instances of an objective formula?

- $\mathrm{KB} \stackrel{\text{def}}{=} (P(\#1) \land P(\#2))$ — $\#1, \#2$ are known P-instances
- $\mathrm{KB} \stackrel{\text{def}}{=} (P(\#1) \lor P(\#2))$ — no known P-instances
- $\mathrm{KB} \stackrel{\text{def}}{=} \forall x P(x)$ — all names are known P-instances
- $\mathrm{KB} \stackrel{\text{def}}{=} \forall x\, (x \neq \#1 \rightarrow P(x))$ — $\#2, \#3, \ldots$ are known P-instances
- $\mathrm{KB} \stackrel{\text{def}}{=} (Q(\#1) \land \forall x\, (Q(x) \rightarrow P(x))$ — $\#1$ is known P-instance

Let $n_1, \ldots, n_j$ be names in KB and let n' be a new one.

$$\begin{aligned} \mathrm{RES}[\mathrm{KB}, P(x)] \stackrel{\text{def}}{=}\ & (x = n_1 \land \text{“}\mathrm{KB} \models P(n_1)\text{”}) \lor \\ & \ldots \\ & (x = n_j \land \text{“}\mathrm{KB} \models P(n_j)\text{”}) \lor \\ & (x \neq n_1 \land \ldots \land x \neq n_j \land \text{“}\mathrm{KB} \models P(n')\text{”}) \end{aligned}$$

Representation Theorem (1)

$$\mathbf{O}\mathrm{KB} \models \exists x\, \mathbf{K} P(x)?$$

How can we represent the known instances of an objective formula?

$\mathrm{KB} \stackrel{\text{def}}{=} (P(\#1) \land P(\#2))$	$x = \#1 \lor x = \#2$
$\mathrm{KB} \stackrel{\text{def}}{=} (P(\#1) \lor P(\#2))$	FALSE
$\mathrm{KB} \stackrel{\text{def}}{=} \forall x P(x)$	TRUE
$\mathrm{KB} \stackrel{\text{def}}{=} \forall x\, (x \neq \#1 \rightarrow P(x))$	$x \neq \#1$
$\mathrm{KB} \stackrel{\text{def}}{=} (Q(\#1) \land \forall x\, (Q(x) \rightarrow P(x))$	$x = \#1$

Let $n_1, \ldots, n_j$ be names in KB and let n' be a new one.

$$\begin{aligned}
\text{RES}[\mathrm{KB}, P(x)] \;\stackrel{\text{def}}{=}\; & (x = n_1 \land \text{“}\mathrm{KB} \models P(n_1)\text{”}) \lor \\
& \ldots \\
& (x = n_j \land \text{“}\mathrm{KB} \models P(n_j)\text{”}) \lor \\
& (x \neq n_1 \land \ldots \land x \neq n_j \land \text{“}\mathrm{KB} \models P(n')\text{”})
\end{aligned}$$

Representation Theorem (1)

$$\mathbf{O}\mathrm{KB} \models \exists x\, \mathbf{K} P(x)?$$

How can we represent the known instances of an objective formula?

- $\mathrm{KB} \stackrel{\text{def}}{=} (P(\#1) \land P(\#2))$ — $x = \#1 \lor x = \#2$
- $\mathrm{KB} \stackrel{\text{def}}{=} (P(\#1) \lor P(\#2))$ — FALSE
- $\mathrm{KB} \stackrel{\text{def}}{=} \forall x P(x)$ — TRUE
- $\mathrm{KB} \stackrel{\text{def}}{=} \forall x (x \neq \#1 \rightarrow P(x))$ — $x \neq \#1$
- $\mathrm{KB} \stackrel{\text{def}}{=} (Q(\#1) \land \forall x (Q(x) \rightarrow P(x))$ — $x = \#1$

Let $n_1, \ldots, n_j$ be names in KB and let n' be a new one.

$$\begin{aligned}\mathrm{RES}[\mathrm{KB}, P(x)] \stackrel{\text{def}}{=} \; & (x = n_1 \land \text{“}\mathrm{KB} \models P(n_1)\text{”}) \lor \\ & \ldots \\ & (x = n_j \land \text{“}\mathrm{KB} \models P(n_j)\text{”}) \lor \\ & (x \neq n_1 \land \ldots \land x \neq n_j \land \text{“}\mathrm{KB} \models P(n')\text{”})\end{aligned}$$

