Knowledge Representation and Reasoning

Part 2: Propositional Languages

Ivan Varzinczak

LIASD, Université Paris 8, France

https://www.ijv.ovh

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Last time

Agents, systems, and the agents' goals

- Agent

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation
- Information

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation
- Information
- Semantics

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation
- Information
- Semantics
- State

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation
- Information
- Semantics
- State
- Symbolic representation

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation
- Information
- Semantics
- State
- Symbolic representation
- System

Last time

Agents, systems, and the agents' goals

- Agent
- Atomic proposition
- Candidate state
- Connective
- Default rule
- Evidence
- Fixed information
- Iconic representation
- Information
- Semantics
- State
- Symbolic representation
- System
- Truth value

Outline

Opaque propositional languages

Semantics

Outline

Opaque propositional languages

Semantics

Syntax

Formal languages

- A formal language is defined recursively
- But, what does this mean?
- Start with basic building blocs and a set of connectives (operators)
- Apply combination rules to build longer, more complex sentences
- Important: only a finite number of applications of the combination rules

Syntax

Formal languages

- A formal language is defined recursively
- But, what does this mean?
- Start with basic building blocs and a set of connectives (operators)
- Apply combination rules to build longer, more complex sentences
- Important: only a finite number of applications of the combination rules

Practical questions

- How big is a language?

Syntax

Formal languages

- A formal language is defined recursively
- But, what does this mean?
- Start with basic building blocs and a set of connectives (operators)
- Apply combination rules to build longer, more complex sentences
- Important: only a finite number of applications of the combination rules

Practical questions

- How big is a language?
- How to handle a possibly infinite language?

Syntax

Formal languages

- A formal language is defined recursively
- But, what does this mean?
- Start with basic building blocs and a set of connectives (operators)
- Apply combination rules to build longer, more complex sentences
- Important: only a finite number of applications of the combination rules

Practical questions

- How big is a language?
- How to handle a possibly infinite language?
- Infinity can often be described finitely

Syntax

Formal languages

- A formal language is defined recursively
- But, what does this mean?
- Start with basic building blocs and a set of connectives (operators)
- Apply combination rules to build longer, more complex sentences
- Important: only a finite number of applications of the combination rules

Practical questions

- How big is a language?
- How to handle a possibly infinite language?
- Infinity can often be described finitely
- In the case of formal languages, we specify a (finite) grammar

Syntax

Definition (Propositional Sentence)

Let $\mathcal{P} \subseteq \{p_0, p_1, \ldots\}$ be a set of propositional atoms, and $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
We say that α is a sentence over $\mathcal{P}$ if one of the following is the case:

- $\alpha = p$ for some $p \in \mathcal{P}$ (note we use $p, q, \ldots$ as meta-variables)
- $\alpha = (\neg\beta)$ for some previously constructed sentence β over $\mathcal{P}$
- $\alpha = (\beta * \gamma)$ where β and γ are previously constructed sentences over $\mathcal{P}$

Syntax

Definition (Propositional Sentence)

Let $\mathcal{P} \subseteq \{p_0, p_1, \ldots\}$ be a set of propositional atoms, and $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. We say that α is a sentence over $\mathcal{P}$ if one of the following is the case:

- $\alpha = p$ for some $p \in \mathcal{P}$ (note we use $p, q, \ldots$ as meta-variables)
- $\alpha = (\neg\beta)$ for some previously constructed sentence β over $\mathcal{P}$
- $\alpha = (\beta * \gamma)$ where β and γ are previously constructed sentences over $\mathcal{P}$

Definition (Language)

The set of all sentences over $\mathcal{P}$ is the language $\mathcal{L}_\mathcal{P}$ generated by $\mathcal{P}$ and $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ through a finite number of steps as above. If $\mathcal{P}$ is finite, we say $\mathcal{L}_\mathcal{P}$ is finitely generated.

Syntax

Example

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$

- p_{17} is a sentence over $\mathcal{P}$, $(\neg p_{17})$ is a sentence over $\mathcal{P}$
- $(p_{22} \wedge (\neg p_{17}))$ is a sentence over $\mathcal{P}$
- p_{222} is not a sentence over $\mathcal{P}$ (it contains an atom not in $\mathcal{P}$)
- $(\neg(\neg \cdots (\neg p_1) \cdots))$ is not a sentence over $\mathcal{P}$ (infinite symbols)

Syntax

Example

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$

- p_{17} is a sentence over $\mathcal{P}$, $(\neg p_{17})$ is a sentence over $\mathcal{P}$
- $(p_{22} \wedge (\neg p_{17}))$ is a sentence over $\mathcal{P}$
- p_{222} is not a sentence over $\mathcal{P}$ (it contains an atom not in $\mathcal{P}$)
- $(\neg(\neg \cdots (\neg p_1) \cdots))$ is not a sentence over $\mathcal{P}$ (infinite symbols)

