Fundamentals of Artificial Intelligence
Exercise 4: Logical Agents

Florian Lercher

Technical University of Munich

Friday 1^{st} December, 2023

Problem 4.1: Model, satisfaction relation, and entailment

Which of the following statements are correct? Prove correctness by reasoning about the models satisfying each sentence.

1. $\mathit{False} \models \mathit{True}$
2. $\mathit{True} \models \mathit{False}$
3. $(A \wedge B) \models (A \Leftrightarrow B)$
4. $(A \Leftrightarrow B) \models (A \vee B)$
5. $(A \Leftrightarrow B) \models (\neg A \vee B)$

Reminder: Entailment

Entailment

Entailment is the relationship between two sentences where the truth of one sentence requires the truth of the other sentence, which is written as

$$\alpha \models \beta$$

if α entails β. Formally, entailment is defined as

$$\alpha \models \beta \text{ if and only if } M(\alpha) \subseteq M(\beta).$$

For instance, the sentence $x = 0$ entails $xy = 0$.

Models:

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

$M(A \vee B) = \{(T,T), (T,F), (F,T)\}$

$\models$ vs. $\Rightarrow$:

Problem 4.1: Model, satisfaction relation, and entailment

1. *False* $\models$ *True*

$M(\text{False}) \subseteq M(\text{True})$

$\Leftrightarrow$ False $\xrightarrow{\text{never}}$ True

$\Leftrightarrow$ $\emptyset \subseteq$ All modle

correct

Problem 4.1: Model, satisfaction relation, and entailment

2. $True \models False$

$\Leftrightarrow M(True) \subseteq M(False)$

$\Leftrightarrow$ All models $\not\subseteq \emptyset$

Incorrect

Problem 4.1: Model, satisfaction relation, and entailment

3. $(A \land B) \models (A \Leftrightarrow B)$

A	B	$A \land B$	$A \Leftrightarrow B$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	T

$M(A \land B) = M\{(T,T)\}$

$M(A \Leftrightarrow B) = M\{(T,T),(F,F)\}$

$M(A \land B) \subseteq M(A \Leftrightarrow B)$

Correct

Problem 4.1: Model, satisfaction relation, and entailment

4. $(A \Leftrightarrow B) \models (A \vee B)$

A	B	$A \Leftrightarrow B$	$A \vee B$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	T	F

$M(A \Leftrightarrow B) =$

$M\{(T,T),(F,F)\}$

$M(A \vee B) =$

$M\{(T,T),(T,F),(F,T)\}$

$\therefore M(A \Leftrightarrow B) \not\subseteq M(A \vee B)$

incorrect

Problem 4.1: Model, satisfaction relation, and entailment

5. $(A \Leftrightarrow B) \models (\neg A \vee B)$

✓ Corret

A	B	$A \Leftrightarrow B$	$\neg A \vee B$
T	T	T	T
F	T	F	T
T	F	F	F
F	F	T	T

Problem 4.2: Validity, satisfiability, and unsatisfiability

Problem 4.2.1: Prove the following two metatheorems:

1. Sentence α is valid if and only if $\alpha \equiv$ *True*,
2. Sentence α is unsatisfiable if and only if $\alpha \equiv$ *False*.

$\alpha \equiv$

Reminder: Validity and satisfiability

Validity

A sentence is valid if it is true in **all** models (e.g. $P \vee \neg P$). Valid sentences are also known as **tautologies**.

Satisfiability

A sentence is satisfiable if it is true in **some** model. E.g. the expression $P_1 \wedge P_2$ is satisfiable for $P_1 = P_2 = \mathit{true}$, whereas $P_1 \wedge \neg P_1$ is not satisfiable.

- The problem of determining the satisfiability of sentences is also called **SAT** problem, which is NP-complete.
- Validity and satisfiability are connected: α is valid if $\neg\alpha$ is unsatisfiable.

Problem 4.2: Validity, satisfiability, and unsatisfiability

1. Sentence α is valid if and only if $\alpha \equiv True$

$\alpha \equiv True$

$\Rightarrow$ if $\alpha \models True$ and $True \models \alpha$

$\therefore M(\alpha) \subseteq M(True)$

if α valid $\Rightarrow M(\alpha) \subseteq M(True)$
all malle

$\therefore M(True) \subseteq M(\alpha)$

same

Correct ✓

Problem 4.2: Validity, satisfiability, and unsatisfiability

2. Sentence α is unsatisfiable if and only if $\alpha \equiv$ *False*

$M(\alpha) = \emptyset = M(False)$ $\Longleftrightarrow$

Problem 4.2: Validity, satisfiability, and unsatisfiability

Problem 4.2.2: Show whether each of the following sentences is valid, satisfiable, or unsatisfiable. To this end, use the two metatheorems above, the standard logical equivalences from the lecture, and the following four logical equivalences:

$$\alpha \lor \neg\alpha \equiv \mathit{True} \qquad \alpha \lor \alpha \equiv \alpha$$

$$\alpha \land \neg\alpha \equiv \mathit{False} \qquad \alpha \land \alpha \equiv \alpha$$