Representation Theorem (2)

Definition: representation of known instances

If ϕ has a free variable x and $n_1, \ldots, n_j$ are the names mentioned in KB, ϕ, and n' is a new name:

$$\begin{aligned}\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \; & (x = n_1 \wedge \mathsf{RES}[\mathrm{KB}, \phi^x_{n_1}]) \vee \\ & \ldots \\ & (x = n_j \wedge \mathsf{RES}[\mathrm{KB}, \phi^x_{n_j}]) \vee \\ & (x \neq n_1 \wedge \ldots \wedge x \neq n_j \wedge \mathsf{RES}[\mathrm{KB}, \phi^x_{n'}]^{n'}_x)\end{aligned}$$

If ϕ has no free variables:

$$\mathsf{RES}[\mathrm{KB}, \phi] \stackrel{\text{def}}{=} \begin{cases} \mathsf{TRUE} & \text{if } \mathrm{KB} \models \phi \\ \mathsf{FALSE} & \text{otherwise} \end{cases}$$

Representation Theorem (3)

$\|\cdot\|$ operator gets a rule for $\exists x\, \alpha$:

Definition: representation operators

- $\|\phi\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \phi$ for objective ϕ
- $\|\neg\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \neg\|\alpha\|_{\mathrm{KB}}$
- $\|(\alpha \vee \beta)\|_{\mathrm{KB}} \stackrel{\text{def}}{=} (\|\alpha\|_{\mathrm{KB}} \vee \|\beta\|_{\mathrm{KB}})$
- $\|\exists x\, \alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \exists x\, \|\alpha\|_{\mathrm{KB}}$
- $\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=}$ RES$[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$

Representation Theorem (3)

$\| \cdot \|$ operator gets a rule for $\exists x\, \alpha$:

Definition: representation operators

- $\|\phi\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \phi$ for objective ϕ
- $\|\neg\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \neg\|\alpha\|_{\mathrm{KB}}$
- $\|(\alpha \vee \beta)\|_{\mathrm{KB}} \stackrel{\text{def}}{=} (\|\alpha\|_{\mathrm{KB}} \vee \|\beta\|_{\mathrm{KB}})$
- $\|\exists x\, \alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \exists x\, \|\alpha\|_{\mathrm{KB}}$
- $\|\mathbf{K}\alpha\|_{\mathrm{KB}} \stackrel{\text{def}}{=} \mathrm{RES}[\mathrm{KB}, \|\alpha\|_{\mathrm{KB}}]$

Theorem: representation theorem

$\mathbf{O}\mathrm{KB} \models \alpha \iff \models \|\alpha\|_{\mathrm{KB}}.$

Overview of the Lecture

- A Logic of Knowledge – The Propositional Fragment
- A Logic of Knowledge – The First-Order Case
- **Extensions of the Logic of Knowledge**
  - Multiple agents
  - Probabilities
  - Conditional belief
  - Limited Belief (week 8)
  - Actions (week 9)

Multi-Agent Belief

Mike does not know what is in the gift box, but he knows that Jane knows what is in there:

$$\mathbf{K}_{\text{Mike}} \exists x \left(\text{InBox}(x) \land \neg \mathbf{K}_{\text{Mike}} \text{InBox}(x) \land \mathbf{K}_{\text{Jane}} \text{InBox}(x) \right)$$

Epistemic states get more complex: in every possible world, Mike considers a whole set of worlds to be possible from Jane's perspective.

Probabilities

I believe that with probability .999, there is no bomb in the gift box:

$$\mathbf{B}\left(\neg\exists x\left(\mathrm{InBox}(x) \wedge \mathrm{Bomb}(x)\right) : 0.999\right)$$

An epistemic state is now probability distribution over possible worlds.

Conditional Belief

I believe that if something is in the gift box, it's probably not a bomb:

$$\mathbf{B}\left(\exists x\, \mathrm{InBox}(x) \Rightarrow \neg \mathrm{Bomb}(x)\right)$$

Epistemic state ranks possible worlds by plausibility and checks if the most-plausible worlds where $\exists x\, \mathrm{InBox}(x)$ is true also satisfy $\mathrm{Bomb}(x)$.

- A knowledge base is now a collection of conditionals
  "if ____, then most likely ____"
- What sort of ranking should these conditionals induce?