Notation

- We can write $\neg\alpha$ instead of $(\neg\alpha)$
- We drop parentheses to write $(\alpha * \beta)$ as $\alpha * \beta$ for all $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$
- Order of precedence: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
- E.g. $p \vee q \wedge \neg r \rightarrow s$ is short for $(p \vee (q \wedge (\neg r))) \rightarrow s)$

Outline

What we mean by 'meaning'

Motivation

- Semantics is concerned with the meaning of words and sentences
- A language without meaning is useless
- Agents build up iconic representations which stand for sentences
- Meaning: the relationship between symbolic and iconic representations

What we mean by ‘meaning’

Motivation

- Semantics is concerned with the meaning of words and sentences
- A language without meaning is useless
- Agents build up iconic representations which stand for sentences
- Meaning: the relationship between symbolic and iconic representations

Truth values

- Indicate the fit between sentences and iconic representations
- A sentence is true if it is faithful to the state; it is false otherwise

What we mean by 'meaning'

Motivation

- Semantics is concerned with the meaning of words and sentences
- A language without meaning is useless
- Agents build up iconic representations which stand for sentences
- Meaning: the relationship between symbolic and iconic representations

Truth values

- Indicate the fit between sentences and iconic representations
- A sentence is true if it is faithful to the state; it is false otherwise

Valuations

- A valuation records the match between sentences and a state
- Each valuation is linked to the iconic representation of a state

Valuations

Definition (Propositional Valuation)

Let $\mathcal{P}$ be a set of atoms. A valuation over $\mathcal{P}$ is a function $v : \mathcal{P} \longrightarrow \{0, 1\}$. The set of all valuations over $\mathcal{P}$ is denoted $\mathcal{U}_{\mathcal{P}}$.

Valuations

Definition (Propositional Valuation)

Let $\mathcal{P}$ be a set of atoms. A valuation over $\mathcal{P}$ is a function $v : \mathcal{P} \longrightarrow \{0, 1\}$. The set of all valuations over $\mathcal{P}$ is denoted $\mathcal{U}_{\mathcal{P}}$.

Example (Light-fan system)

$\mathcal{S} = \{00, 01, 10, 11\}$ (shorthand for the iconic representations)
$\mathcal{P} = \{p, q\}$ (shorthand for 'the light is on' and 'the fan is on')

- State 11 corresponds to the valuation v given by $v(p) = 1 = v(q)$
- State 10 corresponds to the valuation v' given by $v'(p) = 1$ and $v'(q) = 0$
- State 01 corresponds to v'' given by $v''(p) = 0$ and $v''(q) = 1$
- State 00 corresponds to v''' given by $v'''(p) = v'''(q) = 0$

Valuations

Definition (Propositional Valuation)

Let $\mathcal{P}$ be a set of atoms. A valuation over $\mathcal{P}$ is a function $v : \mathcal{P} \longrightarrow \{0, 1\}$. The set of all valuations over $\mathcal{P}$ is denoted $\mathcal{U}_{\mathcal{P}}$.

Example (Light-fan system)

$\mathcal{S} = \{00, 01, 10, 11\}$ (shorthand for the iconic representations)

$\mathcal{P} = \{p, q\}$ (shorthand for 'the light is on' and 'the fan is on')

- State 11 corresponds to the valuation v given by $v(p) = 1 = v(q)$
- State 10 corresponds to the valuation v' given by $v'(p) = 1$ and $v'(q) = 0$
- State 01 corresponds to v'' given by $v''(p) = 0$ and $v''(q) = 1$
- State 00 corresponds to v''' given by $v'''(p) = v'''(q) = 0$

Question: Given $\mathcal{P}$, how many valuations are there in $\mathcal{U}_{\mathcal{P}}$?

Valuations

Notation

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_n\}$. It is convenient to abbreviate each valuation $v : \mathcal{P} \longrightarrow \{0, 1\}$ as $v(p_0)v(p_1)\ldots v(p_n)$.

Valuations

Notation

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_n\}$. It is convenient to abbreviate each valuation $v : \mathcal{P} \longrightarrow \{0, 1\}$ as $v(p_0)v(p_1)\ldots v(p_n)$.

Example

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$

- Let $v : \mathcal{P} \longrightarrow \{0, 1\}$ be s.t. $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$
- We can abbreviate v by the binary string $101010\ldots10$

Valuations

Notation

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_n\}$. It is convenient to abbreviate each valuation $v : \mathcal{P} \longrightarrow \{0, 1\}$ as $v(p_0)v(p_1)\ldots v(p_n)$.