1. $\mathit{Smoke} \Rightarrow \mathit{Smoke}$
2. $(\mathit{Smoke} \Rightarrow \mathit{Fire}) \Rightarrow (\neg\mathit{Smoke} \Rightarrow \neg\mathit{Fire})$
3. $\mathit{Smoke} \lor \mathit{Fire} \lor \neg\mathit{Fire}$
4. $(\mathit{Fire} \Rightarrow \mathit{Smoke}) \land \mathit{Fire} \land \neg\mathit{Smoke}$

Reminder: Logical equivalences

Standard logical equivalences

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Problem 4.2: Validity, satisfiability, and unsatisfiability

1. $Smoke \Rightarrow Smoke$

$\alpha \vee \neg\alpha \equiv True$ $\qquad$ $\alpha \vee \alpha \equiv \alpha$

$\alpha \wedge \neg\alpha \equiv False$ $\qquad$ $\alpha \wedge \alpha \equiv \alpha$

$\Rightarrow \neg$ smoke $\vee$ smoke

$\Rightarrow$ smoke $\vee \neg$ smoke

$\Rightarrow$ True

valid satisficible

Problem 4.2: Validity, satisfiability, and unsatisfiability

2. $(Smoke \Rightarrow Fire) \Rightarrow (\neg Smoke \Rightarrow \neg Fire)$

$\equiv \neg(\neg smoke \vee Fire) \vee (Smoke \vee \neg Fire)$

$\equiv (Smoke \wedge \neg Fire) \vee (Smoke \vee \neg Fire)$

$\equiv (Smoke \vee \neg Fire \vee Smoke) \wedge (Smoke \vee \neg Fire \vee \neg Fire)$

$\equiv (Smoke \vee \neg Fire) \wedge (Smoke \vee \neg Fire)$

$\equiv Smoke \vee \neg Fire$

Problem 4.2: Validity, satisfiability, and unsatisfiability

3. $Smoke \lor Fire \lor \neg Fire$

$\equiv Smoke \lor True$

$\equiv True$

Valid, satisfiable

Problem 4.2: Validity, satisfiability, and unsatisfiability

4. $(Fire \Rightarrow Smoke) \wedge Fire \wedge \neg Smoke$

$(\neg Fire \vee Smoke) \wedge Fire \wedge \neg Smoke$

$\leftrightarrow \quad (\neg Fire \vee Smoke) \wedge Fire \wedge \neg Smoke$

(By distributivity of $\wedge$ over $\vee$, that is $(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$)

$\leftrightarrow \quad ((\neg Fire \wedge Fire) \vee (Smoke \wedge Fire)) \wedge \neg Smoke$

(By the rule $(\alpha \wedge \neg\alpha) \equiv False$)

$\leftrightarrow \quad (False \vee (Smoke \wedge Fire)) \wedge \neg Smoke$

(By the rule $(\alpha \vee False) \equiv \alpha$)

$\leftrightarrow \quad Smoke \wedge Fire \wedge \neg Smoke$

(By commutativity of $\wedge$)

$\leftrightarrow \quad Smoke \wedge \neg Smoke \wedge Fire$

(By the rule $(\alpha \wedge \neg\alpha) \equiv False$)

$\leftrightarrow \quad False \wedge Fire$

(By the rule $(\alpha \wedge False) \equiv False$)

$\leftrightarrow \quad False$

4

Problem 4.3: Knights and Knaves

Suppose we are on an island with two types of inhabitants: "knights" who always tell the truth, and "knaves" who always lie.

> *According to this problem, three of the inhabitants – A, B and C – were standing together in the garden. A stranger passed by and asked A, "Are you a knight or a knave?". A answered, but rather indistinctly, so the stranger could not make out what he said. The stranger then asked B, "What did A say?". B replied, "A said that he is a knave". At this point the third man, C, said "Don't believe B; he's lying!". The question is, what are B and C?*

Model this logic puzzle by introducing three atomic propositions A, B, and C with intended interpretation that A, B, and C are knights.