Example

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$

- Let $v : \mathcal{P} \longrightarrow \{0, 1\}$ be s.t. $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$
- We can abbreviate v by the binary string $101010\ldots10$

Let $\mathcal{P}' = \{p_0, p_1, \ldots\}$

- Let $v : \mathcal{P} \longrightarrow \{0, 1\}$ be s.t. $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$
- We can abbreviate v by the binary string $101010\ldots$

Valuations

Notation

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_n\}$. It is convenient to abbreviate each valuation $v : \mathcal{P} \longrightarrow \{0, 1\}$ as $v(p_0)v(p_1)\ldots v(p_n)$.

Example

Let $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$

- Let $v : \mathcal{P} \longrightarrow \{0, 1\}$ be s.t. $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$
- We can abbreviate v by the binary string $101010\ldots10$

Let $\mathcal{P}' = \{p_0, p_1, \ldots\}$

- Let $v : \mathcal{P} \longrightarrow \{0, 1\}$ be s.t. $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$
- We can abbreviate v by the binary string $101010\ldots$

Question: Is there any difference between a state and a valuation?

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$
- Hence $|\mathcal{U}_{\mathcal{P}}| = 2^9 = 512$ possible valuations

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$
- Hence $|\mathcal{U}_{\mathcal{P}}| = 2^9 = 512$ possible valuations
- How many states of the system are there?

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$
- Hence $|\mathcal{U}_{\mathcal{P}}| = 2^9 = 512$ possible valuations
- How many states of the system are there? Only $3 \cdot 2 \cdot 1 = 6$

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$
- Hence $|\mathcal{U}_{\mathcal{P}}| = 2^9 = 512$ possible valuations
- How many states of the system are there? Only $3 \cdot 2 \cdot 1 = 6$
- Indeed, $\mathcal{S} = \{\text{rgb}, \text{rbg}, \text{grb}, \text{gbr}, \text{brg}, \text{bgr}\}$

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$
- Hence $|\mathcal{U}_{\mathcal{P}}| = 2^9 = 512$ possible valuations
- How many states of the system are there? Only $3 \cdot 2 \cdot 1 = 6$
- Indeed, $\mathcal{S} = \{\text{rgb}, \text{rbg}, \text{grb}, \text{gbr}, \text{brg}, \text{bgr}\}$

As a result

- There are reasons for having a set of states $\mathcal{S}$ different from $\mathcal{U}_{\mathcal{P}}$
- There may be valuations corresponding to no possible state of the system

Difference between states and valuations

Example (The 3-card system)

Assume we have 3 players. Each player is dealt one of 3 cards coloured red, green or blue. A given deal corresponds to a state of the system.

- Let $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$
- Hence $|\mathcal{U}_{\mathcal{P}}| = 2^9 = 512$ possible valuations
- How many states of the system are there? Only $3 \cdot 2 \cdot 1 = 6$
- Indeed, $\mathcal{S} = \{\text{rgb}, \text{rbg}, \text{grb}, \text{gbr}, \text{brg}, \text{bgr}\}$

As a result

- There are reasons for having a set of states $\mathcal{S}$ different from $\mathcal{U}_{\mathcal{P}}$
- There may be valuations corresponding to no possible state of the system

Question: Can there be a state with no valuation associated to it?

Ontologies

Definition (Ontology for $\mathcal{L}_{\mathcal{P}}$)

Assume a system with states $\mathcal{S}$, propositional atoms $\mathcal{P}$, and valuations $\mathcal{U}_{\mathcal{P}}$, and let $\mathcal{V} : \mathcal{S} \longrightarrow \mathcal{U}_{\mathcal{P}}$ denote a labelling function. We call the pair $(\mathcal{S}, \mathcal{V})$ an ontology for the language $\mathcal{L}_{\mathcal{P}}$.

Ontologies

Definition (Ontology for $\mathcal{L}_\mathcal{P}$)

Assume a system with states $\mathcal{S}$, propositional atoms $\mathcal{P}$, and valuations $\mathcal{U}_\mathcal{P}$, and let $\mathcal{V} : \mathcal{S} \longrightarrow \mathcal{U}_\mathcal{P}$ denote a labelling function. We call the pair $(\mathcal{S}, \mathcal{V})$ an ontology for the language $\mathcal{L}_\mathcal{P}$.

Example (Light-fan system)

$\mathcal{S} = \{00, 01, 10, 11\}$, $\mathcal{P} = \{p, q\}$, and $\mathcal{U}_\mathcal{P} = \{00, 01, 10, 11\}$, and let $\mathcal{V}(\cdot)$ be s.t. $\mathcal{V}(00) = 00$, $\mathcal{V}(01) = 01$, $\mathcal{V}(10) = 10$, $\mathcal{V}(11) = 11$. Then $(\mathcal{S}, \mathcal{V})$ is an ontology for $\mathcal{L}_\mathcal{P}$.

Ontologies

Definition (Ontology for $\mathcal{L}_{\mathcal{P}}$)

Assume a system with states $\mathcal{S}$, propositional atoms $\mathcal{P}$, and valuations $\mathcal{U}_{\mathcal{P}}$, and let $\mathcal{V} : \mathcal{S} \longrightarrow \mathcal{U}_{\mathcal{P}}$ denote a labelling function. We call the pair $(\mathcal{S}, \mathcal{V})$ an ontology for the language $\mathcal{L}_{\mathcal{P}}$.

Example (Light-fan system)

$\mathcal{S} = \{00, 01, 10, 11\}$, $\mathcal{P} = \{p, q\}$, and $\mathcal{U}_{\mathcal{P}} = \{00, 01, 10, 11\}$, and let $\mathcal{V}(\cdot)$ be s.t. $\mathcal{V}(00) = 00$, $\mathcal{V}(01) = 01$, $\mathcal{V}(10) = 10$, $\mathcal{V}(11) = 11$. Then $(\mathcal{S}, \mathcal{V})$ is an ontology for $\mathcal{L}_{\mathcal{P}}$.

Example (The 3-card system)

$\mathcal{S} = \{\text{rgb}, \text{rbg}, \text{grb}, \text{gbr}, \text{brg}, \text{bgr}\}$, $\mathcal{P} = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$, and $\mathcal{U}_{\mathcal{P}} = \{000000000, 000000001, \ldots, 111111111\}$. Define an appropriate $\mathcal{V}(\cdot)$.

Satisfaction and models

Definition (Satisfaction)

Let $\alpha \in \mathcal{L}_{\mathcal{P}}$ and $v \in \mathcal{U}_{\mathcal{P}}$. We say v satisfies α, denoted $v \Vdash \alpha$, if one of the following is the case:

- $\alpha = p$ for some $p \in \mathcal{P}$ and $v(p) = 1$
- $\alpha = \neg\beta$ and $v \nVdash \beta$
- $\alpha = \beta \wedge \gamma$ and both $v \Vdash \beta$ and $v \Vdash \gamma$
- $\alpha = \beta \vee \gamma$ and either $v \Vdash \beta$ or $v \Vdash \gamma$ or both
- $\alpha = \beta \rightarrow \gamma$ and either $v \Vdash \gamma$ or $v \nVdash \beta$ or both
- $\alpha = \beta \leftrightarrow \gamma$ and either ($v \Vdash \beta$ and $v \Vdash \gamma$) or ($v \nVdash \beta$ and $v \nVdash \gamma$)

If $v \Vdash \alpha$, we say α is true relative to v, and false (relative to v) otherwise

Satisfaction and models

Definition (Satisfaction)

Let $\alpha \in \mathcal{L}_{\mathcal{P}}$ and $v \in \mathcal{U}_{\mathcal{P}}$. We say v satisfies α, denoted $v \Vdash \alpha$, if one of the following is the case:

- $\alpha = p$ for some $p \in \mathcal{P}$ and $v(p) = 1$
- $\alpha = \neg\beta$ and $v \nVdash \beta$
- $\alpha = \beta \wedge \gamma$ and both $v \Vdash \beta$ and $v \Vdash \gamma$
- $\alpha = \beta \vee \gamma$ and either $v \Vdash \beta$ or $v \Vdash \gamma$ or both
- $\alpha = \beta \rightarrow \gamma$ and either $v \Vdash \gamma$ or $v \nVdash \beta$ or both
- $\alpha = \beta \leftrightarrow \gamma$ and either $(v \Vdash \beta$ and $v \Vdash \gamma)$ or $(v \nVdash \beta$ and $v \nVdash \gamma)$

If $v \Vdash \alpha$, we say α is true relative to v, and false (relative to v) otherwise

For every α and every v, α is either true or false relative to v

Satisfaction and models

Definition (Models, Nonmodels, and Spurious Models)

Let $(\mathcal{S}, \mathcal{V})$ be an ontology for $\mathcal{L}_{\mathcal{P}}$.

- A state $s \in \mathcal{S}$ satisfies $\alpha \in \mathcal{L}_{\mathcal{P}}$, denoted $s \Vdash \alpha$, if $\mathcal{V}(s) \Vdash \alpha$
- If $s \Vdash \alpha$, we say s is a model of α

Satisfaction and models

Definition (Models, Nonmodels, and Spurious Models)

Let $(\mathcal{S}, \mathcal{V})$ be an ontology for $\mathcal{L}_{\mathcal{P}}$.