Problem 4.3: Knights and Knaves

Problem 4.3.1: How can you formalize the sentence "A says that B is a knight"?

case1 A is knight

B is knight

case2 A is thief

B is thief

$A \Leftrightarrow B$

Problem 4.3: Knights and Knaves

Problem 4.3.2: Assume that *Remark* represents what a person says and that we can represent it using propositional logic. Additionally, assume that P could either be A, B, or C. From the previous problem, can you generalize the method to model the sentence "person P says (or replies) *Remark*"?

case 1   P is knight   Remark is True

case 2   P is thief   Remark is False

$$P \Leftrightarrow Remark$$

Problem 4.3: Knights and Knaves

Problem 4.3.3: Model the following facts which are taken from the puzzle:

1. B replies, "A said that he is a knave."
2. C says, "Don't believe B; he's lying!"

① A say A is thief

A is knight ⇒ A is thief

A is thief ⇒ A is knight.

$B \Leftrightarrow (A \Leftrightarrow \neg A)$

② $C \Leftrightarrow \neg B$

Problem 4.3: Knights and Knaves

Problem 4.3.4: By using the following logical equivalences

$$(X \Leftrightarrow \neg X) \equiv \mathit{False}$$
$$(X \Leftrightarrow \mathit{False}) \equiv \neg X$$

and the following deduction (inference) rule

$$\frac{P \Leftrightarrow Q \qquad Q}{P}$$

deduce what B and C are.

Problem 4.3: Knights and Knaves

Problem 4.3.4:

$B \Leftrightarrow \underbrace{(A \Leftrightarrow \neg A)}$

$\equiv B \Leftrightarrow \quad False$

$\equiv \underline{\underline{\neg B}}$

$\underbrace{C \Leftrightarrow \neg B \quad \neg B}$

C

C is a knight.

Problem 4.4: Superman does not exist

If Superman were able and willing to prevent evil, he would do so. If Superman were unable to prevent evil, he would be impotent; if he were unwilling to prevent evil, he would be malevolent. Superman does not prevent evil. If Superman exists, he is neither impotent nor malevolent. Therefore, Superman does not exist.

Assume that we use the following propositions and their meaning:

$$\begin{aligned} A &: \text{Superman is able to prevent evil.} \\ W &: \text{Superman is willing to prevent evil.} \\ I &: \text{Superman is impotent.} \\ M &: \text{Superman is malevolent.} \\ P &: \text{Superman prevents evil.} \\ E &: \text{Superman exists.} \end{aligned}$$

Problem 4.4: Superman does not exist

Problem 4.4.1: Formalize the facts from the text using the propositions defined above.

Problem 4.4: Superman does not exist

1. If Superman were able and willing to prevent evil, he would do so.

   $(A \vee W) \Rightarrow P$

2. If Superman were unable to prevent evil, he would be impotent.

   $\neg A \Rightarrow I$

3. If he were unwilling to prevent evil, he would be malevolent.

   $\neg W \Rightarrow M$

4. Superman does not prevent evil.

   $\neg P$

5. If Superman exists, he is neither impotent nor malevolent.

   $E \Rightarrow (\neg I \wedge \neg M)$

Problem 4.4: Superman does not exist

Problem 4.4.2: Assume we want to prove that "Superman does not exist" using the resolution approach for propositional logic. Identify which sentences belong to the knowledge base KB, and which sentence we want to deduce. How do we need to process these sentences before applying the resolution principle?

Reminder: Resolution algorithm

Inference procedures based on resolution use the principle of **proof by contradiction**:

To show that $KB \models \alpha$, we show that $KB \wedge \neg\alpha$ is unsatisfiable.

Basic procedure

1. $KB \wedge \neg\alpha$ is converted into CNF
2. The resolution rule is applied to the resulting clauses: each pair that contains complementary literals is resolved to produce a new clause, which is added to the others (if not already present)
3. The process continues until
   - there are no new clauses to be added $\Rightarrow KB \not\models \alpha$;
   - two clauses resolve to yield the *empty* clause $\Rightarrow KB \models \alpha$.

Reminder: Resolution rule

Full resolution rule

$$\frac{l_1 \vee \ldots \vee l_k, \quad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n},$$

where l_i and m_j are complementary literals.

Reminder: Conjunctive Normal Form

- The resolution rule only applies to disjunction of literals, which are also called **clauses**.
- Fortunately, every sentence of propositional logic can be reformulated as a conjunction of clauses, which is also referred to as **conjunctive normal form (CNF)**

Conjunctive Normal Form

A sentence with literals x_{ij} of the form $\bigwedge_i \bigvee_j (\neg) x_{ij}$ is in conjunctive normal form.