- A state $s \in \mathcal{S}$ satisfies $\alpha \in \mathcal{L}_{\mathcal{P}}$, denoted $s \Vdash \alpha$, if $\mathcal{V}(s) \Vdash \alpha$
- If $s \Vdash \alpha$, we say s is a model of α
- With $\mathcal{M}(\alpha) \subseteq \mathcal{S}$ we denote the set of all models of α

Satisfaction and models

Definition (Models, Nonmodels, and Spurious Models)

Let $(\mathcal{S}, \mathcal{V})$ be an ontology for $\mathcal{L}_{\mathcal{P}}$.

- A state $s \in \mathcal{S}$ satisfies $\alpha \in \mathcal{L}_{\mathcal{P}}$, denoted $s \Vdash \alpha$, if $\mathcal{V}(s) \Vdash \alpha$
- If $s \Vdash \alpha$, we say s is a model of α
- With $\mathcal{M}(\alpha) \subseteq \mathcal{S}$ we denote the set of all models of α
- With $\mathcal{N}(\alpha) \stackrel{\text{def}}{=} \overline{\mathcal{M}(\alpha)}$ we denote the set of nonmodels of α

Satisfaction and models

Definition (Models, Nonmodels, and Spurious Models)

Let $(\mathcal{S}, \mathcal{V})$ be an ontology for $\mathcal{L}_{\mathcal{P}}$.

- A state $s \in \mathcal{S}$ satisfies $\alpha \in \mathcal{L}_{\mathcal{P}}$, denoted $s \Vdash \alpha$, if $\mathcal{V}(s) \Vdash \alpha$
- If $s \Vdash \alpha$, we say s is a model of α
- With $\mathcal{M}(\alpha) \subseteq \mathcal{S}$ we denote the set of all models of α
- With $\mathcal{N}(\alpha) \stackrel{\text{def}}{=} \overline{\mathcal{M}(\alpha)}$ we denote the set of nonmodels of α
- If $v \Vdash \alpha$ but $v \neq \mathcal{V}(s)$ for any s, then v is a spurious model of α

Satisfaction and models

Definition (Models, Nonmodels, and Spurious Models)

Let $(\mathcal{S}, \mathcal{V})$ be an ontology for $\mathcal{L}_{\mathcal{P}}$.

- A state $s \in \mathcal{S}$ satisfies $\alpha \in \mathcal{L}_{\mathcal{P}}$, denoted $s \Vdash \alpha$, if $\mathcal{V}(s) \Vdash \alpha$
- If $s \Vdash \alpha$, we say s is a model of α
- With $\mathcal{M}(\alpha) \subseteq \mathcal{S}$ we denote the set of all models of α
- With $\mathcal{N}(\alpha) \stackrel{\text{def}}{=} \overline{\mathcal{M}(\alpha)}$ we denote the set of nonmodels of α
- If $v \Vdash \alpha$ but $v \neq \mathcal{V}(s)$ for any s, then v is a spurious model of α
- If $X \subseteq \mathcal{L}_{\mathcal{P}}$, then $s \Vdash X$ if $s \Vdash \alpha$ for every $\alpha \in X$

Satisfaction and models

Definition (Models, Nonmodels, and Spurious Models)

Let $(\mathcal{S}, \mathcal{V})$ be an ontology for $\mathcal{L}_{\mathcal{P}}$.

- A state $s \in \mathcal{S}$ satisfies $\alpha \in \mathcal{L}_{\mathcal{P}}$, denoted $s \Vdash \alpha$, if $\mathcal{V}(s) \Vdash \alpha$
- If $s \Vdash \alpha$, we say s is a model of α
- With $\mathcal{M}(\alpha) \subseteq \mathcal{S}$ we denote the set of all models of α
- With $\mathcal{N}(\alpha) \stackrel{\text{def}}{=} \overline{\mathcal{M}(\alpha)}$ we denote the set of nonmodels of α
- If $v \Vdash \alpha$ but $v \neq \mathcal{V}(s)$ for any s, then v is a spurious model of α
- If $X \subseteq \mathcal{L}_{\mathcal{P}}$, then $s \Vdash X$ if $s \Vdash \alpha$ for every $\alpha \in X$
- If $X \subseteq \mathcal{L}_{\mathcal{P}}$, then $\mathcal{M}(X) \stackrel{\text{def}}{=} \bigcap_{\alpha \in X} \mathcal{M}(\alpha)$

Satisfaction and models

Example (Light-fan system)

Consider $(\mathcal{S}, \mathcal{V})$, with $\mathcal{S} = \{00, 01, 10, 11\}$, $\mathcal{P} = \{p, q\}$, and $\mathcal{V}(00) = 00$, $\mathcal{V}(01) = 01$, $\mathcal{V}(10) = 10$, $\mathcal{V}(11) = 11$.

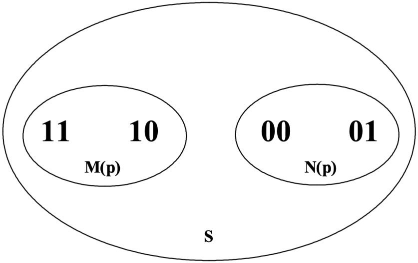
- Take $\alpha = p$
- Then we have $\mathcal{M}(\alpha) = \{11, 10\}$ and $\mathcal{N}(\alpha) = \{00, 01\}$

Satisfaction and models

Example (Light-fan system)

Consider $(\mathcal{S}, \mathcal{V})$, with $\mathcal{S} = \{00, 01, 10, 11\}$, $\mathcal{P} = \{p, q\}$, and $\mathcal{V}(00) = 00$, $\mathcal{V}(01) = 01$, $\mathcal{V}(10) = 10$, $\mathcal{V}(11) = 11$.

- Take $\alpha = p$
- Then we have $\mathcal{M}(\alpha) = \{11, 10\}$ and $\mathcal{N}(\alpha) = \{00, 01\}$

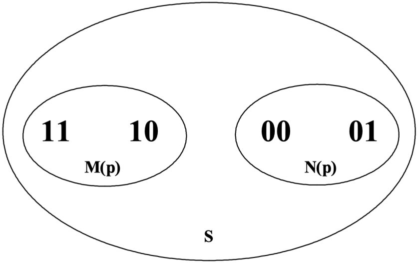


Satisfaction and models

Example (Light-fan system)

Consider $(\mathcal{S}, \mathcal{V})$, with $\mathcal{S} = \{00, 01, 10, 11\}$, $\mathcal{P} = \{p, q\}$, and $\mathcal{V}(00) = 00$, $\mathcal{V}(01) = 01$, $\mathcal{V}(10) = 10$, $\mathcal{V}(11) = 11$.

- Take $\alpha = p$
- Then we have $\mathcal{M}(\alpha) = \{11, 10\}$ and $\mathcal{N}(\alpha) = \{00, 01\}$
- Take $\alpha = p \rightarrow q$. $\mathcal{M}(\alpha) = ?$

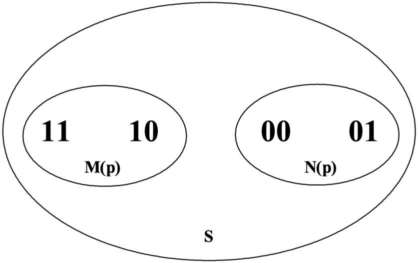


Satisfaction and models

Example (Light-fan system)

Consider $(\mathcal{S}, \mathcal{V})$, with $\mathcal{S} = \{00, 01, 10, 11\}$, $\mathcal{P} = \{p, q\}$, and $\mathcal{V}(00) = 00$, $\mathcal{V}(01) = 01$, $\mathcal{V}(10) = 10$, $\mathcal{V}(11) = 11$.

- Take $\alpha = p$
- Then we have $\mathcal{M}(\alpha) = \{11, 10\}$ and $\mathcal{N}(\alpha) = \{00, 01\}$
- Take $\alpha = p \rightarrow q$. $\mathcal{M}(\alpha) = \{00, 01, 11\}$



Satisfaction and models

Example

Assume $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$. Let s be a state s.t. $\mathcal{V}(s) = v$ and $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$.

- $s \Vdash p_{22}$, because $\mathcal{V}(s) = v$ and $v(p_{22}) = 1$
- $s \nVdash p_{23}$, because $\mathcal{V}(s) = v$ and $v(p_{23}) = 0$
- $s \Vdash p_{23} \rightarrow p_1$, because $\mathcal{V}(s) = v$ and $v \nVdash p_{23}$
- $s \nVdash p_{23} \leftrightarrow \neg p_1$, because $s \nVdash p_{23}$ but $s \Vdash \neg p_1$

Satisfaction and models

Example

Assume $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$. Let s be a state s.t. $\mathcal{V}(s) = v$ and $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$.