Examples:

- $(A \vee B \vee C) \wedge (\neg A \vee B \vee C)$ yes
- $A \wedge B \wedge C \vee (\neg A \wedge B \vee C)$ no
- $A \wedge B \wedge C \wedge (\neg A \vee B \vee C)$ yes

Problem 4.4: Superman does not exist

Problem 4.4.2:

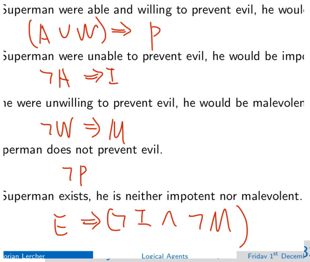
Knowledge base:

1. $\neg(A \lor W) \lor P \equiv \neg A \land \neg W \lor P$
2. $A \lor I$
3. $W \lor M$
4. $\neg P$
5. $\neg E \lor (\neg I \land \neg M) \equiv (\neg E \lor \neg I) \land (\neg E \lor \neg M)$

Goal:

$\neg E$

$\neg\neg E = E$


uperman were able and willing to prevent evil, he woul

$(A \lor W) \Rightarrow P$

uperman were unable to prevent evil, he would be imp

$\neg A \Rightarrow I$

e were unwilling to prevent evil, he would be malevolen

$\neg W \Rightarrow M$

perman does not prevent evil.

$\neg P$

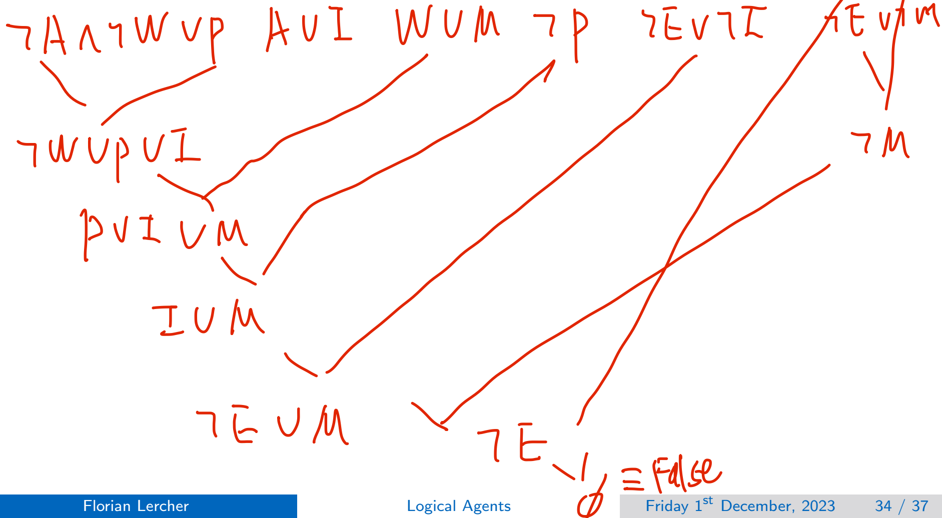
uperman exists, he is neither impotent nor malevolent.

$E \Rightarrow (\neg I \land \neg M)$

orian Lercher Logical Agents Friday 1st Decem

Problem 4.4: Superman does not exist

Problem 4.4.3: Prove diagramatically with the resolution approach that "Superman does not exist."



Problem 4.5: Completeness and soundness

Recall the definition of *completeness* and *soundness*.

Completeness: An inference algorithm is complete if and only if for every entailed sentence $KB \models \alpha$, the inference algorithm will always be able to derive it.

Soundness: An inference algorithm is sound if and only if for every sentence it derives, it is guaranteed that the sentence is entailed $KB \models \alpha$.

Problem 4.5: Completeness and soundness

Problem 4.5.1: Suppose that we have an inference algorithm which will *always* be able to derive a given sentence (regardless whether it is entailed or not). Would this inference algorithm be complete? Sound?

Solution: This inference algorithm is **complete**, because for every entailed sentence, this algorithm will always be able to derive it. However, this inference algorithm is **unsound**, because it can derive a sentence that is not entailed.

Problem 4.5: Completeness and soundness

Problem 4.5.2: Suppose now that we have an inference algorithm which will *never* be able to derive a given sentence (regardless whether it is entailed or not). Would this inference algorithm be complete? Sound?

Solution: This inference algorithm is **incomplete**, because for every entailed sentence, this algorithm will always be unable to derive it. However, this inference algorithm is **sound** since it never derives any sentence.