- $s \Vdash p_{22}$, because $\mathcal{V}(s) = v$ and $v(p_{22}) = 1$
- $s \nVdash p_{23}$, because $\mathcal{V}(s) = v$ and $v(p_{23}) = 0$
- $s \Vdash p_{23} \rightarrow p_1$, because $\mathcal{V}(s) = v$ and $v \nVdash p_{23}$
- $s \nVdash p_{23} \leftrightarrow \neg p_1$, because $s \nVdash p_{23}$ but $s \Vdash \neg p_1$
- Thus $s \in \mathcal{M}(p_{22})$, $s \notin \mathcal{M}(p_{23})$ so that $s \in \mathcal{N}(p_{23})$

Satisfaction and models

Example

Assume $\mathcal{P} = \{p_0, p_1, \ldots, p_{113}\}$. Let s be a state s.t. $\mathcal{V}(s) = v$ and $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$.

- $s \Vdash p_{22}$, because $\mathcal{V}(s) = v$ and $v(p_{22}) = 1$
- $s \nVdash p_{23}$, because $\mathcal{V}(s) = v$ and $v(p_{23}) = 0$
- $s \Vdash p_{23} \rightarrow p_1$, because $\mathcal{V}(s) = v$ and $v \nVdash p_{23}$
- $s \nVdash p_{23} \leftrightarrow \neg p_1$, because $s \nVdash p_{23}$ but $s \Vdash \neg p_1$
- Thus $s \in \mathcal{M}(p_{22})$, $s \notin \mathcal{M}(p_{23})$ so that $s \in \mathcal{N}(p_{23})$
- $s \in \mathcal{M}(p_{23} \rightarrow p_1)$, and $s \notin \mathcal{M}(p_{23} \leftrightarrow \neg p_1)$ so that $s \in \mathcal{N}(p_{23} \leftrightarrow \neg p_1)$

Satisfaction and models

Definition (Classes of sentences)

Let $\alpha \in \mathcal{L}_{\mathcal{P}}$.

- If α has at least one model, we say α is satisfiable
- If α has no model, we say α is unsatisfiable
- If α is satisfied by all states in $\mathcal{S}$, we say α is valid
- If α is satisfied by all valuations in $\mathcal{U}_{\mathcal{P}}$, we say α is a tautology
- If α is satisfied by some states but not by others, we say α is contingent

Satisfaction and models

Definition (Classes of sentences)

Let $\alpha \in \mathcal{L}_\mathcal{P}$.

- If α has at least one model, we say α is satisfiable
- If α has no model, we say α is unsatisfiable
- If α is satisfied by all states in $\mathcal{S}$, we say α is valid
- If α is satisfied by all valuations in $\mathcal{U}_\mathcal{P}$, we say α is a tautology
- If α is satisfied by some states but not by others, we say α is contingent

The problem of deciding whether a propositional sentence is satisfiable is known as the satisfiability problem (SAT)

Information and equivalence

Definition (Information)

The information about the system possessed by an agent is reflected by the selection of a set $\overline{X}$ of excluded states inside $\mathcal{S}$, leaving a complementary set X of included states. We say α expresses the agent's information if $\mathcal{M}(\alpha) = X$. We say $\mathcal{N}(\alpha) = \overline{X}$ is the information content of α.

Information and equivalence

Definition (Information)

The information about the system possessed by an agent is reflected by the selection of a set $\overline{X}$ of excluded states inside $\mathcal{S}$, leaving a complementary set X of included states. We say α expresses the agent's information if $\mathcal{M}(\alpha) = X$. We say $\mathcal{N}(\alpha) = \overline{X}$ is the information content of α.

Definition (Equivalence)

We say $\alpha, \beta \in \mathcal{L}_{\mathcal{P}}$ are equivalent, denoted $\alpha \equiv \beta$, if $\mathcal{M}(\alpha) = \mathcal{M}(\beta)$

Information and equivalence

Definition (Information)

The information about the system possessed by an agent is reflected by the selection of a set $\overline{X}$ of excluded states inside $\mathcal{S}$, leaving a complementary set X of included states. We say α expresses the agent's information if $\mathcal{M}(\alpha) = X$. We say $\mathcal{N}(\alpha) = \overline{X}$ is the information content of α.

Definition (Equivalence)

We say $\alpha, \beta \in \mathcal{L}_{\mathcal{P}}$ are equivalent, denoted $\alpha \equiv \beta$, if $\mathcal{M}(\alpha) = \mathcal{M}(\beta)$

Example (Light-fan system)

$p \to q \equiv \neg p \vee q$ $\quad p \equiv p \wedge p$ $\quad p \wedge q \equiv \neg(\neg p \vee \neg q)$ $\quad \neg\neg p \equiv p$

Information and equivalence

Definition (Information)

The information about the system possessed by an agent is reflected by the selection of a set $\overline{X}$ of excluded states inside $\mathcal{S}$, leaving a complementary set X of included states. We say α expresses the agent's information if $\mathcal{M}(\alpha) = X$. We say $\mathcal{N}(\alpha) = \overline{X}$ is the information content of α.

Definition (Equivalence)

We say $\alpha, \beta \in \mathcal{L}_{\mathcal{P}}$ are equivalent, denoted $\alpha \equiv \beta$, if $\mathcal{M}(\alpha) = \mathcal{M}(\beta)$

Example (Light-fan system)

$p \to q \equiv \neg p \vee q \qquad p \equiv p \wedge p \qquad p \wedge q \equiv \neg(\neg p \vee \neg q) \qquad \neg\neg p \equiv p$

Note that $\equiv$ is not a connective! $\alpha \equiv \beta$ is not a sentence of $\mathcal{L}_{\mathcal{P}}$!
The symbol $\equiv$ belongs to the metalanguage

Entailment

The most important relationship between sentences in logic

- Suppose an agent learns that α is the case
- What is the agent now entitled to believe?
- The sentences that somehow follow from α are called consequences of α

Entailment

The most important relationship between sentences in logic

- Suppose an agent learns that α is the case
- What is the agent now entitled to believe?
- The sentences that somehow follow from α are called consequences of α

Definition (Classical Entailment)

We say α classically entails β, denoted $\alpha \models \beta$, if $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$. With $Cn(\alpha) \stackrel{\text{def}}{=} \{\beta \mid \alpha \models \beta\}$ we denote the set of all classical consequences of α.

Entailment

The most important relationship between sentences in logic

- Suppose an agent learns that α is the case
- What is the agent now entitled to believe?
- The sentences that somehow follow from α are called consequences of α

Definition (Classical Entailment)

We say α classically entails β, denoted $\alpha \models \beta$, if $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$. With $Cn(\alpha) \stackrel{\text{def}}{=} \{\beta \mid \alpha \models \beta\}$ we denote the set of all classical consequences of α.

Example (Light-fan system)

$p \wedge q \models p$ since $\mathcal{M}(p \wedge q) = \{11\} \subseteq \mathcal{M}(p) = \{11, 10\}$

Entailment

The most important relationship between sentences in logic

- Suppose an agent learns that α is the case
- What is the agent now entitled to believe?
- The sentences that somehow follow from α are called consequences of α

Definition (Classical Entailment)

We say α classically entails β, denoted $\alpha \models \beta$, if $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$. With $Cn(\alpha) \stackrel{\text{def}}{=} \{\beta \mid \alpha \models \beta\}$ we denote the set of all classical consequences of α.

Example (Light-fan system)

$p \wedge q \models p$ since $\mathcal{M}(p \wedge q) = \{11\} \subseteq \mathcal{M}(p) = \{11, 10\}$

Generalisation to $X \subseteq \mathcal{L}_{\mathcal{P}}$

- $X \models \alpha$ if $\mathcal{M}(X) \subseteq \mathcal{M}(\alpha)$
- $Cn(X) \stackrel{\text{def}}{=} \{\alpha \mid X \models \alpha\}$

Two views of entailment

Information content

- If $\alpha \models \beta$, then $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$, and therefore $\mathcal{N}(\beta) \subseteq \mathcal{N}(\alpha)$
- β expresses part of the information expressed by α
- E.g. in $p \land q \models p$, $\mathcal{N}(p) = \{01, 00\} \subseteq \mathcal{N}(p \land q) = \{10, 01, 00\}$

Two views of entailment

Information content

- If $\alpha \models \beta$, then $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$, and therefore $\mathcal{N}(\beta) \subseteq \mathcal{N}(\alpha)$
- β expresses part of the information expressed by α
- E.g. in $p \land q \models p$, $\mathcal{N}(p) = \{01, 00\} \subseteq \mathcal{N}(p \land q) = \{10, 01, 00\}$

Conditioning

- In $\alpha \models \beta$, α 'picks out' a subset of $\mathcal{S}$: $\mathcal{M}(\alpha)$
- So $\alpha \models \beta$ means that if we focus on $\mathcal{M}(\alpha)$, β is guaranteed to hold
- The sentence α 'conditions' the question of whether β is true

Epilogue

Summary

- An 'opaque' knowledge representation language: propositional logic
- Semantics: states v. valuations
- Notion of satisfaction of a sentence
- Notions of model and nonmodel
- The foundation of reasoning: entailment

Epilogue

Summary

- An ‘opaque’ knowledge representation language: propositional logic
- Semantics: states v. valuations
- Notion of satisfaction of a sentence
- Notions of model and nonmodel
- The foundation of reasoning: entailment

What next?

- The expressiveness of languages
- A note on meta-languages